

XXII ХАРИТОНОВСКИЕ НАУЧНЫЕ ЧТЕНИЯ

XXII KHARITON SCIENTIFIC TALKS

СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

SUPERCOMPUTER SIMULATIONS AND ARTIFICIAL INTELLIGENCE



СБОРНИК НАУЧНЫХ ТРУДОВ

PROCEEDINGS

24-27 мая 2021 Саров

ФГУП «РОССИЙСКИЙ ФЕДЕРАЛЬНЫЙ ЯДЕРНЫЙ ЦЕНТР –
ВСЕРОССИЙСКИЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ЭКСПЕРИМЕНТАЛЬНОЙ ФИЗИКИ»

СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

SUPERCOMPUTER SIMULATIONS AND ARTIFICIAL INTELLIGENCE

Международная конференция
XXII Харитоновские тематические научные чтения
24–27 мая 2021 года

International conference
XXII Khariton scientific talks
24–27 May 2021

СБОРНИК НАУЧНЫХ ТРУДОВ

PROCEEDINGS

Под редакцией
доктора физ.-мат. наук Р. М. Шагалиева

Саров
2022

УДК 004+519.6
ББК 22.18
С 89
DOI 10.53403/9785951505071_2022

С 89 **Суперкомпьютерное моделирование и искусственный интеллект**. Международная конференция «XXII Харитоновские тематические научные чтения, 24–27 мая 2021 года»: Сборник научных трудов / Под ред. доктора физ.-мат. наук Р. М. Шагалиева. – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2022. – 534 с.: ил.

ISBN 978-5-9515-0507-1

В сборник включены доклады международной конференции «XXII Харитоновские тематические научные чтения. Суперкомпьютерное моделирование и искусственный интеллект». Тематика конференции: методы, модели и алгоритмы численного моделирования физических процессов на высокопроизводительных вычислительных системах; эффективные алгоритмы высокопараллельных вычислений; технология полномасштабных компьютерных испытаний сложных технических систем на основе создания «цифровых двойников»; архитектуры высокопроизводительных вычислительных систем, системное программное обеспечение; искусственный интеллект и нейронные сети; виртуально-имитационное моделирование.

УДК 004+519.6
ББК 22.18

Редакционно-оформительская группа

Ю. М. Белугина, А. А. Гудкова, А. М. Цыганко

ISBN 978-5-9515-0507-1 © ФГУП «РФЯЦ-ВНИИЭФ», 2022

ОРГАНИЗАЦИИ-УЧАСТНИКИ

Госкорпорация по атомной энергии «РОСАТОМ», г. Москва

Российский Федеральный Ядерный Центр – Всероссийский НИИ экспериментальной физики, г. Саров

АО «ТВЭЛ», г. Москва

Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина, г. Воронеж

Военный инновационный технополис «ЭРА», г. Анапа

Волжский государственный университет водного транспорта, г. Нижний Новгород

Всероссийский научно-исследовательский институт автоматики им. Н. Л. Духова, г. Москва

Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН, г. Москва

Дизайн-центр цифровой микроэлектроники, МГУ им. М. В. Ломоносова, г. Москва

Ижевский государственный технический университет им. М. Т. Калашникова, г. Ижевск

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск

Институт гидродинамики им. М. А. Лаврентьева СО РАН, г. Новосибирск

Институт проблем механики им. А. Ю. Ишлинского РАН, г. Москва

Институт проблем проектирования в микроэлектронике РАН, г. Зеленоград

Институт программных систем им. А. К. Айламазяна РАН, г. Москва

Институт радиотехники и электроники им. В. А. Котельникова РАН, г. Москва

Московский государственный университет им. М. В. Ломоносова, г. Москва

Московский физико-технический институт (государственный университет), г. Москва

Научно-исследовательский институт «Квант», г. Москва

Нижегородский государственный технический университет им. Р. Е. Алексеева, г. Нижний Новгород

Нижегородский государственный университет им. Н. И. Лобачевского, г. Нижний Новгород

Объединенный институт высоких температур РАН, г. Москва

ООО «Научно-исследовательский центр супер-ЭВМ и нейрокомпьютеров», г. Таганрог

ПАО «Компания «Сухой» ОКБ «Сухого», г. Москва

Российский Федеральный Ядерный Центр – Всероссийский НИИ технической физики им. акад. Е. И. Забабахина, г. Снежинск

Санкт-Петербургский государственный университет, г. Санк-Петербург

Санкт-Петербургский политехнический университет Петра Великого, г. Санкт-Петербург

Саровский физико-технический институт НИЯУ МИФИ, г. Саров

Сургутский государственный университет, г. Сургут

Сургутский филиал ФНЦ НИИСИ РАН, г. Сургут

Удмуртский государственный университет, г. Ижевск

Федеральный исследовательский центр Инстиут прикладной математики им. М. В. Келдыша РАН, г. Москва

Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва

Федеральный научный центр НИИ системных исследований РАН, г. Москва

Центр по проектированию, производственным технологиям и материалам, Сколковский институт науки и технологии, г. Москва

Южный федеральный университет, г. Ростов-на-Дону

СОДЕРЖАНИЕ

Аблесимов В. Е., Пашарина О. Ю.
Особенности формирования сигнала времяпролетного детектора при измерении излучения плазменного фокуса в замкнутом пространстве 9

Абросимов Н. А., Дьянов Д. Ю., Елесин А. В., Константинов А. Ю.
Моделирование прогрессирующего разрушения ортотропных материалов при динамическом нагружении в пакете программ ЛОГОС 19

Абузяров М. Х., Глазова Е. Г., Кочетков А. В., Крылов С. В.
Численное моделирование трехмерных процессов ударного взаимодействия льда с упругопластической преградой с использованием многосеточных алгоритмов и схемы Годунова повышенной точности 27

Бабанов А. В., Бондаренко С. В., Митрофанов Е. И., Потапкина Л. Ф., Шаров О. О.
Численное моделирование трехмерных задач конверсии лазерного излучения в рентгеновское в сферическом боксе мегаджоульной лазерной установки 38

Баканов Е. К., Алешкин В. А., Красноярова Д. А., Симаков В. Ю., Пищулин И. А., Чайка А. И., Новиков О. В.
Защищенная операционная система «Арамид» для супер-ЭВМ 45

Баранов А. В., Еременко А. Ю., Косарим С. С., Наумов А. О., Ниткина Е. С., Речкин В. Н., Романов В. И., Рябов А. А., Циберев К. В.
Возможности пакета программ ЛОГОС для моделирования отклика конструкций на гармоническую вибрацию 48

Бетелин В. Б., Галкин В. А., Дубовик А. О.
Нейросетевой алгоритм вейвлет-преобразования функций одной переменной 61

Борляев В. В., Афанасьев В. А., Дьянов Д. Ю., Серякова И. В., Беженцева М. В.
Пакет программ ЛОГОС. Методы моделирования кинематики недеформируемых твердых тел и механизмов 66

Быков А. Н., Ветчинников М. В., Сапронов И. С., Дерюгин Ю. Н., Лазарев А. А.
Численное моделирование селективного лазерного спекания порошковой смеси 75

Быков А. Н., Попов В. В., Дерюгин Ю. Н., Ветчинников М. В., Дьянов Д. Ю., Полищук С. Н., Куделькин В. Г., Синицин Е. А., Бахаев А. Н., Журавлева М. В., Британова О. И., Титов М. А., Баранов А. В., Гамов Л. В., Милешин И. Г., Кузнецов В. Ю., Машенькин П. А., Голубкова Е. Ф., Данилов А. С., Медведкина М. В., Каякин А. А., Ломайкин А. И., Сапронова О. В., Веселова Е. А., Емельянов А. Б., Лазарев А. А., Сапронов И. С., Балуева Н. Н., Ларькина О. С.
Проект «Виртуальный 3D-принтер» 90

Габзетдинова Л. Я., Новиков И. Г., Половникова Т. Н., Самодолов А. В., Шиберин И. В., Занегин И. В., Катыков А. Н.
Двумерные и трехмерные расчеты по методике ТИМ моделирования работы взрывного резака на основе метода ударных волн разрежения для резки труб 103

Гордеев А. В., Дюпин В. Н., Еременко А. Ю., Мустаева Н. А., Санталов А. С.
Программное средство импорта внешних нагрузок в расчетную модель ЛОГОС-Прочность, реализованное в рамках препостпроцессора пакета программ «ЛОГОС» 111

Гордеев Д. Г., Жильникова Н. Н., Кидямкина Д. Н., Кудельки н В. Г., Куликова М. В., Шумилина О. Н.
Библиотека УРС-ОФ для векторизованных параллельных вычислений теплофизических функций на общей памяти 116

Губайдулина Е. А., Тюндина А. А., Сурина О. Ю.
Использование интегратора модульной интеграционной платформы ЛОГОС для конфигурирования задач различного типа 125

Гурин А. М., Байкин А. Н.
Эффективное метапрограммирование алгоритмов метода конечных элементов средствами языка C++17 135

Дьянов Д. Ю., Казанцев А. В., Челаков А. А.
Пакет программ ЛОГОС. Использование метода сглаженных частиц при решении задач высокоскоростного динамического деформирования 142

Дьянов Д. Ю., Медведкина М. В., Наумова Е. И., Шувалова Е. В.
Пакет программ ЛОГОС. Методы моделирования поведения композитных материалов при динамическом нагружении 155

Евстифеева Е. О., Панкратов Д. М., Цалко Т. В.
Алгоритмы автоматического исправления пересечений, реализованные в генераторе поверхностных треугольных сеток в пакете программ «ЛОГОС» 170

Елизаров Г. С., Конотопцев В. Н., Корнеев В. В.
Специализированные большие интегральные схемы для реализации нейросетевых выводов 181

Елизаров С. Г., Корнеев В. В.
Архитектура специализированных больших интегральных схем для высокопроизводительных вычислений 185

Змеев Д. Н., Климов А. В., Окунев А. С., Левченко Н. Н.
Особенности реализации теста HPCG для ППВС «БУРАН» 193

Иванов К. В., Галкин М. В., Сайфуллин А. И., Сайфуллина Р. Н., Девятых Д. В.
Проблемы создания инструментальных средств программирования логики поведения агентов в мультиагентных системах имитационного моделирования двухсторонних боевых действий 206

Коваленко О. В., Ежов Д. В., Крючков И. А.
Параллельный алгоритм оптимизации непрерывных функций большой размерности на основе гибридизации алгоритма оптимизации «серых волков» 217

Коротков А. В., Лашкин С. В., Козелков А. С.
Моделирование задач гидродинамики на сетках, содержащих несогласованные сеточные интерфейсы 226

Кошелева Е. В., Сельченкова Н. И., Соколов С. С., Трунин И. Р., Учаев А. Я.
О критическом поведении динамических систем при высокоинтенсивном воздействии 239

Крючков И. А., Ежов Д. В., Огородников А. В.
Средства мультиагентного имитационного моделирования визуализационно-интеграционной платформы «ОптИМУС» 254

Кузнецов В. Ю., Машенькин П. А., Дерюгин Ю. Н., Барабанов Р. А., Филимонкин Е. А., Глазунов В. А., Вишняков А. Ю., Ларькина О. С.
Моделирование процессов теплопроводности и прочности на макроуровне при селективном лазерном плавлении в программном комплексе «Виртуальный 3D-принтер» 261

Лазарев В. В., Борисенко О. Н.
Построение смещенной сетки для выделения пограничного слоя в пакете программ «ЛОГОС» 273

Левин И. И., Доронченко Ю. И., Дордопуло А. И., Левина М. Г.
Высокопроизводительная реконфигурируемая вычислительная система на основе ПЛИС Xilinx UltraScale+ 284

Лобовикова Л. В., Лобовиков П. В., Куркина О. Е., Куркин А. А., Козелков А. С.
Динамические эффекты интенсивных внутренних волн в двухслойной жидкости: лабораторные эксперименты и численное моделирование 297

Ломтев В. В., Жирнов В. В.
Реализация механизма отображения результатов моделирования с использованием объемного рендеринга в параллельном режиме 304

Муратов Р. В.
Алгоритм динамической балансировки нагрузки для адаптивных сеток 308

Надуев А. Г., Черевань А. Д., Кожаев Д. А.
Архитектура модульной интеграционной платформы ЛОГОС 316

Никитин В. А.
Функциональный блок генерации объемных неструктурированных сеток на основе регулярного дробно-адаптивного шаблона в методике ТИМ 323

Никитин В. А., Шурыгин А. В., Новиков И. Г., Егоров А. В., Соколов С. С., Панов А. И.
Программный модуль генерации замкнутой поверхностной триангуляционной сетки в пакете программ ЛОГОС 330

Олесницкая К. К., Дьяков А. В., Побуринная Н. А., Тюндина А. А.
Модульная интеграционная платформа ЛОГОС. Технология подготовки и выполнения мультифизичного моделирования на основе связанно-сопряженного расчета 340

Попов В. В., Быков А. Н., Дерюгин Ю. Н., Ветчинников М. В., Дьянов Д. Ю., Полищук С. Н., Куделькин В. Г., Синицин Е. А., Бахаев А. Н., Журавлева М. В., Бритова О. И., Титов М. А., Баранов А. В., Гамов Л. В., Милешин И. Г., Кузнецов В. Ю., Машенькин П. А., Голубкова Е. Ф., Данилов А. С., Медведкина М. В., Каякин А. А., Ломайкин А. И., Сапронова О. В., Веселова Е. А., Емельянов А. Б., Лазарев А. А.
Программный комплекс «Виртуальный 3D-принтер»: от концепции к реализации 350

Потапов А. А., Кузнецов В. А., Аликулов Е. А.
Мультифрактальное комплексирование многодиапазонных радиолокационных изображений 363

Потапов А. А., Кузнецов В. А., Меньших Н. А.
Возможности ускорения имитационного моделирования эффекта вторичной модуляции сигнала бортовой радиолокационной станции в среде MATLAB 368

Потехин А. Л., Соловьев А. Н.
Препроцессор пакета программ ЛОГОС. Подготовка начальных данных для моделирования задач прочностного анализа с использованием солид элементов 375

Русяк И. Г., Суфиянов В. Г., Королев С. А., Клюкин Д. А.
Математическое моделирование колебаний ствола с учетом теплообмена при стрельбе очередями 386

Саразов А. В., Козелков А. С., Зеленский Д. К., Жучков Р. Н., Резвова Т. В.
Решение задач аэродинамики с подвижными границами в пакете программ ЛОГОС 392

Свидинский А. В., Соколов С. С., Новиков И. Г., Самодолов А. В., Половникова Т. Н.
Результаты численного моделирования бронебойного действия ударника на преграду по методике ТИМ 406

Синицин Е. А., Бритова О. И., Попов В. В., Бахаев А. Н., Гамов Л. В.
Этапы подготовки детали к печати в программе «Виртуальный 3D-принтер» 417

Собанин Д. С., Крючков И. А., Огородников А. В., Коваленко О. В.
Язык программирования агентов Opal++ визуализационно-интеграционной платформы «ОптИМУС» 426

Софронов В. Н., Ветчинников М. В., Демина М. А.
Использование методов гамильтоновой динамики в численных расчетах задач механики сплошной среды 434

Степаненко С. А.
Фотонная вычислительная машина. Оптические логические элементы 451

Стручков А. В., Козелков А. С., Жучков Р. Н.
Особенности применения метода геометрической многоуровневой инициализации для ускорения решения задач аэродинамики на произвольных неструктурированных сетках . . . 461

Суфиянов В. Г., Новиков А. В.
Анализ результатов распараллеливания градиентного алгоритма решения систем нелинейных алгебраических уравнений 473

Тишкин Д. А., Ярулина А. А., Дьяков А. В., Козлов Д. О., Надуев А. Г., Тюхтина П. А., Шемякин А. В.
Клиент-серверная система контроля лицензий ЛОГОС 481

Уткин Д. А., Тятюшкина Е. С., Козелков А. С., Курулин В. В.
Применение неотражающих граничных условий для численного моделирования задач со свободной поверхностью 490

Шанин И. И., Задорожный К. В.
Имитационная модель радиолокационной станции обнаружения и слежения 500

Шумилов И. А., Тимофеев А. Е., Потехин А. Л.
Адаптивная генерация регулярных расчетных сеток на основе решения задач целочисленного линейного программирования 512

Сведения об авторах 519

УДК 539
DOI 10.53403/9785951505071_2022_9

ОСОБЕННОСТИ ФОРМИРОВАНИЯ СИГНАЛА ВРЕМЯПРОЛЕТНОГО ДЕТЕКТОРА ПРИ ИЗМЕРЕНИИ ИЗЛУЧЕНИЯ ПЛАЗМЕННОГО ФОКУСА В ЗАМКНУТОМ ПРОСТРАНСТВЕ

В. Е. Аблесимов, О. Ю. Пашарина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Проведен анализ результатов измерений выхода ДД-нейтронов в экспериментах с плазмофокусной камерой (ПФ) мейзеровского типа. Проведены расчеты формирования сигнала применительно к регистрации излучения ПФ сцинтилляционными детекторами (СД) методом времени пролета для конкретных условий измерений в экспериментальном зале. Постановка эксперимента по измерению излучения ПФ в экспериментальном зале СД смоделирована в расчетах, проведенных по методике С-007 на основе метода Монте-Карло. Условия измерений характеризуются трехмерной геометрией среды, окружающей источник излучения и детектор, заметным вкладом в регистрируемый сигнал нейтронного и гамма-излучения, обусловленного взаимодействием нейтронов с окружающей источник средой.

Проведенный анализ показал, что существенное влияние на формирование сигнала СД оказывает анизотропия нейтронного потока, генерируемого в плазмофокусной камере. Расчеты методом Монте-Карло позволили установить роль вклада в сигнал детектора излучения, рассеянного в окружающей среде, вкупе с анизотропией потока нейтронов из источника.

Ключевые слова: плазменный фокус, анизотропия выхода нейтронов, сцинтилляционные детекторы, метод времени пролета, метод Монте-Карло.

Введение

Рассмотрены результаты измерений характеристик излучения ПФ-камеры методом времени пролета в экспериментах, проводимых в помещении ограниченных размеров. Конструкция камеры, цели эксперимента и некоторые результаты приведены в работе [1]. Описываемые измерения параметров импульсного плазмофокусного источника нейтронов характеризуются существенно трехмерной геометрией среды, окружающей источник излучения и детектор, воздействием на детектор смешанного нейтронного и гамма-излучения, обусловленного взаимодействием нейтронов с окружающей источник средой, неопределенностью спектрального состава рассеянного излучения.

В работе рассмотрены особенности формирования сигнала детекторов в условиях экспериментального зала.

Проведено сравнение результатов измерения излучения ПФ сцинтилляционными детекторами СД методом времени пролета с результатами расчетов аналогичной зависимости методом Монте-Карло.

Рассмотрено влияние анизотропии нейтронного излучения ПФ на формирование вида временной зависимости регистрируемого излучения.

Экспериментальное оборудование и геометрия измерений

Экспериментальный зал имеет размеры 18×12 м и высоту 7,5 м. Источник излучения (ПФ) располагался на высоте 2,06 м от пола. Минимальное расстояние от ближайшей стены составляло 4,5 м, до бетонного потолка ~5,5 м.

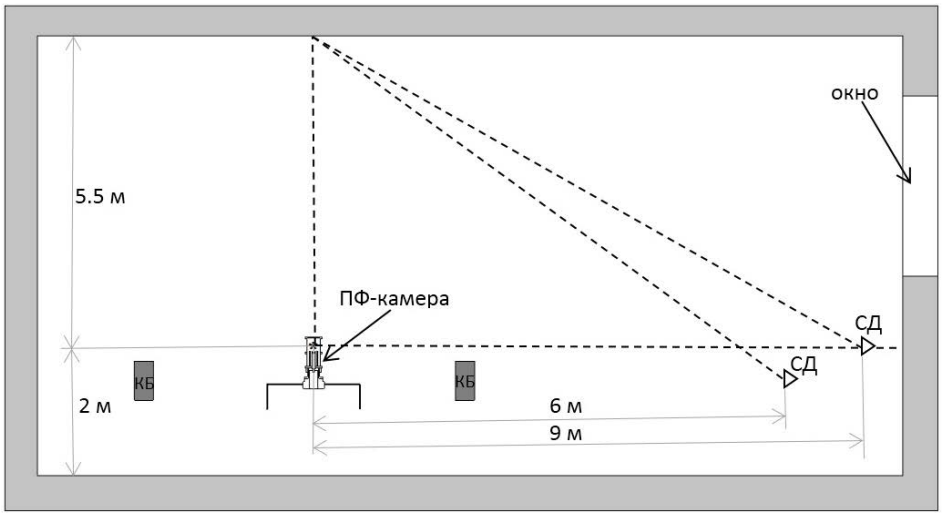


Рис. 1. Схема расположения ПФ-камеры и детекторов СД в экспериментальном зале

Направление от источника на детекторы составляло ~ 90° к оси камеры. Детекторы СД устанавливались на расстоянии 6 и 9 м от источника излучения. Времяпролетный интервал между гамма-излучением и ДД-нейтронами с энергией ~2,5 МэВ для детекторов, установленных на расстоянии 9 м, равняется 390 нс, для 6 м – 260 нс.

В смешанных неколлимированных полях нейтронного и гамма-излучения кроме прямых нейтронов источника в сигнал СД дает вклад фоновое излучение – рассеянные в окружающей среде нейтроны и образующиеся в результате взаимодействия нейтронов с окружением гамма-кванты. Проведенные ранее в том же помещении измерения с экранированием одного из детекторов теневым цилиндром [2] показывают, что вклад фонового излучения в сигнал детектора составляет существенную долю сигнала детектора.

При этом спектральные характеристики того и другого видов фонового излучения неизвестны, а экспериментальное их определение весьма затруднительно и представляет отдельную сложную задачу.

Описание экспериментальных осциллограмм и обсуждение их особенностей

Результатом измерений детекторами СД является временная зависимость тока, вызываемого в детекторе воздействующим на него импульсом излучения. Осциллограммы сигнала с детектора позволяют определить временные характеристики импульса излучения. Интегральный выход нейтронов определяется площадью сигнала детектора.

Осциллограмма сигнала СД (временная зависимость тока детектора от времени) в рассматриваемых экспериментах имеет сложную структуру, соответствующую многообразию процессов генерации излучений в плазменной камере и вкладу в сигнал детектора рассеянных составляющих излучения. Характерная форма импульса детекторов, размещенных на расстояниях 6 и 9 м от ПФ, приведена на рис. 2, 3.

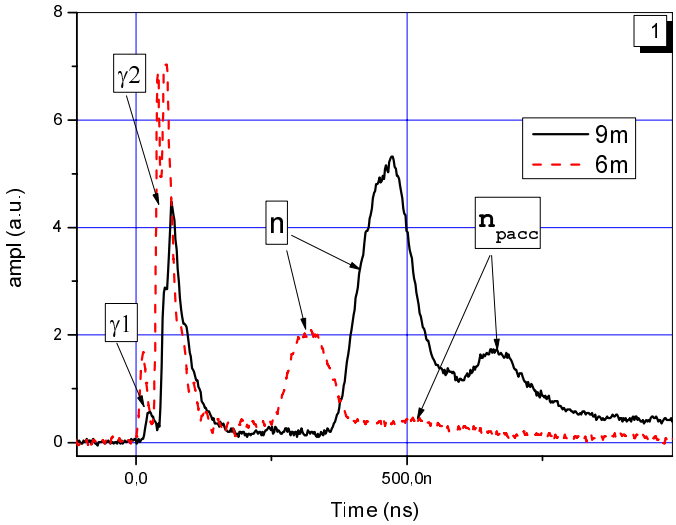


Рис. 2. Характерные особенности сигнала сцинтилляционного детектора (разряд 1)

Для расстояния 9 м первый слабый пик гамма-излучения (**γ1**) отвечает моменту генерации нейтронов, второй (**γ2**), смещенный на ~ 40 нс, генерации жесткого рентгеновского (гамма) излучения (полуширина 20 нс), далее на ~ 390 нс по фронту от первого гамма импульса отстоит пик основного нейтронного импульса **n** (полуширина около 100 нс), и далее в ряде случаев наблюдается более или менее выраженный пик $\mathbf{n}_{\mathbf{pacc}}$ (на отметке около 600–700 нс), наложенный на «хвост» основного импульса, обусловленного рассеянием излучения и простирающегося на интервал более 1,5 мкс после разряда в камере. Вклад в формирование столь длительного сигнала детектора вносит также гамма-излучение, рожденное в процессах неупругого взаимодействия нейтронов с материалами ПФ-камеры и стенами, полом, потолком экспериментального зала. Процессы взаимодействия нейтронов идут также на конденсаторных батареях, других элементах обстановки зала, однако вклад последних, как показали расчеты, существенно меньше.

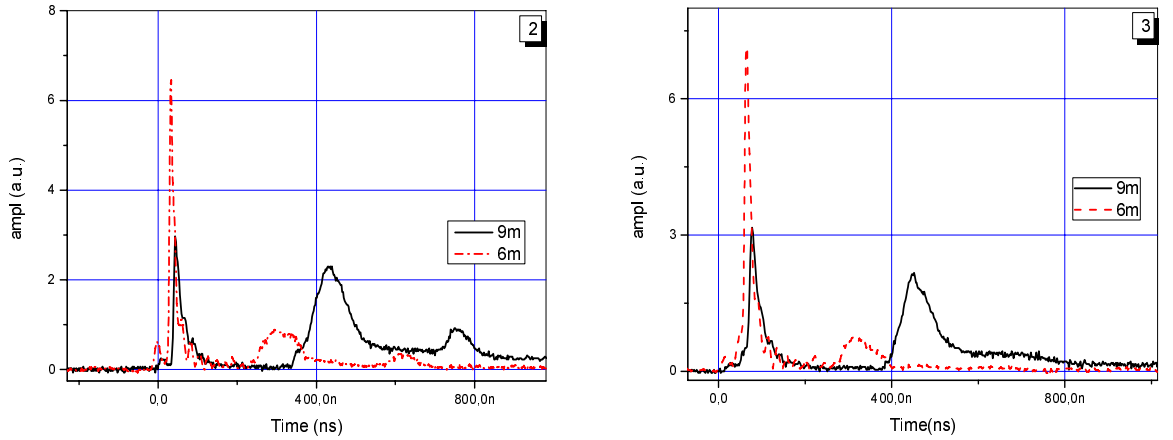


Рис. 3. Осциллограммы сигнала детекторов (разряды 2 и 3)

Для детектора на расстоянии 6 м пики, связанные с нейтронным излучением, смещены в соответствии с меньшим временем пролета ближе к гамма-импульсу (~260 нс для основного и около 500–600 нс для рассеянного нейтронного импульса).

Нет физических механизмов, связанных с процессами в ПФ-камере и отвечающих за возникновение в самой камере нейтронов с временем пролета до детектора 600–700 нс (9 м) и 500–600 нс (6 м) – разряд уже закончился.

В качестве возможной причины возникновения пика на осциллограммах была рассмотрена гипотеза рассеяния нейтронов от потолка экспериментального зала (помещения). В пользу этого предположения говорит зависимость положения указанного пика на осциллограмме от места размещения детектора, что можно связать с временем пролета упруго рассеянных нейтронов до детектора. Если предположить, что этот пик отвечает упругому рассеянию нейтронов от потолка экспериментального зала, то время появления этих пиков (~650 нс для 9 м и ~550 нс для 6 м) соответствует времени пролета нейтронами с энергией 2,9 МэВ расстояния 5,5 м от источника излучения до потолка помещения (катет) и расстояния от области упругого рассеяния на потолке до места расположения детекторов. Для обоснования этой гипотезы было проведено сравнение результатов измерения излучения ПФ методом времени пролета с результатами расчетов аналогичной зависимости методом Монте-Карло.

Заметим, что пик рассеянных нейтронов выражен на разных осциллограммах в разной степени (рис. 1 и 2). В частности он может быть практически неразличим в структуре сигнала – рис. 3, разряд 3. Это зависит от величины анизотропии выхода нейтронов из плазменного фокуса.

Влияние анизотропии нейтронного излучения ПФ на зависимость от времени сигнала детектора

Выше отмечено, что величина пика рассеянных нейтронов от импульса к импульсу варьируется (осциллограммы разрядов 1, 2 и 3). Возможной причиной изменения вклада этой составляющей сигнала может служить различная степень анизотропии излучения ПФ. Анизотропия излучения, как известно, свойственна процессу генерации нейтронов в ПФ и проявляется в зависимости значений выхода и энергии генерируемых нейтронов от направления вылета излучения. Как правило, анизотропия измеряется значением коэффициента анизотропии $K_a = Y(0^o)/Y(90^o)$ как отношение выхода нейтронов по оси камеры (0^o) и перпендикулярно оси (90^o). Она приписывается ускорительному механизму образования нейтронов, согласно которому ускоренные в малой области фокуса дейтроны бомбардируют области невозмущенного газа и разогретой плазменной оболочки (модель «пучок-мишень»). Наряду с ускорительным механизмом в ПФ-камере в области фокуса (максимального сжатия плазмы) реализуется термоядерный механизм образования нейтронов с изотропным распределением генерируемых нейтронов.

Вклад ускорительного механизма генерации нейтронов сильно варьируется от разряда к разряду в силу случайного стохастического характера газового разряда в камере в процессе экспериментов. Исследование анизотропии выхода нейтронов из ПФ-камеры представляет интерес для изучения вопроса о соотношении термоядерного и ускорительного механизмов генерации нейтронов в исследуемых устройствах и деталей этих процессов.

Анизотропия характеристик нейтронного потока, возникающего в ДД-реакции в плотном плазменном фокусе, является предметом исследований в ряде работ [3, 4, 5]. В работах [3] измеренная величина анизотропии выхода нейтронов составляла 1,28+–0,04, в работе [4] от 1,28 до 1,48, в работе [5] в узком конусе $\pm 5^0$ от оси камеры зафиксирована анизотропия до 5,4–6. Такую величину анизотропии авторы связывают с зарегистрированным экспериментально с помощью камеры Фарадея возникновением узких ионных пучков, распространяющихся из области фокуса в том же интервале углов. В работе авторов с сотрудниками [6] измеренная величина анизотропии варьировалась в диапазоне 1,3–2 от импульса к импульсу, а энергия нейтронов составляла 2,9 МэВ под 0^o по отношению к 2,5 МэВ под углом 90^o к оси камеры. В работе [7] значение K_a изменяется в диапазоне 1,5–3,2 и зависит от давления ДД-газа в камере (5–15 мбар), в работе [8] K_a определен как 1,33±0,18, а энергия нейтронов в осевом и радиальном направлениях составляла 2,9 МэВ и 2,58 МэВ соответственно, в работе [9] анизотропия выхода нейтронов была оценена как 1,59±0,12. Обширный обзор работ, связанных с генерацией нейтронного излучения в ПФ и с анизотропией генерируемого излучения, приведен в работе [10].

Расчетное исследование формирования сигнала сцинтилляционного детектора

Современные возможности метода Монте-Карло [11] и соответствующего вычислительного комплекса позволяют проводить расчеты в трехмерной геометрии, практически не зависящие от ее сложности.

Для обоснования нашей гипотезы и объяснения особенностей, наблюдаемых на экспериментальных осциллограммах, в частности для выяснения влияния анизотропии выхода нейтронов из источника на форму сигнала сцинтилляционного детектора были проведены модельные расчеты по программе МК, в которых угловое распределение источника нейтронов задавалось: а) изотропным с энергией 2,5 МэВ и б) равномерно распределенным в конусе с углом раствора ±3градуса от оси камеры с энергией 2,9 МэВ [6, 8] – сильная степень анизотропии. Обоснованием для этого служит явно выраженная локализация рассеянных нейтронов в довольно узком временном диапазоне, что позволяет предположить генерацию по оси устройства направленного потока нейтронов с малой угловой расходимостью. Для более наглядной интерпретации процессов генерации рассеянного излучения источник нейтронов задавался мгновенным,

Эти расчеты позволили установить временную привязку и относительный вклад в сигнал детектора составляющих излучения, рассеянного от окружающей источник и детектор среды. Результатом расчетов является выход света из сцинтиллятора, рассчитываемый с учетом модели световыхода, описанной в [12], пропорциональный выходному току детектора (напряжению), регистрируемому на осциллограмме.

В расчетах учитывалась геометрия экспериментального зала и конструкционные (основные) элементы размещенного в нем оборудования (конденсаторная батарея, сама конструкция камеры, подводящие кабели), учитывалось рассеяние в воздухе. Оценивался парциальный вклад этих элементов в сигнал детектора от рассеянных нейтронов.

Результаты указанных расчетов, с расшифровкой вклада отдельных процессов, приведены на рис. 4, 5.

На рис. 4 приведена расчетная зависимость световыхода сцинтиллятора, установленного на расстоянии 9 м, для направленного по оси камеры источника нейтронов.

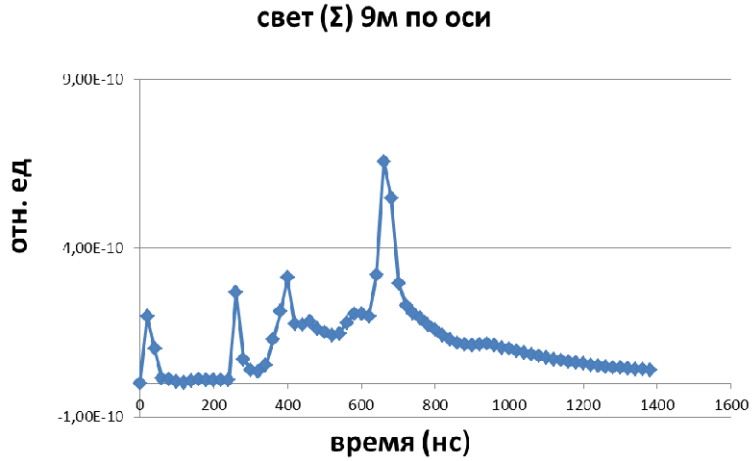


Рис. 4. Расчетная зависимость световыхода сцинтиллятора 9 м (источник нейтронов, направленный по оси камеры)

По горизонтальной оси время в нс, по вертикали – величина, пропорциональная свету, возникающему в сцинтилляторе под воздействием падающего на детектор излучения. Здесь первый пик – гамма-кванты от неупругого взаимодействия нейтронов с материалами плазмофокусной камеры (время пролета до детектора ~30 нс), второй пик 270–280 нс – гамма-кванты от неупругого взаимодействия нейтронов с потолком экспериментального зала (на расстоянии 5,5 м), куда направлен поток нейтронов, третий пик 380–420 нс – нейтроны, рассеянные материалами ПФ-камеры, четвертый пик 700 нс – нейтроны, рассеянные в месте взаимодействия нейтронного потока с потолком экспериментального зала.

Аналогичные расчеты были проведены и для точечного изотропного источника нейтронов. На рис. 5 приведены расчетные зависимости суммарного световыхода в детекторе на расстоянии 9 м.

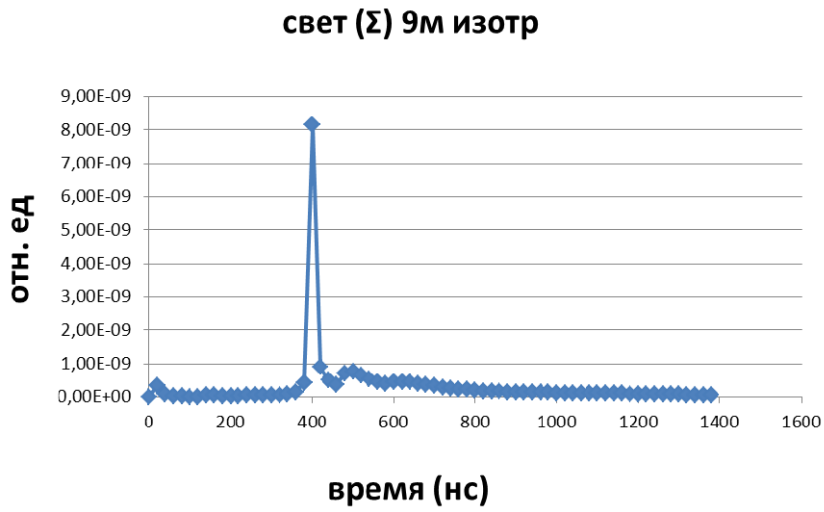


Рис. 5. Расчетная зависимость световыхода сцинтиллятора (источник нейтронов изотропный)

Здесь также явно просматривается вклад в сигнал детектора процессов рассеяния нейтронов источника на элементах окружающей обстановки, но в случае изотропного источника временная зависимость рассеянного излучения более сглажена за счет распределения области рассеяния нейтронов по всей окружающей источник и детектор среде.

Приведем также несколько результатов для детектора, установленного на 6 м. Для случая направленного по оси источника расчетная зависимость сигнала детектора показана на рис. 6.

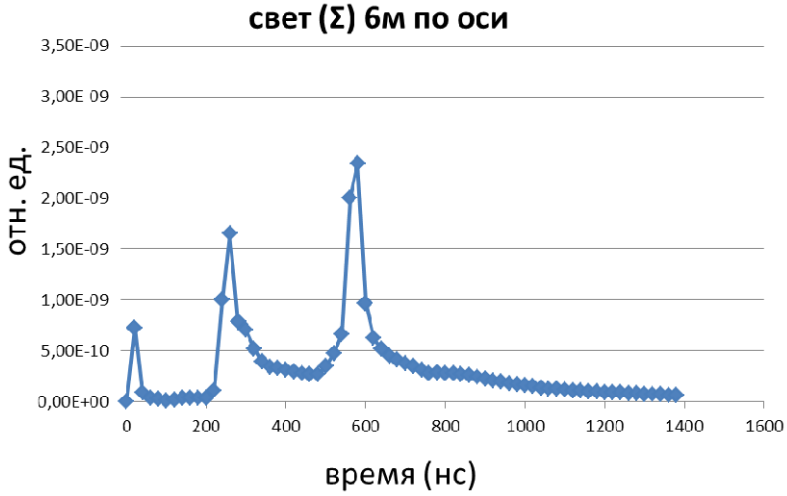


Рис. 6. Расчетная зависимость световыхода сцинтиллятора 6 м (источник нейтронов, направленный по оси камеры)

Для случая изотропного источника расчетная зависимость сигнала детектора показана на рис. 7).

В реальном эксперименте сигнал детектора является суперпозицией этих двух крайних случаев с изменяющейся от импульса к импульсу относительной долей. Примеры такой суперпозиции для относительного вклада 50/50 изотропной и направленной по оси камеры части приведены на рис. 8 и 9.

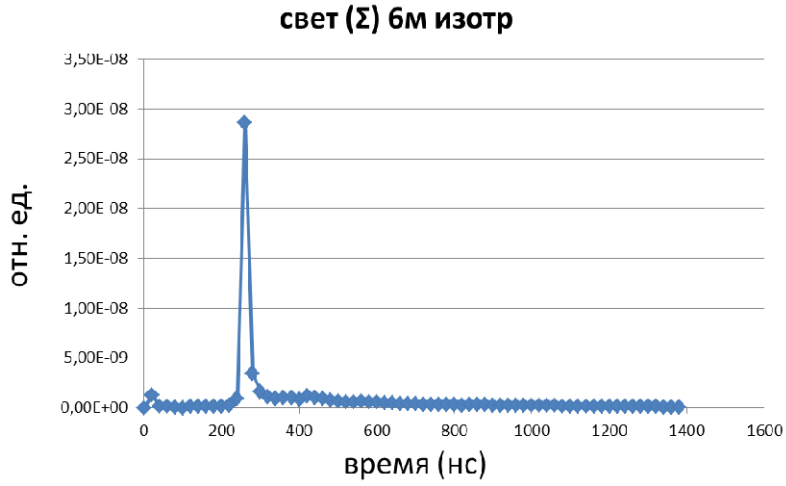


Рис. 7. Расчетная зависимость световыхода сцинтиллятора (изотропный источник нейтронов)

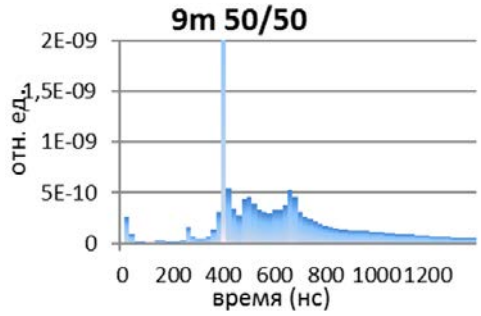


Рис. 8. Расчетная зависимость световыхода сцинтиллятора 9 м

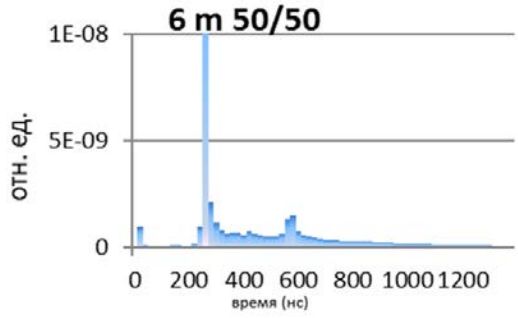


Рис. 9. Расчетная зависимость световыхода сцинтиллятора 6 м

Ясно видно, что характерные особенности расчетных временных зависимостей совпадают с аналогичными особенностями экспериментальной осциллограммы. Как общая картина, так и временное положение характерных пиков на экспериментальных осциллограммах сигналов (рис. 2 и 3) в целом соответствуют расчетной картине, видно заметное влияние анизотропии на форму временной зависимости сигналов детекторов на разных расстояниях. Напомним, что пример практического отсутствия анизотропии в одном из разрядов (3) иллюстрирует рис. 3.

В обзоре [10] указывается, что «по имеющимся оценкам, доля анизотропной нейтронной эмиссии может составлять от нескольких до 42,5 % от полного нейтронного выхода», т. е. может наблюдаться сильное различие доли анизотропии от разряда к разряду. Соответственно, вклад рассеянных нейтронов в сигнал детектора может быть ярко выраженным или почти неразличимым, обусловливая различие регистрируемых зависимостей.

Обсуждение результатов

Расчеты позволили воспроизвести временную картину зависимости сигнала детектора для конкретных условий эксперимента, совпадающую с аналогичными особенностями экспериментальной осциллограммы, в частности подтвердить гипотезу рассеяния от потолка узконаправленного пучка нейтронов.

Отметим, что многокомпонентная смесь элементов (бетон), рассеяние от которой воспроизводилось в расчетах, затрудняет детальный анализ – это скорее качественное рассмотрение с выделением характерных черт процесса. Возможность точных количественных измерений открывает спе-

циальная редакция измерений, где рассеиватель (графит) с фиксированными размерами, установленный определенным образом, позволяет провести рассеяние осевого пучка нейтронов только в упругом процессе и тем самым упростить получение количественных результатов.

На практике измерения выхода нейтронов ведутся под определенным углом к оси камеры, а затем результаты пересчитываются в значение полного выхода (как правило, в предположении изотропного распределения излучения источника). При определении истинного значения полного выхода нейтронов из источника надо учитывать влияние анизотропии либо расширением неопределенности результата, либо проводить подробные расчеты, добиваясь совпадения расчетной и экспериментальной осциллограмм.

Возможно, но весьма трудоемко и существенно удорожает эксперимент проведение измерений, например, несколькими СД под разными углами при тщательной коллимации излучения от источника в направлении детектора. В частности в экспериментах, описанных в работе [13], выход и спектр нейтронов были зарегистрированы под углами 0^0, 90^0, и 180^0, что расширяет представления о характеристиках исследуемого устройства. Строгая коллимация выходящего из источника излучения, как правило, неоправданно удорожает постановку измерений.

В конкретных приложениях нас всегда интересует выход нейтронов в определенном направлении, поэтому с практической точки зрения возможно проведение измерений в определенном направлении в неколлимированном пучке с вычетом вклада рассеянных нейтронов по результатам «теневых» измерений, согласно методике, рассмотренной в [2].

Заключение

Проведенный анализ формирования сигнала сцинтилляционных детекторов, регистрирующих выход нейтронов из ПФ-камеры методом времени пролета, указывает на существенное влияние на формирование сигнала анизотропии нейтронного потока. Условия измерений характеризуются трехмерной геометрией среды, окружающей источник излучения и детектор, существенным вкладом фонового нейтронного и гамма-излучения в регистрируемый детекторами СД сигнал.

Рассмотрены особенности формирования сигнала детекторов СД в условиях ограниченного пространства экспериментального зала, обусловленные как рассеянным излучением, так и режимом работы ПФ.

Для объяснения особенностей, наблюдаемых на экспериментальных осциллограммах, была предложена гипотеза о влиянии анизотропии выхода нейтронов из источника на форму сигнала СД. Для обоснования этого предположения были проведены модельные расчеты по программе Монте-Карло, в которых угловое распределение источника нейтронов задавалось: а) изотропным и б) направленным по оси камеры (сильная степень анизотропии). Для более наглядной интерпретации процессов генерации рассеянного излучения источник нейтронов задавался мгновенным. Как общая картина, так и временное положение характерных пиков на экспериментальных осциллограммах в целом соответствуют расчетной картине и подтверждают гипотезу заметного влияния на формирование сигнала детектора рассеянного излучения вкупе с анизотропией потока нейтронов из источника.

Результаты расчетов можно применить для анализа проводимых экспериментов, в том числе для оценки наличия анизотропии выхода нейтронов в конкретном разряде.

Литература

1. Аблесимов В. Е., Андрианов А. В., Базанов А. А., Глыбин А. М., Долин Ю. Н., Дудай П. В., Зименков А. А., Иванов В. А., Ивановский А. В., Калинычев А. Е., Карпов Г. В., Краев А. И., Ломтев С. С., Нудиков В. Н., Пак С. В., Поздов Н. И., Полюшко С. М., Рыбаков А. Ф., Скобелев А. Н., Туров А. Н., Февралев А. Ю. Разработка плазмофокусного нейтронного источника, запитываемого от взрывомагнитного генератора // ПМТФ, № 1, 2015 г., с. 94–103. Новосибирск.

2. Аблесимов В. Е., Цибиков З. С., Пашко О. В., Долин Ю. Н. Комплексные измерения выхода нейтронов импульсного источника типа «плазменный фокус». Методы и средства физических измерений, Т. 1. Сборник докладов VI отраслевой научно-технической конференции / Саров: РФЯЦ-ВНИИЭФ, 2008. С. 24–33.

3. Bernstein M. J., Meskan D. A., van Paasen H. L. L. Space, time, and energy distribution of neutrons and X-rays from a focused plasma discharge.//Physics of Fluids. 1969. V.12. N. 10. P. 2193.

4. Lee J. H., Shomo L. P., Williams M. D., Hermansdorfer H. Neutron production mechanism in a plasma focus // Physics of Fluids. 1971. V. 14. N. 10. P. 2217.

5. Castillo Mejia F., Milanese M., Moroso R., Pouso J. Some experimental research on anisotropic effects in the neutron emission of dense plasma-focus devices // J. Phys.D: Appl.Phys. 1997. V. 30. P. 1499–1506.

6. Аблесимов В. Е., Долин Ю. Н., Пашко О. В., Цибиков З. С. Корреляция анизотропии выхода нейтронного излучения с электрическими характеристиками разряда типа «плазменный фокус» // Физика плазмы, том 36, № 5, Май 2010, С. 436–439.

7. Roshan M. V., Rawat R. S., Talebitaher A., Lee P., and Springham S. V. Neutron and high energy deuteron anisotropy investigations in plasma focus device // Physics of Plasmas, 16, (2009); https://doi.org/10.1063/1.3133189

8. Ram Niranjana, R. K. Rout, R. Srivastava, T. C. Kaushik, and Satish C. Gupta. A 109 neutrons/pulse transportable pulsed D-D neutron source based on flexible head plasma focus unit. //Review of Scientific Instruments 87, 033504 (2016); https://doi.org/10.1063/1.4942666

9. Milanese1 M., Moroso1 R., Castillo F., Herrera J. J. E., Golzarri J. I., and Espinosa G. Time Integrated and Time Resolved Neutron Measurements in a Plasma Focus Device // AIP Conference Proceedings 875, 423 (2006); https://doi.org/10.1063/1.2405979

10. Дубинов А. Е., Сенилов Л. А. Исследования на плазменных фокусах в развивающихся странах / Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2013.

11. Житник А. К., Донской Е. Н., Огнев С. П., Горбунов А. В. и др. Методика С-007 решения методом Монте-Карло связанных линейных уравнений переноса нейтронов, гамма-квантов, электронов и позитронов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2011. Вып. 1. С. 17–24.

12. Шмаров А. Е., Семенов В. И. Модель световыхода органических жидких и пластмассовых сцинтилляторов для протонов // Труды РФЯЦ-ВНИИЭФ. 2014. Вып. 19. Ч. 1. С. 182–193.

13. Голубев В. И., Макеев Н. Г., Пащенко Е. С., Рубцов Н. В., Спирин А. А., Сурский О. К., Цукерман В. А., Черемухин Г. Н. // Физика и техника импульсных источников ионизирующих излучений для исследования быстропротекающих процессов: Труды ученых ядерных центров России. Саров: РФЯЦ–ВНИИЭФ, 1996. Вып. 5. С. 297–303.

THE SPECIFICS OF THE TIME-OF-FLIGHT DETECTOR SIGNAL GENERATION IN MEASUREMENTS OF THE PLASMA FOCUS RADIATION IN A CLOSED SPACE

V. E. Ablesimov, O. Yu. Pasharina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

Measuring results of DD-neutron yield in experiments with plasma-focus chamber (PF-chamber) of Meiser type were analyzed. Signal forming was computed with regard to PF-radiation registration with scintillation detector (SD) with time-of-flight method for specific measuring conditions in the experimental room. The setup of the experiment on

PF-radiation measurements in experimental SD room was simulated in computations performed by C-007 procedure based on Monter-Carlo method. Measuring conditions are characterized by a 3D geometry of the radiation source and detector surrounding medium, a significant neutron and gamma radiation contribution into the registered signal conditioned by the interaction of neutrons with the source surrounding medium.

The performed analysis has shown that anisotropy of the neutron flux generated in plasma-focus chamber influences the formation of SD signal essentially. Monte-Carlo computation made it possible to find the role of radiation dissipated in the surrounding medium along with anisotropy of the neutron flux from the source as they contribute to the signal.

Key words: plasma focus, neutron yield anisotropy, scintillation detectors, time-of-flight method, Monte Carlo method.

УДК 539.4
DOI 10.53403/9785951505071_2022_19

МОДЕЛИРОВАНИЕ ПРОГРЕССИРУЮЩЕГО РАЗРУШЕНИЯ ОРТОТРОПНЫХ МАТЕРИАЛОВ ПРИ ДИНАМИЧЕСКОМ НАГРУЖЕНИИ В ПАКЕТЕ ПРОГРАММ ЛОГОС

Н. А. Абросимов, Д. Ю. Дьянов, А. В. Елесин, А. Ю. Константинов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Композитные материалы находят все большее применение при разработке современных конструкций различного назначения, подвергающихся действию интенсивных импульсных нагрузок. При этом одной из ключевых проблем является выбор расчетной модели оценки несущей способности элемента конструкции с учетом совокупности нескольких факторов деформирования: сложного напряженно-деформированного состояния, влияния прогрессирующего разрушения, неоднородности полей напряжений и деформаций, нестационарности деформирования, влияния скорости деформирования и т. д.

Решение данной многоплановой задачи осложняется тем, что разрушение композитных материалов является многофакторным и многостадийным процессом, реализующимся на различных масштабных и структурных уровнях композитных материалов. Кроме того, характерная особенность конструкций из композитных материалов состоит в том, что материал и конструкция создаются одновременно – в рамках единого технологического процесса. Поэтому разработка моделей деформирования и разрушения композитных материалов является актуальной проблемой.

Для расчета напряженно-деформированного состояния и оценки динамической прочности элементов конструкций, выполненных из композиционных материалов, используются различные модели прогрессирующего разрушения ортотропных материалов [1-8]. Целью настоящего исследования является разработка 3D модели прогрессирующего разрушения ортотропных материалов при динамическом нагружении (аналог MAT_ORTHOTROPIC_SIMPLIFIED_DAMAGE в коммерческом пакете программ LS-DYNA [9,10]). Разработанная модель вводилась через возможность реализации пользовательской модели материала в ПП ЛОГОС [11–13].

Ключевые слова: моделирование, ортотропные материалы, ЛОГОС, композитные материалы.

Определяющие соотношения модели

Определяющие соотношения композитного ортотропного материала устанавливаются на основе закона Гука в сочетании с теорией эффективных модулей [14].

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{bmatrix}=\begin{bmatrix}C_{11} & C_{12} & C_{13} & & & \\ C_{12} & C_{22} & C_{23} & & & \\ C_{13} & C_{23} & C_{33} & & & \\ & & & C_{44} & & \\ & & & & C_{55} & \\ & & & & & C_{66}\end{bmatrix}\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \chi_{12}\\ \chi_{23}\\ \chi_{31}\end{bmatrix}.$$

Элементы матрицы жесткости $C_{ij}\ \left(i,j=\overline{1,6}\right)$ для ортотропного тела находятся согласно следующим соотношениям:

$$C_{11}=\frac{1-\nu_{32}\nu_{23}}{E_{22}E_{33}\Delta};\quad C_{12}=\frac{\nu_{21}+\nu_{31}\nu_{21}}{E_{22}E_{33}\Delta}=\frac{\nu_{12}+\nu_{32}\nu_{13}}{E_{11}E_{33}\Delta};$$

$$C_{13}=\frac{\nu_{31}+\nu_{21}\nu_{32}}{E_{22}E_{33}\Delta}=\frac{\nu_{13}+\nu_{12}\nu_{23}}{E_{11}E_{22}\Delta};\quad C_{22}=\frac{1-\nu_{13}\nu_{31}}{E_{11}E_{33}\Delta};$$

$$C_{23}=\frac{\nu_{32}+\nu_{12}\nu_{31}}{E_{11}E_{33}\Delta}=\frac{\nu_{23}+\nu_{21}\nu_{13}}{E_{11}E_{22}\Delta};\quad C_{33}=\frac{1-\nu_{12}\nu_{21}}{E_{11}E_{22}\Delta};$$

$$C_{44}=G_{12};\quad C_{55}=G_{23};\quad C_{66}=G_{31},$$

где $\Delta=\dfrac{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{13}\nu_{31}-2\nu_{21}\nu_{32}\nu_{13}}{E_{11}E_{22}E_{33}}$, $E_{ii}\left(i=\overline{1,3}\right)$ – модули упругости, $G_{ij}\ \left(i,j=\overline{1,3}\right)$ – модули сдвига, $\nu_{ij}\ \left(i,j=\overline{1,3}\right)$ – коэффициенты Пуассона ортотропного материала.

В случае ортотропного материала между коэффициентами упругости $C_{ij}\ \left(i,j=\overline{1,6}\right)$ и коэффициентами податливости $s_{ij}\ \left(i,j=\overline{1,6}\right)$ существуют следующие зависимости [15]:

$$C_{11}=\frac{s_{22}s_{33}-s_{23}^2}{s};\qquad C_{12}=\frac{s_{13}s_{23}-s_{12}s_{33}}{s};$$

$$C_{22}=\frac{s_{33}s_{11}-s_{13}^2}{s};\qquad C_{13}=\frac{s_{12}s_{23}-s_{13}s_{22}}{s};$$

$$C_{33}=\frac{s_{11}s_{22}-s_{12}^2}{s};\qquad C_{23}=\frac{s_{12}s_{13}-s_{23}s_{11}}{s};$$

$$C_{44}=\frac{1}{s_{44}};\quad C_{55}=\frac{1}{s_{55}};\quad C_{66}=\frac{1}{s_{66}},$$

где $s=s_{11}s_{22}s_{33}-s_{11}s_{23}^2-s_{22}s_{13}^2-s_{33}s_{12}^2+2s_{12}s_{23}s_{13}$.

Рассматриваемая модель ортотропного материала включает в себя модель повреждаемости и критерий разрушения. Модель учитывает взаимное влияние различных механизмов разрушения через зависимость прочностных свойств от вида напряженно-деформированного состояния и ортотропию параметров, управляющих этой зависимостью. Модель можно использовать только с объемными конечными элементами.

Для расчета развития повреждений в матрицу податливости вводятся шесть параметров повреждаемости для корректировки трех модулей упругости E11, E22, E33 и трех модулей сдвига G12, G23, G31 и вычисляется матрица податливости поврежденного материала, которая связывает тензоры эффективных деформаций и напряжений.

Принятая модель учитывает девять взаимосвязанных механизмов разрушения: шесть критериев разрушения при растяжении и сжатии по трем локальным осям и три критерия разрушения при сдвиге в 3-х координатных плоскостях (12, 13 и 23). В модели принимается линейный закон эволюции повреждений. До достижения деформации возникновения повреждений монослоев εS (точка A, рис. 1) связь между напряжениями и деформациями принимается линейно-упругой.

Затем происходит линейное разупрочнение материала (линия AC) до полного разрушения материала (элемента) εC (точка C). Разгрузка материала из произвольного состояния на линии AC также принимается линейной (линия BO). Повторное нагружение материала происходит по линии OB. Разница в тангенсах угла наклона OA и OB количественно характеризует деградацию упругих свойств материала. Значения величин εS и εC определяются из экспериментальных диаграмм деформирования растяжения, сжатия и сдвига элементарных образцов с однонаправленной укладкой слоев при нагружении вдоль и поперек волокон.

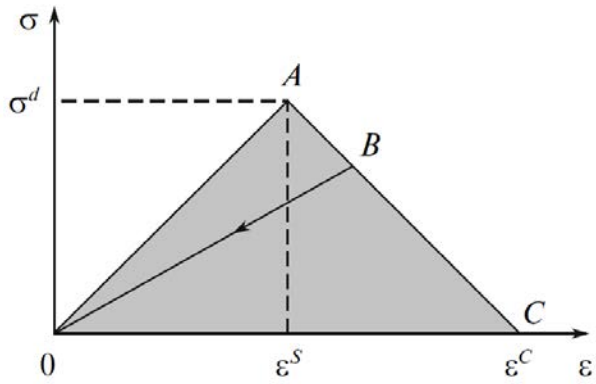


Рис. 1. Эволюция повреждений

Учет деградации материала осуществляется путем корректировки трех модулей упругости E11, E22, E33 и трех модулей сдвига G12, G23, G31, для чего в матрицу податливости вводятся шесть параметров повреждаемости и вычисляется матрица податливости поврежденного материала, которая связывает тензоры эффективных деформаций и напряжений $\{\varepsilon\} = [\mathrm{Sdam}]\{\sigma\}$, где

$$S^{dam} = \begin{pmatrix} \frac{1}{E_{11}\left(1-d_{1t/c}\right)} & \frac{-\nu_{21}}{E_{22}} & \frac{-\nu_{31}}{E_{33}} & 0 & 0 & 0 \\ \frac{-\nu_{21}}{E_{22}} & \frac{1}{E_{22}\left(1-d_{2t/c}\right)} & \frac{-\nu_{32}}{E_{33}} & 0 & 0 & 0 \\ \frac{-\nu_{31}}{E_{33}} & \frac{-\nu_{32}}{E_{33}} & \frac{1}{E_{33}\left(1-d_{3t/c}\right)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}\left(1-d_{12}\right)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}\left(1-d_{23}\right)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}\left(1-d_{31}\right)} \end{pmatrix}.$$

Здесь $d_k = \max\left(d_k ; D_k^c \left\langle \frac{\varepsilon_k - \varepsilon_k^s}{\varepsilon_k^c - \varepsilon_k^s} \right\rangle_+ \right)$, $k = 1t, 2t, 3t, 1c, 2c, 3c, 12, 23, 31$ – индексы направления нагрузки (индекс t – растяжение вдоль указанного направления, индекс c – сжатие), в угловых скобках $\langle\ \rangle_+$ берут положительную часть числа. Коэффициенты D принимаются равными единице, до возникновения повреждений все параметры повреждаемости d_k равны нулю.

В данной модели материала считается, что до начала процесса повреждения материал ведет себя линейно упруго. Чтобы отследить начало повреждения, в этой модели материала используется предельная деформация разрушения – критерий разрушения, основанный на виде разрушения. Такой критерий разрушения является простым способом предсказания начала разрушения без анализа зависимости между напряжением и деформацией в различных направлениях:

при растяжении в направлении оси 1: $\varepsilon_{11} > \varepsilon_{11}^t$ $(\varepsilon_{11} > 0)$;

при сжатии в направлении оси 1: $|\varepsilon_{11}| > \varepsilon_{11}^c$ $(\varepsilon_{11} < 0)$;

при растяжении в направлении осей 2 и 3: $\varepsilon_{22} > \varepsilon_{22}^t$ $(\varepsilon_{22} > 0)$; $\varepsilon_{33} > \varepsilon_{33}^t$ $(\varepsilon_{33} > 0)$;

при сжатии в направлении осей 2 и 3: $|\varepsilon_{22}| > \varepsilon_{22}^c$ $(\varepsilon_{22} < 0)$; $|\varepsilon_{33}| > \varepsilon_{33}^c$ $(\varepsilon_{33} < 0)$;

при плоском сдвиге: $|\chi_{12}| > \Gamma_{12}$;

при неплоском сдвиге: $|\chi_{13}| > \Gamma_{13}$; $|\chi_{23}| > \Gamma_{23}$.

Когда несколько или все деформации отвечают критерию разрушения, начинается соответствующий процесс повреждения.

Для моделирования прогрессирующего разрушения разработанная модель содержит 9 критериев. Разрушение может проходить по трем ортотропным направлениям, при растяжении, при сжатии и при сдвиговом поведении. Разрушением управляют девять деформаций разрушения:

при растяжении в направлении оси 1: $\varepsilon_{11t} \geq \varepsilon_{11tf}$ $(\varepsilon_{11t} > 0)$;

при сжатии в направлении оси 1: $|\varepsilon_{11c}| \geq \varepsilon_{11cf}$ $(\varepsilon_{11c} < 0)$;

при растяжении в направлении осей 2 и 3: $\varepsilon_{22tf} \geq \varepsilon_{22tf}$ $(\varepsilon_{22t} > 0)$; $\varepsilon_{33tf} \geq \varepsilon_{33tf}$ $(\varepsilon_{33t} > 0)$;

при сжатии в направлении осей 2 и 3: $|\varepsilon_{22c}| \geq \varepsilon_{22cf}$ $(\varepsilon_{22c} < 0)$; $|\varepsilon_{33c}| \geq \varepsilon_{33cf}$ $(\varepsilon_{33c} < 0)$;

при плоском сдвиге: $|\chi_{12}| \geq \Gamma_{12f}$;

при неплоском сдвиге: $|\chi_{13}| \geq \Gamma_{13f}$; $|\chi_{23}| \geq \Gamma_{23f}$.

При наступлении разрушения элементы уничтожаются (эрозия). Разрушение возможно при разрушении только одной, нескольких или всех точек интегрирования конечно-элементной модели.

Параметры модели для конкретного композитного материала определяются экспериментальным путем.

Таким образом, модель позволяет учесть деградацию жесткостных характеристик композитного материала при растяжении, сжатии и сдвиге. При превышении пороговых значений деформаций происходит полное разрушение материала и удаление элемента расчетной области.

Программная реализация модели в ПП ЛОГОС

На основе разработанной 3D модели прогрессирующего разрушения ортотропных материалов при динамическом нагружении были сформулированы требования к заданию параметров модели в ЛОГОС-ПреПост, необходимые для реализация пользовательских интерфейсов. В соответствии с имеющейся технологией добавления в пакет программ ЛОГОС нового пользовательского материала был создан модуль, реализующий 3D модель прогрессирующего разрушения ортотропных материалов.

Блок-схема разработанного модуля представлена на рис. 2, где Cmr – массив, содержащий константы, описывающие материал.



Рис. 2. Блок-схема разработанного модуля

Для проверки работы процедуры, отвечающей за расчет тензора напряжений в материале (процедуры, реализующей модель материала), использовались задачи с заданным изменением компонент тензора деформаций в элементарной ячейке. Тестирование проводилось на элементарном объеме материала при реализации простого и сложного напряженно-деформированного состояния. Указанный подход позволяет верифицировать именно реализацию модели материала, исключая влияние остальных модулей расчетного комплекса, способных повлиять на точность решения: схему интегрирования уравнений, формулировку конечных элементов, контактные алгоритмы, схемы стабилизации решения и пр.

Значения параметров модели материала в задачах тестирования были взяты из работы [16], полученные результаты решения 15-ти задач подтверждают работоспособность реализованного модуля.

Решены две исследовательские задачи. Рассматривалась балка квадратного сечения 10×10 мм длиной 30 мм, консольно закрепленная с одного торца (ориентация балки – вдоль оси *Z*, рис. 3). На другом торце балки задавалось перемещение в направлении оси *Y* по закону заданному таблично (табл. 1).

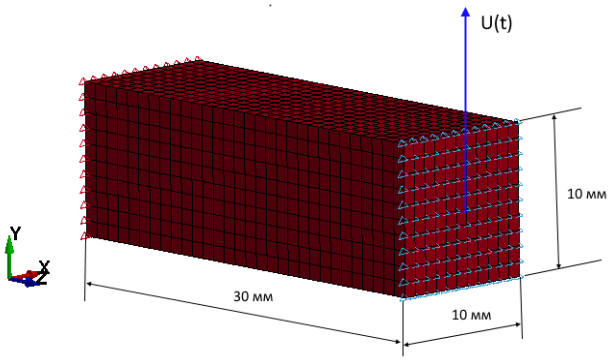


Рис. 3. Постановка задачи

Таблица 1

Закон смещения узлов торца балки

Время, мс	Смещение, мм
0	0
1	2

На рис. 4 приводится поле нормальных напряжений (МПа) в направлении оси *Z* в различные моменты времени. Ось *Y* на рисунках направлена вниз. Видно, что разрушение начинается на плоскости консольного закрепление в зоне сжатия, поскольку критическая деформация при сжатии в 2 раза меньше, чем деформация разрушения при растяжении.

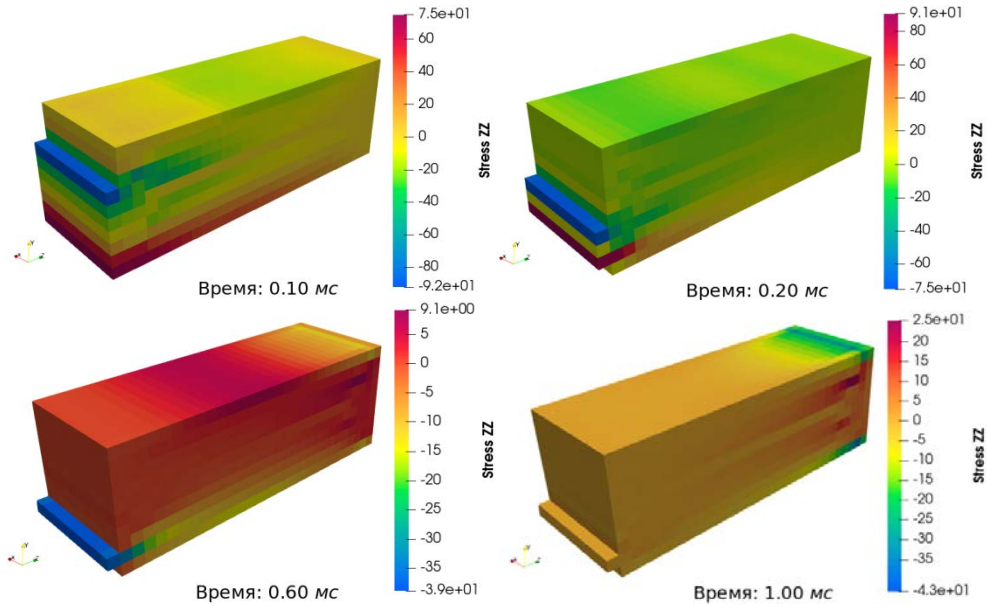


Рис. 4. Поле нормальных напряжений в направлении *Z*, Мпа

Рис. 5 иллюстрирует относительную массу удаленных элементов с течением времени.

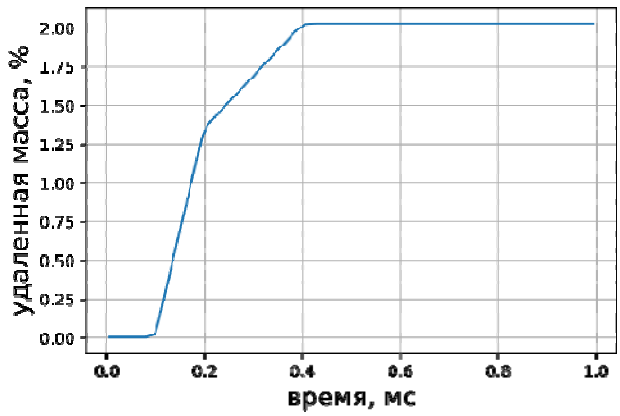


Рис. 5. Масса удаленных элементов

Далее рассматривалась задача о низкоскоростном ударе жесткого шарика по композиционной пластине. Диаметр шарика составлял 60 мм. Начальная скорость шарика 20 м/с. Размеры пластины 500×500 мм. Толщина пластины равна 2 мм. Пластина жестко защемлена по контуру.

На рис. 6 приводятся результаты расчета на разные моменты времени, отображен процесс деформирования и эрозии элементов.

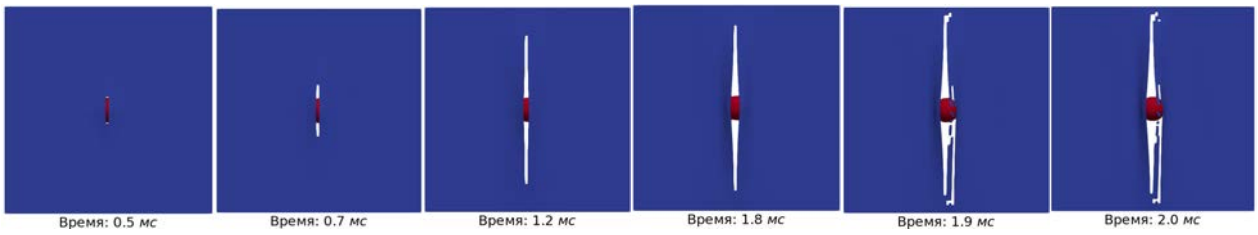


Рис. 6. Результаты решения задачи об ударном нагружении пластины шариком

Рис. 7 иллюстрирует относительную массу удаленных элементов с течением времени, определенную в расчетах с использованием ПП ЛОГОС.

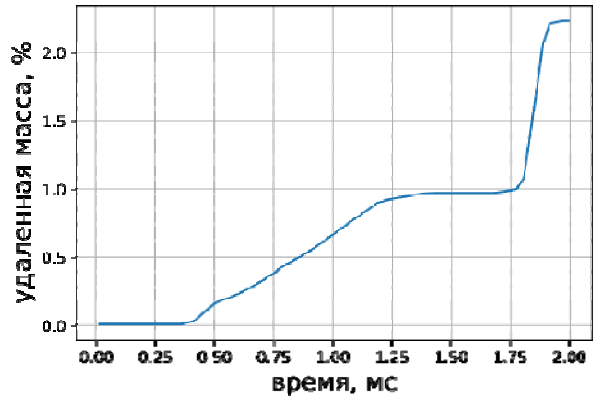


Рис. 7. Масса удаленных элементов

Заключение

Разработанная 3D модель прогрессирующего разрушения ортотропных материалов при динамическом нагружении позволяет учесть ортотропию деформационных и прочностных характери-

стик композиционного материала, а также различную скорость накопления поврежденности. Модель используется с объемными конечными элементами.

Модель достаточно хорошо описывает динамическое хрупкое разрушение ортотропного материала, однако не учитывает упрочнение материала и ряд иных нелинейных эффектов, сопровождающих процесс прогрессирующего разрушения композитных материалов. К недостаткам модели следует также отнести несвязанность процессов накопления повреждений и деградации свойств материала в разных направлениях и для разных знаков нагружения. Указанная модель может быть успешно использована для оценки прочности конструкций из композиционных материалов при нагружениях, для которых за точкой зарождения поврежденности наблюдается одномерное напряженно-деформированное состояние и сохраняется знак до достижения деформаций удаления элемента из расчета.

В соответствии с имеющейся технологией добавления в пакет программ ЛОГОС нового пользовательского материала был создан модуль, содержащий программный код, реализующий разработанную 3D модель прогрессирующего разрушения ортотропных материалов при динамическом нагружении. Решены контрольные и исследовательские задачи.

Литература

1. Gunel M., Kayran A. Non-linear progressive failure analysis of open-hole composite laminates under combined loading / Journal of Sandwich Structures & Materials. 2013. V. 15, No. 3. P. 309–339.
2. Эллул Б., Камиллери Д., Беттс Дж. Анализ прогрессирующего разрушения пластин из волокнисто-армированных композитов, подверженных изгибу из плоскости. Механика композитных материалов. 2013. Т. 49, № 6. С. 911–932.
3. Chen X., Li Z., and Wang H. Progressive failure analysis of an open-hole composite laminate by using the s-version finite-element method . Mechanics of Composite Materials. 2014. V. 50, No. 3. P. 279–294.
4. Чернякин С. А., Скворцов Ю. В. Анализ роста расслоений в композитных конструкциях // Вестн. Сиб. гос. аэрокосм. ун-та им. акад. Решетнева. 2014. Вып. 4(56). С. 249–258.
5. Shroff S., Kassapoglou C. Progressive failure modelling of impacted composite panels under compression // Journal of Reinforced Plastics and Composites. 2015. V. 34, No. 19. P. 1603–1614.
6. Козлов М. В., Шешенин С. В. Моделирование прогрессирующего разрушения слоистых композитов. Механика композитных материалов. 2015. Т. 51, № 6. С. 991–1006.
7. Абросимов Н. А., Елесин А. В. Численное моделирование нелинейного деформирования и прогрессирующего разрушения композитных цилиндрических оболочек при неосесимметричных взрывных воздействиях. Механика композитных материалов. 2016. Т. 52, № 3. С. 547–562.
8. Беклемышева К. А., Васюков А. В., Ермаков А. С., Петров И. Б. Численное моделирование при помощи сеточно-характеристического метода разрушения композиционных материалов // Матем. моделирование, – 28:2 (2016), С. 97–110.
9. LS-DYNA® Keyword User's Manual Volume II Material Models March 26, 2012 (revision: 1275) Version 971 (LS-DYNA_Manual_Volume_II_ R10.0.pdf) С. 879–886.
10. LS-DYNA® Keyword User's Manual Volume II Material Models LS-DYNA R10.0 10/16/17 (r:9024). С. 1114–1122. http://ftp.lstc.com/anonymous/outgoing/jday/manuals/LS-DYNA_Manual_Volume_II_R10.0.pdf.
11. Циберев К. В., Авдеев П. А., Артамонов М. В. и др. Пакет программ ЛОГОС. Обзор текущих возможностей для решения задач прочности // XIV международный семинар «Супервычисления и математическое моделирование»: Сборник научных трудов. – Саров. 2013. С. 599–609.
12. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В и др. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов 2018. Вып.1. С. 3–14.
13. Дьянов Д. Ю., Казанцев А. В., Циберев К. В и др. Пакет программ ЛОГОС. Функциональные возможности для решения динамических задач прочности // Супервычисления и математическое моделирование. Труды XVII Международной конференции / Под ред. Р. М. Шагалиева. – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2019. С. 210–219.

14. Абросимов Н. А., Баженов В. Г. Нелинейные задачи динамики композитных конструкций. Н. Новгород: Изд-во ННГУ, 2002.

15 Скудра А. М., Булавс Ф. Я. Прочность армированных пластиков. – М.: Химия, 1982.

16. Сергеичев И. В., Антонов Ф. К., Сафонов А. А., Ушаков А. Е. Оценка остаточной прочности элементов композитных конструкций после низкоскоростного удара // Проблемы машиностроения и надежности машин. 2013. № 1. С. 36–44.

PROGRESSIVE FRACTURE SIMULATION OF ORTHOTROPIC MATERIALS UNDER DYNAMIC LOADING IN LOGOS SOFTWARE PACKAGE

N. A. Abrosimov, D. Yu. Dyanov, A. V. Elesin, A. Yu. Konstantinov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

Composite materials are more and more widely used when developing modern structures of different purpose that undergo intensive pulse loading. And one of the basic problems here is selection of a computational model to evaluate the strength of the structural element with the account for several strain factors: complex stress-and strain state, the effect of progressing fracture, non-uniformities in the stress and strain fields, strain non-stationarity, the effect of strain rates etc.

The solution of this extensive problem is complicated by the fact that the destruction of composite materials is a multi-factor and multi-stage process that is realized at different scale and structure levels of composite materials. Besides, a characteristic specific feature of composite material structures lies in the fact that both the material and the structure are created at the same time – within one technological process. So, the development of strain and destruction models for composite materials is an urgent problem.

Different models of progressive fracture of orthotropic materials are used for computations on stress-and-strain state and evaluation of dynamic strength of structural elements of composite materials [1–8]. The goal of the present research is the development of 3D model of progressive fracture of orthotropic materials under dynamic loading (analogous to MAT_ORTHOTROPIC_SIMPLIFIED_DAMAGE LS-DYNA commercial software package [9, 10]). The developed model was introduced using the possibility to realize a user model of the material in LOGOS software package [11–13].

Key words: *simulation, orthotropic materials, LOGOS, composite materials.*

УДК 539.3
DOI 10.53403/9785951505071_2022_27

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ПРОЦЕССОВ УДАРНОГО ВЗАИМОДЕЙСТВИЯ ЛЬДА С УПРУГОПЛАСТИЧЕСКОЙ ПРЕГРАДОЙ С ИСПОЛЬЗОВАНИЕМ МНОГОСЕТОЧНЫХ АЛГОРИТМОВ И СХЕМЫ ГОДУНОВА ПОВЫШЕННОЙ ТОЧНОСТИ

М. Х. Абузяров, Е. Г. Глазова, А. В. Кочетков, С. В. Крылов

Нижегородский государственный университет им. Н. И. Лобачевского,
Нижний Новгород

Излагается численная методика решения трехмерных связанных задач взаимодействия ледяных осколков с упругопластическими преградами. Методика построена на основе модифицированного разностного метода С. К. Годунова повышенной точности для расчета динамического деформирования упругопластических сред и многосеточных алгоритмов. Методика реализует эйлерово-лагранжев подход с явным выделением подвижных контактных поверхностей льда и элементов конструкций. Приводятся результаты численных исследований процессов соударения ледяных осколков сферической и цилиндрической формы с защемленной титановой пластиной.

Ключевые слова: численное моделирование, ледяные ударники, схема Годунова, повышенная точность, многосеточный подход, трехмерная задача, упругопластическая преграда, взаимодействие.

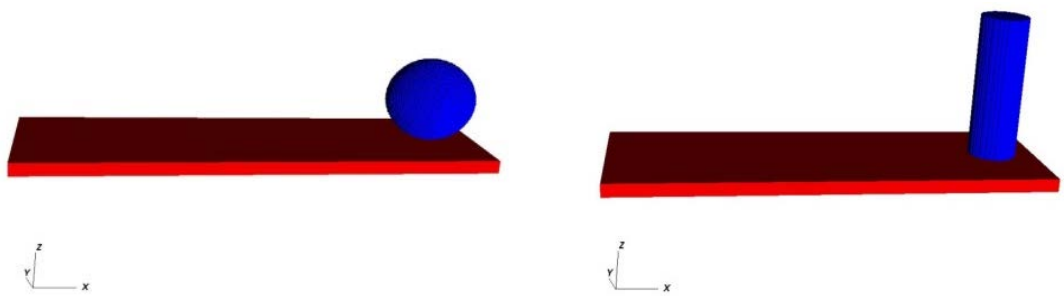
Введение

Исследования процессов динамического взаимодействия осколков льда с элементами конструкций активно развиваются в настоящее время, как зарубежом, так и в нашей стране. Их актуальность связана прежде всего с необходимостью обеспечения безопасной эксплуатации технических объектов в воздухе (авиация), в космосе (спутники и космические станции). При проектировании таких объектов требуются оценки воздействия ударных нагрузок от ледяных осколков различной формы. Актуальны расчетные методы анализа этих процессов, так как эксперименты дороги и в ряде случаев невозможны. Процессы ударного взаимодействия конструкций с ледяными осколками сопровождаются большими деформациями и перемещениями. В силу значительной разнородности взаимодействующих сред при численном моделировании важнейшее значение имеет описание процессов их совместного движения с корректным отслеживанием контактных границ. В настоящее время существует два подхода для описания пространственного движения контактных границ. Первый подход [1–3] включает в себя «точное» (Sharp Interface Method – SIM) выделение и отслеживание движения поверхности контакта, часто с использованием техники подсеточного разбиения (Adaptive Mesh Refinement - AMR) для лучшего разрешения. Этот подход был успешно применен для решения двумерных задач, но в трехмерном случае вызывает значительные трудности, связанные с отслеживанием и восстановлением поверхности контакта. Второй подход [4–7], применяющийся на эйлеровых сетках, не предполагает «точного» выделения контактной поверхности и допускает использование ячеек, содержащих смесь различных сред (Diffusive Interface Method – DIM). При этом подходе приходится конструировать искусственное (нефизичное) уравнение со-

стояния для смеси. Поверхность контакта в этом подходе явно не определяется. Этот подход удобен для решения трехмерных задач, но он имеет значительную численную вязкость и не обладает необходимой точностью при описании сложных контактных явлений, таких как трение, отрыв и т. д. В вышеупомянутых работах для моделирования ударных и волновых процессов используются схемы типа Годунова, что позволяет проводить вычисления как в эйлеровых, так и в лагранжевых координатах, основываясь на использовании точного решения задачи распада разрыва. Основным недостатком оригинального метода Годунова является первый порядок точности. В данной работе для моделирования трехмерных процессов в деформируемом теле используется модифицированный метод С. К. Годунова [8–13] основанный на интегрировании законов сохранения, имеющий второй порядок аппроксимации на гладких решениях и монотонный на разрывных. Повышение точности достигается только за счет модификации решения задачи распада разрыва на шаге «предиктор» схемы путем сближения областей влияния разностной и дифференциальной задач. Пластическое поведение твердых тел учитывается на этапе корректора схемы в соответствии с методом расщепления, предложенным и обоснованным в [14] для схем типа предиктор-корректор. В работе [15] предложен многосеточный SIM-подход на этой модификации схемы Годунова, использующий три типа сеток. Первый тип используемых сеток – это лагранжева поверхностная сетка в виде непрерывного набора треугольников (файл STL), которая применяется как для задания начальной геометрии объекта, так и для его сопровождения в процессе расчета, и два вида объемных трехмерных сеток. Это базовая декартова фиксированная сетка для каждого объекта, и вспомогательные подвижные локальные эйлерово-лагранжевы сетки, связанные с каждым треугольником поверхностной сетки. Количество треугольников поверхностной сетки и ячеек базовой декартовой сетки в процессе счета может меняться. Физические величины в этих сетках связаны взаимной интерполяцией. Подробное описание методики приведено в [15].

Математическая постановка задачи

Постановка задачи приведена на рис. 1. На защемленную с левого торца титановую пластину ударяется ледяной шар (рис. 1,а) и цилиндр (рис. 1,б). Требуется описать в связанной постановке процессы взаимодействия ледяных ударников с упругопластической деформируемой пластиной.



а б

Рис. 1. Постановка задачи

Для моделирования используются уравнения динамики сплошных сред в виде [13, 14]:

$$\frac{\partial}{\partial t}\mathbf{u} + \frac{\partial}{\partial x}\mathbf{f} + \frac{\partial}{\partial y}\mathbf{g} + \frac{\partial}{\partial z}\mathbf{h} = \mathbf{k} \qquad (1)$$

где $\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \\ s_{xx} \\ s_{yy} \\ s_{zz} \\ s_{xy} \\ s_{xz} \\ s_{yz} \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p - s_{xx} \\ \rho uv - s_{xy} \\ \rho uw - s_{xz} \\ (e + p - s_{xx})u - s_{xy}v - s_{xz}w \\ u\left(s_{xx} - \frac{4}{3}\mu\right) \\ u\left(s_{yy} + \frac{2}{3}\mu\right) \\ u\left(s_{zz} + \frac{2}{3}\mu\right) \\ us_{xy} - \mu v \\ us_{xz} - \mu w \\ us_{yz} \end{bmatrix}$,

$\mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv - s_{xy} \\ \rho v^2 + p - s_{yy} \\ \rho vw - s_{yz} \\ (e + p - s_{yy})v - s_{xy}u - s_{yz}w \\ v\left(s_{xx} + \frac{2}{3}\mu\right) \\ v\left(s_{yy} - \frac{4}{3}\mu\right) \\ v\left(s_{zz} + \frac{2}{3}\mu\right) \\ vs_{xy} - \mu u \\ vs_{xz} \\ vs_{yz} - \mu w \end{bmatrix}$, $\mathbf{h} = \begin{bmatrix} \rho w \\ \rho uw - s_{xz} \\ \rho vw - s_{yz} \\ \rho w^2 + p - s_{zz} \\ (e + p - s_{zz})w - s_{yz}v - s_{xz}u \\ w\left(s_{xx} + \frac{2}{3}\mu\right) \\ w\left(s_{yy} + \frac{2}{3}\mu\right) \\ w\left(s_{zz} - \frac{4}{3}\mu\right) \\ ws_{xy} \\ ws_{xz} - \mu u \\ ws_{yz} - \mu v \end{bmatrix}$,

$\mathbf{k}$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ s_{xx}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + s_{xy}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + s_{xz}\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) - \lambda s_{xx} \\ s_{xy}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) + s_{yy}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + s_{yz}\left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right) - \lambda s_{yy} \\ s_{xz}\left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right) + s_{yz}\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) + s_{zz}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) - \lambda s_{zz} \\ s_{xy}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) - \frac{1}{2}(s_{xx} - s_{yy})\left(\frac{\partial u}{\partial y} - \frac{\partial w}{\partial x}\right) + \frac{1}{2}s_{xz}\left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right) + \frac{1}{2}s_{yz}\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) - \lambda s_{xy} \\ s_{xz}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) - \frac{1}{2}(s_{xx} - s_{zz})\left(\frac{\partial u}{\partial y} - \frac{\partial w}{\partial x}\right) + \frac{1}{2}s_{xy}\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) + \frac{1}{2}s_{yz}\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) - \lambda s_{xy} \\ s_{yz}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) - \frac{1}{2}(s_{yy} - s_{zz})\left(\frac{\partial u}{\partial y} - \frac{\partial w}{\partial x}\right) + \frac{1}{2}s_{xz}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) + \frac{1}{2}s_{xy}\left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right) - \lambda s_{yz} \end{bmatrix}$$

Здесь $e = \rho(\varepsilon + 0.5(u^2 + v^2 + w^2))$. Система (1) замыкается уравнением состояния (УРС) в форме

$$\varepsilon = \varepsilon(p, \rho) \qquad (2)$$

В соотношениях (1), (2): p – давление, ρ – плотность, u, v, w – компоненты скоростей по x, y и z; ε – внутренняя энергия единицы массы, e – полная механическая энергия единицы объема сплошной среды, s_{xx}, s_{yy}, s_{zz}, s_{xy}, s_{xz}, s_{yz} – компоненты девиатора тензора истинных напряжений Эйлера, μ – модуль сдвига. Первые пять уравнений системы (1) представляют собой законы сохранения массы, импульса и энергии. Следующие шесть уравнений – физические соотношения упругости и пластичности с учетом поворота тензора напряжений в эйлеровых координатах (производная Яумана), записанные в дифференциальной форме. К системе (1), (2) добавляются начальные и краевые условия. В случае отсутствия сдвиговых напряжений система (1) переходит в уравнения Эйлера для движения сжимаемого газа [3]. Для описания процессов в плотных сжимаемых средах (металл) применяется баротропное УРС вида $p = p(\rho)$, позволяющее избежать интегрирования уравнения сохранения энергии. Для металлической пластины плотностью ρ_0 это будет УРС идеального упругопластического тела $p = \mathbf{K}\varepsilon_v$, где $\mathbf{K}$ – модуль объемного сжатия, $\varepsilon_v = 1 - \rho_0/\rho$ – объемная деформация. Критерием перехода из упругого напряженно-деформированного состояния в пластическое является условие текучести Мизеса: $J2 = 1/2\, S_{ij} S_{ij} \geq 1/3\, \sigma_T{}^2$, где J2 – второй инвариант девиатора тензора напряжений S_{ij}, σ_T – предел текучести. В соответствии с [14] в этом случае происходит коррекция компонент девиатора напряжений умножением на $\lambda = \sigma_T / \sqrt{3J_2}$.

Интегральная форма системы (1), на базе которой строится разностная схема, имеет вид

$$\oiint_{\omega} \mathbf{u}dxdydz + \mathbf{f}dydzdt + \mathbf{g}dxdzdt + \mathbf{h}dxdydt = \int\int\int_{\Omega}\int \mathbf{k}dxdydzdt, \qquad (3)$$

где Ω – любой замкнутый объем, поверхность ω которого гомеоморфна сфере в четырехмерном пространстве (x, y, z, t). На границах контакта льда с пластиной ставится условие непроникания. В начальный момент времени все среды покоятся, напряжения и деформации отсутствуют, $p_0 = 0{,}1$ МПа.

Метод численного решения

Решение уравнений (1) – (3) производится методом Годунова повышенной точности [8–12], единым, как для газодинамических, так и упругопластических течений, модифицированного для решения задач динамики сжимаемых сред с необратимыми сдвиговыми деформациями на эйлерово-лагранжевых сетках. Данная модификация позволяет повысить порядок аппроксимации схемы до второго на гладких решениях, сохранив монотонность на разрывных без изменения разностного шаблона явной двухшаговой схемы Годунова. При этом используется новая реализация шага «предиктор».

Модификация задачи распада разрыва в деформируемом твердом теле

В [14] Кукуджановым В. Н. было показано, что для численного моделирования динамических упругопластических уравнений со вторым порядком аппроксимации для схем типа «предиктор-корректор» на этапе «предиктор» достаточно решения упругих уравнений со вторым порядком аппроксимации с использованием линеаризованных уравнений (1)–(2). При этом учет пластического поведения среды происходит на этапе «корректор» после интегрирования линеаризованных уравнений и сводится к «посадке» девиаторов на поверхность текучести [14]. В соответствии с этим решение задачи распада разрыва проводится на основе упрощенной системы уравнений в упругом приближении, получаемой из (1), если положить λ = 0. Исходную систему (1), (2) линеаризуем в окрестности некоторого состояния и предполагаем, что движение происходит вдоль оси x и все неизвестные функции зависят только от x. Получим систему из одиннадцати уравнений. После введения автомодельной переменной $\xi = x/t$ система преобразуется к системе обыкновенных дифференциальных уравнений [15]. Нетривиальные решения этой системы $\xi_1 = u + a$, $\xi_2 = u - a$,

$\xi_3 = u + \beta_y$, $\xi_4 = u - \beta_y$, $\xi_5 = u + \beta_z$, $\xi_6 = u - \beta_z$, $\xi_7 = u$, $\xi_8 = u$, $\xi_9 = u$, $\xi_{10} = u$, $\xi_{11} = u$ представляют собой скорости разрывов, образующих устойчивую конфигурацию (рис. 2).

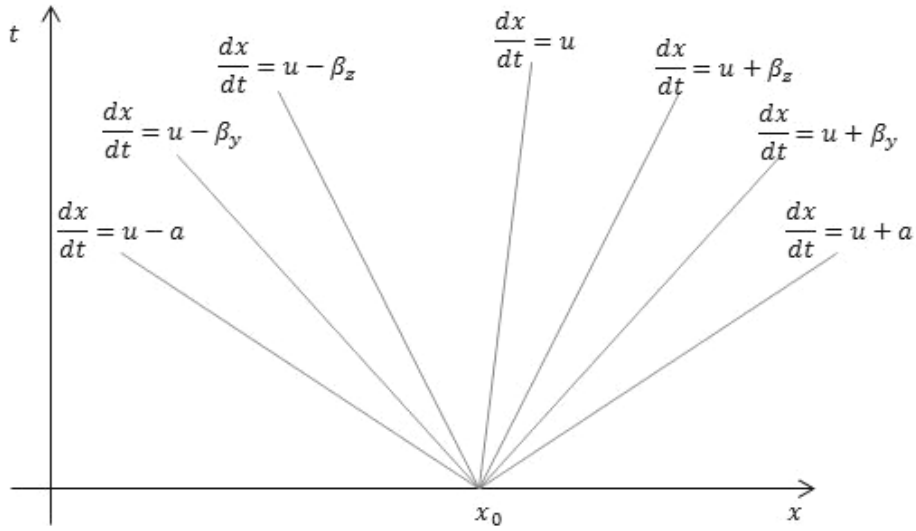


Рис. 2. Конфигурация разрывов для упругого распада разрыва

Здесь $a^2 = c^2 + (4/3\,\mu - fS_{xx})/\rho$, $c^2 = (\partial p/\partial \rho)_s$, $f = 1/\rho(\partial \varepsilon/\partial p)_\rho$,

$$\beta_y = \sqrt{(\mu + 3/4S_{xx})/\rho - 0.5\sqrt{\left(0.25 * (S_{yy} - S_{zz})^2 + S_{yz}^2\right)/\rho^2}},$$

$$\beta_z = \sqrt{(\mu + 3/4S_{xx})/\rho + 0.5\sqrt{\left(0.25 * (S_{yy} - S_{zz})^2 + S_{yz}^2\right)/\rho^2}},$$

В плоскости (x, t) траектории разрывов (характеристики) изображаются лучами, исходящими из точки $x = x_0$, и делят полуплоскость $t > 0$ на 8 зон (рис. 2).

Соотношения на этих характеристиках R_l, $l = 1, 11$ имеют следующий вид

$$R_1 = [\rho a]u - \left[a(1+f)\frac{S_{xy}(a^2 - b_3^2) - S_{xz} * 0.5S_{yz}/\rho}{(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2/\rho^2}\right]v - \left[a(1+f)\frac{S_{xz}(a^2 - b_2^2) - S_{xy} * 0.5S_{yz}/\rho}{(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2/\rho^2}\right]w +$$

$$+p - S_{xx} + \left[(1+f)\frac{\rho S_{xy}(a^2 - b_3^2) - S_{xz} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]S_{xy} + \left[(1+f)\frac{\rho S_{xz}(a^2 - b_2^2) - S_{xy} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]S_{xz}$$

$$b_2^2 = \left(\mu + 0.5(S_{xx} - S_{yy})\right)/\rho, \qquad b_3^2 = \left(\mu + 0.5(S_{xx} - S_{zz})\right)/\rho,$$

$$R_2 = [-\rho a]u + \left[\rho a(1+f)\frac{\rho S_{xy}(a^2 - b_3^2) - S_{xz} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]v + \left[\rho a(1+f)\frac{\rho S_{xz}(a^2 - b_2^2) - S_{xy} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]w +$$

$$+p - S_{xx} + \left[(1+f)\frac{\rho S_{xy}(a^2 - b_3^2) - S_{xz} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]S_{xy} + \left[(1+f)\frac{\rho S_{xz}(a^2 - b_2^2) - S_{xy} * 0.5S_{yz}}{\rho^2(a^2 - b_2^2)(a^2 - b_3^2) - 0.25S_{yz}^2}\right]S_{xz}$$

$$R_3 = [\beta_y \rho]v - [\beta_y \rho C]w - S_{xy} + [C]S_{xz}$$

$$R_4 = -[\beta_y \rho]v + [\beta_y \rho C]w - S_{xy} + [C]S_{xz}$$

$$R_5 = [\beta_z \rho C]v - [\beta_z \rho]w - [C]S_{xy} + S_{xz}$$

$$C = \sqrt{1 - 2\Big/\left(1 + \sqrt{1 + 4S_{yz}^2/\left(S_{yy} - S_{zz}\right)^2}\right)}, \quad 0 \le C \le 1$$

$$R_6 = -[\beta_z \rho C]v + [\beta_z \rho]w - [C]S_{xy} + S_{xz}$$

$$R_7 = \left[a^2\right]\rho - p + S_{xx} + \left[\frac{(1+f)(S_{xy}\rho b_3^2 + 0.5S_{yz}S_{xz})}{\rho b_2^2 \rho b_3^2 - 0.25S_{yx}^2}\right]S_{xy} + \left[\frac{(1+f)(\rho b_2^2 S_{xz} + 0.5S_{yz}S_{xy})}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xz}$$

$$R_8 = \left[4/3\frac{\mu}{\rho}\right]\rho + S_{xx} + \left[\frac{S_{xy}\rho b_3^2 + 0.5S_{yz}S_{xz}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xy} + \left[\frac{\rho b_2^2 S_{xz} + 0.5S_{yz}S_{xy}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xz}$$

$$R_9 = \left[-2/3\frac{\mu}{\rho}\right]\rho + S_{yy} + \left[\frac{-S_{xy}\rho b_3^2}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xy} + \left[\frac{-0.5S_{yz}S_{xy}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xz}$$

$$R_{10} = \left[-2/3\frac{\mu}{\rho}\right]\rho + S_{zz} + \left[\frac{-0.5S_{yz}S_{xz}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xy} + \left[\frac{-\rho b_2^2 S_{xz}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xz}$$

$$R_{11} = \left[\frac{-0.5S_{xz}\rho b_3^2 - 0.25S_{yz}S_{xy}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xy} + \left[\frac{-0.5\rho b_2^2 S_{xy} - 0.25S_{yz}S_{xz}}{\rho b_2^2 \rho b_3^2 - 0.25S_{yz}^2}\right]S_{xz} + S_{yz}$$

Здесь величины в квадратных скобках представляют собой коэффициенты линеаризации, определяемые по усредненным параметрам в ячейках. Для получения схемы второго порядка аппроксимации из центров ячеек линейно интерполируются инварианты Римана [11, 12]. Координаты точек интерполяции определяются как точки, из которых приходят соответствующие инварианты на положение грани в момент времени $\Delta t/2$, где w скорость грани (обозначена пунктиром на рис. 3), следующим образом:

$$x_n = (x_{i-1} + x_i)/2 - (c_n - w)\,\Delta t/2, \quad n = 1, \ldots, 11,$$

где $c_1 = u + a; c_2 = u - a; c_3 = u + \beta_y; c_4 = u - \beta_y; c_5 = u + \beta_z; c_6 = u - \beta_z; c_7 = c_8 = c_9 = c_{10} = c_{11} = u;$

Обозначим интерполированные инварианты индексом ‘m’, соответственно

$$R_n^m = R_n^{i-1} + \left(R_n^i - R_n^{i-1}\right)(x_n - x_{i-1})/(x_i - x_{i-1}), n = 1, \ldots, 11.$$

Полученные значения инвариантов используются для определения «распадных» и «потоковых» значений, в зависимости от того, в какую зону попадает грань ячейки (рис. 3). Этап численного интегрирования уравнений (этап корректор) остается неизменным [8] .

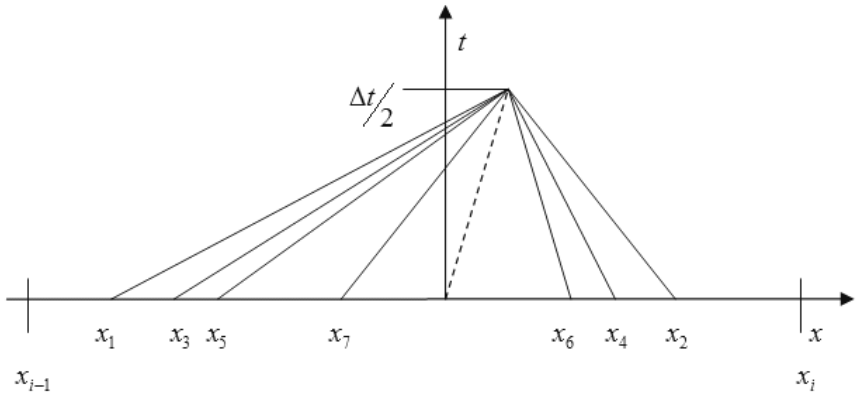


Рис. 3. Координаты приходящих инвариантов для грани интегрируемой ячейки на момент $\Delta t/2$

Монотонность на разрывных решениях

Решение по схеме второго порядка будет испытывать дисперсионные колебания на разрывах. Чтобы обеспечить монотонность, необходимо строить решение гибридным образом – в областях гладкости по соотношениям, обеспечивающим второй порядок аппроксимации, а на разрывах – по соотношениям схемы первого порядка [8]. Критерии оценки области перехода для газов изложены в [8–10]. Для упругопластических течений для получения монотонных решений достаточно анализа поля нормальных напряжений [11, 12]. Критерии перехода основаны на анализе квадратичных сплайнов. Например, при расчете задачи распада разрыва на ребре $x_{i-1/2}$ строится левый квадратичный сплайн по давлениям p_{i-2}, p_{i-1}, p_i и правый по давлениям p_{i-1}, p_i, p_{i+1}. Несмотря на то, что численная функция монотонна, сплайны могут быть как монотонными, так и иметь экстремумы. В случае монотонных сплайнов применяется схема второго порядка. Если левый сплайн имеет максимум, и этот максимум расположен между центрами ячеек с координатами x_{i-2} и x_i, или правый сплайн имеет максимум, расположенный между центрами ячеек с координатами x_{i-1} и x_{i+1}, то для решения задачи распада разрыва берутся параметры из центров ячеек и определяется решение этой задачи как в классической схеме Годунова. Можно ослабить этот критерий и брать параметры из центров только слева или справа, анализируя положение соответствующего экстремума. При этом можно добиться размазывания на разрыве на 2–3 ячейки.

Многосеточный алгоритм для расчета контактных задач

Широко используемые в решении двумерных задач взаимодействия сред и конструкций эйлерово-лагранжевые подходы, основанные на использовании подвижных криволинейных сеток для описания движения лагранжевых контактных границ с соответствующими перестройками эйлеровых сеток внутри однородной области [16, 17], оказались, практически непригодными для решения трехмерных задач этого класса. Причинами этого является сложности построения трехмерных подвижных эйлерово-лагранжевых сеток, связанных с деформируемыми контактными лагранжевыми поверхностями. Кроме того большое влияние на точность решения оказывает точность вычисления интегралов по подвижным пространственно-временным ячейкам, особенно для деформируемых твердых тел. Поэтому в данной работе используется эйлерово-лагранжевый подход на сетках типа «химера» [15, 18]. Подход является многосеточным и использует три типа расчетных сеток [15]. Первый тип сеток – лагранжевые сетки в виде STL файлов, задающие и сопровождающие деформирующиеся поверхности тел. Внутри однородных областей используются неподвижные регулярные эйлеровые сетки с кубическими ячейками. Третий тип сеток – вспомогательные локальные подвижные эйлерово-лагранжевые сетки, связанные с поверхностями тел. В целом алгоритм расчета контактного взаимодействия сред и конструкций состоит из последовательности следующих шагов.

1) Конструкции и среды задаются в виде поверхностей из наборов треугольников с необходимой точностью – в виде файлов STL формата, содержащих внешние нормали и координаты вершин треугольников. На рис. 4,а и б красным и синим цветами отмечены такие объекты, на рис. 4,б приведен случай контакта двух объектов.

2) Каждая расчетная область (газ, твердое тело) с криволинейными границами заключается в окаймляющий прямоугольный параллелепипед и покрывается декартовой сеткой. На рис. 4,в приведено сечение такой расчетной области (черная сплошная кривая) с окаймляющим параллелепипедом. Получаем четыре вида ячеек для расчетной области, первый вид – ячейки, секущиеся треугольниками поверхности или граничные ячейки, окрашены зеленым, второй вид – ячейки снаружи поверхности, третий и четвертый вид – ячейки внутри поверхности, для интегрирования которых хватает (отмечены коричневым) или не хватает (отмечены черными точками) разностного шаблона из целых ячеек, находящихся внутри поверхности.

3) На каждом треугольнике поверхности строится вспомогательная локальная декартова сетка 3×3×3 внутрь объема по нормали от этой поверхности, на рис. 4,а эти сетки коричневого цвета. Размеры ячеек этой локальной трехмерной сетки берутся близкими к размерам ячеек внутренней

сетки. В случае контакта с другой подобластью или граничных условий, требующих дополнительных параметров, локальная сетка симметрично достраивается и в эту подобласть от плоскости треугольника, на рис. 4,а и б эта достроенная сетка отмечена зеленым. На рис. 4,г эта вспомогательная сетка приведена для одного треугольника. Значения параметров в построенной локальной сетке определяются интерполяцией параметров с предыдущей локальной сетки и параметров из основной сетки. Данного шаблона достаточно для интегрирования центральных ячеек (на рис. 4,г выделено темным и на рис. 4,д эти ячейки приведены отдельно и помечены крестиками), примыкающих к лагранжевой контактной поверхности, с повышенной точностью по модифицированной схеме С. К. Годунова. Для центральных ячеек решается задача распада разрыва на контактных границах сред, эти ячейки и соответствующий треугольник приведены на рис. 4,ж. Результатом ее решения являются скорости и силы на половинном временном слое в центре этого треугольника. С нормальной скоростью двигаем эту контактную границу и получаем локальную сетку на новом временном слое. На рис. 4,д и е приведены эти центральные ячейки до и после подвижки контактной границы. Проводим стандартное интегрирование параметров этих центральных ячеек в подвижных сетках. Так как движение локальной сетки одномерное, объемные и поверхностные интегралы по ячейкам при этом считаются точно.

4) Используя скорости в центре каждого треугольника, полученные из решения задачи распада разрыва на этапе 3, вычисляем скорости в вершинах треугольников STL файла с весами, пропорциональными площадям треугольников. С этими скоростями двигаем вершины и получаем положение поверхности на новом временном слое (новое положение STL файла).

5) Производим перестройку окаймляющего параллелепипеда с возможным добавлением или уменьшением слоев ячеек в соответствии с новым положением STL файла. Сечение этой поверхности с окаймляющим параллелепипедом приведено на рис. 4,з. Красная кривая соответствует положению расчетной области на новом временном слое. К ячейкам четвертого типа, оставшимся в расчетной области на новом временном слое (отмечены крестиками на белом фоне), добавляем непроинтегрированные ячейки, захваченные при подвижке поверхности (отмечено крестиками на зеленом фоне).

6) В ячейки четвертого вида (помеченные крестиками на белом и зеленом фоне) интерполируем параметры из проинтегрированных ячеек третьего вида и из проинтегрированных ячеек локальных сеток, таким образом завершив расчетный шаг.

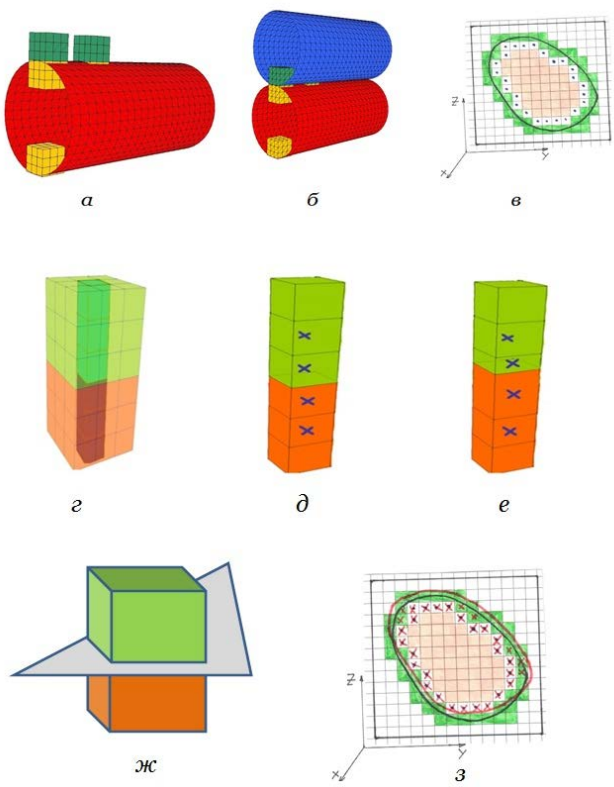


Рис. 4. Лагранжева поверхностная и вспомогательная эйлерово-лагранжева локальная сетка и центральные ячейки локальной сетки до и после подвижки контактной границы и интегрирования

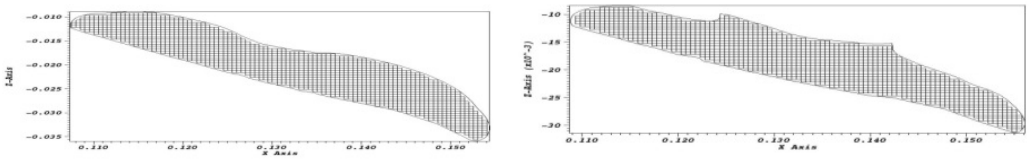
Результаты численного исследования взаимодействия ледяных ударников с защемленной титановой пластиной

Постановка трехмерной задачи показана на рис. 1,а – удар шара и 1,б – удар цилиндра. Размеры пластины от начала координат: длина по оси X 150 мм, ширина по оси Y 50 мм, высота по оси Z 5 мм, пластина жестко закреплена в плоскости X = 0, материал – титан ВТ6 с механическими характеристиками: плотность 4.51г/см^3, модуль объемной деформации; $1.12 \cdot 10^5$ МПа, модуль сдвига $4.1 \cdot 10^4$ МПа, предел текучести 1450 МПа. В момент удара координаты точки касания шара и оси симметрии цилиндра X = 134 мм, Y = 25 мм, Z = 5 мм. Ледяной шар – диаметр 28 мм, цилиндр – диаметр 17.1 мм, высота 50 мм, массы одинаковые 10.35 г, механические характеристики льда: плотность 0.9 г/см^3 , модуль объемной деформации 9196 МПа, модуль сдвига 3526.3 МПа, предел текучести 5.2 МПа. Начальная скорость ударников в обоих случая 350 м/с. На внешних границах металлической преграды и ледяных ударников выполняются условия как на «свободной границе» с заданным давлением p = 0.1 МПа. Размеры ячеек по льду и преграде брались 0.5 мм, что потребовало порядка шестьсот тысяч ячеек основной и трех миллионов ячеек вспомогательной сеток.



а б

Рис. 5. Формоизменение объектов 800 мкс после удара



а б

Рис. 6. Расчетная сетка для шарового и цилиндрического ударников, сечение вдоль оси X, 800 мкс после удара

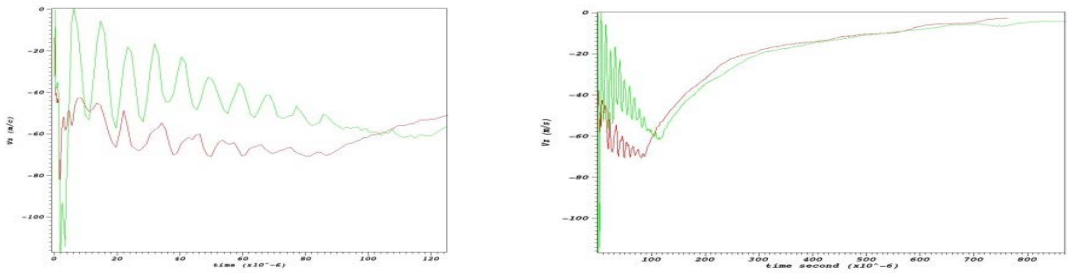


Рис. 7. Вертикальные скорости в точке удара от времени, шар - красным, цилиндр зеленым

В процессе удара градины и пластина претерпевают значительные перемещения и деформации, происходит разрушение градин. На рис. 5 и 6 приведены также положения объектов и расчетная сетка в продольной плоскости симметрии по шару и цилиндру на момент 800 мкс. На рис. 7 приведены скорости пластины от времени на оси симметрии шара, отмечено красным и цилиндра – зеленым, в левой части от начала взаимодействия до 120 мкс, в правой соответственно до 800 мкс. Различие в начальный момент связано с большей начальной площадью взаимодействия для цилиндрического объекта.

Заключение

Описанная в данной статье численная методика решения трехмерных связанных задач взаимодействия ледяных осколков с упругопластическими преградами позволяет получать результаты при решении нелинейных задач с большими перемещениями и деформациями. Использование STL файлов для построения всех типов сеток и описания движения лагранжевых контактных и граничных поверхностей в рамках данного многосеточного подхода, а также применение регулярных неподвижных эйлеровых сеток позволяет резко упростить подготовку начальных данных для решения сложных задач и повысить эффективность и точность расчетов за счет исключения искажений и перестроений расчетных сеток в традиционных эйлерово-лагранжевых методиках.

Работа выполнена при финансовой поддержке РФФИ проект №19-08-00320.

Литература

1.LS-DYNA, User's manual, http://www.lstc.com, 2020.

2.EUROPLEXUS, User's manual, http://www-epx.cea.fr/, 2020.

3.Nonlinear Finite Elements for Continua and Structures | Ted Belytschko, Wing Kam Liu, Brian Moran, Khalil I Elkhodary, Wiley, 2014.

4.DJ Benson. Computer methods in Applied mechanics and Engineering 99 (2-3), 235-394, 1992. 1528, 1992.

5. Miller G. H. and Colella P. A Conservative Three-Dimensional Eulerian Method for Coupled Solid???Fluid Shock Capturing, Journal of Computational Physics, vol.183, issue.1, pp.26-82, 2002.

6. Barton P. T., Obadia B., and Drikakis D. A conservative level-set based method for compressible solid/fluid problems on fixed grids, Journal of Computational Physics, vol.230, issue.21, pp.7867-7890, 2011.

7. Barton P. T., Deiterding R., Meiron D., and Pullin D. Eulerian adaptive finite difference method for impact and penetration problems, Journal of Computational Physics, vol. 240, issue.C 3, pp.76-99, 2013.

8. Годунов С. К., Забродин А. В., Иванов М. Я. и др. Численное решение многомерных задач газовой динамики. – М.: Наука, 1976. 400 с.

9. Абузяров М. Х., Баженов В. Г., Кочетков А. . О новом эффективном подходе к повышению точности схемы Годунова // Прикладные проблемы прочности и пластичности. Методы решения. Межвуз.сб. Горьк. ун-т. 1987. С. 43–49.

10. Абузяров М. Х., Баженов В. Г., Кочетков А. В. О монотонизации схемы Годунова второго порядка точности введением схемной вязкости // ППП. Исследование и оптимизация конструкций: Всесоюз. межвуз. сб. / Горький: Изд-во ГГУ. 1987. С. 85–90.

11. Абузяров М. Х. О повышении точности схемы Годунова для решения задач гидрогазодинамики // XIII конференция молодых ученых Московского физико-технического института. 1988. Т. 2. С. 30–37.

12. Abouziarov H. Aiso, An application of retroactive characteristic method to conservative scheme for structure problems (elastic-plastic flows) // Hyperbolic Problems, Theories, Numerics, Applications. Tenth International Conference in Osaka. September 2004, Copiright 2006 by Yokohama Publishers, Inc. P. 223–230.

13. Abouziarov M., Aiso H., Takahashi T. An application of conservative scheme to structure problems // Series from Research Institute of Mathematics of Kyoto University. Mathematical Analysis in Fluid and Gas Dynamics. 2004. № 1353. P. 192–201.

14. Кукуджанов В. Н. Метод расщепления упругопластических уравнений // МТТ. 2004. № 1. С. 98–108.

15. Abuziarov K. M., Abuziarov M. H., Kochetkov A. V. 3D fluid structure interaction problem solving method in euler variables based on the modified Godunov scheme. Materials Physics and Mechanics. 2016. Т. 28. № 1–2. С. 1–5.

16. Баженов В. Г., Зефиров С. В., Кочетков А. В., Крылов С. В., Фельдгун В. Р. Пакет программ «Динамика-2» для решения плоских и осесимметричных задач нестационарного взаимодействия конструкций со сжимаемыми средами // Мат. моделирование. 2000. Т. 12, № 6. С. 67–72.

17. Абузяров М. Х., Крылов С. В., Цветкова Е. В. Моделирование упругопластического взаимодействия с помощью программного комплекса UPSGOD. Проблемы прочности и пластичности. 2013. Вып. 75(1). С. 25–32.

18. Дерюгин, Саразов А. В., Жучков В. Н.. Особенности построения методики расчета на сетках типа «Химера» для неструктурированных сеток // Мат. Моделирование. 2017. Т 29, № 2. С. 106–118.

NUMERICAL SIMULATION OF THREE-DIMENSIONAL PROCESSES OF IMPACT INTERACTION OF ICE WITH ELASTOPLASTIC BARRIER USING MULTI-NETWORK ALGORITHMS AND GODUNOV'S SCHEME OF INCREASED ACCURACY

M. H. Abuziarov, E. G. Glazova, A. V. Kochetkov, S. V. Krylov

Nizhny Novgorod Lobachevsky State University, Nizhny Novgorod

A numerical technique for solving three-dimensional coupled problems of the interaction of ice fragments with elastoplastic barriers is presented. The technique is based on the modified finite volume method of S.K. Godunov of high accuracy for calculating the dynamic deformation of elastoplastic media and multigrid algorithms. The method implements the Eulerian-Lagrangian approach with an explicit identification of movable contact surfaces of ice and structural elements. The results of numerical studies of the processes of collision of spherical and cylindrical ice fragments with a clamped titanium plate are presented.

Key words: numerical modeling, ice strikers, Godunov's scheme, increased accuracy, multigrid approach, three-dimensional problem, elastoplastic barrier, interaction.

УДК 519.6
DOI 10.53403/9785951505071_2022_38

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ЗАДАЧ КОНВЕРСИИ ЛАЗЕРНОГО ИЗЛУЧЕНИЯ В РЕНТГЕНОВСКОЕ В СФЕРИЧЕСКОМ БОКСЕ МЕГАДЖОУЛЬНОЙ ЛАЗЕРНОЙ УСТАНОВКИ

А. В. Бабанов, С. В. Бондаренко, Е. И. Митрофанов, Л. Ф. Потапкина, О. О. Шаров

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Одним из ведущих направлений исследований по термоядерному синтезу является инерциальный синтез. Данный подход базируется на использовании инерции слоя

термоядерного топлива, предварительно сжатого и ускоренного до высоких скоростей в результате абляционного ускорения оболочечной мишени интенсивным излучением. Рассматривается численное моделирование задач по оптимизации конструкции криогенных мишеней непрямого облучения для получения зажигания, по подбору рентгеновского импульса, для определения динамики поглощения лазерного излучения и характеристики неоднородной генерации рентгеновского излучения на стенках бокса.

В данной работе приведены результаты трехмерных расчетов конверсии лазерного излучения в рентгеновское по методике МИМОЗА-НД3Д.

Ключевые слова: непрямое облучение мишени, сферический бокс-конвертор, лазерное излучение, рентгеновское излучение, пластиковый (CH) аблятор, HDC аблятор.

В данной работе рассматривается трехмерная методика МИМОЗА-НД3Д[1], которая позволяет проводить расчеты движения горячей излучающей плазмы совместно с расчетом переноса излучения в приближении спектральной многогрупповой диффузии и кинетикой ионизации. Рассмотрены основные моменты организации параллельных вычислений. Приведены результаты трехмерных расчетов по конверсии лазерного излучения в рентгеновское в рамках обоснования выбора мишеней, которые предполагается использовать при проведении опытов на лазерной установке [2].

Список моделируемых процессов в трехмерной методике МИМОЗА-НД3Д:

- газодинамическое движение плазмы в многотемпературном приближении,
- поглощение и рассеяние лазерного излучения,
- электронная и ионная теплопроводность,
- спектральный перенос рентгеновского излучения,
- обмен энергией между электронами, ионами и излучением,
- кинетика ионизации в приближении модели среднего иона [3].

Для ускорения времени счета задач использовались следующие методы распараллеливания численной методики МИМОЗА-НД3Д:

- метод трехмерной матричной декомпозиции,
- распараллеливание программы спектральной диффузии по группам спектра,
- мелко-блочное распараллеливание программы кинетики ионизации,
- распараллеливание программы расчета поглощения и рассеяния лазерного излучения.

Распараллеливание программы спектральной диффузии организовано таким образом, что группы спектра рассчитывались на разных процессорах. Предположим, что после геометрической декомпозиции задача считается на M процессорах, тогда в ней M-1 подобласть, так как один процессор управляющий. Затем для каждой подобласти добавлялось определенное количество дополнительных P-1 процессоров для того, чтобы распределить все спектральные группы между этими P процессорами подобласти. Количество всех процессоров становится N=(M-1)*P+1.

Для проверки правильности реализации и анализа эффективности распараллеливания была выбрана задача об обжатии стеклянной капсулы, заполненной водородом (рис. 1). На внешней поверхности капсулы задавалась температура излучения.

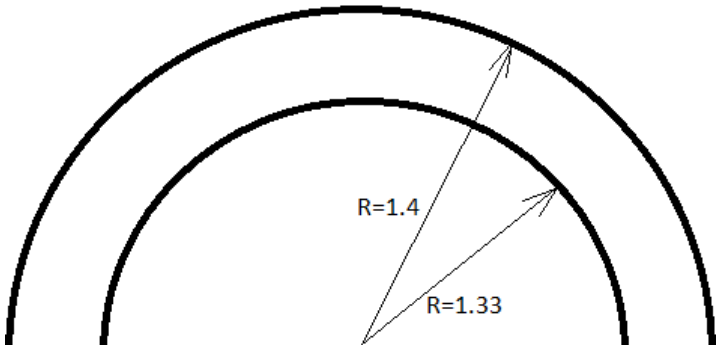


Рис. 1. Начальная геометрия тестовой задачи

Энергетический спектр разбивался равномерно на 46 групп. Пространственная сетка состояла из 90 столбцов, 90 строк и 90 листов (всего 729000 ячеек). Тестовые замеры проводились на момент фокусировки ударной волны в центре капсулы.

Проводилось 3 типа расчетов: 1 – только пространственная декомпозиция, 2 – только декомпозиция энергетического излучения на 2 слоя, 3 – пространственная декомпозиция и декомпозиция энергетического излучения.

Ускорение и эффективность распараллеливания вычислялись по следующим формулам:

$$Sp = \frac{t_N}{t_M}; \quad Eff = Sp \cdot \frac{N}{M} \cdot 100\%$$

где, t_N – время счета задачи на N процессорах, t_M – время счета задачи на M процессорах.

Наибольшее ускорение 1.8 и эффективность 89.84 получено в расчетах типа 3.

В программе кинетики ионизации расчет состояния плазмы осуществляется локально в каждой пространственной точке, поэтому здесь был применен метод мелко-блочного распараллеливания. Все точки подобласти распределялись между P процессорами.

Тестовые расчеты показали, что распараллеливание программ спектральной диффузии и кинетики ионизации не влияет на результаты счета, но позволяет уменьшить календарные сроки расчетов.

Для расчета лазерного энерговыделения использовался метод лучевых траекторий, известный в литературе также под названием «ray tracing». В этом методе начальное распределение интенсивности лазерного пучка разбивается на множество отдельных лучей, каждому из которых присваивается соответствующая доля энергии пучка.

Затем на каждом временном шаге рассчитывается траектория каждого луча внутри системы с учетом преломления и отражения от критических поверхностей.

Траекторию лазерного луча предсказать невозможно, луч неоднократно меняет процессор, поэтому потребовалась разработка алгоритмов для передачи энергии лазера через границу параобластей.

На каждом процессоре определялось количество лазерных лучей, их положение, затем рассчитывались траектории каждого луча до тех пор, пока ослабление из-за поглощения не составит величину 10^{-18} или все они не покинут параобласть. После того как все лучи покинут свои параобласти или ослабнут, определялось новое положение лучей и расчет продолжался до полного ослабления всех лучей или их выхода за пределы геометрии.

При проведении опытов на лазерной установке, при сжатии и генерации нейтронного выхода при непрямом лазерном облучении, предполагается в основном использовать сферический бокс с шестью отверстиями, предназначенными для ввода лазерного излучения. На стенках бокса и самой мишени будет происходить конверсия лазерного излучения в рентгеновское излучение.

Первые расчеты по программам параллельного варианта методики МИМОЗА-НД3Д проводились в замкнутой технологии, когда в качестве входного параметра задавались временная форма лазерного пучка и его энергия.

Бокс-конвертор представлял собой сферическую золотую оболочку толщиной 20 мкм и наружным диаметром 9 мм с 6-ю отверстиями, предназначенными для ввода лазерного излучения (рис. 2). Все 48 лазерных пучков (кластеров) имели одинаковую энергию и полная энергия лазерного излучения, введенного в бокс, была равна 1.4 МДж.

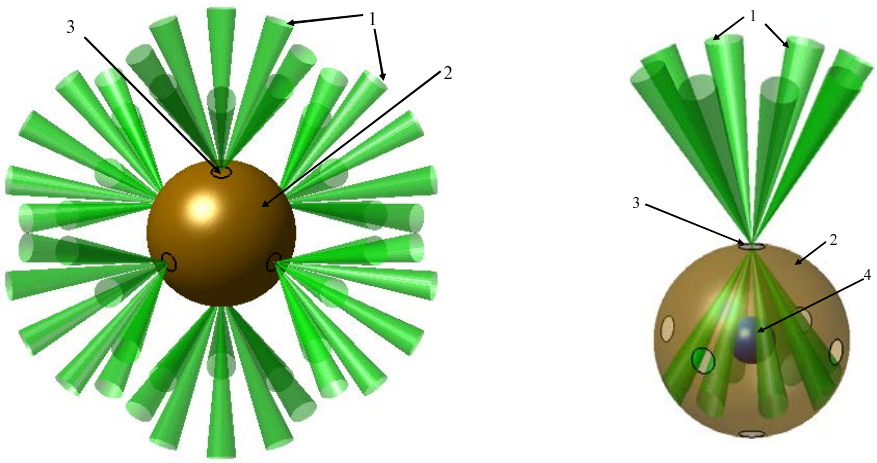


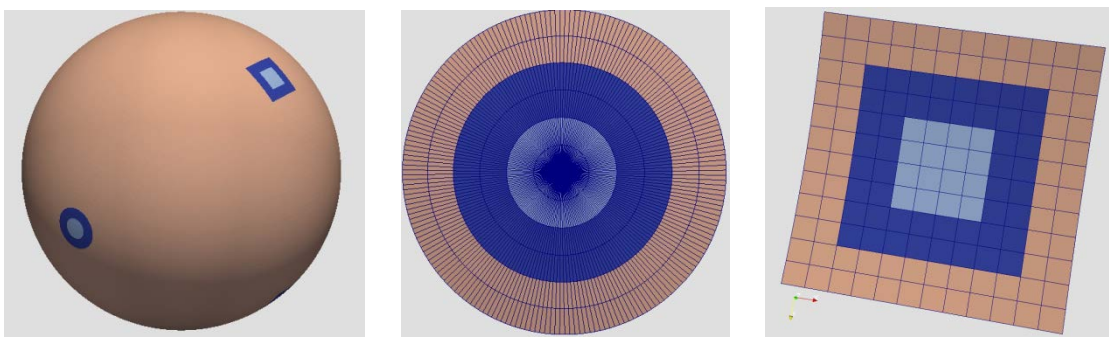
Рис. 2. Схема ввода лазерного излучения внутрь сферического бокса-конвертора: 1 – кластеры лазерных пучков, 2 – сферический бокс, 3 – отверстие для ввода ЛИ, 4 – капсула с термоядерным топливом

Временная форма мощности, подобранная в результате серии расчетов по одномерной методике СНДП [3], у всех лазерных пучков также была одинаковой и совпадала с временной формой мощности лазерного излучения в тестовых расчетах.

В трехмерных расчетах использовалась равномерная сетка, как по полярному θ, так и по азимутальному углу φ: по θ от 0 до 180° – 100 интервалов, по φ от 0 до 360° – 200 интервалов.

При построении сетки два отверстия для ввода лазерного излучения, центры которых лежали на полюсах, получались круглыми, а другие четыре отверстия, центры которых лежали на экваторе, были квадратными.

Геометрия отверстий в трехмерном расчете схематично представлена на рис. 3. Синим цветом обозначена область отверстия, голубым – область лазерного пятна в отверстии.



а б в

Рис. 3. Геометрия отверстий в трехмерном расчете: а – полная геометрия, б – область лазерного пятна в районе полюса, в – область лазерного пятна в районе экватора

Как видно из рис. 3, количество счетных ячеек в квадратных отверстиях не равно количеству ячеек в круглых отверстиях, поэтому круглым и квадратным отверстиям соответствовало разное количество лучей.

Проведенные трехмерные расчеты для полной геометрии имеют свои достоинства и недостатки. К положительным моментам расчетов в полной постановке можно отнести лучшую наглядность и представительность численных результатов, возможность проявления неочевидных, на первый взгляд, счетных эффектов, связанных с несовершенством используемых физических моделей или индивидуальных особенностей счетной методики.

Недостатками расчетов в полной постановке являются огромные вычислительные ресурсы и возможные неоднозначности с построением расчетной сетки и ее качеством на этапе счета задачи. Первые расчеты бокса с полной геометрией длились в среднем около 8 месяцев.

Проект мегаджоульной лазерной установки предусматривает схему облучения мишени в совокупности обладающую симметриями куба [4]. В этом случае размеры счетной области могут быть значительно уменьшены – в 48 раз, а, следовательно, и время счета задачи существенно сокращается.

Например, такая область может быть получена путем рассечения шара радиуса R тремя плоскостями: $z = 0$, $x = y$, $z = y$ (в качестве полярной оси выбрана ось X). В сферической системе координат неравенства, определяющие границы такой уменьшенной счетной области, имеют вид:

$$\begin{cases} 0 \le \varphi \le \dfrac{\pi}{4} \\ 0 \le \theta \le \arccos\left(\dfrac{1}{\sqrt{2 + \left(tg\varphi\right)^2}} \right), \\ 0 \le r \le R \end{cases} \qquad (1)$$

где θ – полярный угол, отсчитываемый от оси X, φ – азимутальный угол, отсчитываемый от оси Z.

Были проведены сравнительные трехмерные расчеты «1/48» для фрагмента бокса, в котором количество точек было уменьшено в 32 раза. Календарное время счета сократилось до трех недель.

По методике МИМОЗА-НДЗД был проведен трехмерный расчет конверсии лазерного излучения в рентгеновское в боксе для капсулы с аблятором из высокоплотного углерода (HDC – high-density carbon).

Счетная область была ограничена в соответствии с неравенствами (1), а ось отверстия для ввода лазерного излучения совпадала с полярной осью X.

На рис. 4,а, б приведена геометрия отверстия, начальная расстановка точек по полярному углу θ, направления осей и разностная сетка в трехмерной задаче на начальный момент времени.

На внешней границе счетной области задавались граничные условия в виде свободной границы как по спектральной диффузии, так и по газовой динамике. По электронной и ионной теплопроводности в качестве граничного условия использовалось условие равенства нулю потока тепла.

На плоскостях: $z = 0$, $x = y$, $z = y$ в качестве граничного условия по теплопроводности задавалась условие равенства нулю потока тепла, по газовой динамике задавалось условие: скорость равна 0.

Лазерный луч при распространении внутри бокса при попадании на плоскости $z = 0$, $x = y$, $z = y$ испытывал зеркальное отражение.

Временная зависимость мощности лазерного импульса с длиной волны $\lambda_L = 0{,}53$ мкм, введенного внутрь бокса, имела вид: $\dfrac{1}{48} P_L\left(t\right)$, где $P_L(t)$ – была подобрана в результате серии расчетов по одномерной методике СНДП.

Лазерное излучение задавалось в центре отверстия 384 лучами, объединенными в 48 пучков по 8 лучей, каждому из которых присваивалась соответствующая доля энергии пучка. Угол между направлением ввода лазерного излучения и осью отверстия был равен 30°.

Спектр излучения рассчитывался на сетке из 45 спектральных групп на интервале от 0 до 4 кэV. Разбиение диапазона спектра на спектральные группы произведено следующим образом: от 0 до 0.5 – 10 групп равномерно, то 0.5 до 4 – 35 групп равномерно.

При счете 45 спектральных групп были разделены на 5 интервалов, таким образом, в программе спектральной диффузии на каждом счетном процессоре рассчитывалась определенная часть спектра, состоящая из 9 групп. В программе кинетики ионизации на каждом счетном процессоре рассчитывалась 1/5 часть точек подобласти.

При таком распараллеливании программ спектральной диффузии и кинетики ионизации получено ускорение счета задачи 3.3475, эффективность распараллеливания составила 71 %.

На рис. 4,а, б, в приведены распределения электронной температуры в одном из сечений ($\varphi = \varphi^* = 31{,}875°$) на 3 момента времени: $t = 0{,}8$, $t = 4$ и $t = 6{,}4$ нс.

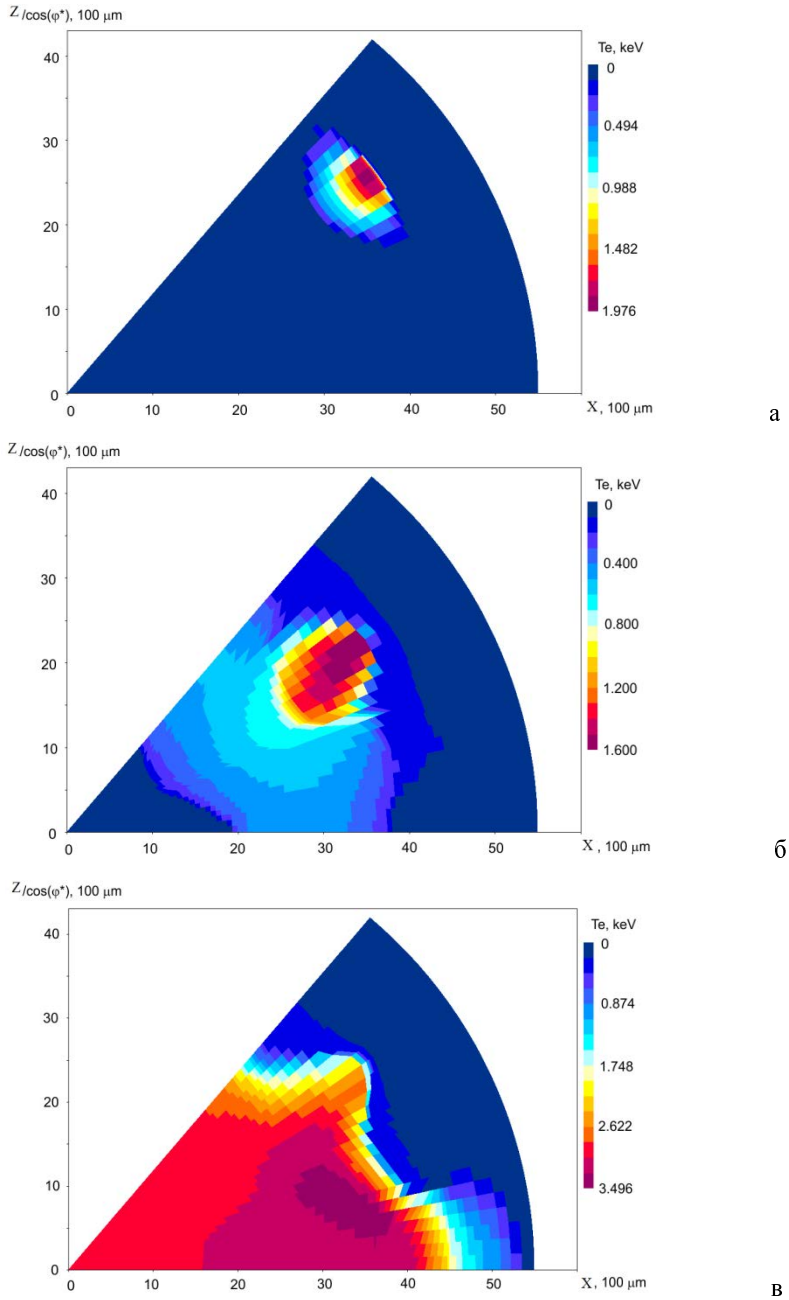
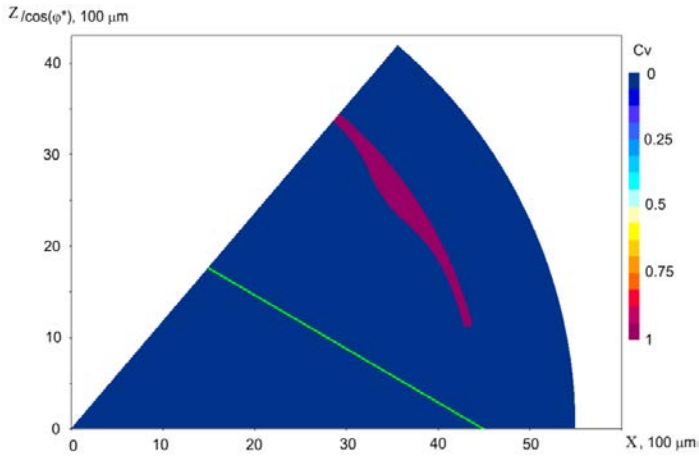
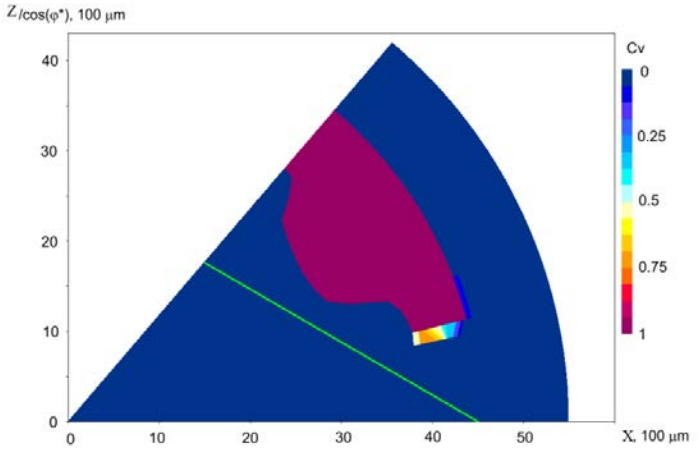


Рис. 4. Пространственные распределения электронной температуры T_e в одном из сечений ($\varphi = \varphi^* = 31{,}875°$) на 3 момента времени: а – $t = 0{,}8$ нс, б – $t = 4$ нс, в – $t = 6{,}4$ нс

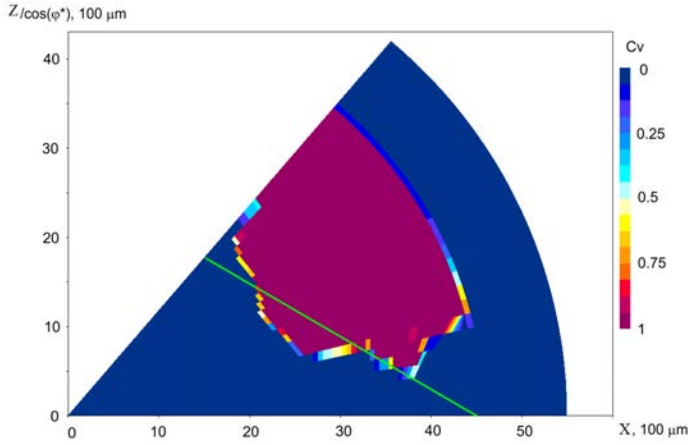
На рис. 5,а, б, в приведены распределения объемной концентрации золота в одном из сечений ($\varphi = \varphi^* = 31{,}875°$) на 3 момента времени: $t = 0{,}8$, $t = 4$ и $t = 6{,}4$ нс.



а



б



в

Рис. 5. Пространственные распределения объемной концентрации золота C_v в одном из сечений ($\varphi = \varphi^* = 31{,}875°$) на 3 момента времени: а – $t = 0{,}8$ нс, б – $t = 4$ нс, в – $t = 6{,}4$ нс

Как видно из рис. 6,а, б, в, к моменту времени $t = 6{,}4$ нс золотая плазма, прилетевшая со стенок бокса, оказывается на пути лазерного излучения, которое в основном поглощается вблизи отверстия, что видно из распределения температуры электронов на рис. 5,в.

Результаты трехмерных расчетов согласуются с ранее полученными одномерными расчетами, а также выявляют наличие еще одного фактора, влияющего на симметрию рентгеновского излучения.

Трехмерные расчеты показали, что внутренний объем бокса заполняется золотой плазмой, испаренной с его стенок, и эта плазма становится непрозрачной для лазерного излучения. Основная часть импульса поглощается внутри бокса вблизи отверстий ввода, что может привести к снижению симметрии рентгеновского излучения в центре бокса по сравнению со случаем, когда все лазерные пучки доходят до стенки бокса.

Следует предпринять дополнительные меры для сдерживания разлета плазмы со стенок бокса, например, заполнять бокс буферным газом. Необходимо также учесть в расчетах влияние потоков плазмы с капсулы и активную динамику разлета пленки, закрывающей отверстия.

Для подбора различных параметров в задаче конверсии лазерного излучения однозначно рекомендуется проводить расчеты для фрагмента «1/48» задачи, однако, для получения полной картины все же необходимо проведение чистового расчета с полной геометрией.

Литература

1. Бабанов А. В., Бельков С. А. и др. Методика МИМОЗА-НД3Д. Расчет трехмерных задач спектрального переноса излучения // Вопросы атомной науки и техники. Сер.: Математическое моделирование физических процессов. 2012. Вып. 2. С. 64–72.

2. Гаранин С. Г. Мощные лазеры и их применение в исследованиях физики высоких плотностей энергии // УФН. 2011. Т. 181, № 4. С. 434–441.

3. Бельков С. А., Долголева Г. В. Модель среднего иона для расчета кинетики ионизации, населенностей возбуждения уровней и спектральных коэффициентов переноса излучения в программе СНДП // Вопросы атомной науки и техники. Сер.: Математическое моделирование физических процессов. 1992. Вып. 1. С. 59–61.

4. Бельков С. А., Бондаренко С. В., Вергунова Г. А. и др. Влияние пространственной неоднородности нагрева на сжатие и горение термоядерной мишени при прямом многопучковом облучении лазерным импульсом мегаджоульного уровня // ЖЭТФ. 2017. Т. 151. Вып. 2. С. 396–408.

NUMERICAL SIMULATION OF 3D PROBLEMS OF LASER-INTO-X-RAY RADIATION CONVERSION IN A SPHERICAL HOHLRAUM OF MEGA-JOULE LASER FACILITY

A. V. Babanov, S. V. Bondarenko, E. I. Mitrofanov, L. F. Potapkina, O. O. Sharov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

One of the leading research trends in thermonuclear fusion is the inertial fusion. This approach is based on implementation of the inertia of the layer of thermonuclear fuel that was preliminary compressed and accelerated up to high velocities as a result of ablation acceleration of the shell target with intensive radiation.

Numerical simulation of the problems to optimize the design of cryogenic targets for indirect irradiation to produce the ignition, to select an X-ray pulse, to determine the laser radiation absorption dynamics and to characterize heterogeneous generation of the R-ray radiation at the walls of the hohlraum is considered.

The presentation describes the results of 3D computations on laser radiation conversion into the X-ray radiation using MIMOZA-ND3D-method.

Key words: indirect target irradiation, spherical hohlraum, laser radiation, X-ray radiation, plastic (CH) ablator, HDC ablator.

УДК 004.4
DOI 10.53403/9785951505071_2022_45

ЗАЩИЩЕННАЯ ОПЕРАЦИОННАЯ СИСТЕМА «АРАМИД» ДЛЯ СУПЕР-ЭВМ

Е. К. Баканов, В. А. Алешкин, Д. А. Красноярова, В. Ю. Симаков, И. А. Пищулин, А. И. Чайка, О. В. Новиков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Защищенная операционная система «Арамид» предназначена для организации распределенных высокопроизводительных вычислений, содержащих сведения, составляющие государственную тайну. Обеспечивает создание вычислительной среды с повышенными возможностями обеспечения информационной безопасности. Позволяет проводить работы с секретными данными в АСЗИ класса защищенности не ниже 2А по классификации ФСТЭК и Министерства Обороны Российской Федерации.

Ключевые слова: Арамид, система, защищенная, вычисления, супер-ЭВМ, дистрибутив.

В настоящее время высокопроизводительные вычисления (High-Performance Computing, HPC) играют ключевую роль в решении крупных задач математического моделирования и отработки перспективных программно-аппаратных решений.

Высокопроизводительные системы в ряде случаев применяются в приложениях специального назначения[1]. Для организации сбора и обработки больших объемов данных и выполнения вычислений над ними требуется операционная система, удовлетворяющая требованиям государственного регулятора[2] и адаптированная для работы на многопроцессорных системах.

Сегодня в России используется несколько сертифицированных ФСТЭК и МО РФ операционных систем, созданных на основе различных дистрибутивов ОС Linux: МСВС на базе RedHat, AstraLinux на базе Debian и ряд других. Как правило, при использовании этих ОС с включенными средствами разграничения доступа наблюдается снижение быстродействия при работе на супер-ЭВМ. Кроме этого в них отсутствуют либо не адаптированы (с точки зрения поддержки аппаратных платформ и высокой масштабируемости) сертифицированные ключевые для супер-ЭВМ программные компоненты, такие как: система управления вычислительными ресурсами и распределенными вычислениями, параллельная файловая система, система мониторинга, система подготовки легковесных бездисковых образов ОС, библиотеки межпроцессных обменов и т.п. Это делает невозможным применение данных операционных систем при разработке и эксплуатации супер-ЭВМ, предназначенных для обработки данных с ограниченным доступом, хотя и не исключает их использование в качестве сертифицированных ОС для рабочих мест и серверов.

В связи с этим специалистами ФГУП «РФЯЦ-ВНИИЭФ» ведется работа по созданию дистрибутива специализированной защищенной операционной системы для супер-ЭВМ (ЗОС «Арамид»), удовлетворяющей требованиям ФСТЭК России и МО РФ по обработке информации, составляющей государственную тайну.

---

[1] Приложения специального назначения предназначены для производственных и специальных потребностей органов государственной власти, крупных государственных предприятий и госкорпораций.

[2] В качестве государственного регулятора могут выступать ФСТЭК и МО РФ.

Входящие в дистрибутив ЗОС «Арамид» программные компоненты обеспечивают полнофункциональную работу супер-ЭВМ, что позволяет использовать ЗОС «Арамид» для решения таких задач, как:

– создание защищенных супер-ЭВМ нового поколения;

– построение автоматизированных систем управления (АСУ) специального назначения;

– сбор и обработка больших объемов структурированных и неструктурированных данных;

– внедрение систем поддержки принятия решений, требующих интенсивных вычислений и т. п.

Разрабатываемые средства защиты информации дистрибутива ЗОС «Арамид» предназначены для выполнения следующих функций:

– идентификация и аутентификация пользователей;

– дискреционное и мандатное разграничение доступа;

– регистрация и сигнализация о событиях безопасности;

– организация и контроль доменной и мандатной структуры разграничения потоков информации;

– контроль целостности средств защиты информации;

– создание замкнутой среды обработки данных;

– контроль подключаемых устройств;

– очистка высвобождаемой дисковой и оперативной памяти;

– создание единого пользовательского пространства.

Разрабатываемые средства защиты информации дистрибутива ЗОС «Арамид» реализуют «Требования безопасности информации к операционным системам», предъявляемые к операционным системам типа «А» (ОС общего назначения) 2-го класса защиты (обработка информации, содержащей совершенно секретные сведения) [1] – следовательно, ЗОС «Арамид» может применяться в автоматизированных системах класса защищенности 1Б.

В настоящее время уже пройдены сертификационные испытания дистрибутива ЗОС «Арамид» версии 1.0, а также проходит сертификационные испытания версий 1.1 и 3.0 дистрибутива защищенной операционной системы «Арамид». В ЗОС «Арамид» 1.1 и 3.0 выполнен переход на актуальные версии компонентов ОС, реализован механизм мандатного разграничения доступа в СУБД PostgreSQL, работающий согласованно с моделью доступа ЗОС «Арамид», а также доработана технология виртуализации для функционирования в условиях защищенной среды. Ведутся работы по разработке дистрибутива версии 3.1, в котором будет реализована новая кодовая база и ядро.

Заключение

Разрабатываемый во ФГУП «РФЯЦ-ВНИИЭФ» дистрибутив защищенной ОС «Арамид» планируется к эксплуатации на ряде защищенных комплексов и высоконагруженных параллельных супер-ЭВМ, применяемых для оперативного решения задач предприятий госкорпорации «Росатом», Минобороны России, а также предприятий ОПК (авиастроение, судостроение, ракетно-космическая отрасль и др.). Поставленные цели разработки дистрибутива ЗОС «Арамид» достигаются за счет акцентирования сил разработчиков на следующих функциональных требованиях:

– поддержка современных гибридных архитектур;

– предоставление вычислительных ресурсов пользователям с разными правами доступа;

– разработка защищенных механизмов управления вычислительными ресурсами и распределенными вычислениями с учетом современных подходов и технологий.

Литература

1. «Требования безопасности информации к операционным системам». ФСТЭК России. 19.08.2016 г.

ECURE OPERATING SYSTEM «ARAMID»: THE CURRENT STATE AND PROSPECTS FOR FURTHER DEVELOPMENT

E. K. Bakanov, V. A. Aleshkin, D. A. Krasnoyarova, V. Yu. Simakov, I. A. Pishchulin, A. I. Chayka, O. V. Novikov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The secure operating system «Aramid» was developed for the arrangement of distributed high-performance computations and operations with data constituting State secrets. It provides a computing environment with a higher level of information security. The system allows operating classified data in the automated information security system of the protection class not lower than 2A, according to the classification of FSTEC and the Russian Federation Ministry of Defense.

Key words: secure system, Aramid, computations, supercomputer, distribution kit.

УДК 519.6
DOI 10.53403/9785951505071_2022_48

ВОЗМОЖНОСТИ ПАКЕТА ПРОГРАММ ЛОГОС ДЛЯ МОДЕЛИРОВАНИЯ ОТКЛИКА КОНСТРУКЦИЙ НА ГАРМОНИЧЕСКУЮ ВИБРАЦИЮ

А. В. Баранов[1], *А. Ю. Еременко*[1], *С. С. Косарим*[1], *А. О. Наумов*[1,2], *Е. С. Ниткина*[1], *В. Н. Речкин*[1,2], *В. И. Романов*[1], *А. А. Рябов*[1], *К. В. Циберев*[1]

[1]Российский федеральный ядерный центр – Всероссийский НИИ экспериментальной физики, Саров
[2]Саровский физико-технический институт НИЯУ МИФИ, Саров

Приводится методика численного решения задач по расчету отклика конструкций на действие гармонической вибрации с учетом их начального квазистатического напряженно-деформированного состояния /НДС/. Представлено краткое описание базовых соотношений, вычислительных методов и алгоритмов, реализованных в пакете программ ЛОГОС. Описаны особенности учета квазистатического напряженно-деформированного состояния конструкции, общего конструкционного демпфирования колебаний и внутреннего рассеяния энергии в материалах при моделировании вибрационного отклика изделий. Приводятся результаты решения верификационных задач по определению вибрационного отклика типовых элементов конструкций. Выполнен анализ точности и сходимости результатов численных расчетов на основе сравнения с аналитическими решениями. Приведено решение практической задачи по определению отклика конструкции транспортного упаковочного комплекта (ТУК-159) на действие транспортной вибрации.

Ключевые слова: гармоническая вибрация, демпфирование, конечно-элементная модель, напряженно-деформированное состояние, ПП ЛОГОС.

Введение

Одним из наиболее востребованных направлений развития ПП ЛОГОС [1,2] является разработка конечно-элементной методики и программных средств для решения задач гармонического анализа конструкций. Решение таких задач является важной составной частью комплекса расчетных исследований отклика конструкций при действии вибрационных и динамических нагрузок [3–6].

Обычно при решении задач на определение отклика конструкции на действие синусоидальной нагрузки находится ее реакция в некотором диапазоне частот и затем строятся графики зависимости определенных параметров (обычно перемещений) от частоты для нескольких контрольных точек. Максимум реакции конструкции, найденный по графикам, обычно соответствует максимуму напряжений в силовых элементах и приходится на частоты возбуждения, совпадающие с собственными частотами системы. Далее по найденным максимальным значениям напряжений для основных элементов конструкции выполняется анализ усталостной прочности. Такой анализ при необходимости может выполняться и с учетом влияния предварительного напряженно-деформированного состояния конструкций.

В статье приводится описание основных особенностей вычислительных методов и алгоритмов, реализованных в текущей версии ПП ЛОГОС. Показаны результаты верификации методики на тестовых задачах гармонического анализа типовых элементов конструкций. Также приводятся результаты решения практической задачи по моделированию отклика транспортного упаковочного комплекта ТУК-159 на действие транспортной вибрации.

1. Основные функциональные возможности ПП ЛОГОС для решения задач гармонического анализа

Основные функциональные возможности разработанной методики, реализованной в составе ПП ЛОГОС версии 2020 года, для решения задач гармонического анализа конструкций:

– поддерживается более 20 типов конечных элементов (трехмерные изопараметрические элементы сплошной среды 1 и 2-го порядка – параллелепипеды, треугольные призмы, тетраэдры, пирамиды; «гибридные» элементы сплошной среды 1-го порядка с расширенными функциями формы – параллелепипеды; оболочечные и балочные элементы; специальные элементы, имитирующие связи узлов, заклепочные и сварные соединения);

– можно формировать граничные условия «жесткой заделки» узлов по одной или нескольким степеням свободы, связи двух групп узлов по определенным степеням свободы, условия контактного взаимодействия в зонах предполагаемого контакта деталей и условия циклической симметрии;

– возможно задание свойств для изотропных, ортотропных и анизотропных материалов с учетом зависимости материальных параметров от температуры;

– решение задач можно выполнять, как на согласованных, так и на несогласованных расчетных сетках контактирующих деталей путем прямого моделирования их контактного взаимодействия на начальном этапе расчета статического НДС и учета выявленных особенностей силового замыкания на последующем этапе при решении задачи гармонического анализа;

– расчет отклика конструкций можно проводить с учетом изменения жесткости деталей в зависимости от уровня квазистатических напряжений в них (с учетом преднапряженного состояния);

– гармонический анализ выполняется в линейной постановке и, соответственно, все виды нелинейности, заданные в модели на начальном этапе определения квазистатического НДС (физическая нелинейность, геометрическая нелинейность, контактные элементы), игнорируются;

– возможно распараллеливание вычислительного процесса на общей и на распределенной памяти ЭВМ (OpenMP+MPI).

Разработанные методика и программные средства по функциональным и вычислительным возможностям для решения задач гармонического анализа не уступают наиболее распространенным коммерческим ПП инженерного анализа и к настоящему времени прошли верификацию и валидацию на множестве тестовых задач, результаты решения которых частично приведены в [7–9].

Допущения и ограничения

При решении задач определения установившегося отклика конструкций на действие гармонической вибрации на основе разработанной методики вводятся следующие допущения и ограничения:

1. Используется линейно-упругая модель материалов;
2. Перемещения систем рассматриваются, как малые;
3. Матрица демпфирования в общем случае может формироваться, как сумма нескольких матриц, соответствующих различным моделям диссипации энергии. Чаще всего из-за отсутствия достаточного объема информации по демпфированию исследуемой системы матрица демпфирования формируется, как линейная комбинация матриц жесткости и масс (демпфирование по Релею). При использовании метода суперпозиции форм колебаний матрица демпфирования не строится явно, а демпфирование учитывается введением коэффициентов рассеяния энергии для различных собственных форм колебаний системы;
4. Кинематические граничные условия стационарны. Можно задавать граничные условия в виде кинематических закреплений и кинематических связей (простейших кинематических соотношений, моделирующих шарниры, упругие скользящие опоры и т. д.);
5. Все нагрузки должны изменяться во времени по гармоническому закону с одной и той же частотой;
6. Любые нелинейности (пластические свойства материала, условия контактного взаимодействия и др.) игнорируются, даже если они заданы в модели. Условия контактного взаимодействия деталей, определенные при расчете предварительного напряженно-деформированного состояния конструкции, принимаются постоянными при вычислении ее установившегося отклика;
7. Матрицы $[M]$ и $[K]$ постоянны во всем частотном диапазоне.

Если постановка задачи такова, что многие из этих ограничений и допущений недопустимы, то для ее решения можно использовать метод прямого интегрирования уравнений динамики, но при этом потребуются значительные вычислительные ресурсы.

2. Технология решения задач на основе разработанной методики

Технология решения задач гармонического анализа конструкций с учетом их квазистатического НДС включает три основных этапа:

1. Численное моделирование квазистатического НДС при необходимости с учетом геометрической и физической нелинейности, а также условий контактного взаимодействия деталей;
2. Изменение (корректировка) матрицы жесткости системы с учетом вычисленных параметров ее квазистатического НДС и заданных параметров нагрузок;
3. Численное моделирование вибрационного отклика конструкции с учетом скорректированной матрицы жесткости в линейной постановке.

Аналогичный подход используется также в ПП ЛОГОС при анализе устойчивости конструкций в линейной постановке.

Алгоритмы и особенности численного моделирования квазистатического НДС конструкций в ПП ЛОГОС достаточно подробно описаны в литературных источниках, например, в [10] и в данной работе не рассматриваются.

Для учета в расчетах вибрационного отклика конструкции изменения жесткости ее элементов в зависимости от разных видов нелинейности и заданного уровня статических нагрузок разработаны специальные алгоритмы и программные средства. В текущей версии ПП ЛОГОС изменение жесткости системы учитывается посредством вычисления ряда дополнительных матриц на основе параметров ее квазистатического НДС, заданных нагрузок и суммирования этих матриц с основной матрицей жесткости перед выполнением гармонического анализа.

В текущей версии ПП ЛОГОС реализовано два метода моделирования отклика конструкций на гармоническое воздействие в линейной постановке: «полный» метод и метод суперпозиции форм колебаний [5, 6, 11]. При решении задач указанными методами все виды нелинейностей такие, как нелинейное поведение материала, контактные граничные условия и др. игнорируются при решении, даже если они были определены на предыдущих этапах.

При необходимости учета в задаче каких-либо видов нелинейностей можно применить метод неявного «прямого» интегрирования уравнений [5, 11]. Термин «прямое» означает, что перед интегрированием не выполняется никаких преобразований уравнений. «Прямое» численное интегрирование основано на двух идеях. Во-первых, достаточно, чтобы условия равновесия системы удовлетворялись на отдельных коротких отрезках времени Δt, а не в любой момент времени t. Это означает, что равновесие с учетом сил инерции и демпфирования рассматривается в дискретных точках временного интервала, и, следовательно, возможно эффективное использование в методах «прямого» интегрирования всей вычислительной процедуры статического анализа. Во-вторых, внутри каждого временного интервала учитывается изменение перемещений, скоростей и ускорений, и способ учета этих изменений определяет точность и устойчивость численного решения. Однако при больших порядках матриц (большой размерности задач) эти процедуры становятся неэффективными для расчета установившегося гармонического отклика конструкций. Кроме того, для расчета поведения конструкций в некотором диапазоне частот гармонического воздействия на основе таких методов возникает необходимость решения большого количества отдельных задач для каждого конкретного значения частоты возбуждения. Поэтому этот метод является сравнительно «дорогим», требует больших затрат времени на решение и значительных вычислительных ресурсов.

Для решения систем алгебраических уравнений, формируемых в процессе решения указанных задач, в рамках разработанных программных средств используются: библиотека решателей PMLP/Parsol [12], разработанная в РФЯЦ-ВНИИЭФ, прямой решатель ParDiSo из библиотеки Intel® Math Kernel Library и решатель ParFBSS разработки ООО ТЕСИС.

3. Особенности корректировки жесткости систем в зависимости от начального напряженно-деформированного состояния

В некоторых случаях, когда деформирование конструкций под действием каких-либо статических нагрузок сопровождается значительным изменением жесткости и/или силового замыкания отдельных элементов, расчеты осложняются необходимостью учета этого эффекта. Например, при расчете отклика тонкостенных конструкций или ферм, изгибная жесткость отдельных элементов которых мала по сравнению с их жесткостью в продольном направлении, необходимо учитывать изменение их изгибной жесткости в зависимости от уровня действующих растягивающих или сжимающих сил [3, 4, 7]. Учет начального НДС при решении задач гармонического анализа конструкций необходим также в тех случаях, когда под действием приложенных сил значительно изменяется их исходная геометрия.

Для учета возможного изменения жесткости конструкций (или отдельных их элементов) под действием квазистатических нагрузок при численном исследовании их вибрационного отклика выполняется корректировка исходной матрицы жесткости расчетной модели в общем случае по формуле

$$[K] = [K_0] + [K_\sigma] + [K_L] + [K_R], \qquad (1)$$

где $[K_0]$ – касательная матрица жесткости, построенная на первом шаге решения и при необходимости скорректированная с учетом физической нелинейности материалов в предположении, что перемещения малы.

Матрицы $[K_\sigma]$, $[K_L]$ и $[K_R]$ формируются на основе вычисленных значений параметров начального (квазистатического) НДС конструкции и заданных значений действующих нагрузок. Особенности вычисления этих матриц приведены в работах [8, 9]. Суммирование указанных матриц с касательной матрицей жесткости позволяет учесть изменение жесткости конструкции под влиянием заданных квазистатических нагрузок.

4. Особенности формирования матрицы демпфирования

Динамический отклик любых конструкций сопровождается диссипацией энергии [11, 13]. При численном решении задач рассматриваемого класса в текущей версии ПП ЛОГОС диссипация энергии системы может быть учтена введением сил диссипации, пропорциональных скорости движения. В уравнениях равновесия системы, находящейся в движении под действием некоторой нагрузки, силы диссипации (демпфирования) представляются слагаемым $[C]\dfrac{d\{u\}}{dt}$, поэтому демпфирование вынужденных колебаний расчетной модели определяется заданием матрицы демпфирования $[C]$. Необходимо отметить, что в общем случае матрица *[C]* не может быть построена из матриц демпфирования отдельных конечных элементов, как матрица жесткости и матрица масс.

При численном решении задач для сложных конструкций в большинстве случаев трудно определить все необходимые параметры демпфирования. Наиболее частыми причинами этого являются неполная информация о зависимости параметров демпфирования от частоты колебаний или отсутствие данных о демпфирующих свойствах некоторых материалов в сборке и др. В таких случаях можно упростить задачу и матрицу $[C]$ задать постоянной во всем диапазоне частот возбуждения. Одним из вариантов является построение матрицы $[C]$ на основе глобальных матриц $[K]$ и $[M]$ системы (демпфирование по Релею) [3, 11]:

$$[C] = \alpha \cdot [M] + \beta \cdot [K], \qquad (2)$$

где $\alpha[M]$ – составляющая, так называемого, инерционного демпфирования; $\beta[K]$ – составляющая конструкционного демпфирования.

Значения коэффициентов α и β, как правило, неизвестны заранее, но они могут быть определены по известным коэффициентам рассеяния энергии ξ_i для нескольких (как минимум, двух) основных собственных форм колебаний конструкции. Предполагается, что сумма значений данных ко-

эффициентов постоянна в рассматриваемом диапазоне частот воздействия. Таким образом, при известных значениях двух коэффициентов ξ_1 и ξ_2, соответствующих двум различным собственным частотам колебаний конструкции ω_1 и ω_2, коэффициенты α и β могут быть определены из системы двух уравнений:

$$\alpha = \frac{2\omega_1\omega_2\left(\xi_1\omega_2 - \xi_2\omega_1\right)}{\omega_2^2 - \omega_1^2},\ \beta = \frac{2\left(\xi_2\omega_2 - \xi_1\omega_1\right)}{\omega_2^2 - \omega_1^2}. \quad (3)$$

Если известно несколько коэффициентов рассеяния энергии какой-то конструкции может применить более сложную процедуру вычисления матрицы демпфирования. Пусть количество известных коэффициентов рассеяния энергии равно p. Тогда матрица $[C]$ находится с помощью ряда [11]:

$$[C] = [M]\sum_{k=0}^{p-1} a_k \left[[M]^{-1}[K]\right]^k, \quad (4)$$

где коэффициенты a_k определяются из решения системы p уравнений

$$\xi_i = \frac{1}{2}\left(\frac{a_0}{\omega_i} + a_1 \cdot \omega_i + a_2 \cdot \omega_i^3 + \ldots + a_{p-1} \cdot \omega_i^{2p-3}\right). \quad (5)$$

Необходимо отметить, что при $p = 2$ формула (4) приводится к формуле Релеевского демпфирования. При $p > 2$ матрица демпфирования, вычисляемая по формуле (4), в общем случае является полной. Поскольку с использованием полной матрицы $[C]$ вычислительные затраты значительно возрастают, то при решении задач большой размерности стремятся ограничиться только Релеевским демпфированием. Недостатком Релеевской модели демпфирования является значительный уровень диссипации энергии на высших собственных формах колебаний по сравнению с низшими формами, для которых обычно и подбираются релеевские константы α, β.

Во многих практических случаях составляющая инерционного демпфирования $\alpha[M]$ игнорируется ($\alpha = 0$), тогда коэффициент β определяется из известных значений ξ_i и ω_i по формуле

$$\beta = 2 \cdot \xi_i / \omega_i\ . \quad (6)$$

Для построения глобальной матрицы $[C]$ используется только одно значение коэффициента β, поэтому для его вычисления должна использоваться доминирующая собственная частота системы.

Выше предполагалось, что демпфирование конструкций может быть удовлетворительно учтено в расчетах моделью пропорционального демпфирования. Во многих случаях предположение о пропорциональном демпфировании является адекватным. Однако, при расчете конструкций с широким диапазоном свойств материалов или, например, с встроенными локальными демпферами может оказаться необходимым применение модели непропорционального демпфирования. В таких случаях при формировании общей матрицы демпфирования используются различные значения коэффициента β для различных частей (материалов) конструкции. Зависящие от свойств материалов коэффициенты β_i задаются в качестве одного из параметров используемых моделей материалов, а общее для системы конструкционное демпфирование определяется выражением

$$\sum_{j=1}^{M} \beta_j [K]_j, \quad (7)$$

где $[K]_j$ – часть матрицы жесткости системы, соответствующая j-му материалу, M – количество различных материалов в исследуемой системе.

При таком подходе результирующая матрица $[C]$ имеет сложный вид. Коэффициенты β_j вычисляются, по-прежнему, по формуле (6), но вместо коэффициента рассеяния энергии ξ_j используется коэффициент затухания γ_j [11].

Непропорциональное демпфирование может быть также учтено заданием специфических конечных элементов с определенными характеристиками демпфирования. В качестве таких элементов чаще всего используется пружинный элемент с демпфированием.

Таким образом, полное выражение для глобальной матрицы демпфирования *[C]*, соответствующей в общем случае модели непропорционального демпфирования можно представить в следующем виде

$$[C] = \alpha [M] + \beta [K] + \sum_{j=1}^{M} \beta_j [K]_j + \sum_{k=1}^{nel} [C]_k \ , \qquad (8)$$

где $[C]_k$ – матрица демпфирования *k*-го специфического конечного элемента, *nel* – количество специфических элементов.

В общем случае при моделировании отклика конструкций на гармоническое воздействие можно учесть пять разных моделей демпфирования:

– постоянное пропорциональное демпфирование;

– «модальное» демпфирование, когда коэффициенты рассеяния энергии ξ_i задаются отдельно для каждой учитываемой собственной формы колебаний (используется только при решении методом суперпозиции форм колебаний);

– релеевское демпфирование;

– демпфирование в определенном материале (непропорциональное);

– демпфирование специфических элементов (непропорциональное).

В табл. 1 приведена информация о доступных моделях демпфирования при решении задач на определение гармонического отклика конструкций в ПП ЛОГОС.

Таблица 1

Доступные модели демпфирования при решении задач гармонического анализа с использованием различных методов

Метод решения задачи	Постоянное пропорциональное демпфирование	Релеевское демпфирование	«Модальное» демпфирование	Демпфирование, в определенном материале	Демпфирование специальных элементов
Полный	+	+	–	+	+
Суперпозиция форм колебаний	+	+	+	–	–
Метод неявного «прямого» интегрирования	+	+	–	+	+

5. Результаты решения некоторых верификационных задач

Проводятся результаты верификации разработанных программных средств на трех задачах по определению установившегося отклика типовых систем на гармоническое силовое воздействие. Выполнен анализ точности и сходимости результатов численных решений. Решение задач выполнено на пространственных конечно-элементных моделях из восьмиузловых элементов сплошной среды 1-го порядка с расширенными функциями формы. Результаты численных расчетов сравнивались с аналитическими решениями и с данными экспериментов [3, 4, 14].

Расчет установившегося отклика консольно-закрепленного бруса с грузом на конце при действии осевой силы, изменяющейся по гармоническому закону

Рассматривается консольно закрепленный брус квадратного поперечного сечения с грузом на конце (рис. 1). К грузу приложена гармонически изменяющаяся сила $P_1 = P_0 \cdot \cos \omega t$ с амплитудой $P_0 = 10$ кг, направленная вдоль оси балки. Необходимо определить АЧХ и ФЧХ данной системы в диапазоне частот изменения силы от 0 до 240 Гц с учетом демпфирования колебаний за счет внутреннего рассеивания энергии в материале бруса.

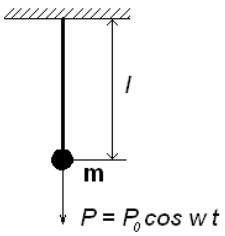
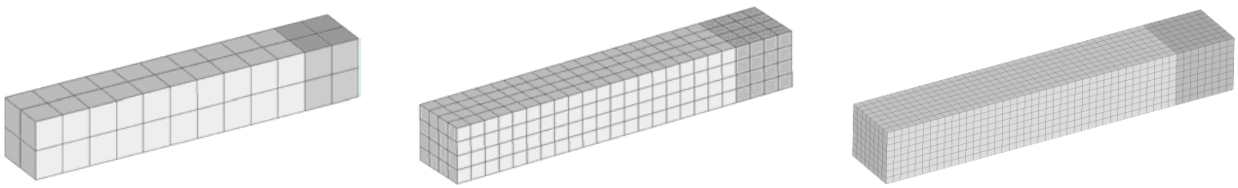


Рис. 1. Консольно закрепленный брус с грузом на конце

Исходные данные по геометрическим параметрам системы и свойствам материала бруса: длина бруса – $l = 50$ мм, ширина сечения бруса – $a = 10$ мм, масса груза – $m = 104{,}5$ кг, коэффициент Пуассона материала бруса – $\nu = 0{,}3$, модуль упругости – $E = 8400$ кг/мм^2, коэффициент внутреннего рассеивания (затухания) – $\gamma = 0{,}0032$.

Уравнения для аналитического решения задачи представлены в [3, 14].

Для численного решения задачи и анализа сходимости результатов использовались три конечно-элементные модели /КЭМ/ разной размерности (рис. 2): КЭМ-1, КЭМ-2 и КЭМ-3 со средним размером элементов 5,0 × 5,0 × 5,0 мм, 2,5 × 2,5 × 2,5 мм и 1,25 × 1,25 × 1,25 мм соответственно.



КЭМ-1 КЭМ-2 КЭМ-3

Рис. 2. КЭМ системы

Низшая собственная частота осевых колебаний рассматриваемой системы, согласно аналитическим расчетам, составляет 200 Гц, а по результатам численных расчетов на КЭМ-2 – 200,5 Гц. Отличие значений низшей собственной частоты осевых колебаний системы, полученных численно и аналитически, составляет ~ 0,2 %.

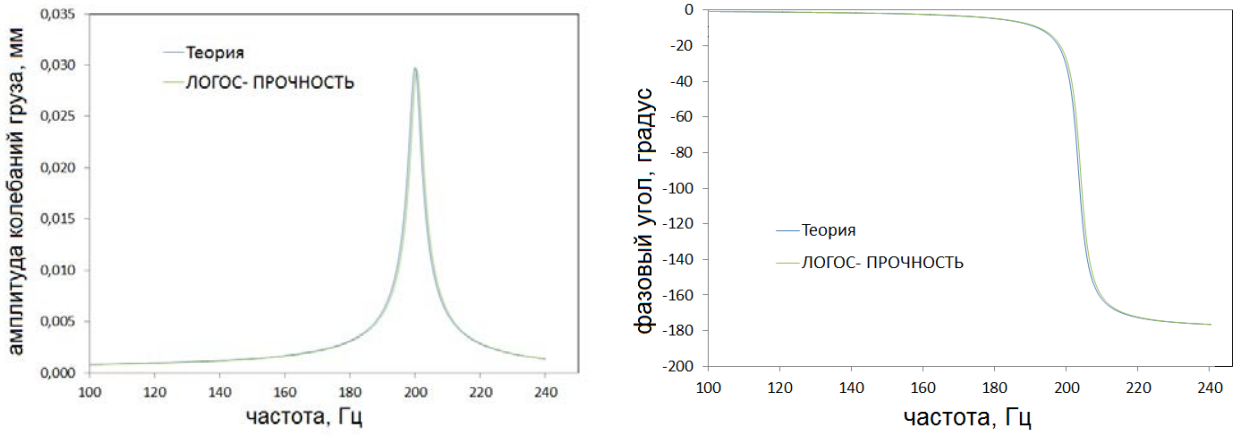
Численное решение задачи определения отклика системы на действие гармонической вибрации выполнялось «полным» методом. Результаты численных и аналитических расчетов амплитуды перемещений груза при действии осевой силы, изменяющейся по гармоническому закону на некоторых фиксированных частотах, представлены для сравнения в табл. 2. Погрешность результатов численных расчетов вычисляется относительно аналитического решения.

Таблица 2

Результаты численных и аналитических расчетов амплитуды перемещений груза

Частота, Гц	Теория, мм	ПП ЛОГОС					
		КЭМ-1		КЭМ-2		КЭМ-3	
		мм	%	мм	%	мм	%
100	0,000794	0,0074	2,52	0,000786	1,01	0,00787	0,83
150	0,001360	0,001312	3,53	0,001329	1,54	0,001342	1,34
200	0,029763	0,028647	3,75	0,029135	2,11	0,029550	0,72
230	0,001841	0,001924	3,42	0,001876	1,90	0,001871	1,61

На рис. 3 приведены графики зависимости амплитуды перемещений (АЧХ) и сдвига фаз (ФЧХ) груза от частоты воздействия (изменения) силы в диапазоне от 0 до 240 Гц, построенные по результатам численных и аналитических расчетов для КЭМ-1 системы. Из графиков видно, что результаты расчетов качественно согласуются с приемлемой для инженерных расчетов точностью даже на относительно грубой сетке.



АЧХ ФЧХ

Рис. 3. АЧХ и ФЧХ груза для КЭМ-1

Для анализа сходимости результатов численного решения задачи на КЭМ разной размерности построен график изменения значений перемещений груза на частоте 150 Гц в зависимости от степени сгущения расчетной сетки (рис. 4). На основе сравнения с аналитическим решением график наглядно показывает точность и характер сходимости результатов численных решений.

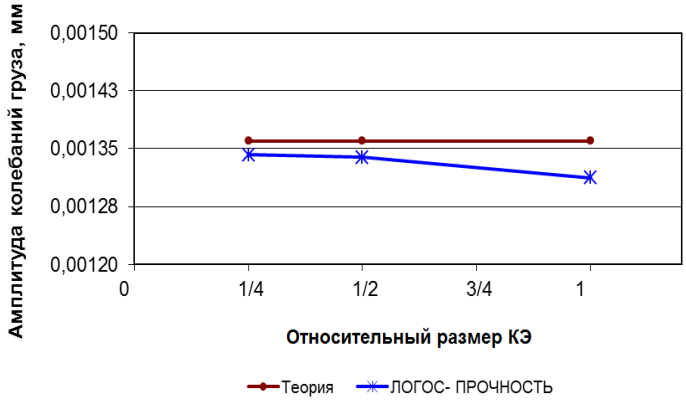


Рис. 4. График сходимости результатов численных расчетов
по перемещениям груза на частоте 150 Гц

Расчет установившегося отклика шарнирно опертой балки постоянного поперечного сечения при действии поперечной силы, изменяющейся по гармоническому закону

Рассматривается балка постоянного квадратного поперечного сечения, шарнирно закрепленная на двух опорах (рис. 5). В срединном сечении балки приложена возмущающая сила $P(t) = P_0 \cos\omega t$ с амплитудой $P_0 = 0{,}5$ кг. Необходимо определить величину прогиба в точке приложения силы при установившихся вынужденных колебаниях балки в зависимости от частоты воздействия в диапазоне от 3 до 8 Гц.

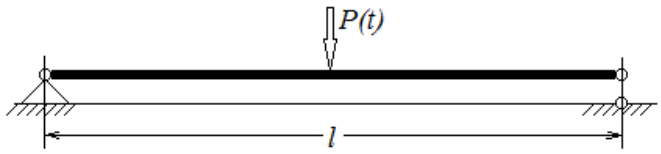


Рис. 5. Расчетная схема шарнирно опертой балки
квадратного поперечного сечения, нагруженной
в срединном сечении поперечной силой $P(t) = P_0 \cos\omega t$

Исходные данные по геометрическим параметрам балки и свойствам материала: длина балки – l = 2 м, высота и ширина поперечного сечения – $a = b$ = 0,01 м, коэффициент Пуассона – ν = 0,3, модуль упругости – $E = 2 \cdot 10^{11}$ Па, плотность материала – ρ = 7800 кг/м^3.

Прогиб в срединном сечении балки от действия периодически изменяющейся возмущающей силы можно вычислить по формуле. [14]

$$\eta = \frac{2P_0}{ql} \cos \omega t \sum_{k=1,3,5\ldots}^{\infty} \frac{1}{k^4 f_1^2 - \omega^2}, \qquad (9)$$

где P_0 – амплитуда возмущающей силы, ω – частота изменения возмущающей силы, q – погонная масса балки, f_1 – низшая собственная частота балки.

Для численного решения задачи и анализа сходимости результатов использовались три КЭМ разной размерности со средним размером элементов 5,0 × 5,0 × 10,0, 2,5 × 2,5 × 5,0 и 1,5 × 1,5 × 2,0 мм соответственно.

Низшая собственная частота поперечных колебаний рассматриваемой балки, согласно аналитическим расчетам, составляет 5,68 Гц. Численное решение такой задачи на КЭМ-2 дает значение низшей собственной частоты – 5,72 Гц. Отличие значений низшей собственной частоты поперечных колебаний балки, полученных численно на КЭМ-2 и аналитически, составляет ~ 0,5 %.

Расчет отклика системы выполнялся «полным» методом и методом суперпозиции собственных форм колебаний. Точного аналитического решения этой задачи с учетом демпфирования колебаний не существует. В связи с этим решение задачи проводится без учета демпфирования для диапазона частот изменения возмущающей силы от 3 до 8 Гц за исключением околорезонансной области в окрестности низшей собственной частоты балки, где решение, соответствующее ее резонансному возбуждению, должно стремиться к бесконечности.

Результаты расчетов амплитуды колебаний срединного сечения балки в дорезонансной области в диапазоне частот от 2,39 до 3,16 Гц представлены для сравнения в табл. 3. Незначительное увеличение отличий результатов численных расчетов от аналитического решения по мере приближения к низшей резонансной частоте объясняется тем, что при численном решении задачи задавалось демпфирование колебаний с коэффициентом внутреннего трения материала (затухания) 0,0002 % и при приближении к околорезонансной области это начинает сказываться на результатах.

Таблица 3

Результаты численных и аналитических расчетов амплитуды колебаний срединного сечения балки

Частота, Гц	Теория, мм	ПП ЛОГОС					
		КЭМ-1		КЭМ-2		КЭМ-3	
		мм	%	мм	%	мм	%
2,39	6,1487	6,0357	1,84	6,0361	1,83	6,0363	1,82
2,53	6,3029	6,1860	1,85	6,1866	1,84	6,1868	1,84
2,71	6,5358	6,4117	1,90	6,4122	1,89	6,4123	1,88
2,89	6,8049	6,6714	1,96	6,6720	1,95	6,6721	1,95
3,02	7,0347	6,8925	2,02	6,8930	2,01	6,8931	2,01
3,16	7,2926	7,1405	2,09	7,1411	2,08	7,1413	2,07

На рис. 6 приведены графики зависимости амплитуды колебаний срединного сечения балки от частоты воздействия (изменения) силы в диапазоне от 3 до 8 Гц, построенные по результатам численных (на КЭМ-1) и аналитических расчетов. Из рисунков видно, что результаты расчетов качественно согласуются с приемлемой для инженерных расчетов точностью.

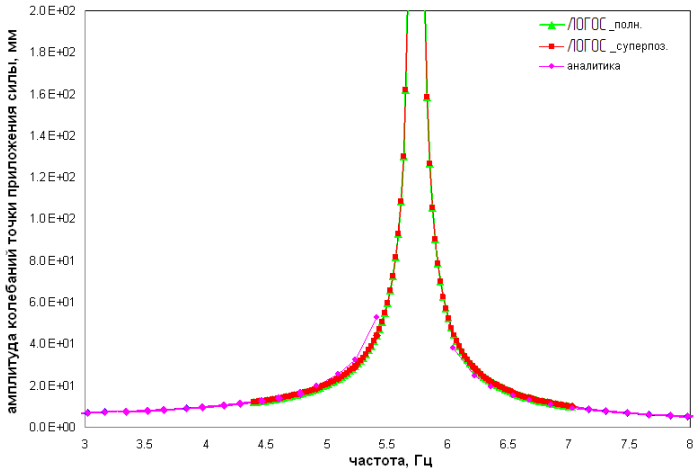


Рис. 6. АЧХ для КЭМ-1

6. Расчет напряженно-деформированного состояния конструкции ТУК-159 при действии транспортной вибрации

На рис. 7 показана в разрезе геометрическая модель конструкции транспортного упаковочного комплекта ТУК-159, предназначенной для перевозки ураносодержащих материалов всеми видами транспорта. Для обоснования вибропрочности конструкции выполнен численный расчет ее напряженно-деформированного состояния и отклика при действии транспортных нагрузок, заданных в виде синусоидальной вибрации. Нагрузка на упаковку действует в вертикальном направлении. Для учета демпфирования колебаний задавался постоянный коэффициент конструкционного демпфирования $\xi = 3$ %.

Для проведения численных исследований средствами препроцессора ПП ЛОГОС разработана подробная трехмерная КЭМ конструкции (рис. 8). На рис. 9 показаны отдельные фрагменты КЭМ. КЭ-дискретизация расчетной области задачи по пространству выполнена на основе 8-узловых конечных элементов сплошной среды с расширенными функциями формы. Размерность модели составляет ~ 670 тыс. конечных элементов.

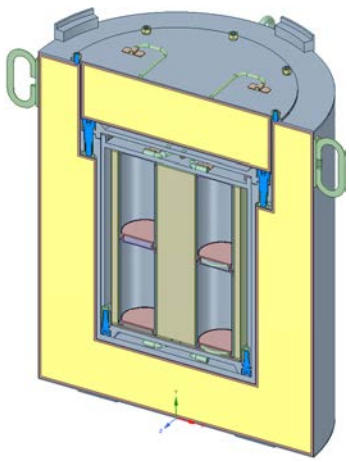


Рис. 7. Общий вид конструкции

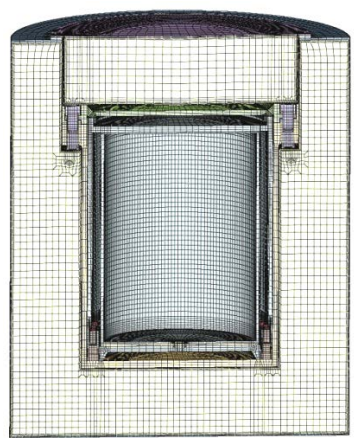


Рис. 8. КЭМ конструкции в разрезе

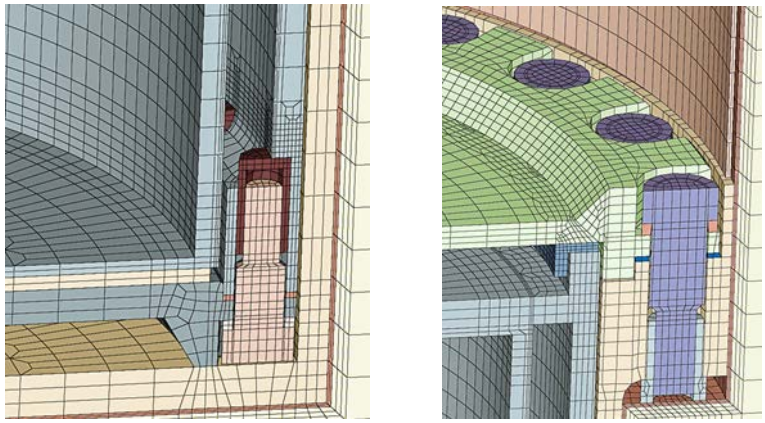


Рис. 9. Отдельные фрагменты КЭМ

Расчет отклика конструкции на действие транспортной вибрации выполнен «полным» методом. Результаты расчета показали, что низшая собственная частота боковых колебаний конструкции составляет 277 Гц. Все собственные частоты колебаний конструкции выходят за диапазон частот вибрационного воздействия, характерного для условий транспортирования, поэтому резонансного возбуждения элементов конструкции в процессе транспортирования не произойдет.

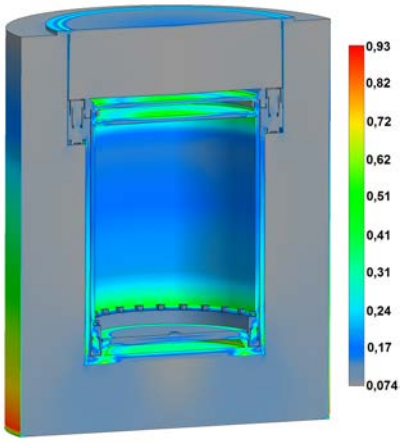


Рис. 10. Интенсивность напряжений в конструкции при действии транспортной вибрации в вертикальном направлении, МПа

Все силовые детали упаковки в условиях перевозки деформируются упруго. Максимальный уровень интенсивности напряжений реализуется во внешней облицовке при колебаниях конструкции на максимальной частоте из диапазона частот транспортной вибрации и составляет 0,93 МПа (см. рис. 10).

Заключение

В работе приведено описание функциональных возможностей реализованной в составе ПП ЛОГОС методики численного решения задач по расчету отклика конструкций на действие гармонической вибрации с учетом их начального квазистатического напряженно-деформированного состояния. Показаны особенности реализации вычислительных методов и основных алгоритмов методики, а также учета общего конструкционного демпфирования колебаний и внутреннего рассеяния энергии в материалах при моделировании вибрационного отклика изделий.

Работоспособность разработанных программных средств подтверждена результатами решения верификационных задач гармонического анализа типовых элементов конструкций с приемлемой точностью и одной практической задачи по расчету отклика конструкции ТУК-159 на действие транспортной вибрации.

Литература

1. Дерюгин Ю. Н., Козелков А. С., Спиридонов В. Ф., Циберев К. В., Шагалиев Р. М. Многофункциональный высокопараллельный пакет программ ЛОГОС для решения задач тепломассопереноса и прочности // Сборник тезисов докладов Санкт-Петербургского научного форума “Наука и общество”. – Санкт-Петербург, 2012.

2. Свидетельство о государственной регистрации программы для ЭВМ № 2019611831 от 04.02.2019 г. Программный модуль пакета программ «ЛОГОС», версия 2017 года.

3. Вибрации в технике: Справочник. В 6-ти т. // Ред. В. Н. Челомей (пред). – М.: Машиностроение, 1980. – Т.3. Колебания машин, конструкций и их элементов // Под ред. Ф. М. Диментберга и К.С. Колесникова. – 1980. С. 544.

4. Прочность, устойчивость, колебания. Справочник в трех томах. Под ред. Биргера И. А. и Пановко Я. Г. – М.: Машиностроение, 1968.

5. http://www.ansys.com [Electronic resource].

6. http://www.simulia.com [Electronic resource].

7. Рябов А. А., Спиридонов В. Ф., Циберев К. В., Речкин В. Н. и др. Результаты верификации пакета программ ЛОГОС на задачах определения основных собственных частот и форм колебаний типовых конструкций // Сборник докладов XIII-го международного семинара «Супервычисления и математическое моделирование». – Саров, 2011.

8. Циберев К. В., Речкин В. Н., Чембаров А. И., Александрова О. Л. Исследования точности результатов модального анализа конструкций в пакете программ ЛОГОС на основе конечных элементов сплошной среды // Сборник докладов XII-й научно-технической конференции «Молодежь в науке». – Саров, 2013.

9. Агафонов С. К., Речкин В. Н., Чембаров А. И., Кремчеева Л. Р. Исследования точности решения задач гармонического анализа конструкций в ПП ЛОГОС-Прочность на двух верификационных задачах // Сборник материалов VII-й всероссийской молодежной научно-инновационной школы «Математика и математическое моделирование». – Саров: СарФТИ НИЯУ-МИФИ, 2013.

10. Барабанов Р. А., Дьянов Д. Ю., Спиридонов В. Ф., Филимонкин Е. А., Циберев К. В. и др. Пакет программ ЛОГОС. Функциональные возможности для решения задач статической прочности конструкций с учетом физической и геометрической нелинейностей // Сборник докладов XI-й научно-технической конференции «Молодежь в науке» // секция «Теоретическая и математическая физика». – Саров, 2012.

11. Оден Дж. Конечные элементы в нелинейной механике сплошных сред. Перевод с английского. – М.: МИР. 1976.

12. Бартенев Ю. Г., Бондаренко Ю. А., Ерзунов В. А. и др. Комплекс LParSol для решения СЛАУ// Труды XIII международного семинара «Супервычисления и математическое моделирование». – Саров, 2011. С.34–36.

13. Зенкевич О. Метод конечных элементов в технике. – М.: Мир, 1975.

14. Расчеты на прочность в машиностроении: Справочник в 3-х томах. Под ред. заслуженного деятеля науки и техники, д-ра техн. наук проф. Пономарева С. Д. – М.: МАШГИЗ, 1959. – Т. 3. Инерционные нагрузки. Колебания и ударные нагрузки. Выносливость. Устойчивость. – 1959.

THE POSSIBILITIES OF THE LOGOS SOFTWARE FOR SIMULATION OF THE CONSTRUCTIONS RESPONSE TO HARMONIC VIBRATION

A. V. Baranov[1], *A. Yu. Eremenko*[1], *S. S. Kosarim*[1], *A. O. Naumov*[1,2], *E. S. Nitkina*[1], *V. N. Rechkin*[1,2], *V. N. Romanov*[1], *A. A. Ryabov*[1], *K. V. Tsiberev*[1]

[1] Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov
[2] Sarov Institute of Physics and Engineering, National Research Nuclear University MEPhI, Sarov

The paper describes a method of numerical solution of the problem to compute structures response to harmonic vibration with the account for their initial quasi-static stress and strain state (SSS). Brief description of basic ratios, computational methods and

algorithms realized in LOGOS software package is given. Specific features to account for quasi-static stress and strain state of the structure and general structure damping of oscillations and internal energy dissipation in materials when simulating vibration response of structures are described. Solution results of verification problems when finding vibration response of typical structural elements are shown. Accuracy and convergence of the results were analyzed when comparing computations with analytical solutions. The results of solving a practical problem of modeling the response of the TUK-159 transport pack set to the effect of transport vibration are presented.

Key words: harmonic vibration, damping, finite-element model, stress and strain state, LOGOS software package.

УДК 519.65
DOI 10.53403/9785951505071_2022_61

НЕЙРОСЕТЕВОЙ АЛГОРИТМ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ

В. Б. Бетелин[1], *В. А. Галкин*[2], *А. О. Дубовик*[2]

[1]Федеральный научный центр НИИ системных исследований РАН, Москва
[2]Сургутский филиал ФНЦ НИИСИ РАН, Сургут

В данной работе исследуется аспекты построения нейронных сетей на примере решения задачи о поиске вейвлет-разложения для функций одной переменной. Нейронная сеть строится с помощью библиотеки машинного обучения TensoFlow. Результаты работы показали, что созданная нейронная сеть более точно справляется с решением задачи построения вейвлет-разложения, чем имеющиеся готовые библиотеки при небольшом наборе входных данных.

Ключевые слова: нейронные сети, вейвлет-преобразование.

Исследование эффективных методов нахождения вейвлет-преобразования связано с поиском наиболее экономичных алгоритмов сжатия данных различных типов, при этом, данные необходимо восстанавливать с наименьшими потерями. В качестве приложении вейвлет-преобразований также необходимо указать исследование нестационарных сигналов, неоднородных полей и изображений, распознавание образов и т. д. [1].

Вейвлет преобразованием функции одной переменной называется представление ее в виде обобщенного ряда по системе базисных функций вида

$$\psi_{ab}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{1}$$

обладающего определенными свойствами за счет сдвига по времени и изменения временного масштаба, множитель $1/\sqrt{a}$ обеспечивает независимость нормы функции от масштабирующего коэффициента [1]. Функция $\psi(t)$ называется материнским вейвлетом и может быть выбрана произвольно. Обычно на нее накладывают следующие свойства: ограниченности нормы, локализации, среднее значение равно нулю и самоподобность [1–3].

В данной работе поиск вейвлет-разложения функций одной переменной осуществляется с помощью нейронной сети, а накладываемые обычно ограничения на материнские вейвлеты при таком подходе являются слишком обременительными и их выполнение в дальнейшем не предполагается.

Поиск вейвлет-разложения для произвольных непрерывных функций $y = f(x)$, определенных на отрезке [–1;1]. В качестве материнских вейвлетов использовались следующие функции:

$$\psi_{ab}(x) = \sin(ax+b), \tag{2}$$

$$\psi_{ab}(x) = (ax+b)\exp\left\{\frac{(ax+b)^2}{2}\right\}. \tag{3}$$

Работа любой нейронной сети может рассматриваться как решение задачи нелинейной оптимизации. Нейронная сеть состоит, как правило, из нескольких слоев, каждый из которых, в свою очередь,

состоит из нескольких нейронов. Вся нейронная сеть представляет собой суперпозицию отображений. Каждый слой сети функционирует как отображение в конечномерном векторном пространстве $X \to f\left(AX + B\right)$, где $X = \left\{x_1, ..., x_n\right\}^T$ – входной сигнал или вектор входных данных, $AX + B = \left\{a_{ij} x_j + b_i\right\}^T$ – линейное преобразование над вектором входных данных, $A = \left\{a_{ij}\right\}$ – матрица весовых коэффициентов, $B = \left\{b_1, ..., b_n\right\}^T$ – сдвиг, $f = \left\{f_1\left(a_{1j} x_j + b_1\right), ..., f_n\left(a_{nj} x_j + b_n\right)\right\}$ – вектор-функция активация для слоя. Слой состоит из n нейронов, каждый из которых функционирует как отображение $X \to f_i\left(a_{ij} x_j + b_i\right)$, т. е. отображение $R^n \to R$. В общем случае для каждого слоя матрица линейного преобразования A, сдвиг B и вектор-функция активация f различны.

Известно, что решение задачи поиска вейвлет-разложения функций одной переменной может быть реализовано с помощью двухслойной нейронной сети [2]. При этом для первого слоя компоненты вектор-функции активации совпадают с материнским вейвлетом (2) или (3), т. е. $f_1 = ... = f_n = \psi_{ab}$. При решении задачи разложения по системе функций вида (3) использовалась библиотеки TensorFlow (https://www.tensorflow.org), реализующей работу нейронной сети. Нейросетевой алгоритм разложение по системе функций (2) был реализован самостоятельно. В обоих случаях алгоритм работы нейронной сети идентичен и представлен на рис. 1.



Рис. 1. Алгоритм работы нейронной сети

На вход нейронной сети (нулевой слой) подается таблица значений функции, подлежащей аппроксимации. Инициализируются весовые коэффициенты случайным образом. Если при решении задачи аппроксимации известны K значений исходной функции, то матрица A – квадратная и диагональная порядка $2K$. Количество нейронов на первом слое равняется $N \times 2K$, где N – количество членов разложения в ряд. На втором слое количество нейронов равняется $2K$ и вычисляется приближенное значение аппроксимируемой функции для каждой компоненты вектора входных данных X. Итого на каждой итерации (эпохе) вычисляется приближенное выражение

$$f_{calc}\left(x\right) = \sum_{i=1}^{N} c_i \psi_{a_i b_i}\left(x\right).$$

Оценивается ошибка разложения

$$\|f - f_{calc}\|_{L_2} = \sqrt{\int_{-1}^{1} \left(f(x) - f_{calc}(x)\right)^2 dx}. \quad (4)$$

Если погрешность является удовлетворительный, то процесс останавливается, если нет, то весовые коэффициенты, к которым относятся параметры $\{a_i, b_i\}$, изменяются. При этом решается задача минимизации функции многих переменных градиентным методом. В случае вейвлет-разложения по системе функций (2) для решения задачи многомерной оптимизации метод градиентного спуска реализовался самостоятельно, а при разложении по системе функций (3) использовался метод из библиотеки TensorFlow.

Тестовые расчеты с материнским вейвлетом (2) проведены при $f(x) = x$ (рис. 2), и $f(x) = x^2$ (рис. 3), ряд ограничивался в обоих случаях 11 членами. В обоих случаях потребовалась ≈300, погрешность составила 0.02. Сходимость алгоритма сильно зависит от начального приближения.

Если сравнивать с разложением в ряд Фурье (рис. 2, 3) (коэффициенты вычислены аналитически), то при том же количестве членов разложения 11, ошибка разложения будет составлять 1.7 и 0.02 соответственно. Чтобы добиться точности разложения порядка 0.02 как при расчетах (рис. 2), необходимо использовать 5000 членов разложения в ряд Фурье.

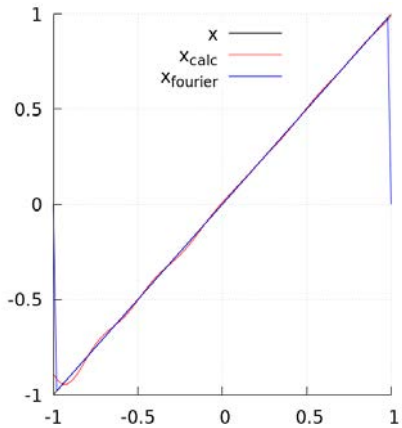


Рис. 2. Результаты расчетов разложения $f(x) = x$

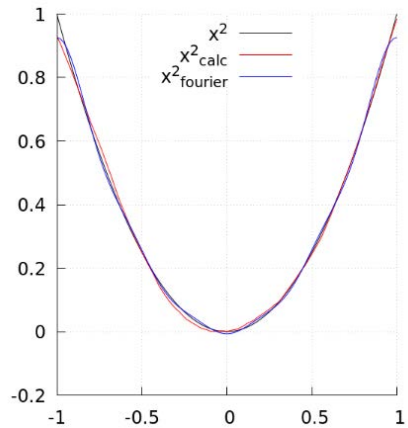


Рис. 3. Результаты расчетов разложения $f(x) = x^2$

Далее приводятся результаты апробации алгоритма (рис. 1) для материнского вейвлета (3), полученные при использовании библиотеки TensorFlow. Результаты работы алгоритма сравнивались с работой готовой библиотекой поиска вейвлет-разложения PyWavelets (https://pywavelets.readthedocs.io), в частности, в этой работе использовалась возможность библиотеки нахождения дискретного вейвлет-разложения. Результаты представлены в табл. 1 и сравнивались по нескольким показателям: $h = \dfrac{b-a}{K-1}$ – шаг сетки, $b-a$ есть длина интервала области определения аппроксимируемой функции, а K – число узлов разбиения отрезка $[a, b]$; точность вычисления ε_{HC} для нейронной сети и ε_{PW} для библиотеки PyWavelets , оценивалась согласно (4); M – количество эпох (итераций алгоритма). Количество членов разложения в ряд $N = 16$.

На рис. 4 представлены результаты работы нейронной сети по решению задачи аппроксимации, соответствующей последней строчке табл. 1, а на рис. 5 – аппроксимация с помощью библиотеки PyWavelets. Результаты табл. показывают, что точность разложения, получаемого нейронной сетью совсем немного уступает работе библиотеки PyWavelets на мелких сетках, но в отличие от нее на более крупной сетке (меньшее число нейронов) ее точность практически не меняется, в то время как, для библиотеки PyWavelets точность ухудшается в десять раз.

Таблица 1

Погрешность решения задачи аппроксимации нейронной сетью

Функция	h	M	ε_{HC}	ε_{PW}
$f(x)=x$	0.002	100	0.001	0.001
$f(x)=x$	0.02	1000	0.002	0.01
$f(x)=x^2$	0.002	100	0.004	0.001
$f(x)=x^2$	0.02	1000	0.004	0.01
$f(x)=x^3$	0.002	100	0.005	0.002
$f(x)=x^3$	0.02	1000	0.01	0.02
$f(x)=\sin x$	0.002	100	0.004	0.0007
$f(x)=\sin x$	0.02	1000	0.006	0.007

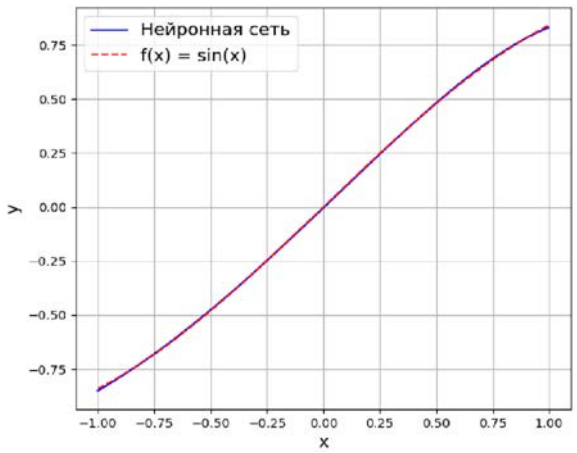


Рис. 4. Аппроксимация нейронной сетью функции $f(x)=\sin x$

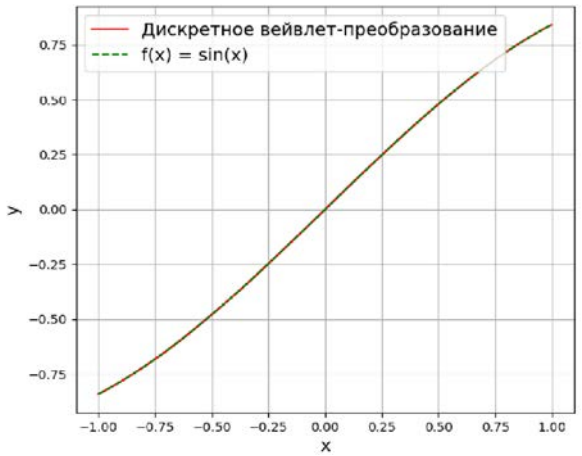


Рис. 5. Аппроксимация библиотекой PyWavelets функции $f(x)=\sin x$

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (Проведение фундаментальных научных исследований (47 ГП) по теме № 0065-2019-0007 "36.20 Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления." (№АААА-А19-119011590093-3)).

Литература

1. Яковлев А. Н. Введение в вейвлет-преобразование: Учеб. пособие. – Новосибирск: Изд-во. НГТУ. 2003.

2. Терехов С. А. Вейвлеты и нейронные сети // Научная сессия МИФИ – 2001. III Всерос. начн.-техн. конф. «Нейроинформатика-2001»: лекции по нейроинформатике. М.: МИФИ. 2001. С. 142–181.

3. Нагорнов. О. В. и. др. Вейвлет-анализ в примерах: учебн. пособие. М.: НИЯУ МИФИ. 2010.

NEURAL NETWORK ALGORITHM FOR WAVELET TRANSFORMATION OF FUNCTIONS OF ONE VARIABLE

V. B. Betelin[1], *V. A. Galkin*[2], *A. O. Dubovik*[2]

[1] Scientific Research Institute for System Analysis RAS, Moscow
[2] Surgut Branch, SRISA RAS, Surgut

This paper examines aspects of the construction of neural networks by the example of solving the problem of finding a wavelet decomposition for functions of one variable. The neural network is built using the TensoFlow machine learning library. The results of the work showed that the created neural network copes more accurately with the solution of the problem of constructing a wavelet decomposition than the existing ready-made libraries with a small set of input data.

Key words: neural networks, wavelet transform.

УДК 519.6
DOI 10.53403/9785951505071_2022_66

ПАКЕТ ПРОГРАММ ЛОГОС. МЕТОДЫ МОДЕЛИРОВАНИЯ КИНЕМАТИКИ НЕДЕФОРМИРУЕМЫХ ТВЕРДЫХ ТЕЛ И МЕХАНИЗМОВ

В. В. Борляев, В. А. Афанасьев, Д. Ю. Дьянов, И. В. Серякова, М. В. Беженцева

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Доклад посвящен описанию подхода моделирования недеформируемых твердых тел и механизмов в пакете программ ЛОГОС. В докладе приведено описание общих принципов построения аппроксимаций для моделирования твердых тел и механизмов, описаны подходы для моделирования шарнирных соединений, а также методы сопряжения деформируемых и недеформируемых объектов.

Для демонстрации работоспособности реализованного функционала в докладе представлены результаты численных расчетов ряда тестовых и верификационных задач, проведенных с использованием пакета программ ЛОГОС.

Ключевые слова: пакет программ ЛОГОС, программный модуль ЛОГОС-ПА, метод конечных элементов, недеформируемое твердое тело, шарнирные соединения.

Введение

Во ФГУП «РФЯЦ-ВНИИЭФ» ведется разработка расчетно-математического пакета программ ЛОГОС, предназначенного для инженерного анализа и суперкомпьютерного моделирования в интересах высокотехнологичных отраслей промышленности. Одной из составных частей пакета программ ЛОГОС является программный модуль (далее – ЛОГОС-ПА) для моделирования двумерных и трехмерных задач анализа напряженно-деформированное состояние конструкций при динамических, статических, вибрационных нагрузках [1, 2].

Одним из условий успешного внедрения пакета программ ЛОГОС в полный цикл разработки изделий на предприятиях является наличие широкого набора функциональных возможностей для моделирования задач динамического деформирования. Базовым способом численного решения указанного класса задач в ЛОГОС-ПА является использование лагранжевого подхода на основе метода конечных элементов для аппроксимации по пространству при явной конечно-разностной аппроксимации по времени [3, 4]. При решении динамических задач деформирования конструкций часто возникает потребность в моделировании кинематики движения твердых тел и механизмов, а также возможность связного моделирования движения деформируемых и недеформируемых твердых тел.

На текущий момент в программном модуле ЛОГОС-ПА существуют два подхода моделирования недеформируемых твердых тел и механизмов. Первый подход основан на использовании программного комплекса EULER [5]. Данный программный комплекс имеет богатый функционал для моделирования кинематики недеформируемых твердых тел и механизмов и связан с пакетом программ ЛОГОС средствами интеграционной платформы ЛОГОС для проведения, в том числе, связанных расчетов. Ограничением для использования данного подхода является то, что связь осуществляется только через специальные интерфейсные узлы и соответственно не учитывается контактное взаимодействие недеформируемых тел с деформируемой частью модели ЛОГОС. Второй подход основан на собственной реализации расчета кинематики недеформируемых твердых тел непосредственно в ЛОГОС-ПА. В данной работе речь пойдет о втором подходе.

В 2018 году в рамках контракта с КНИТУ-КАИ был проведен анализ возможностей пакета программ ЛОГОС для расчетов задач по моделированию крэш-тестов грузовых автомобилей. По результатам этого анализа была создана верификационная база данных различных задач, правильное решение которых необходимо для полноценного моделирования крэш-теста автомобиля. Данная база данных содержит постановки и результаты решения различных задач по пакету программ ЛОГОС и LS-Dyna [6]. Все результаты, приводимые в статье, по пакету программ LS-Dyna взяты из базы данных верификационных задач.

На текущий момент в пакете ЛОГОС-ПА реализован следующий функционал по работе с недеформируемыми объектами:

– объединение нескольких твердых тел в одно;
– расчет узлового твердого тела;
– различные шарнирные соединения между недеформируемыми телами;
– ограничение поступательного или вращательного движения;
– навязанное перемещения или вращения;
– навязанное положение центра масс, тензора инерции;
– инерционная нагрузка.

Расчет движения недеформируемого тела

Уравнения движения недеформируемого тела записываются через движение центра масс и выглядят следующим образом [7]:

$$M\ddot{\vec{x}} = \overrightarrow{f^x}, \tag{1}$$

$$J\dot{\vec{\omega}} + \vec{\omega} \times J\vec{\omega} = \overrightarrow{f^\omega}. \tag{2}$$

Здесь M – масса недеформируемого объекта, J – тензор инерции, $\overrightarrow{f^x}$ – результирующая сила, $\overrightarrow{f^\omega}$ – результирующий момент, $\vec{x}$ – координаты центра масс, $\vec{\omega}$ – угловая скорость.

$$I_{ij} = \Sigma_k \left(m_k \left(\delta_{ij} r_k^2 - r_{ik} r_{jk} \right) \right). \tag{3}$$

Здесь m_k – масса узла k, r_{ik} – i-ая компонента радиус-вектора узла k, δ_{ij} – символ Кронекера.

При переходе в локальную систему координат, расположенную в точке центра масс, уравнения (1) и (2) примут вид:

$$M\ddot{\vec{x}} = \overrightarrow{f^x}, \tag{4}$$

$$J\dot{\vec{\omega}_c} = \overrightarrow{f_c^\omega}. \tag{5}$$

Положение центра масс вычисляется следующим образом:

$$\overrightarrow{x_c} = \frac{\Sigma_k m_k \overrightarrow{r_k}}{M}. \tag{6}$$

Из уравнений (4), (5) получаем поступательные и вращательные ускорения для центра масс:

$$\ddot{\vec{x}} = \overrightarrow{f^x}/M, \tag{7}$$

$$\dot{\vec{\omega}_c} = J^{-1}\overrightarrow{f_c^\omega}. \tag{8}$$

Далее вычисляются изменения поступательной и угловой координаты Δx и $\Delta\theta$ для находится новое положение центра масс и всех точек недеформируемого тела.

$$x_i^{n+1} = x_i^n + \Delta x + (A - I) r_i^n, \tag{9}$$

где A – матрица преобразования, рассчитываемая по формуле Hughes-Winget [6, 8]:

$$A = I + \frac{1}{1 + \frac{\Delta\theta^T \Delta\theta}{4}} \left(I + \frac{1}{2}\Delta S \right) \Delta S, \tag{10}$$

$$\Delta Sr = \Delta\theta \times r. \tag{11}$$

Новые скорости в узлах тела находятся по формуле

$$\dot{x}_i = \frac{(x_i^{n+1} - x_i^n)}{\Delta t}. \tag{12}$$

На каждом шаге по времени происходит изменение тензора инерции:

$$J^{n+1} = AJ^nA^T, \tag{13}$$

Шарнирные соединения

Шарнирное соединение – кинематическая пара, то есть подвижное соединение двух частей конструкции, которое обеспечивает им вращательное и/или поступательное движение относительно друг друга [6, 9]. Шарниры используются для задания кинематических ограничений между различными твердыми телами. Основные виды шарнирных соединений представлены на рис. 1: сферический (шаровой) (а), поворотный (шарнирная петля) (б), цилиндрический (в), плоский (г), универсальный (д) (кардан) и поступательный (е).

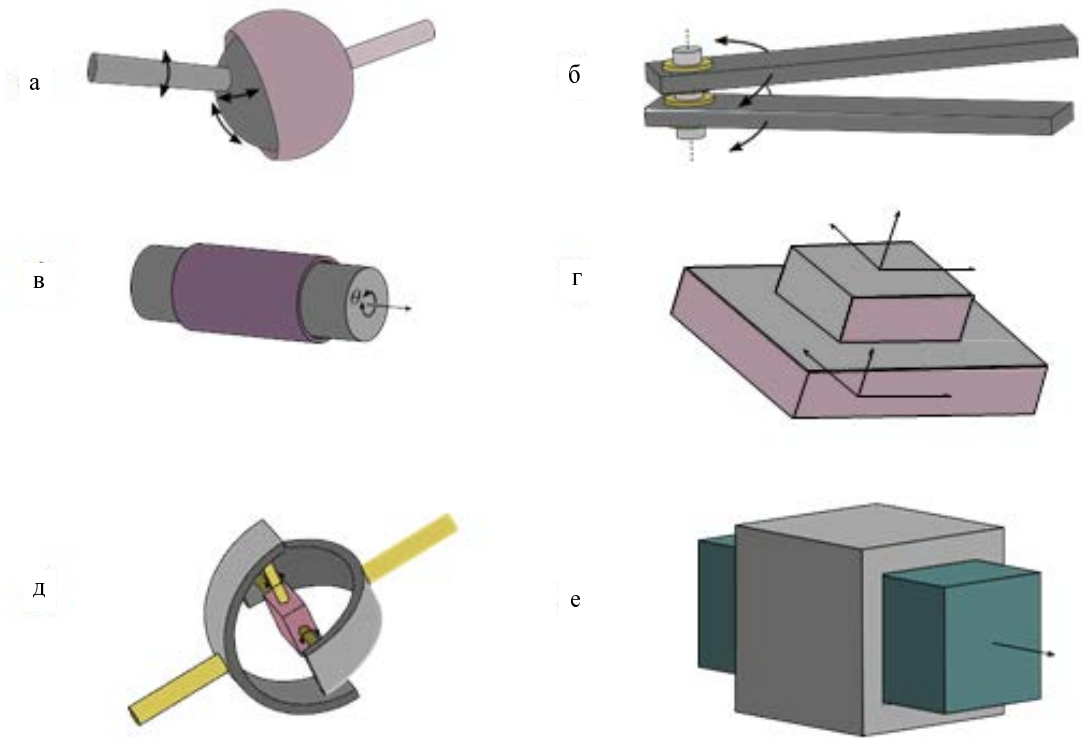


Рис. 1. Примеры реализованных шарнирных соединений

Основным методом моделирования шарнирного соединения в явных схемах является использование метода штрафных функций (метод штрафа [6]). Сила штрафа для узла k, возникающая в результате работы шарнирного соединения, вычисляется по формуле

$$f_k = \frac{K \cdot d_k}{\tau^2}, \tag{14}$$

где K – коэффициент жесткости, d_k – расстояние отклонения узла шарнира от заданного положения, τ – шаг интегрирования по времени.

Задание жесткостных и демпфирующих ограничений для шарнирных соединений

В ЛОГОС-ПА реализован метод, который позволяет отслеживать изменение положения недеформируемых тел при их вращении друг относительно друга [6]. Это бывает необходимо, например, для корректного описания поведения суставов манекена, задания трения при работе шарнирного соединения.

Для каждой из трех осей координат можно задать допустимые углы поворота, при превышении которых будет появляться момент в обратном направлении. Диаграмма зависимости момента от угла приведена на рис. 2.

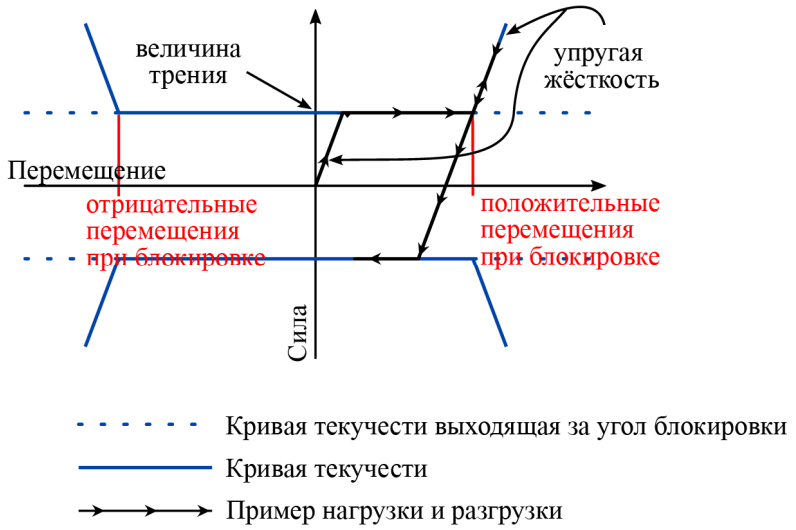


Рис. 2. Модель трения и жесткости для шарнирных соединений

Пример расчета недеформируемых тел с использованием шарниров

На рис. 3 приведена система недеформируемых тел, соединенных шарнирами.

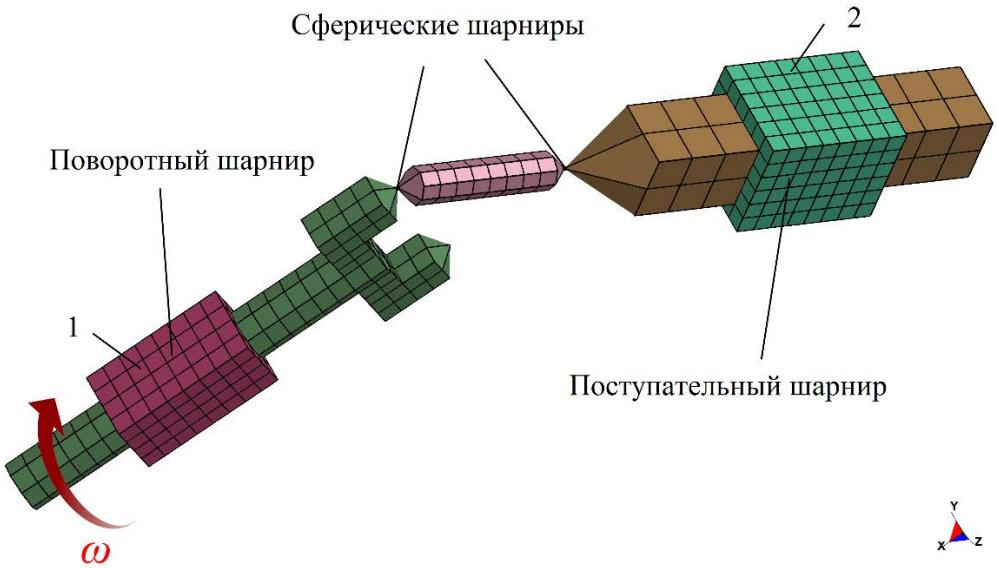


Рис. 3. Модель для тестирования шарниров

Детали 1 и 2 неподвижны. В данной модели используются два сферических, поворотный и поступательный шарниры. На входном валу (деталь слева) задается начальная угловая скорость $\omega = 50$ рад/с. В результате работы данного механизма вращательное движение преобразуется в поступательное. На рис. 4 приведена зависимости результирующей поступательной скорости на выходном звене (деталь справа), полученные по LS-DYNA [6] и ЛОГОС-ПА.

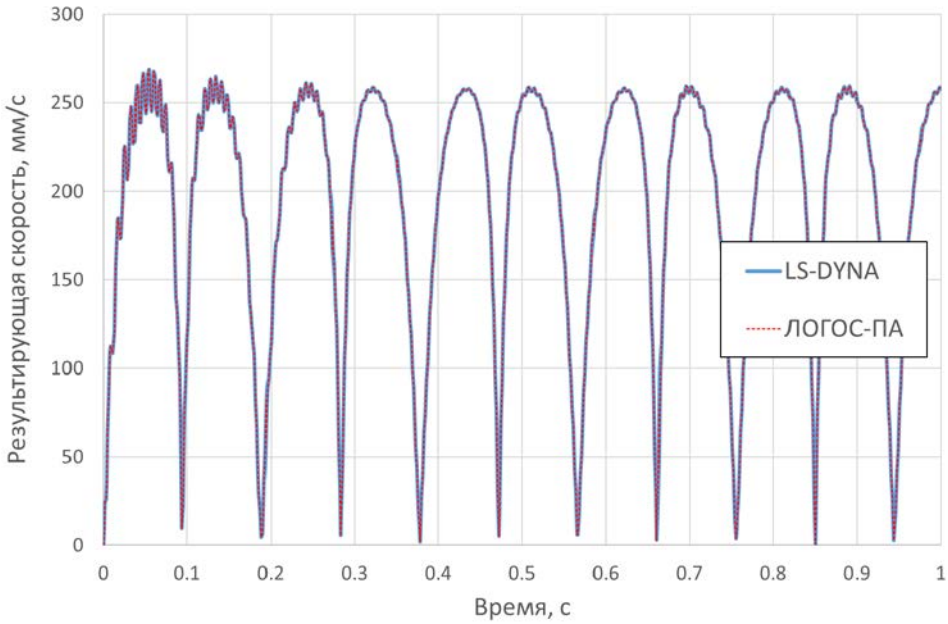


Рис. 4. Зависимость поступательной скорости выходного звена механизма, полученные в LS-DYNA и ЛОГОС

Из рис. 4 видно, что результаты моделирования по LS-DYNA и ЛОГОС-ПА имеют хорошее совпадение. Анализ графиков показал, что разница результатов не превышает 0.01%.

Численное моделирование крэш-теста прототипа автомобиля-тягача

В задаче из верификационной базы данных рассматривается моделирование крэш-теста прототипа автомобиля-тягача. Автомобиль движется с начальной скоростью 30 км/ч и взаимодействует с неподвижным жестким препятствием, расположенным под углом 60° к направлению движения. Постановка задачи представлена на рис. 5.

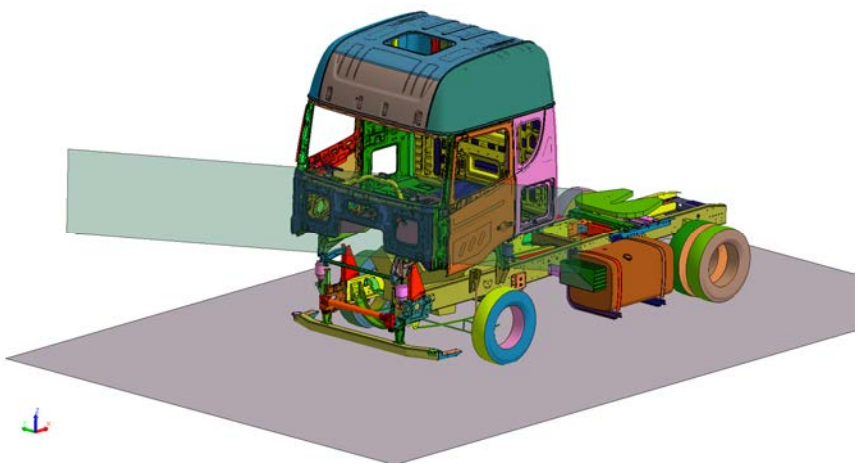


Рис. 5. Постановка задачи

Модель содержит 1 131 089 узел, 1149338 элементов, из которых оболочечных – 927 344, балочных – 4 449, объемных – 217539, дискретных – 6, массовых – 78, жестких узловых связей – 998. В задаче используются как недеформируемые, так и деформируемые элементы конструкции.

Шарниры навесов дверей кабины были смоделированы путем задания недеформируемых систем узлов, представленных на рис. 6.

В каждую такую систему входили узлы кронштейнов навесов дверей и пары узлов на осях шарниров. Пары узлов на осях шарниров навесов использовались для задания цилиндрических шарниров, изображенных на рис. 7.

Болтовые соединения моделировались путем объединения узлов по контуру отверстий жесткой связью, как показано на рис. 8. Болтовые соединения литых кронштейнов (моделировались трехмерными элементами) задавались аналогично, жестким связыванием всех узлов по поверхности отверстий.

Конечно-элементная модель несущей системы с силовым агрегатом, передним противоподкатным защитным устройством, седельным устройством, топливными баками, инструментальным ящиком и колесами изображены на рис. 9. Штампованные детали конструкции и шины моделировались оболочечными конечными элементами, литые кронштейны, двигатель и седло – объемными конечными элементами.

Двигатель с кронштейнами и его крепления на раме задавалась моделями абсолютно жесткого тела.

Расчет выполнялся на 84 процессорах в параллельном режиме в модели распределенной памяти с использованием интерфейса передачи сообщений MPI [10], время счета 2 суток.

Деформирование грузового автомобиля, полученные по LS-DYNA и ЛОГОС-ПА на моменты времени t = 0.06 с и t = 0.09 с показаны на рис. 10–11 и 12–13 соответственно. Из приведенных рисунков видно, что результаты расчетов качественно совпадают между собой.

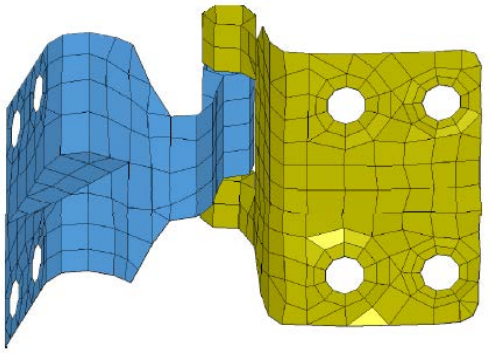


Рис. 6. Конечно-элементная модель навеса двери на кабину

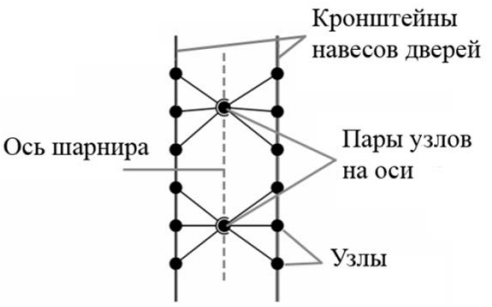


Рис. 7. Схема моделирования шарниров навески дверей

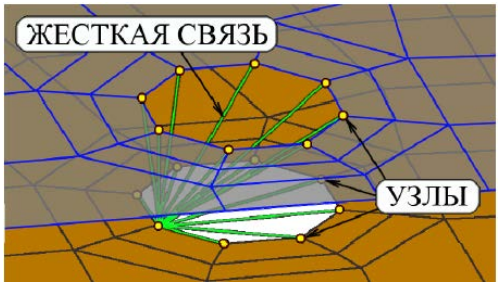


Рис. 8. Моделирование болтового соединения тонкостенных деталей

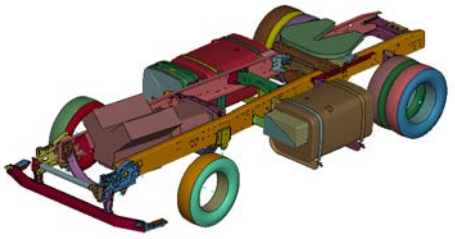


Рис. 9. Конечно-элементная модель несущей системы с силовым агрегатом, ППЗУ, Баками

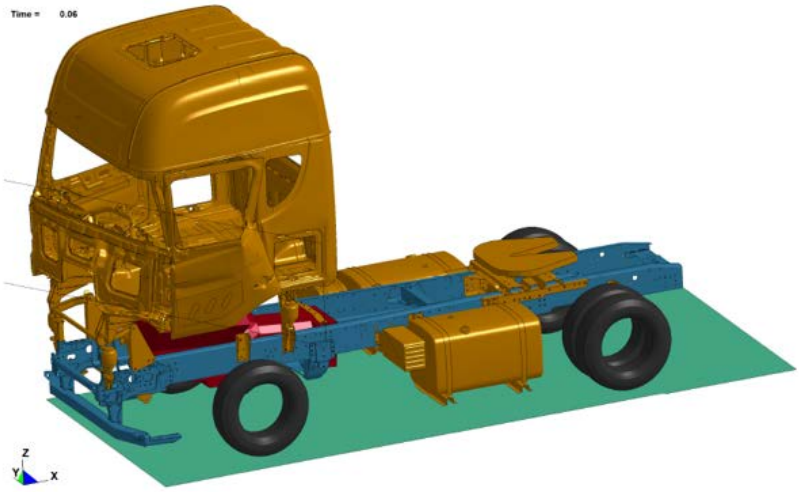


Рис. 10 . Деформированное состояние на момент времени t = 0.06 с, LS-DYNA

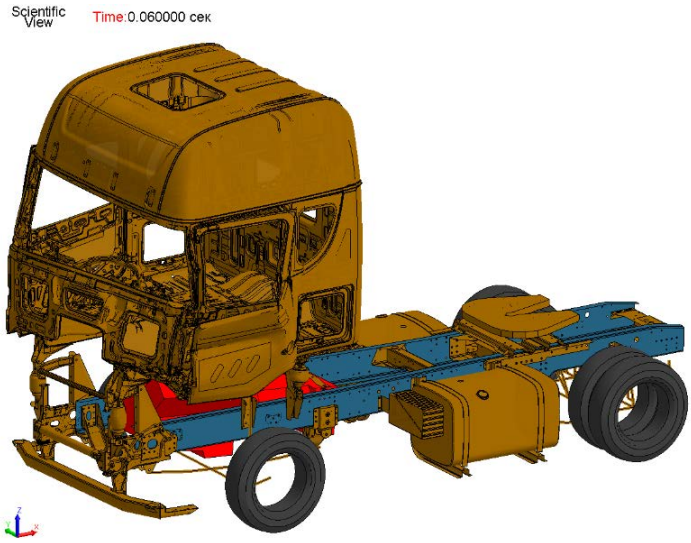


Рис. 11. Деформированное состояние на момент времени t = 0.06с, ЛОГОС-ПА

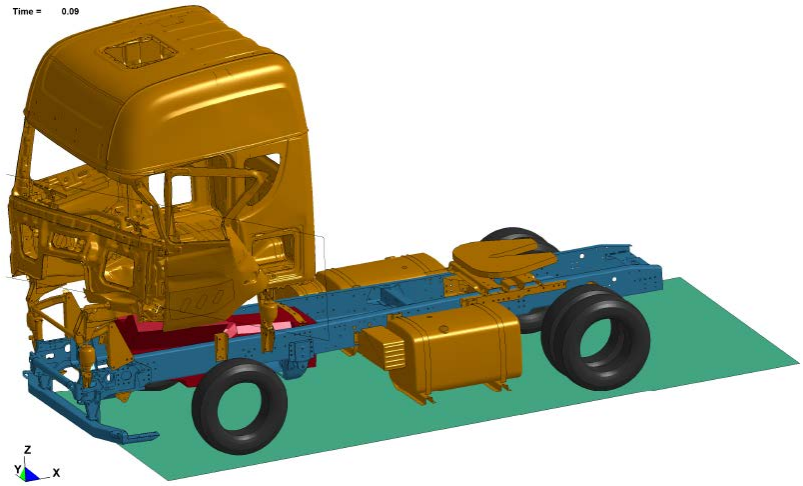


Рис. 12. Деформированное состояние на момент времени t = 0.09с, LS-DYNA

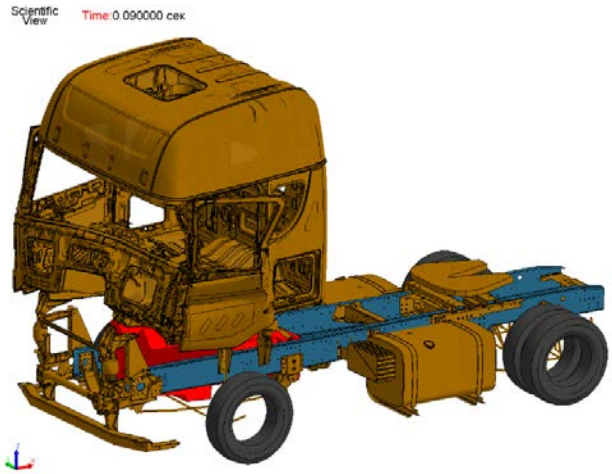


Рис. 13. Деформированное состояние на момент времени $t = 0.09$ с, ЛОГОС-ПА

Заключение

Методы и алгоритмы, описанные в рамках данной работы, предназначены для моделирования в программном модуле ЛОГОС-ПА движения недеформируемых тел как отдельно, так и в составе единой конструкций.

В работе приведено краткое описание алгоритмов моделирования движения недеформируемых тел и шарнирных соединений в пакете программ ЛОГОС. Приведены результаты расчета верификационных задач, подтверждающих работоспособность реализованных алгоритмов.

Реализованные возможности позволяют существенно расширить класс решаемых задач.

Литература

1. Циберев К. В., Авдеев П. А., Артамонов М. В., Борляев В. В., Величко С. В., Волков А. Ю., Володина Н. А., Дьянов Д. Ю., Корсакова Е. И., Косарим С. С., Кулыгина О. Н., Мышкина И. Ю., Наумов А. О., Присташ М. М., Резвова Т. В., Резяпов А. А., Родионов А. В., Симонов Г. П., Спиридонов В. Ф., Стародубов С. В., Тарадай И. Ю., Филимонкин Е. А., Челаков А. А., Шувалова Е. В., и др. Пакет программ ЛОГОС. Обзор текущих возможностей для решения задач прочности // XIV международный семинар «Супервычисления и математическое моделирование»: Сборник научных трудов. – Саров, 2013. С. 599–609.

2. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В., Наумова Е. И., Борляев В. В., Стародубов С. В., Шувалова Е. В., Медведкина М. В., Артемова Е. О., Челаков А. А., Казанцев А. В., Рябов А. А., Романов В. И., Куканов С. С. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. 2018. Вып.1. С. 3–13.

3. Дьянов Д. Ю., Казанцев А. В., Циберев К. В. и др. Пакет программ ЛОГОС. Функциональные возможности для решения динамических задач прочности // Супервычисления и математическое моделирование: Труды XVII международного семинара, Саров, 15-19 октября, 2018. С. 210–219.

4. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В. и др. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 2. С. 3–13.

5. Бойков В. Г., Юдаков А. А. Моделирование динамики системы твердых и упругих тел в программном комплексе EULER // Информационные технологии и вычислительные системы. 2011. № 1. С. 42–52

6. LS-DYNA [Electronic resource]: Livermore Software Technology Corp. –Mode of Access: http://www.lstc.com/products/ls-dyna.

7. Бутенин Н. В., Лунц Я. Л., Меркин Д. Р. Курс теоретической механики: Учебник. В двух томах. Т. II.: Динамика. – 2-е изд., перераб. и доп. М.: Наука. Главная редакция физико математической литературы, 1979. С. 544.

8. Benson D. J., Hallquist J. O. A simple rigid body algorithm for structural dynamics programs // Int. J. Numer. Meth. Eng. 1986. Vol. 22. P. 723–749.

9. Артоболевский И. И. Теория механизмов и машин: Учеб. для втузов. – 4-е изд., перераб. и доп. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 640 с.

10. MPIDocuments [Electronic resource]. Mode of Access: http://www.mpi-forum.org/docs/docs.html.

LOGOS SOFTWARE PACKAGE: SIMULATION METHODS FOR THE KINEMATICS OF RIGID BODIES AND DEVICES

V. V. Borlyaev, V. A. Afanasyev, D. Yu., Dyanov, I. V. Seryakova, M. V. Bezhentseva

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents an approach to simulating rigid bodies and devices in the LOGOS software package. General ideas of approximations used to simulate rigid bodies and devices and approaches to simulating hinge joints are described, as well as the methods of conjugating deformable and non-deformable parts.

To demonstrate the efficiency of the implemented functionality, the paper presents the LOGOS software package results of numerical simulations for a number of test and varication problems.

Key words: the LOGOS software package, the LOGOS-SA program module, the finite element method, rigid body, hinge joints.

УДК 519.6
DOI 10.53403/9785951505071_2022_75

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ СЕЛЕКТИВНОГО ЛАЗЕРНОГО СПЕКАНИЯ ПОРОШКОВОЙ СМЕСИ

А. Н. Быков, М. В. Ветчинников, И. С. Сатронов, Ю. Н. Дерюгин, А. А. Лазарев

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В работе сформулирована математическая модель для описания процессов, протекающих в аддитивном производстве. Модель основана на континуальном описании плавления порошкового материала под действием лазерного излучения. Численное решение находится с использованием метода сглаженных частиц. Формирование рабочего слоя порошка выполняется по одному из двух методов: простой геометрический метод и метод дискретных элементов.

Ключевые слова: селективное лазерное спекание порошковой смеси, слой порошка, геометрический метод, метод дискретных элементов, математическая модель, уравнения Навье-Стокса, метод сглаженных частиц.

Введение

В основе метода селективного лазерного спекания (СЛС) порошков лежит плавление порошковой смеси лучом лазера, когда смесь порошков с различными температурами плавления подвергают тепловой обработке. В результате лазерного нагрева создается материал со сложной структурой, в которой тугоплавкие керамические или металлические частицы связываются посредством более легкоплавкой органической или металлической матрицы. Для обеспечения высокого качества изготавливаемых изделий и высоких механических прочностных и износостойких характеристик используется метод лазерного плавления.

При разработке программного обеспечения для моделирования физических процессов, протекающих при СЛС, одним из необходимых этапов является создание модуля формирования рабочего слоя порошка. От качества моделирования насыпки порошка и соответствия полученных физических свойств реальных порошков во многом зависит качество дальнейшего моделирования СЛС. Рассматриваются два метода формирования рабочего слоя порошка: простой геометрический метод (модификация так называемого «метода дождя») и метод дискретных элементов (Discrete Element Method – DEM), которые были реализованы в рамках проекта по созданию программного комплекса «Виртуальный 3D принтер».

Для численного моделирования СЛС порошков используется метод сглаженных частиц (метод SPH). Рассматриваются слабо сжимаемые течения жидкой фазы, образующейся в результате плавления. При описании данных течений используются уравнения Навье-Стокса, в которых учитываются силы давления, вязкие силы и силы поверхностного натяжения на контактных поверхностях жидкой и твердой фаз, жидкой фазы и внешней среды. Тепловое состояние определяется из уравнения сохранения энергии, в котором учитываются тепловые процессы, объемное поглощение энергии лазерного излучения, конвективный теплообмен с внешней средой, радиационное излучение и работа вязких сил. Фазовые переходы между твердой и жидкой фазами описываются в рамках обобщенной формулировки задачи Стефана.

1. Формирование слоя порошка

Для моделирования плавления слоя порошка методом SPH (или другими) на каждом шаге послойного выпекания требуется разработка и реализация модуля формирования слоя порошка. Алгоритмы, реализующие насыпку порошка, могут быть различными. Наиболее простым является геометрический подход – такой, при котором гранулы порошка считаются несжимаемыми сферами (вообще говоря, различного диаметра), а процесс их насыпки описывается только из соображений геометрии и статики. Другой, более точной, группой методов являются методы насыпки, учитывающие некоторые физические свойства насыпаемого порошка. Одним из вариантов таких подходов является DEM (Discrete Element Method) [1].

1.1. Геометрический подход

В свою очередь, геометрические алгоритмы насыпки могут различаться подходами к формированию порошкового слоя. С 80-х гг. XX в. известен так называемый «rain-метод» [2]. Суть метода состоит в генерации сфер над областью их насыпки в произвольном месте и моделировании падения сферы итерациями по времени с небольшим временным шагом до касания падающей сферы первого встреченной сферы, затем ее поворотом в вертикальной плоскости вокруг встреченной сферы до касания второй сферы, затем ее поворотом относительно отрезка, соединяющего две точки касания, до касания третьей сферы. Если после третьего касания проекция центра тяжести падающей сферы окажется внутри треугольника, образованного точками касания, сфера считается упавшей окончательно. В противном случае она продолжает «скатываться» в направлении градиента силы тяжести. Аналогично, сфера считается окончательно упавшей в случае ее касания с плоскостью подложки (обычно это плоскость $Z = 0$).

В нашем случае реализован вариант «rain-метода», отличие которого от оригинального состоит в том, что очередная сфера генерируется не в произвольном месте над подложкой, а в некотором наперед заданном объеме над плоскостью подложки. Это позволяет, задавая толщину данного объема (Zmin <= Zцентра_частицы <= Zmax), регулировать толщину насыпаемого слоя и позволяет сделать алгоритм более эффективным по времени работы за счет исключения расчета падения сферы до подложки.

В программе геометрической насыпки реализована возможность выбора распределения диаметров сфер (константа, равномерное случайное распределение по отрезку, Гауссово нормальное (одномодальное и двухмодальное) распределение с заданными математическими ожиданиями и дисперсиями), размеров заполняемой сферами области насыпки и требуемой насыпной плотности порошка.

Для проверки работоспособности геометрического метода насыпки были взяты данные из статьи LLNL [3]. Начальные данные формирования слоя порошка следующие:

– насыпка порошка в 1 слой;

– размер области насыпки: 1000×300 μm;

– гранулометрический состав порошка - Гауссово нормальное распределение диаметров сфер с математическим ожиданием $\mu = 27$ μm и дисперсией $\sigma = 0{,}1$;

насыпная плотность ≈ 40 %.

Результаты сравнения приведены на рис. 1:

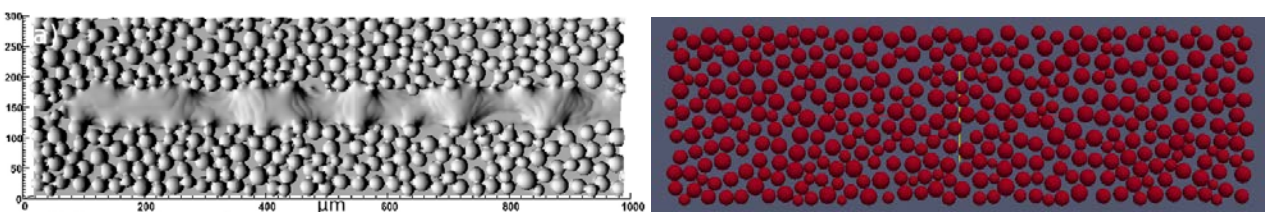


Рис. 1. Результаты работы геометрического метода насыпки (слева) в сравнении с данными из статьи [3] (справа)

1.2. Метод дискретного элемента (DEM)

Метод дискретного элемента (DEM) основан на моделировании сплошного вещества посредством твердых частиц с 6 степенями свободы (3 вращательных и 3 поступательных). Законы взаимодействия между дискретными элементами определяются макроскопическим поведением моделируемого материала. В ранних работах взаимодействия частиц определялись законами трения для не связанных материалов типа песка [4]. Законы взаимодействия для связанных материалов были определены в дальнейших работах.

Метод дискретного элемента весьма эффективно описывает поведение гранулированных сред, состоящих из частиц произвольного размера и формы. В настоящее время данный метод получил широкое распространение при исследованиях в механике сыпучих сред и горных пород.

В работе [5] материал представляется в виде совокупности из N обособленных частиц сферической формы с заданным распределением радиусов $R_i (i = 1,\ldots \mathrm{N})$ и набором физических свойств (плотность ρ_i, упругие и вязкие константы, трение и т. д.). Напряженно-деформированное состояние такой системы в каждый момент времени зависит от положения каждой сферической частицы, от сил, возникающих на контактах между ними, а также от приложенных к системе внешних сил.

Взаимодействие между упругими сферическими частицами приводит к их деформации и искажению формы, но в методе дискретного элемента форма частиц на протяжении всего времени моделирования остается неизменной, а степень их деформации представляется величиной перекрытия между контактирующими элементами (рис. 2), где R_{ij} – расстояние между центрами масс сфер. При этом предполагается, что размер такого перекрытия намного меньше размеров самих сфер.

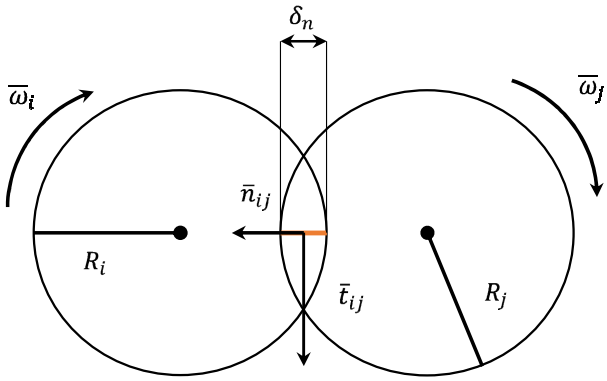


Рис. 2. Схематическое изображение контактирующих частиц сферической формы

Геометрия контакта описывается векторами $\mathbf{n}_{ij}$ и $\mathbf{t}_{ij}$, где $\mathbf{n}_{ij}$ – единичный вектор, определяющий плоскость контакта между двумя сферами (вектор нормали к плоскости пересечения сфер, направленный вдоль линии, соединяющей центры); $\mathbf{t}_{ij}$ – единичный вектор, принадлежащий плоскости контакта:

$$\bar{t}_{ij} = \frac{\bar{u}_{ij} - (\bar{u}_{ij}\bar{n}_{ij})\bar{n}_{ij}}{\left|\bar{u}_{ij} - (\bar{u}_{ij}\bar{n}_{ij})\bar{n}_{ij}\right|},$$

где $\bar{u}_{ij} = \bar{u}_i - \bar{u}_j$.

В общем случае движение каждой i-й сферы состоит из поступательного и вращательного и в момент времени t описывается в рамках второго закона Ньютона:

$$m_i \frac{d^2 \bar{r}_i}{\partial t^2} = \sum_j \left(\bar{F}_{ij}^n + \bar{F}_{ij}^t + \bar{F}_{ij}^v\right) + m_i \bar{g}$$

Здесь $\mathbf{r}_i$ – координаты центра масс, $\boldsymbol{\omega}_i$ – угол поворота, m_i – масса, I_i – момент инерции i-й сферы.

$$I_i \frac{d\bar{\omega}_i}{\partial t} = \sum_j \left(\bar{T}_{ij}^t + \bar{T}_{ij}^r \right)$$

Правые части уравнений состоят из нормальной силы $\bar{F}_{ij}^n$ (действующей вдоль вектора $\mathbf{n}_{ij}$), тангенциальной силы $\bar{F}_{ij}^t$ (действующей вдоль вектора $\mathbf{t}_{ij}$), и сил Ван-дер-Ваальса $\bar{F}_{ij}^v$, действующих на контактах, а также внешних сил mig (сила тяжести) и записываются следующим образом:

$$\bar{F}_{ij}^n = \left[k_{ij}^n \delta_n^{3/2} - \gamma_{ij}^n (\bar{u}_{ij} \bar{n}_{ij}) \right] \bar{n}_{ij}$$

$$\bar{F}_{ij}^t = \left[-k_{ij}^t |\xi_t| - \gamma_{ij}^t (\bar{u}_{ij} \bar{t}_{ij}) \right] \bar{t}_{ij}$$

$$\bar{F}_{ij}^v = -\frac{H_a}{6} \frac{64 R_i^3 R_j^3 (h + R_i + R_j)}{(h^2 + 2R_i h + 2R_j h)^2 (h^2 + 2R_i h + 2R_j h + 4R_j R_i)^2} \bar{n}_{ij}$$

где $k_{ij}^t = \frac{2E}{(2+\nu)(1-\nu)} \sqrt{R^* \delta_n}$, $k_{ij}^n = \frac{2E}{3(1-\nu^2)} \sqrt{R^*}$, $R^* = \frac{R_i R_j}{R_i + R_j}$, $\gamma_{ij}^n = -2\sqrt{\frac{5}{6}} \beta \sqrt{S_n \frac{m_i}{2}}$, $\gamma_{ij}^t = -2\sqrt{\frac{5}{6}} \beta \sqrt{S_t \frac{m_i}{2}}$, $S_n = \frac{E}{1-\nu^2} \sqrt{R^* \delta_n}$, $S_t = \frac{2E}{(2+\nu)(1-\nu)} \sqrt{R^* \delta_n}$, $\beta = \frac{\ln(e)}{\sqrt{\ln^2(e) + \pi^2}}$, $h = R_{ij} - R_i - R_j$,

E – модуль Юнга, ν – коэффициент Пуассона, e – коэффициент восстановления.

Причем, если $|\bar{F}_{ij}^t| \geq \mu |\bar{F}_{ij}^n|$, тогда для сферы i тангенциальная сила вычисляется как $\bar{F}_{ij}^t = -\mu |\bar{F}_{ij}^n| \bar{\tau}_{ij}$, где μ – коэффициент трения скольжения.

Моменты вращения $\bar{T}_{ij}^t$, $\bar{T}_{ij}^r$ характеризуют вращательное воздействие тангенциальной силы $\bar{F}_{ij}^t$ и сил трения качения на i-ю сферу со стороны j-й и записываются следующим образом:

$$\bar{T}_{ij}^t = -\bar{R}_i (\bar{n}_{ij} \times \bar{F}_{ij}^t),$$

$$\bar{T}_{ij}^r = -\mu_r \bar{R}_i \left| \bar{F}_{ij}^n \right| \varpi_i,$$

где μ_r – коэффициент трения качения, а $\bar{\omega}_t = \bar{\omega}_t / |\bar{\omega}_t|$.

Была реализована тестовая двумерная программа и в последующем проведена реализация в комплексе программ молекулярной динамики MoDyS[6].

Результаты моделирования «насыпания» сфер с различным радиусом и одинаковыми физическими свойствами в прямоугольник по тестовой программе приведены на рис. 3.

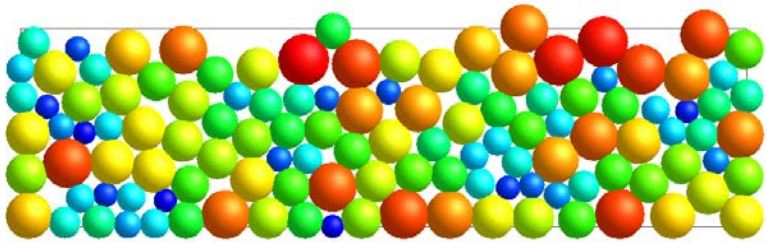


Рис. 3. Результат работы DEM на тестовой двумерной программе, цветовая гамма соответствует радиусу сферы

Параметры для моделируемого вещества приведены в табл. 1.

Таблица 1

Параметры моделируемого вещества

E, Н/м^2	ν	μ	e	ρ, кг/м^3	На, Дж
10^7	0.3	0.3	0.95	2500	$6.5 \cdot 10^{-20}$

На рис. 4 и 5 представлены результаты работы программы, реализующей DEM, для сфер с одинаковыми и 3 различными радиусами, реализованной в комплексе MoDyS.

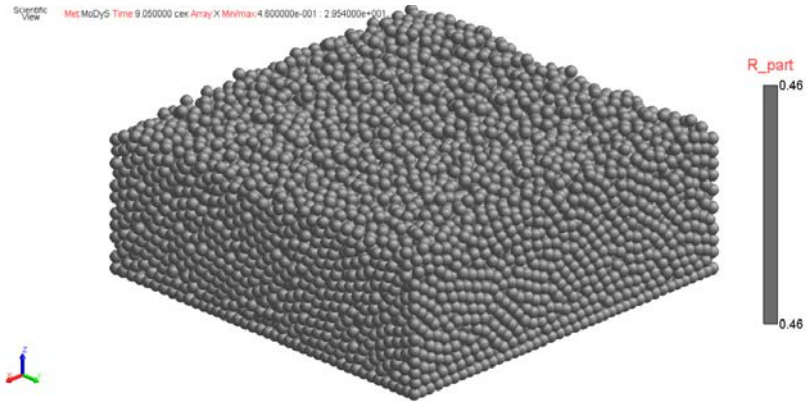


Рис. 4. Результат работы DEM для сфер с одинаковыми радиусами, реализованного в рамках комплекса MoDyS

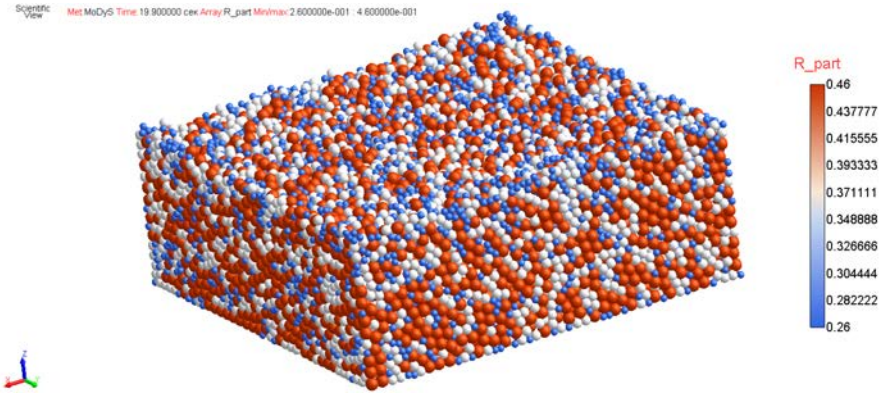


Рис. 5. Результат работы DEM для сфер с 3 различными радиусами, реализованного в рамках комплекса MoDyS

2. Математическая модель

Совокупность частиц и воздуха будем рассматривать, как континуальную двухфазную среду. Нестационарное течение многофазной среды рассматривается как течение несжимаемой вязкой жидкости. В декартовых координатах в трехмерной геометрии уравнения Навье-Стокса [7] описывающие течение несжимаемой жидкости имеют следующий вид:

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0,\\
&\frac{\partial \rho u}{\partial t}+\frac{\partial(\rho u^2+p)}{\partial x}+\frac{\partial \rho uv}{\partial y}+\frac{\partial \rho uw}{\partial z}=\rho g_x+\mu\left(\frac{\partial^2 \mathrm{u}}{\partial x^2}+\frac{\partial^2 \mathrm{u}}{\partial y^2}+\frac{\partial^2 \mathrm{u}}{\partial z^2}\right)+\rho F_x,\\
&\frac{\partial \rho v}{\partial t}+\frac{\partial \rho uv}{\partial x}+\frac{\partial(\rho v^2+p)}{\partial y}+\frac{\partial \rho vw}{\partial z}=\rho g_y+\mu\left(\frac{\partial^2 \mathrm{v}}{\partial x^2}+\frac{\partial^2 \mathrm{v}}{\partial y^2}+\frac{\partial^2 \mathrm{v}}{\partial z^2}\right)+\rho F_y,\\
&\frac{\partial \rho w}{\partial t}+\frac{\partial \rho uw}{\partial x}+\frac{\partial \rho vw}{\partial y}+\frac{\partial(\rho w^2+p)}{\partial z}=\rho g_z+\mu\left(\frac{\partial^2 \mathrm{w}}{\partial x^2}+\frac{\partial^2 \mathrm{w}}{\partial y^2}+\frac{\partial^2 \mathrm{w}}{\partial z^2}\right)+\rho F_z.
\end{aligned}
\qquad (1)
$$

Здесь (u, v, w) – вектор скорость, ρ – плотность, μ – коэффициент динамической вязкости, p – давление, g – вектор ускорения свободного падения и F – массовые силы.

Несжимаемые течения будем рассматривать как слабо – сжимаемые течения с баротропным уравнением состояния, которое обычно берется в форме Тета [8]:

$$p = A\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) + p_0,$$

где $A = \frac{\rho_0 c_0^2}{\gamma}$, ρ_0 – начальная плотность, c_0 – скорость звука, γ – показатель адиабаты, p_0 – давление при комнатной температуре.

С увеличением температуры происходит увеличения размера тела. Это явление характеризуется температурным коэффициентом объемного расширения [9]:

$$V = V_0\left(1 + \beta_T\left(T - T_0\right)\right).$$

Тогда для слабо-сжимаемых течений уравнение состояние может быть записано в виде:

$$p = A\left(\left(\frac{1}{1 + \beta_T\left(T - T_0\right)}\right)^{\gamma} - 1\right) + p_0.$$

Давление в газе определяется уравнением состояния идеального газа, которое представим в виде:

$$p = \left(\gamma - 1\right)\rho C_v\left(T - T^0\right) + p_0.$$

На контактах жидкой фазы с твердой фазой и с газовой средой возникает дополнительное давление, обусловленное поверхностным натяжением и силой Марангони [10]:

$$\Delta p = \sigma_n k\vec{n} + \nabla\sigma_n,$$

где σ_n – коэффициент поверхностного натяжения для жидкой фазы, k – кривизна поверхности жидкой фазы, $\vec{n}$ – вектор внешней нормали к поверхности жидкой фазы, $\nabla\sigma_n$ – тангенциальный поверхностный градиент. Дополнительное давление моделируется массовой силой.

Уравнение сохранения энергии для слабо-сжимаемой среды имеет вид [7]:

$$\frac{\partial\rho\varepsilon}{\partial t} + \frac{\partial\rho u\varepsilon}{\partial x} + \frac{\partial\rho v\varepsilon}{\partial y} + \frac{\partial\rho w\varepsilon}{\partial z} = -\frac{\partial}{\partial x}\left(q_x\right) - \frac{\partial}{\partial y}\left(q_y\right) - \frac{\partial}{\partial z}\left(q_z\right) + \Phi + \rho S, \qquad (2)$$

где ε – внутренняя энергия, q – тепловой поток, Φ – функция описывающая диссипацию энергии за счет работы вязких сил, S – источники тепла.

В уравнении энергии (2) в силу малости скорости движения жидкой фазы и изменения объема пренебрегается изменением кинетической энергии и работой сил давления. Внутренняя энергия и коэффициент теплопроводности для каждого вещества считаются функциями температуры:

$$\varepsilon = \varepsilon(T), \quad \chi = \chi(T).$$

Молекулярный вектор потока тепла выражается по закону Фурье через градиент температуры:

$$q = -\chi\nabla T,$$

где χ – коэффициент теплопроводности.

Диссипация энергии за счет работы вязких сил для несжимаемых течений является положительно определенной функцией:

$$\Phi = \frac{\mu}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)^2.$$

В задачах плавления и затвердевания (задача Стефана) на каждый момент времени определяется криволинейная подвижная поверхность $\left(F(t,r)=0\right)$, на которой происходит фазовый переход. Положение границы фазового перехода определяется из условия:

$$T\left(t,r\right)=T^{*},\ (t,r)\in F,$$

где T^{*} – температура фазового перехода.

Фазовый переход сопровождается выделением либо поглощением определенного количества тепла. На границе фазового перехода разность тепловых потоков определяет количество тепла, затрачиваемое на фазовый переход, поэтому условия сопряжения принимают вид [12-14]:

$$\left[T\right]=0,\ (t,\mathrm{r})\in F, \tag{3}$$

$$\left[\chi\frac{\partial T}{\partial n}\right]=\rho\Delta h V_{n},\ (t,\mathrm{r})\in F, \tag{4}$$

где Δh – удельная энтальпия фазового перехода, а V_n – скорость движения границы фазового перехода по нормали к поверхности. Теплофизические свойства различных агрегатных состояний вещества с разных сторон от фазового фронта в общем случае различны.

Задача Стефана допускает обобщенную формулировку [12–14], при которой условия (3), (4) включается в уравнение теплопроводности:

$$\rho\left(C_P+\Delta h\delta\left(T-T^{*}\right)\right)\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}\left(\chi\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\chi\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(\chi\frac{\partial T}{\partial z}\right)+\rho S, \tag{5}$$

где $\delta\left(T-T^{*}\right)$ – дельта функция Дирака. При этом значения теплофизических свойств вещества в различных агрегатных состояниях определяются с учетом аппроксимации дельта функции ступенчатой функцией, следующим образом [12–14]:

$$C_P=\begin{cases}C_{P1}(T), & \text{если } T<T^{*}-\Delta T,\\ C_{P1}(T^{*})+\left(C_{P2}(T^{*})-C_{P1}(T^{*})+\Delta h\right)\dfrac{T-T^{*}+\Delta T}{2\Delta T}, & \text{если } T^{*}-\Delta T\le T\le T^{*}+\Delta T,\\ C_{P2}(T), & \text{если } T>T^{*}+\Delta T;\end{cases}$$

$$\chi=\begin{cases}\chi_1(T), & \text{если } T<T^{*}-\Delta T,\\ \chi_1(T^{*})+\left(\chi_2(T^{*})-\chi_1(T^{*})\right)\dfrac{T-T^{*}+\Delta T}{2\Delta T}, & \text{если } T^{*}-\Delta T\le T\le T^{*}+\Delta T,\\ \chi_2(T), & \text{если } T>T^{*}+\Delta T.\end{cases}$$

Источники тепла, упитываются как распределенные массовые источники, которые складываются из энерговыделения лазерного луча S_L, нагревающего материал, поверхностной конвекции S_C и радиационного излучения S_R:

$$S=S_L-S_C-S_R.$$

Поглощение веществом лазерного излучения по закону Ламберта описывается следующей зависимостью

$$I\left(z\right)=\left(1-K\right)I_0\left(r\right)\exp\left\{-\int_0^{z}\alpha\left(z\right)dz\right\},$$

где K – коэффициент отражения излучения, $I_0\left(r\right)$ – интенсивность излучения на поверхности тела и $\alpha\left(z\right)$ – коэффициент поглощения. Величина поглощенной энергии излучения объемом dV мате-

риала есть $P = \alpha IdV$. Поэтому величину источника энерговыделения от лазерного излучения, можно выразить так:

$$S_L(z) = (1-K) I_0(r) \alpha \exp\left\{ -\int_0^z \alpha(z)\, dz \right\}.$$

Интенсивность лазерного луча на поверхности описывается Гауссовым профилем:

$$I(r) = \frac{2P_L}{\pi r_L^2} \exp\left\{ -4\left(\frac{r}{r_L} \right)^2 \right\},$$

где P_L – мощность лазерного излучения, а r_L – радиус пятна лазерного излучения.

Конвективный обмен энергией с окружающим веществом

$$S_C = \alpha_C (T_s - T_\infty),$$

где α_C – коэффициент конвективного теплообмена, T_s – температура на поверхности материала, T_∞ – температура окружающей среды (обычно полагаемая 293 K).

Радиационное излучение описывается зависимостью Стефана-Больцманна:

$$S_R = \sigma\varepsilon (T_s - T_\infty)^4,$$

где σ – постоянная Больцмана, ε – степень черноты поверхности.

Деформация твердой фазы в данной работе не рассматривается. Считается, что движение твердой фазы незначительно по сравнению с движение жидкой фазы и твердая фаза никак не влияет на движение жидкой фазы. Определение остаточных тепловых и механических нагрузок в данной работе не рассматривается.

3. SPH формулировка уравнений

Численное решение системы уравнений описывающей СЛС будем находить с использованием метода SPH [15-16]. В этом методе значение расчетных величин и их градиентов вычисляются по формулам:

$$\left\langle f(\vec{r}) \right\rangle = \mathring{a}_j \frac{m_j}{\rho(\vec{r}_j)} f(\vec{r}_j) W(\vec{r} - \vec{r}_j, h),$$

$$\left\langle \nabla f(\vec{r}) \right\rangle = \sum_j \frac{m_j}{\rho(\vec{r}_j)} f(\vec{r}_j) \nabla W(\vec{r} - \vec{r}_j, h),$$

$$\nabla W(\vec{r} - \vec{r}_j, h) = W'(\vec{r} - \vec{r}_j, h) \frac{\vec{r} - \vec{r}_j}{|\vec{r} - \vec{r}_j|},$$

где f – искомые функции, h – длина сглаживания, W – интерполяционное ядро, в качестве которого используется сплайн третьей степени [17]:

$$W(\vec{r} - \vec{r}', h) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}R^2 + \frac{3}{4}R^3, & 0 < R < 1, \\ \frac{1}{4}(2-R)^3, & 1 \le R < 2, \\ 0. & R \ge 2. \end{cases}$$

Приведем SPH аппроксимацию исходной системы уравнений. Уравнение неразрывности запишется следующим образом:

$$\left(\frac{d\rho}{dt}\right)_i = -\left(\rho\nabla\vec{u}\right)_i = -\rho_i\sum_j\frac{m_j}{\rho_j}\left(\left(\vec{u}_i-\vec{u}_j\right)\nabla W_{ij}\right)+\rho_i\sum_j\frac{m_j}{\rho_j}\left(R_{ij}\nabla W_{ij}\right).$$

Существуют различные формы аппроксимации диффузионного члена в SPH [18]. В данной работе используется аппроксимация из [19], которая совместима с изотермически-несжимаемой средой:

$$R_{ij} = -2\delta h\frac{c_0}{\rho_0}\left(\left(\overline{\rho}_i-\overline{\rho}_j\right)\frac{r_{ij}}{\left|r_{ij}\right|}\right),$$

где $\overline{\rho}_i = \rho_i - \rho_0$, а h и δ являются сглаживающими параметрами.

Аппроксимация уравнений сохранения количества движений на частицах следующая:

$$\left(\frac{du}{dt}\right)_i = \left(g_x\right)_i - \sum_j m_j\left(\frac{p_i}{\rho_i^2}+\frac{p_j}{\rho_j^2}\right)\left(\nabla_x W_{ij}\right)+\sum_{j=1}^{N}\frac{m_j\left(\mu_i+\mu_j\right)}{\rho_i\rho_j}\frac{\left(u_i-u_j\right)}{\left|r_i-r_j\right|}\omega_{ij}+\left(F_x\right)_i,$$

$$\left(\frac{dv}{dt}\right)_i = \left(g_y\right)_i - \sum_j m_j\left(\frac{p_i}{\rho_i^2}+\frac{p_j}{\rho_j^2}\right)\left(\nabla_y W_{ij}\right)+\sum_{j=1}^{N}\frac{m_j\left(\mu_i+\mu_j\right)}{\rho_i\rho_j}\frac{\left(v_i-v_j\right)}{\left|r_i-r_j\right|}\omega_{ij}+\left(F_y\right)_i,$$

$$\left(\frac{dw}{dt}\right)_i = \left(g_z\right)_i - \sum_j m_j\left(\frac{p_i}{\rho_i^2}+\frac{p_j}{\rho_j^2}\right)\left(\nabla_z W_{ij}\right)+\sum_{j=1}^{N}\frac{m_j\left(\mu_i+\mu_j\right)}{\rho_i\rho_j}\frac{\left(w_i-w_j\right)}{\left|r_i-r_j\right|}\omega_{ij}+\left(F_z\right)_i.$$

Схема аппроксимации на частицах поверхностной силы состоит из следующих шагов:

1. Определение нормализованного градиента от функции фазы на частицах:

$$\left(\nabla C\right)_i = \sum_j\frac{m_j}{\rho_j}\left(\frac{\rho_i}{\Gamma_j}+\frac{\rho_j}{\Gamma_i}\right)\nabla W_{ij}.$$

2. Определение вектора нормали на интерфейсной поверхности с ограничением, чтобы избежать ошибок в определении искривления поверхности контакта:

$$\hat{n}_i = \begin{cases}\dfrac{\left(\nabla C\right)_i}{\left|\left(\nabla C_f\right)_i\right|}, & \text{если } \dfrac{\left(\nabla C\right)_i}{h_i} > 0.1\\ 0, & \text{в противном случае.}\end{cases}$$

3. Вычисление нового значения коэффициента нормализации Γ_i для частиц с отличным от нуля значением $\hat{n}_i$:

$$\Gamma_i = \sum_j\frac{m_j}{\rho_j}\nabla W_{ij}.$$

4. Вычисление кривизны поверхности для частиц с отличным от нуля значением $\hat{n}_i$, используя повторно нормализованную формулировку:

$$\left(k\right)_\alpha = \frac{1}{\Gamma_i}\sum_j\frac{m_j}{\rho_j}\left(\hat{n}_j-\hat{n}_i\right)\nabla W_{ij}.$$

5. Ограничение кривизны поверхности:

$$k_i = \max\left\{k_{\max}, k_i\right\},\quad k_{\max} = \frac{1}{2h_{ij}}.$$

6. Вычисление нормальной силы поверхностного натяжения:

$$\left(F_n\right)_i = -\sigma_i k_i\hat{n}_i.$$

7. Вычисление теплового градиента:

$$\left(\nabla T\right)_i = \frac{1}{\Gamma_i}\sum_j \frac{m_j}{\rho_j}\left(T_i - T_j\right)\nabla W_{ij}.$$

8. Вычисление касательной силы поверхностного натяжения (силы Марангони):

$$\left(F_\tau\right)_i = \left(\frac{d\sigma}{dT}\right)_i \left(\left(\nabla T\right)_i - \left(\left(\nabla T\right)_i \vec{n}_i\right)\vec{n}_i\right)\left|\left(\nabla C\right)\right|_i.$$

Рассмотрим теперь аппроксимацию уравнения сохранения энергии на частицах. Изменение энергии описывается пятью процессами: работа вязких сил, конвективный теплообмен с внешней средой, радиационное излучение, поглощение энергии лазерного излучения и теплообменом. SPH аппроксимация работы вязких сил следующая:

$$\left(\frac{1}{\rho}\Phi\right)_i = \sum_{j=1}^{N} \frac{m_j}{\rho_i\rho_j}\frac{4\mu_i\mu_4}{\mu_i+\mu_j}\left(\vec{u}_i - \vec{u}_j\right)\pi_{ij}\left(\nabla W_{ij}\right)_i,$$

где

$$\pi_{ij} = \frac{\left(\vec{u}_i - \vec{u}_j\right)\left(r_i - r_j\right)}{\left|r_i - r_j\right|^2 + \eta}.$$

Конвективный теплообмен происходит только в области контакта металла и окружающей среды, поэтому для его определения вводится функция G, описывающая материал:

$$G_i = \begin{cases} 1, & \text{если } r_i \in \text{материал}, \\ 0, & \text{если } r_i = \text{газ}. \end{cases}$$

Определяется градиент этой функции на частицах:

$$\left(\nabla G\right)_i = \sum_j \frac{m_j}{\rho_j}\left(G_i - G_j\right)\nabla W_{ij}.$$

Определяется максимальное значение по всем частицам:

$$D = \max_i\left\{\left|\nabla G\right|_i\right\}.$$

Величина энергии, уходящая из системы в результате конвективного теплообмена, определяется на частицах по формуле:

$$\left(S_C\right)_i = \alpha_C\left(T_i - T^0_i\right)\frac{\left|\nabla G\right|_i}{D+\eta}.$$

Аналогично определяется величина энергии, уходящая из системы в результате радиационного излучения:

$$\left(S_R\right)_i = \sigma_B\varepsilon\left(T_i - T_\infty\right)^4\frac{\left|\nabla G\right|_i}{D+\eta}.$$

Учет поглощения лазерной энергии производится только для частиц, координаты которых попадают в текущий момент времени в область поглощения лазерной энергии.

Наконец, теплообмен на частицах аппроксимируется по следующей формуле

$$\left(\frac{1}{\rho}\nabla\left(\chi\nabla T\right)\right)_i = \sum_{j=1}^{N}\frac{m_j}{\rho_i\rho_j}\frac{4\chi_i\chi_j}{\chi_i+\chi_j}\left(\left(\nabla T\right)_{ij}\left(\nabla W_{ij}\right)_i\right).$$

Термодинамические свойства порошкового материала различны в разных фазах. Это может быть интерпретировано как преодоление барьера материалом, для того чтобы перейти из одной фа-

зы в другую фазу. В формулировке SPH процесс перехода частиц из одной фазы в другую фазу учитывается с помощью функции C:

$$C=\begin{cases}0, & T\le T^*-\dfrac{\Delta T}{2},\\ \dfrac{T-\left(T^*-\Delta T/2\right)}{\Delta T}, & T^*-\dfrac{\Delta T}{2}<T<T^*+\dfrac{\Delta T}{2},\\ 1, & T\ge T^*+\dfrac{\Delta T}{2}.\end{cases}$$

где $C=1$ отвечает жидкому состоянию, $C=0$ твердому состоянию, а $0<C<1$ части перехода между твердой и жидкой фазой.

4. Численный метод

Численно система уравнений в SPH формулировке решается методом расщепления по физическим процессам. Расщепление производится на решение уравнений газовой динамики и на решение уравнения теплопереноса. Метод интегрирования уравнений газовой динамики основан на методе предиктор-корректор. На этапе предиктор в этом методе, определяются предварительные значения плотности и координат частиц:

$$\tilde{\rho}_i=\rho_i^n+\frac{\tau}{2}\left(\frac{d\rho}{dt}\right)_i^n;$$

$$\tilde{\vec{r}}_i=\vec{r}_i^{\,n}+\frac{\tau}{2}\vec{u}_i^{\,n}.$$

После этого вычисляется давление по уравнению состояния:

$$\tilde{p}_i=P\left(\tilde{\rho}_i,\mathrm{T}_i\right).$$

Этап корректор (второй этап) в рассматриваемом методе, осуществляется в несколько проходов. На первом проходе определяются:

нормализованный градиент от функции фазы

$$\left(\nabla C\right)_i=\sum_j\frac{m_j}{\rho_j}\left(\frac{\rho_i}{\Gamma_j}+\frac{\rho_j}{\Gamma_i}\right)\nabla W_{ij}$$

и вектор нормали на поверхности раздела фаз

$$\hat{n}_i=\begin{cases}\dfrac{\left(\nabla C\right)_i}{\left|\left(\nabla C_f\right)_i\right|}, & \text{если } \dfrac{\left(\nabla C\right)_i}{h_i}>0.1\\ 0, & \text{в противном случае.}\end{cases}$$

На втором проходе определяются коэффициент нормализации

$$\Gamma_i=\sum_j\frac{m_j}{\rho_j}\nabla W_{ij}$$

и кривизна поверхности

$$\left(k\right)_\alpha=\frac{1}{\Gamma_i}\sum_j\frac{m_j}{\rho_j}\left(\hat{n}_j-\hat{n}_i\right)\nabla W_{ij}.$$

На третьем проходе определяется ограничение кривизны поверхности

$$k_i = \max\left\{k_{\max}, k_i\right\}, \quad k_{\max} = \frac{1}{2h_{ij}}.$$

На четвертом проходе определяется тепловой градиент

$$\left(\nabla T\right)_i = \frac{1}{\Gamma_i}\sum_j \frac{m_j}{\rho_j}\left(T_i - T_j\right)\nabla W_{ij},$$

нормальная

$$\left(F_n\right)_i = -\sigma_i k_i \hat{n}_i$$

и касательная

$$\left(F_\tau\right)_i = \left(\frac{d\sigma}{dT}\right)_i \left(\left(\nabla T\right)_i - \left(\left(\nabla T\right)_i \cdot \hat{n}_i\right)\hat{n}_i\right)\left|\left(\nabla C\right)\right|_i$$

составляющие силы поверхностного натяжения,

На пятом проходе определяются компоненты вектора скорости:

$$\left(u_i\right)^{n+1} = \left(u_i\right)^n + \tau\left(\frac{du}{dt}\right)_i,$$

$$\left(v_i\right)^{n+1} = \left(v_i\right)^n + \tau\left(\frac{dv}{dt}\right)_i,$$

$$\left(w_i\right)^{n+1} = \left(w_i\right)^n + \tau\left(\frac{dw}{dt}\right)_i,$$

где плотность, давление берутся с промежуточного слоя, а компоненты скорости при расчете вязких напряжений в уравнениях движения с n–ого слоя.

Затем определяется средняя скорость частиц:

$$\tilde{u}_i^\alpha = \frac{\left(u_i^\alpha\right)^{n+1} + \left(u_i^\alpha\right)^n}{2}$$

и определяются плотность, внутренняя энергии и координаты частицы на новом временном слое:

$$\rho_i^{n+1} = \rho_i^n + \tau\left(\frac{d\rho\left(\tilde{u}^\alpha\right)}{dt}\right)_i,$$

$$T_i^{n+1} = T_i^n + \frac{\tau}{\rho C_V}\left(\frac{de\left(\tilde{u}^\alpha, \tilde{\rho}, \tilde{s}^{\alpha\beta}\right)}{dt}\right)_i,$$

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,n} + \tau\tilde{\vec{u}}_i.$$

Отметим, что производные от компонент вектора скорости, входящие в определение компонент диссипативной функции в уравнение энергии, вычисляются по средним компонентам вектора скорости.

После решения уравнений газовой динамики решается уравнение теплопереноса, которое аппроксимируется неявной разностной схемой на частицах:

$$\left(C_P\right)_i \frac{T_i^{n+1} - \overline{T}_i}{\tau} = \sum_{j=1}^{N} \frac{m_j}{\rho_i \rho_j} \frac{4\chi_i\chi_i}{\chi_i + \chi_j}\left(\frac{T_j^{n+1} - T_i^{n+1}}{\left|r_j - r_i\right|}\right)\omega_{ij} -$$

$$-\alpha_C\left(T_i^{n+1} - T^0\right)\frac{\left|\nabla G\right|_i}{D+\eta} - \sigma_B\varepsilon\left(T_i^{n+1} - T_\infty\right)^4\frac{\left|\nabla G\right|_i}{D+\eta} + S_L.$$

В системе уравнений входит неизвестная температура в частицах на шаге по времени с номером n+1. Используя процедуру линеаризации уравнения состояния по Ньютону и радиационного теплообмена, систему разностных уравнений относительно температуры $T^{\gamma+1}$ (здесь γ номер итерации) приобретает вид:

$$\left(C_P\right)_i^{\gamma}\left(T_i^{\gamma+1}-\overline{T}_i\right)+2\tau\sum_{j=1}^{N}\frac{m_j}{\rho_i\rho_j}\chi_{ij}\left(\frac{T_i^{\gamma+1}-T_j^{\gamma+1}}{\left|r_j-r_i\right|}\right)\omega_{ij}+\tau\alpha_C T_i^{\gamma+1}\frac{\left|\nabla G\right|_i}{D+\eta}+$$

$$+\tau\sigma_B\varepsilon\left(T_i^{\gamma}-T_\infty\right)^3 T_i^{\gamma+1}\frac{\left|\nabla G\right|_i}{D+\eta}=\tau\alpha_C\overline{T}_i\frac{\left|\nabla G\right|_i}{D+\eta}+\tau\sigma_B\varepsilon\left(T_i^{\gamma}-T_\infty\right)^3 T_\infty\frac{\left|\nabla G\right|_i}{D+\eta}+\tau S_L.$$

На первой итерации полагается

$$e_i^{\gamma}\Big|_{\gamma=0}=\overline{e}_i,\quad T_i^{\gamma}\Big|_{\gamma=0}=\overline{T}_i,$$

Итерационный процесс завершается при выполнении по неравенства:

$$\left|T_i^{\gamma+1}-T_i^{\gamma}\right|<\varepsilon_{\text{отн}}\left|T_i^{\gamma}\right|+\varepsilon_{\text{абс}},$$

где $\varepsilon_{\text{отн}}=10^{-4}$ и $\varepsilon_{\text{абс}}=10^{-8}$ – значения, соответственно, относительной и абсолютной погрешностей.

Систему уравнений относительно нового значения температуры представляется в виде:

$$AT^{\gamma+1}=B.$$

Для решения этой системы уравнений используются параллельные решатели из библиотеки LParSol [20].

Заключение

В работе сформулирована математическая модель для описания процессов, протекающих в аддитивном производстве. Математическая модель основана на континуальном описании плавления порошкового материала под действием лазерного излучения. Для описания слабо сжимаемых течений жидкой фазы, образующейся в результате плавления, используются уравнения Навье-Стокса, в которых учитываются силы давления, вязкие силы и силы поверхностного натяжения на контактных поверхностях жидкой и твердой фаз, жидкой фазы и внешней среды. Тепловое состояние определяется из уравнения сохранения энергии, в котором учитываются тепловые процессы, объемное поглощение энергии лазерного излучения, конвективный теплообмен с внешней средой, радиационное излучение и работа вязких сил. Фазовые переходы между твердой и жидкой фазами описываются в рамках обобщенной формулировки задачи Стефана. Численное решение находится с помощью метода SPH.

Рассмотрены два метода формирования рабочего слоя порошка: простой геометрический метод (модификация так называемого «метода дождя») и метод дискретных элементов (Discrete Element Method – DEM), которые были реализованы в рамках проекта по созданию программного комплекса «Виртуальный 3D принтер». Важнейшим фактором, определяющий какой из методов использовать, является необходимость учета сил взаимодействия между гранулами порошка. Если данными силами можно пренебречь, то следует использовать более быстрый геометрический подход. Для получения более реальной картины рабочего слоя порошка необходимо использовать DEM.

Литература

1. Zohdi T. I. Additive particle deposition and selective laser processing-a computational manufacturing framework. Comput. Mech., 2014, 54, pp. 171-191.

2. Meakin P., Jullien R.. Restructuring e_ects in the rain model for random deposition. Journal de Physique, 1987, 48 (10), pp.1651-1662.

3. Saad A. Khairallah, Andy Anderson, 2014. Mesoscopic simulation model of selective laser melting of stainless steel powder . J. Mat. Proc. Techn. 214(2014), pp. 2627–2636.

4. Cundall P. A., Strack O. D. L. “A discrete numerical model for granular assemblies”, Geotechnique, vol. 29, n 1, p. 47-65, 1979.

5. Zhaowei Xiang, Ming Yin, Zhenbo Deng, Xiaoqin Mei, Guofu Yin “Simulation of Forming Process of Powder Bed for Additive Manufacturing” / Journal of Manufacturing Science and Engineering, Vol. 138, March 2016.

6. Анисимов А. Н., Грушин С. А., Воронин Б. Л., Копкин С. В., Ерофеев А. М., Демин Д. А., Демина М. А., Здорова М. В., Ветчинников М. В., Еричева Н. С., Коваленко Н. О., Крючков И. А., Кечин А. Г., Дегтярев В. А., Урм В. Я. Комплекс программ молекулярно-динамического моделирования (MoDyS). Свидетельство о государственной регистрации программы для ЭВМ № 2010614974. 2010.

7. Ландау Л. Д., Лифшиц Е. М. Гидродинамика. М.: Наука, 1986.

8. Физика взрыва. Том 1 / Под ред. Л. П. Орленко. М.: Физматлит, 2002.

9. Физические величины: Справочник / Под ред. И. С. Григорьева, А. М. Братковского. М.: Энергоатомиздат,1991.

10. Савельев И. В. Курс общей физики. Том I. М.: Наука, 1970.

11. Карслоу У., Егер Д. Теплопроводность твердых тел. М.: Наука, 1964.

12. Самарский А. А., Вабищевич П. Н. Вычислительная теплопередача. М.: Книжный дом “ЛИБРОКОМ”, 2009. 784с.

13. Самарский А. А., Моисеенко Б. Д. Экономичная схема сквозного счета для многомерной задачи Стефана. ЖВММФ, 1965. Т. 5, № 5. С. 816–827.

14. Будак Б. М., Соловьев Е. Н., Успенский А. Б. Разностный метод со сглаживанием коэффициентов для решения задачи Стефана. ЖВММФ, 1965. Т. 5, № 5. С. 828–840.

15. Gingold R. A., Monaghan J. J. Smoothed Particle Hydrodynamics: theory and application to non-spherical stars // Monthly Notices of the Royal Astronomical Society, 1977. Vol. 181. P. 375–389.

16. Lucy L. A numerical approach to the testing of the fission hypothesis // Astronom. J. 1977. Vol. 82. P. 1013.

17. Monaghan J.J. Smoothed particle hydrodynamics // Annual Review of Astronomy and Astrophysics, Clayton, 1992. P. 543–574.

18. Cercos-Pita J., Dalrymple R., Herault A. Diffusive terms for the conservation of mass equation in SPH, Appl. Math. Model. 40 (19) (2016) 8722–8736. http://dx.doi.org/10.1016/j.apm.2016.05.016

19. Antuono M., Colagrossi A., Marrone S., Molteni D. Free-surface flows solved by means of SPH schemes with numerical diffusive terms, Comput. Phys. Comm. 181 (3) (2010) 532–549. http://dx.doi.org/10.1016/j.cpc.2009.11.002.

20. Бартенев Ю. Г., Ерзунов В. А., Карпов А. П., Петров Д. А., Пищулин И. А., Стаканов А. Н., Щаникова Е. Б., Капорин И. Е., Милюкова О. Ю., Харченко С. А., Коньшин И. Н., Сысоев А. В., Мееров И. Б. Комплекс библиотек параллельных решателей СЛАУ LParSol версии 3.: в сб. докладов XIV международной конференции “Высокопроизводительные параллельные вычисления на кластерных системах”. ПНИПУ, Пермь, 2014. С. 49–53.

NUMERICAL SIMULATION OF SELECTIVE LASER SINTERING OF A POWDER MIXTURE

A. N. Bykov, M. V. Vetchinnikov, I. S. Sapronov, Yu. N. Deryugin, A. A. Lazarev

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The work formulates a mathematical model to describe the processes in additive manufacturing. The model is based on continual description of powder material melting with a laser beam. The numerical solution is found using method of smoothed particles. Operating layer of the powder is formed using either of two methods: a simple geometrical method and a method of discrete elements.

Key words: selective laser sintering of the powder mixture, powder layer, geometric method, method of discrete elements, mathematical model, Navier-Stocks equations, method of smoothed particles.

УДК 004.4
DOI 10.53403/9785951505071_2022_90

ПРОЕКТ «ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР»

А. Н. Быков, В. В. Попов, Ю. Н. Дерюгин, М. В. Ветчинников, Д. Ю. Дьянов, С. Н. Полищук, В. Г. Куделькин, Е. А. Синицин, А. Н. Бахаев, М. В. Журавлева, О. И. Бритова, М. А. Титов, А. В. Баранов, Л. В. Гамов, И. Г. Милешин, В. Ю. Кузнецов, П. А. Машенькин, Е. Ф. Голубкова, А. С. Данилов, М. В. Медведкина, А. А. Каякин, А. И. Ломайкин, О. В. Сапронова, Е. А. Веселова, А. Б. Емельянов, А. А. Лазарев, И. С. Сапронов, Н. Н. Балуева, О. С. Ларькина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе приведено описание проекта «Виртуальный 3D-принтер». Цель проекта – создание комплекса программного обеспечения (ПО), обеспечивающего функциональность и широкие возможности для оптимизации формы деталей на основе методов топологической оптимизации (бионический дизайн), в том числе и с использованием решетчатых структур, управления оборудованием для лазерного аддитивного производства, функциональность для определения оптимальных параметров работы оборудования на основе математического моделирования.

Ключевые слова: аддитивные технологии, селективное лазерное плавление, многоуровневое математическое моделирование, топологическая оптимизация, решетчатые структуры, генеративный дизайн, неявное представление геометрии, управление технологическим оборудованием.

Введение

С 90-х годов 20-го века в мире активно развиваются аддитивные технологии производства, которые обладают рядом преимуществ (изготовление точных прототипов и готовых деталей практически любой степени сложности внутренней и внешней структуры поверхности, возможность формировать элементы деталей размерами десятые доли миллиметра, возможность получать градиентные свойства материалов) по сравнению с традиционными технологиям производства в ряде областей применения (аэрокосмическая и военная техника; тяжелое и легкое машиностроение; ядерная, нефтяная и газовая энергетика; медицина и фармацевтика; строительство и архитектура).

Одной из перспективных и востребованных промышленностью технологий является технология селективного лазерного сплавления (СЛС), позволяющая изготавливать детали из металлов и сплавов. Особенностями лазерных аддитивных технологий, в частности, СЛС, являются наличие множества происходящих физических процессов (около двух десятков) и большое количество технологических параметров (более ста), влияющих на протекание этих физических процессов. Полный экспериментальный перебор всех сочетаний параметров с целью получения наилучшего результата изготовления деталей является длительным, трудоемким и привязанным к конкретному материалу процессом. Распространенными недостатками готовых деталей являются анизотропия свойств, пористость и высокие остаточные напряжения, которые могут приводить к деформации формы деталей и к снижению механических свойств. Описанные практические трудности существенно сдерживают развитие и широкое внедрение данных технологий в мире, в том числе затрудняют расширение номенклатуры используемых материалов. Понимание взаимосвязей и закономер-

ностей протекания физических процессов с помощью математического моделирования позволит продвинуться в решении технологических проблем лазерных аддитивных технологий.

Для реализации этого в 2019 году в Госкорпорации “Росатом” в рамках Единого отраслевого тематического плана НИОКР начат проект “Виртуальный 3D-принтер”. Задачи проекта заключаются в разработке специализированного программного обеспечения для:

– получения возможности проектировать новые изделия с улучшенными массовыми характеристиками при сохранении эксплуатационных свойств посредством топологической оптимизации (бионический дизайн);

– выбора и оптимизации технологических режимов работы лазерных 3D-принтеров на основе многоуровневого математического моделирования;

– получения возможности проведения подготовки моделей деталей для изготовления с помощью лазерных аддитивных технологий;

– предсказания физико-механических свойств синтезируемых изделий.

Головным исполнителем проекта является ФГУП «РФЯЦ-ВНИИЭФ», соисполнители – ФГБОУ ВО «Удмуртский государственный университет», ФГБУ НИЦ «Курчатовский институт», ООО «Кинтех Лаб», АНОО ВО «Сколтех», ФГБОУ ВО "Казанский национальный исследовательский технический университет им. АН. Туполева-КАИ".

В докладе освещены различные вопросы создания такого программного обеспечения и отражены результаты выполнения первого этапа работ 2019 года.

Составные части проекта

Задачи проекта реализуются посредством разработки, верификации и интеграции двадцати расчетных модулей. Модули сгруппированы по трем блокам – «Проектирование», «Моделирование», «Производство».

Блок «Проектирование»

В модулях блока «Проектирование» реализована возможность:

– импорта детали в фасеточном виде или построение детали на основе геометрических примитивов и булевых операций с ними;

– генерации решетчатых структур;

– выполнения топологической оптимизации (ТО);

– заполнения детали решетчатыми структурами;

– проведение прочностных расчетов (виртуальные испытания).

На рис. 1 приведены примеры четырех деталей, построенных в программе.

Одним из преимуществ аддитивных технологий является свобода конструктора при проектировании деталей от технологических ограничений традиционных способов производства деталей. Перспективный подход при проектировании деталей для аддитивного производства (в литературе часто используется термин «генеративный дизайн») заключается в том, что конструктор задает условия работы детали (нагрузки, закрепления), свойства материала, области геометрии, которые нельзя изменять, выбирает алгоритм топологической оптимизации и параметры целевой функции (например, процент уменьшения массы). Далее работает программа топологической оптимизации, которая рассчитывает геометрию детали, удовлетворяющей целевой функции. На рис. 2 приведены примеры геометрий деталей, полученных в результате топологической оптимизации.

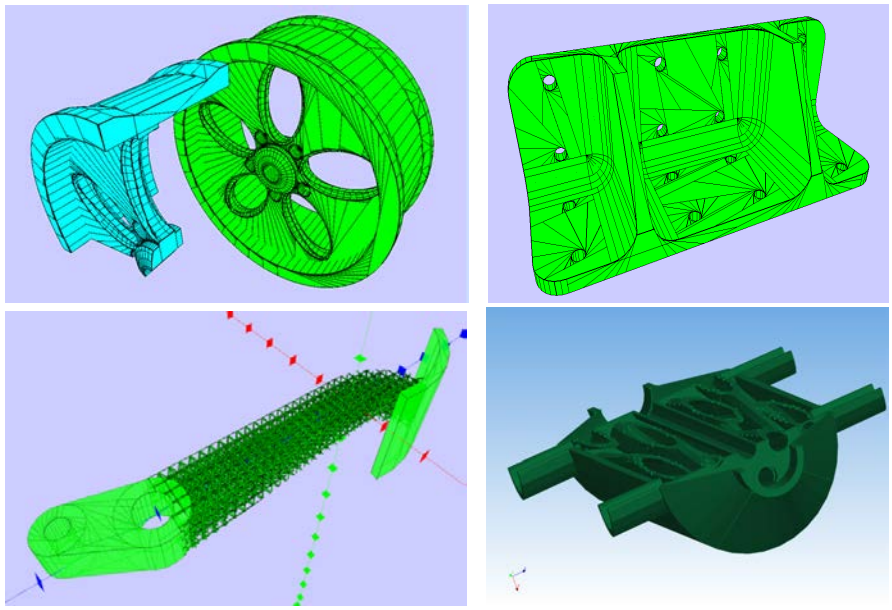


Рис. 1. Примеры деталей, построенных в программе

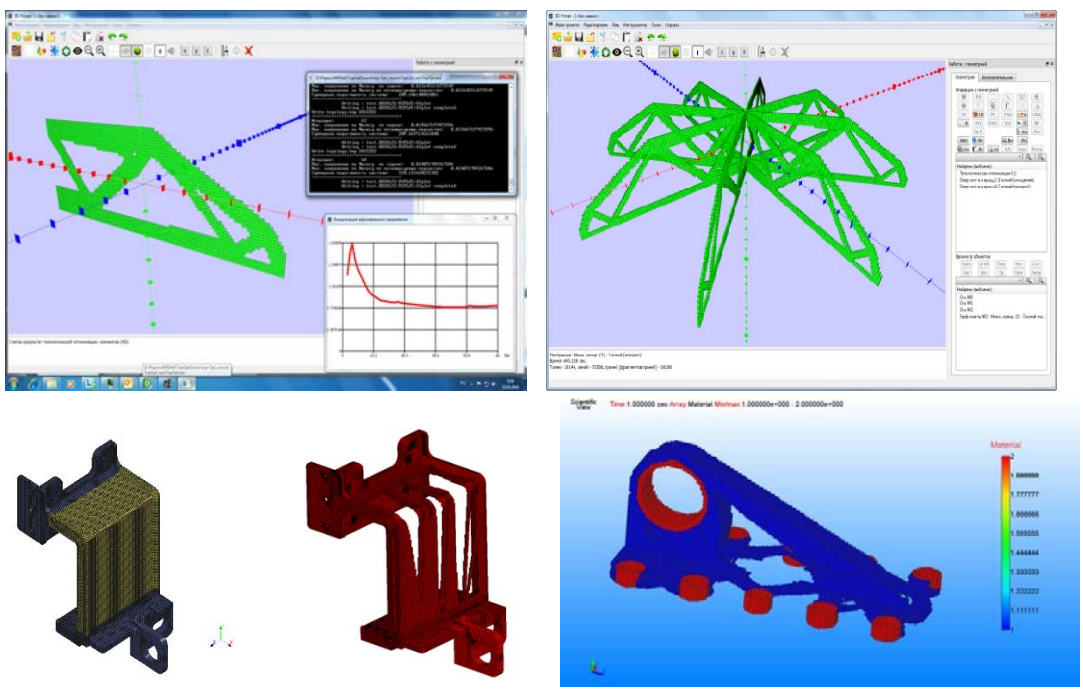


Рис. 2. Примеры геометрий деталей, полученных в результате топологической оптимизации

Другим важным преимуществом аддитивных технологий является возможность замены сплошного материала на решетчатые структуры с сохранением прочностных свойств. Это востребовано в авиакосмической, автомобильной отраслях промышленности для создания легковесных деталей, теплообменников с высокими удельными характеристиками, в медицине для повышения характеристик имплантируемых протезов.

Привлекательным для практического использования является объединение этих двух подходов при проектировании деталей для аддитивного производства, то есть сначала получается геометрия детали посредством топологической оптимизации, а затем, на основе анализа полученных полей различных величин (например, напряжений) производится замена областей детали (например, с наименьшими напряжениями) на решетчатые структуры.

Для этого в модуле «Генерация решеток» реализована возможность генерации следующих типов решеток:

1) Трехмерные решетки, вписанные в объемные ячейки разной формы:

32 типа решеток, вписанных в параллелепипед;

– 4 типа решеток, вписанных в 6 угольную призму «сота»;

– 4 типа решеток, вписанных в тетраэдр;

– 4 типа решеток, вписанных в октаэдр.

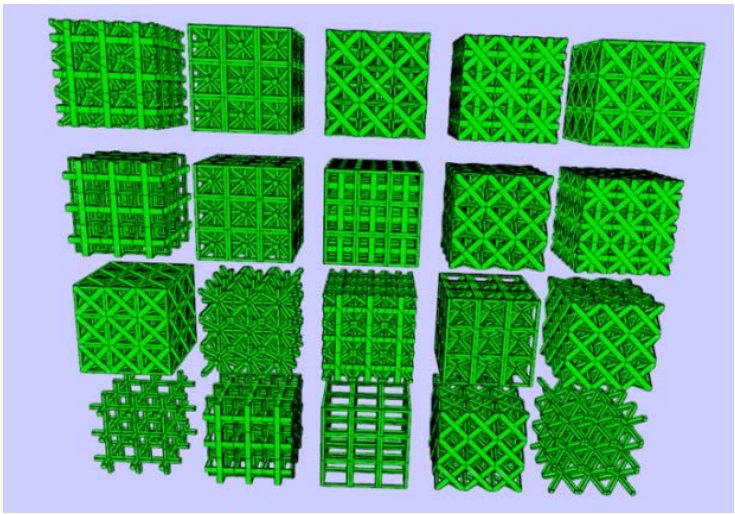


Рис. 3. Примеры решеток, построенных на основе объемных ячеек

2) Двумерные решетки, вписанные в плоские ячейки разной формы и вытянутые по третьему направлению:

– 3 типа решеток, вписанных в шестиугольник;

– 3 типа решеток, вписанных в прямоугольник;

– 3 типа решеток, вписанных в треугольник;

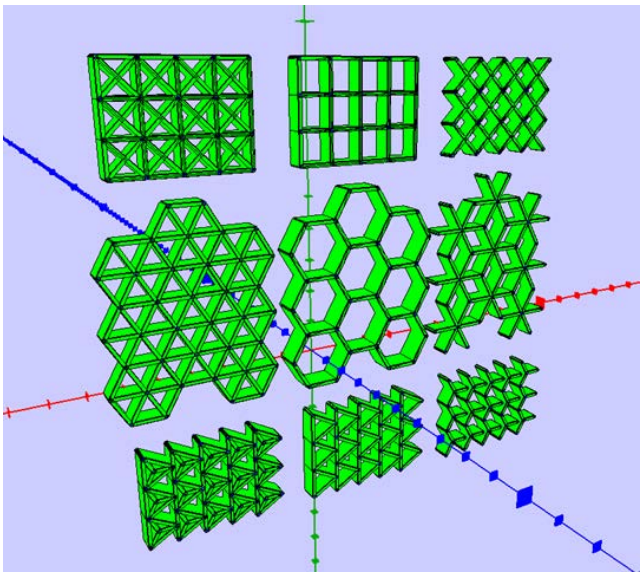


Рис. 4. Примеры решеток, построенных на основе плоских ячеек

3) Решетки на основе поверхности суммарной нулевой кривизны (TPMS):

– 3 типа основных решеток;

– 3 типа производных решеток от TPMS решеток.

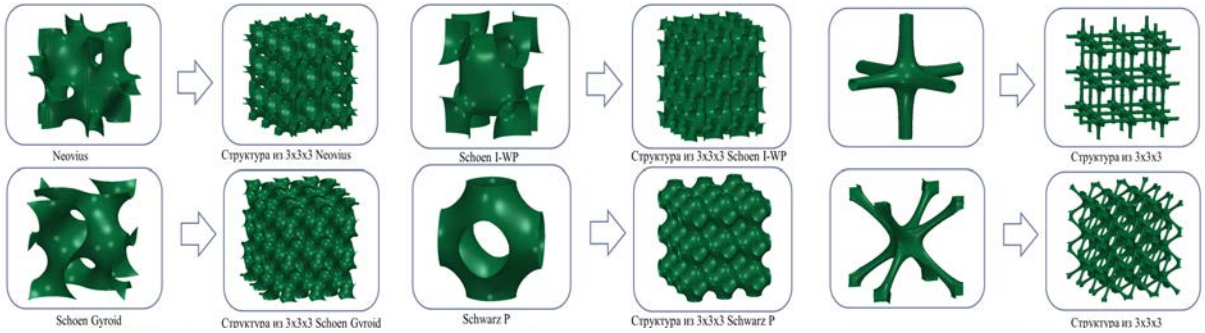


Рис. 5. Примеры TPMS решеток

4) Решетка, полученная генерацией призм со случайными параметрами (расположение, длина, диаметр, количество ребер).

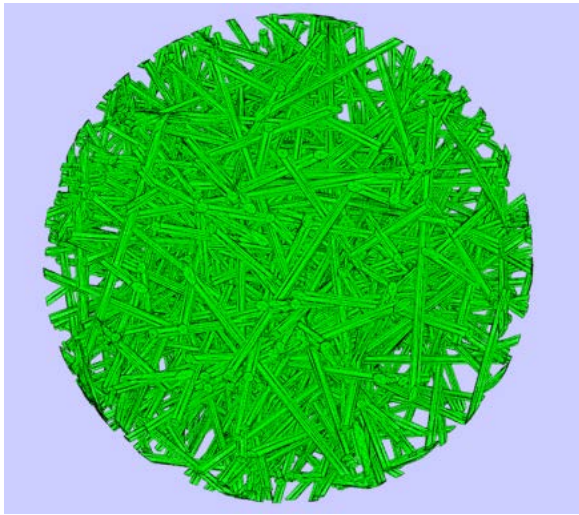


Рис. 6. Примеры решетки со случайными параметрами

5) Решетка, созданная с учетом градиента величин (плотности).

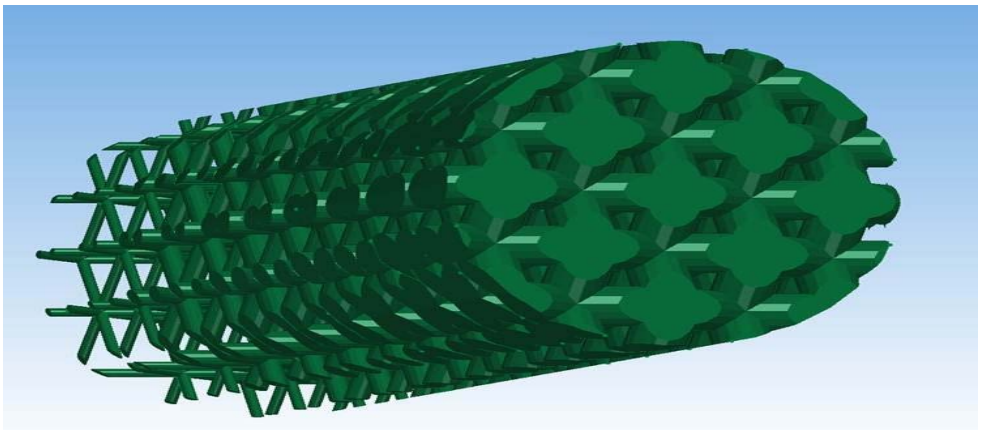


Рис. 7. Пример градиентной решетки

Для конструктора важна возможность не только построить различные типы решеток, но и заполнить ими часть геометрии. Для этого реализован модуль “Заполнение геометрии решетками”. Результаты использования этого модуля показаны на рис. 8 и 9.

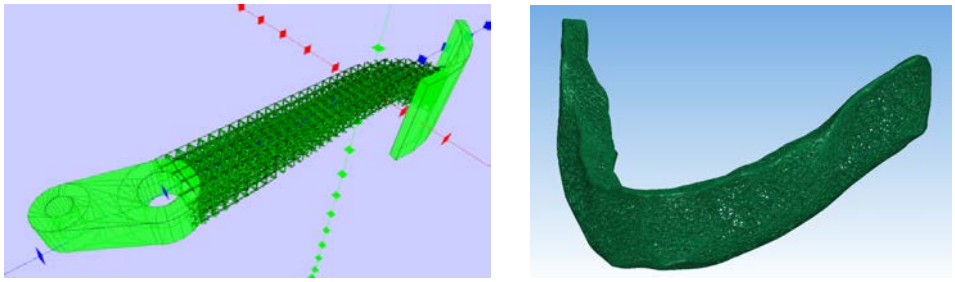


Рис. 8. Пример заполнения решетками деталей «педаль» и «имплант»

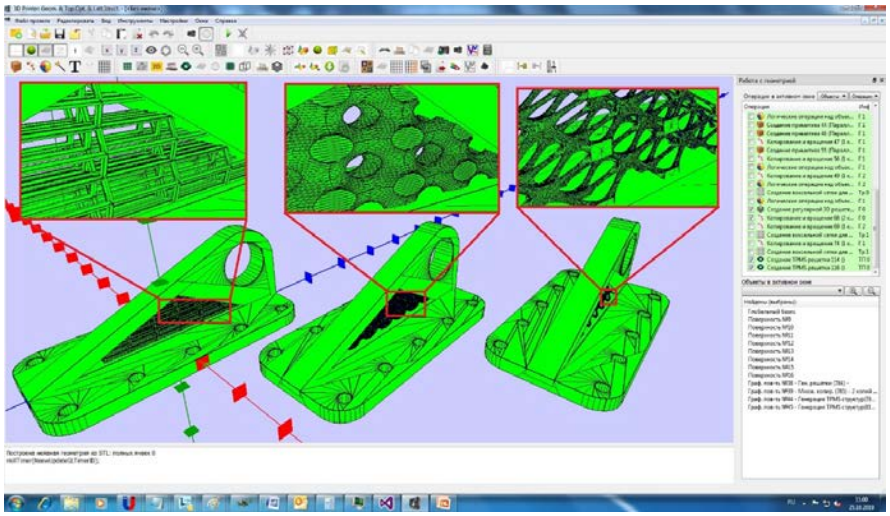


Рис. 9. Пример заполнения различными типами решеток детали «кронштейн»

Другой важная возможность для конструктора – это заполнение детали градиентной решеткой, например, в тех областях детали, где высоки напряжения будут более толстые ребра, а там, где низкие напряжения – более тонкие ребра, тем самым, меняя прочностные свойства детали. Пример заполнения детали «кронштейн» решеткой градиентного типа показан на рис. 10.

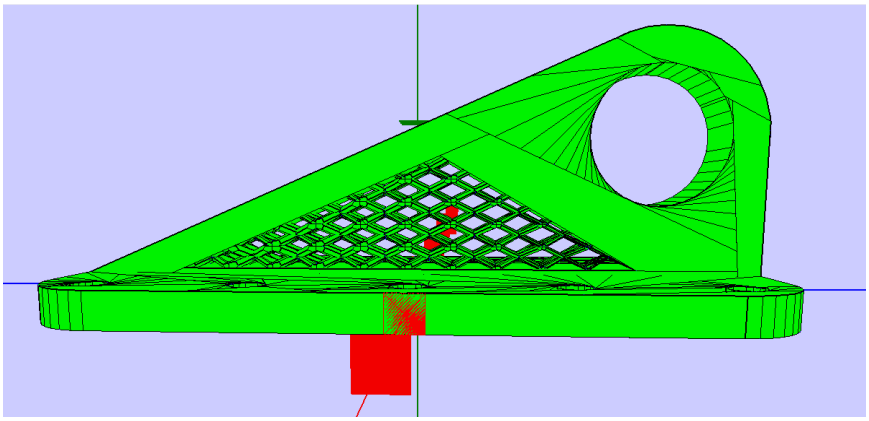


Рис. 10. Пример заполнения детали «кронштейн» решеткой градиентного типа

Блок «Моделирование»

Множество происходящих физических процессов (около двух десятков) и большое количество технологических параметров (более ста), влияющих на протекание этих физических процессов в лазерных аддитивных технологиях приводят к невозможности полного экспериментальный перебор

всех сочетаний параметров с целью получения наилучшего результата изготовления деталей, поэтому очевидна необходимость использования математического моделирования для сокращения времени и стоимости экспериментальной отработки. Причем это моделирование должно быть на нескольких масштабных уровнях. С одной стороны, конструктора интересует успешно или нет произойдет выпекание всей детали, а это десятки сантиметров (макроуровень). С другой стороны, свойства материала зависят от параметров технологического процесса и определяются на уровне зерен (мезоуровень). Поэтому в проекте реализуются модули для моделирования физических процессов на макро- и мезоуровнях. На макроуровне определяющими процессами являются тепловые и прочностные процессы. На мезоуровне тепломассоперенос в ванне расплава, образование микроструктуры.

В модуле «Тепло макроуровень» реализована возможность расчета процесса теплопроводности с учетом энерговклада от лазерного источника и роста детали в процессе выпекания. На рис. 11 приведены распределение температуры на один из моментов времени, на рис. 12 Распределение фаз внутри спеченной металлической конструкции.

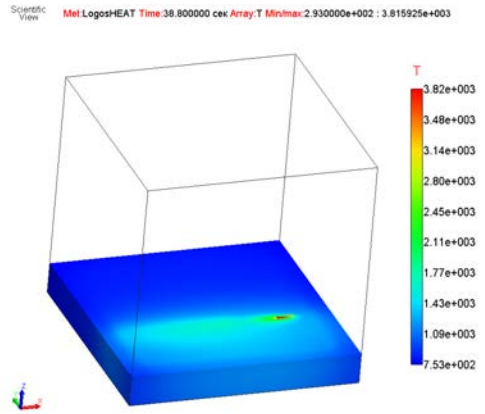


Рис. 11. Распределение температуры при t = 38.8 с (сечение)

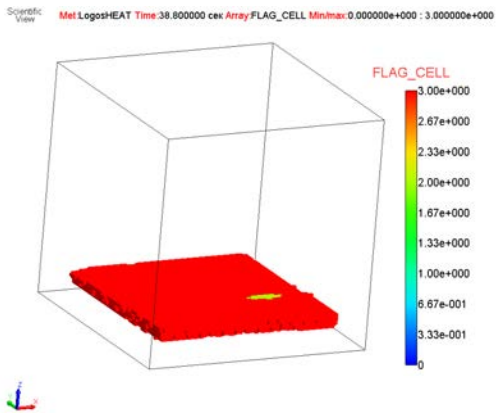


Рис. 12. Распределение фаз внутри спеченной металлической конструкции: твердый металл (красный), расплавленный металл (желтый)

В модуле «Прочность макроуровень» реализована возможность анализа напряженно-деформируемого состояния детали с помощью статического анализа при квазистатических термосиловых воздействиях. Поле температуры является результатом решения «тепловой» задачи. На рис. 13 приведены поля напряжений и деформаций для решетчатой конструкции, на рис. 14 аналогичные поля для детали “спинной кейдж”.

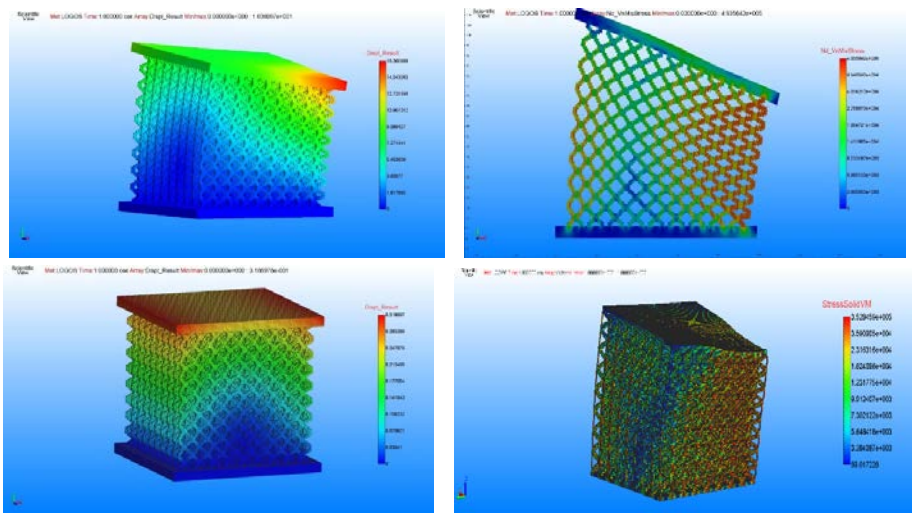


Рис. 13. Поля напряжений и деформаций для решетчатой конструкции

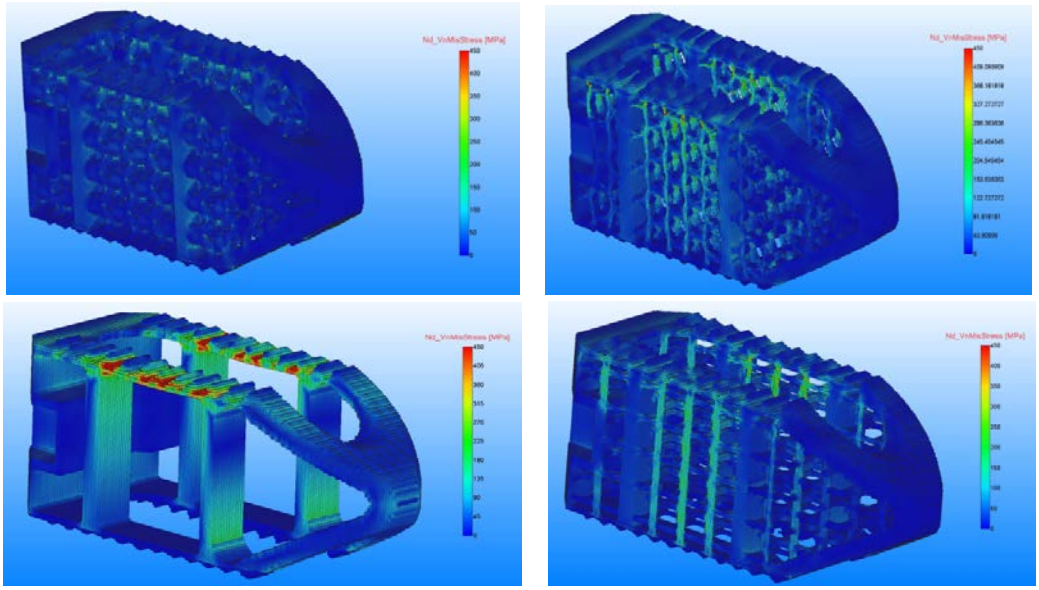


Рис. 14. Поля напряжений и деформаций для детали «спинной кейдж»

Для понимания зависимостей свойств материала от технологических параметров лазерных аддитивных технологий необходимо моделирование тепломассопереноса с учетом различных сил (силы давления, вязкие силы и силы поверхностного натяжения) в одном или нескольких слоях порошка. Начальными данными для такого моделирования является формирование слоя порошка. Алгоритмы, реализующие насыпку порошка, могут быть различными. Наиболее простым является геометрический подход – такой, при котором гранулы порошка считаются несжимаемыми сферами (вообще говоря, различного диаметра), а процесс их насыпки описывается только из соображений геометрии и статики. На рис. 15 приведен результат формирования одного слоя порошка геометрическим методом в сравнении с результатами работы [2].

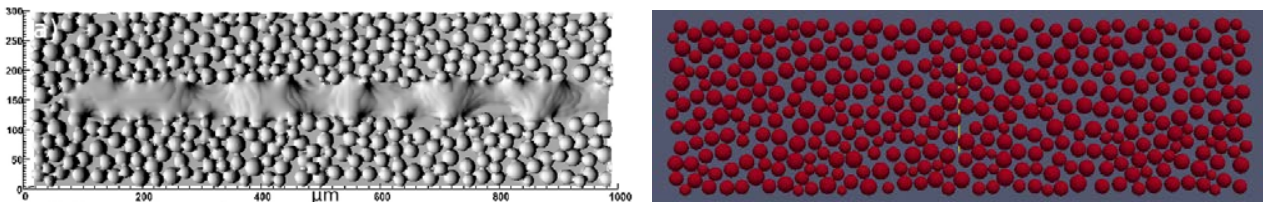


Рис. 15. Результаты работы геометрического метода насыпки (слева) в сравнении с данными из статьи [2] (справа)

Другой, более точной, группой методов являются методы насыпки, учитывающие некоторые физические свойства насыпаемого порошка. Одним из вариантов таких подходов является DEM (Discrete Element Method) [1].

На рис. 16 показан результат формирования нескольких слоев порошка методом DEM

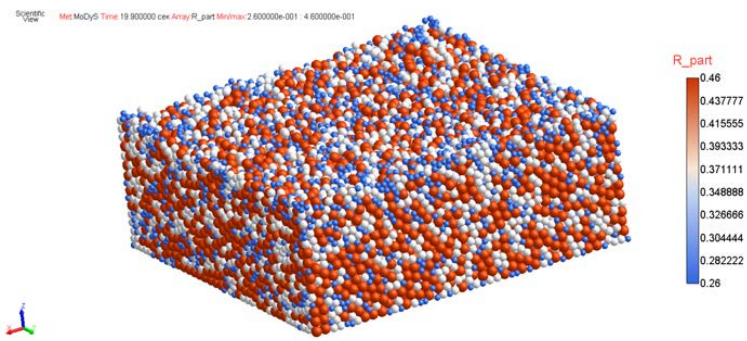


Рис. 16. Результат формирования нескольких слоев порошка методом DEM

Для моделирования тепломассопереноса на мезоуровне совокупность частиц и воздуха рассматривается, как континуальная двухфазная среда. Нестационарное течение многофазной среды рассматривается как течение несжимаемой вязкой жидкости. Численное решение системы уравнений, описывающей СЛС, ищется с использованием метода SPH.

На рис. 17 показано распределение фаз и температуры в тестовом расчете

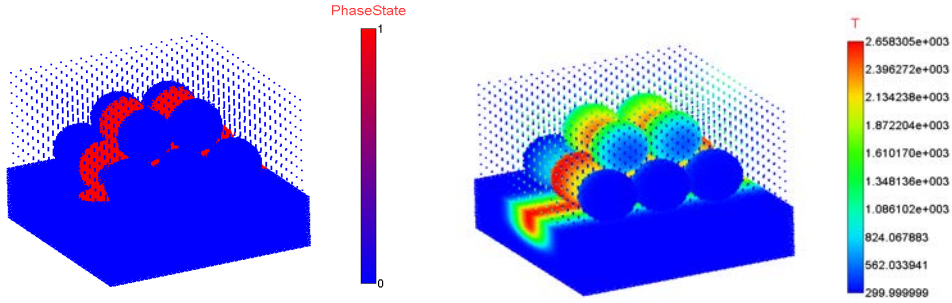


Рис. 17. Распределение фаз и температуры в тестовом расчете методом SPH

Блок «Производство»

Важной частью программного комплекса является модуль подготовки деталей к печати, который также используется для подготовки начальных данных для моделирования физических процессов на макроуровне. Процесс подготовки деталей к печати состоит из нескольких стандартных этапов:

– компоновка (размещение) деталей на платформе;
– генерация поддержек;
– формирование печатных слоев («слайсинг», «хэтчинг»);
– генерация управляющего кода.

На рис. 18–20 показаны результаты выполнения отдельных этапов.

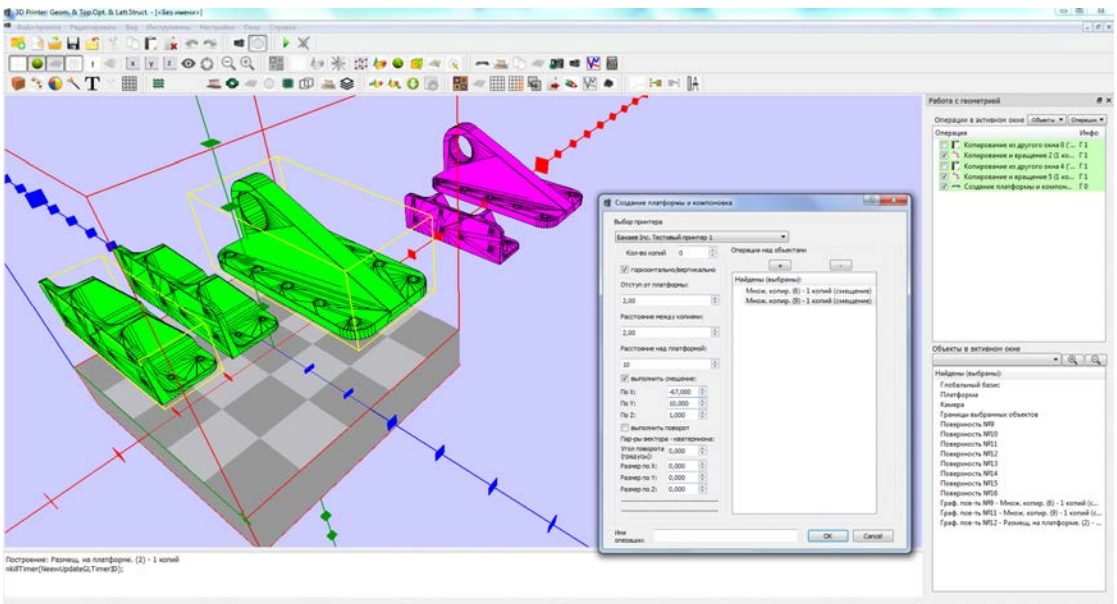


Рис. 18. Компоновка (размещение) деталей на платформе

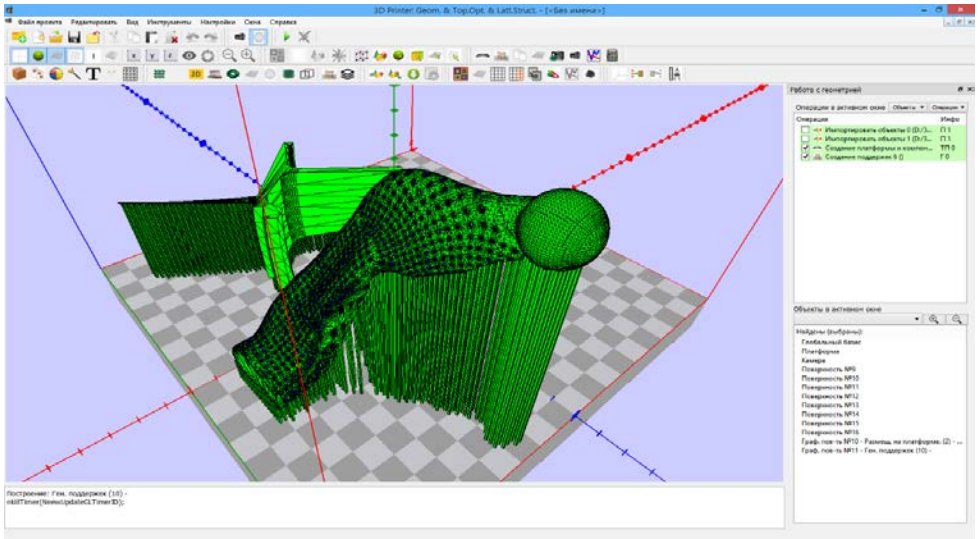


Рис. 19. Генерация поддержек

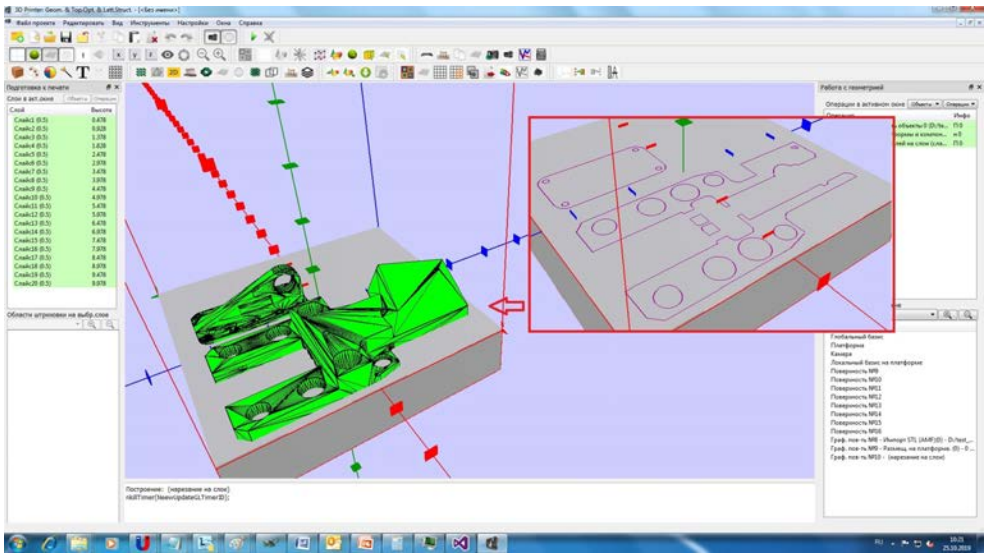


Рис. 20. Формирование печатных слоев («слайсинг», «хэтчинг»)

Блок «Интегрирующая оболочка»

Для моделирования на мезо- и макроуровнях необходимо знать механические и теплофизические свойства материалов. В проекте должна быть проведена верификация программ на двух металлах – сталь 03Х17Н14М2 (316L) и титановый сплав ВТ6 (Ti-6Al-4V). Для этого реализован модуль «Урсы». В 2019 году:

1. Собраны актуальные справочные и экспериментальные данные по теплофизическим свойствам стали 03Х17Н14М2 (316L) и титанового сплава ВТ6 (Ti-6Al-4V)

• на нормальной изобаре:

– теплоемкость,

– энтальпия,

– коэффициент температуропроводности,

– коэффициент теплопроводности,

– электрическое сопротивление,

– плотность,

– коэффициент линейного расширения,
– теплоты фазовых переходов,
– упругие модули и коэффициент Пуассона.
• зависимость давления от плотности на изотерме.

2. Выполнены анализ и статистическая обработка собранных данных.

3. Обработанные данные подготовлены в виде таблиц и введены в базу данных в качестве рекомендованных значений для проведения моделирования.

4. На основе экспериментальных данных разработаны уравнения состояния веществ в форме модели Тэта.

На рис. 21 и 22 показаны зависимости теплофизических свойств от температуры.

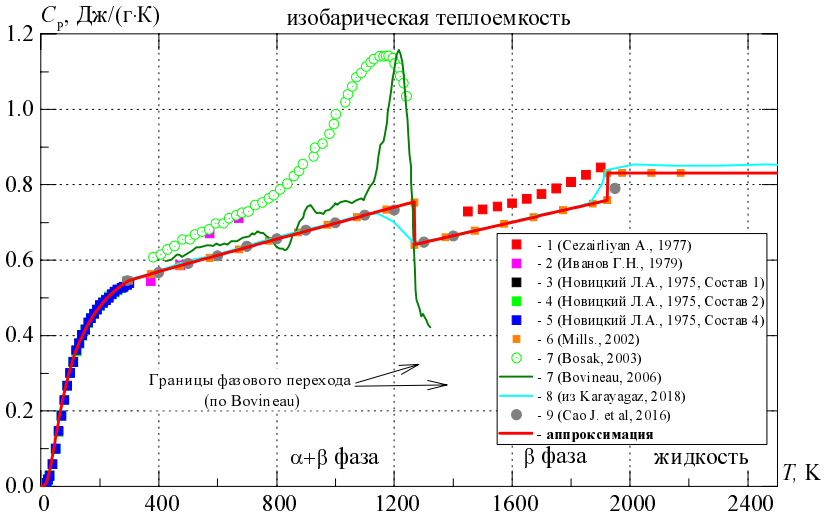


Рис. 21.Зависимость изобарической теплоемкости от температуры

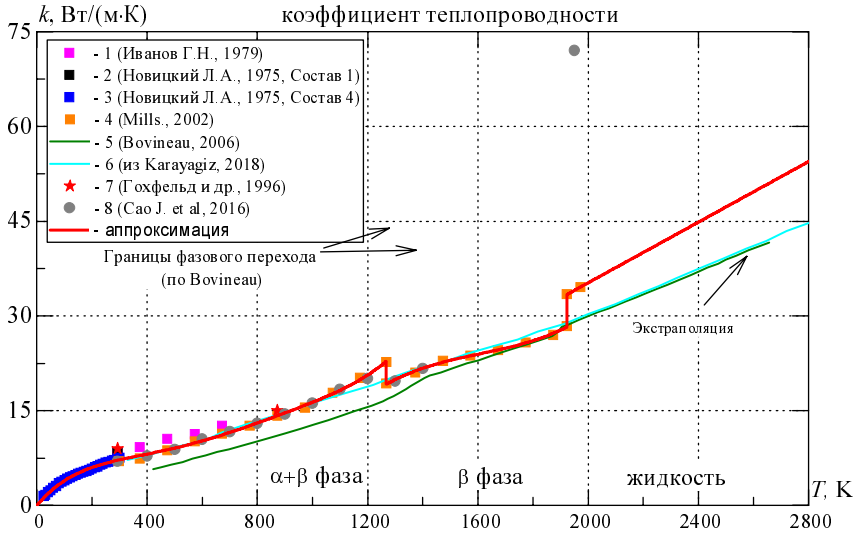


Рис. 22. Зависимость коэффициента теплопроводности от температуры

Для хранения свойств материалов разработана структура базы данных и начато ее заполнение. На рис. 23 показана структура базы данных.



На данный момент в базу данных введены:

- 2 класса материалов
- 2 материала (3 состава)
- 5 литературных источников
- 8 типов тепловых свойств материалов
- 4 типа прочностных свойств материалов

Рис. 23. Структура базы данных

Для работы со всеми модулями разработана интегрирующая графическая оболочка пользователя. На рис. 24 показан внешний вид интегрирующей оболочки.

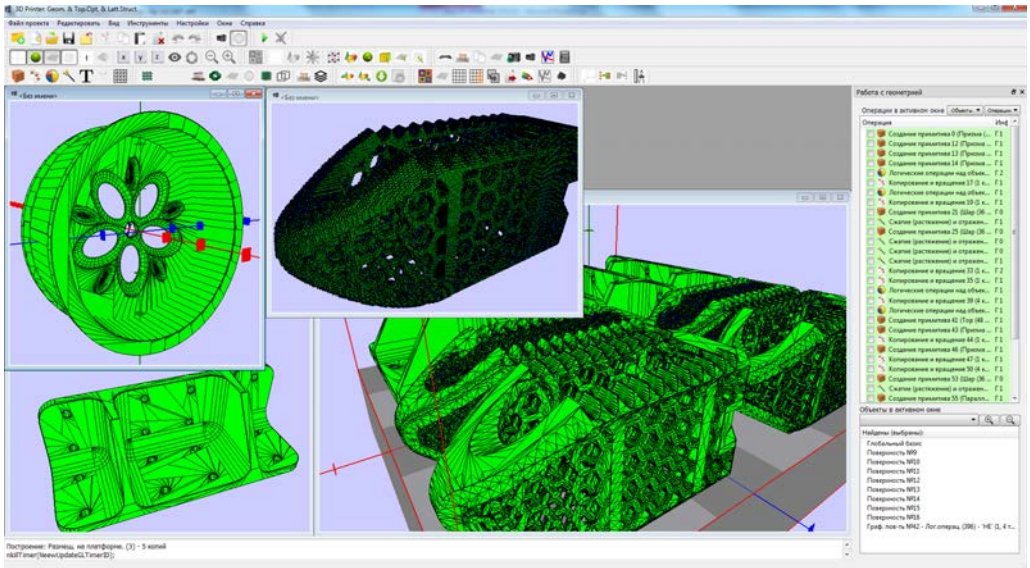


Рис. 24. Внешний вид интегрирующей оболочки

Заключение

Во ФГУП «РФЯЦ-ВНИИЭФ» реализуется проект «Виртуальный 3D-принтер». Цель проекта – создание комплекса программного обеспечения (ПО), обеспечивающего функциональность и широкие возможности для оптимизации формы деталей на основе методов топологической оптимизации (бионический дизайн), в том числе и с использованием решетчатых структур, управления обо-

рудованием для лазерного аддитивного производства, функциональность для определения оптимальных параметров работы оборудования на основе математического моделирования.

В докладе приведены общие сведения о проекте и результаты работы по проекту в 2019 году.

В 2020 году планируется продолжение проекта. Задачи 2020 года – продолжение развития функциональности модулей, интеграция разработанных модулей в единый программный продукт, верификация и валидация на конкретных деталях.

Литература

1. Zohdi T. I. Additive particle deposition and selective laser processing-a computational manufacturing framework. Comput. Mech., 2014, 54, pp. 171-191.

2. Saad A. Khairallah, Andy Anderson, 2014. Mesoscopic simulation model of selective laser melting of stainless steel powder . J. Mat. Proc. Techn. 214(2014), pp. 2627–2636.

"VIRTUAL 3D-PRINTER" PROJECT

A. N. Bykov, V. V. Popov, Yu. N. Deryugin, M. V. Vetchinnikov, D. Yu Dyanov, S. N. Polishchuk, V. G. Kudelkin, E. A. Sinitsin, A. N. Bakhaev, M. V. Zhuravleva, O. I. Britova, M. A. Titov, A. V., Baranov, L. V. Gamov, I. G. Mileshin, V. Yu. Kuznetsov, P. A. Mashenkin, E. F. Golubkova, A. S. Danilov, M. V. Medvedkina, A. A. Kayakin, A. I. Lomaykin, O. V. Sapronova, E. A. Veselova, A. B. Emelyanov, A. A. Lazarev, I. S. Sapronov, N. N. Balueva, O. S. Larkina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The report describes Virtual 3D-Printer Project. The goal of the project is to develop a software complex that would provide functionality and wide capabilities to optimize the shapes of the parts on the basis of topological optimization methods (bionic design), including the ones with grid structures, the control over the equipment for laser additive production, and functionality to find optimum operation parameters on the basis of mathematical simulation.

Key words: additive technologies, selective laser melting, multi-level mathematical simulation, topological optimization, grid structures, generative design, implicit representation of geometry, control over technological equipment.

УДК 519.6
DOI 10.53403/9785951505071_2022_103

ДВУМЕРНЫЕ И ТРЕХМЕРНЫЕ РАСЧЕТЫ ПО МЕТОДИКЕ ТИМ МОДЕЛИРОВАНИЯ РАБОТЫ ВЗРЫВНОГО РЕЗАКА НА ОСНОВЕ МЕТОДА УДАРНЫХ ВОЛН РАЗРЕЖЕНИЯ ДЛЯ РЕЗКИ ТРУБ

Л. Я. Габзетдинова, И. Г. Новиков, Т. Н. Половникова, А. В. Самодолов, И. В. Шиберин, И. В. Занегин, А. Н. Катыков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Объектом исследования является подход проведения численного моделирования работы взрывного резака в трехмерной постановке по лагранжевой методике ТИМ. В докладе приведены результаты расчетов работы взрывного резака для резки труб изнутри в двумерной и трехмерной постановках. Показано сравнение полученных результатов между собой и сравнение с экспериментальными данными. Трехмерные расчеты проведены с использованием смешанных ячеек, появляющихся на границах раздела разных веществ, двумерный расчет проведен без смешанных ячеек с явно выделенными подвижными контактными границами. Целью работы является отработка подхода проведения подобного класса расчетов в трехмерном случае, влияния смешанных ячеек на результаты расчетов.

Ключевые слова: трехмерные и двумерные расчеты, численное моделирование, методика тим, взрывной резак, резка труб изнутри, смешанные ячейки.

Трехмерные и двумерные расчеты, численное моделирование, методика ТИМ, взрывной резак, резка труб изнутри, смешанные ячейки

В РФЯЦ-ВНИИЭФ для резки толстостенных стальных опор предложен сотрудниками ИФВ метод [1, 2], основанный на образовании и взаимодействии ударных волн разрежения (УВР) в железе или стали [3]. Работоспособность этого метода при резке толстостенных стальных конструкций была показана более 10 лет назад.

Причиной образования ударных волн разрежения в железе является наличие аномального участка адиабаты разгрузки в области (α-Fe $\Leftrightarrow$ ε-Fe) фазового перехода [4, 5]. При взаимодействии двух ударных волн разрежения растягивающие напряжения возникают в очень узкой зоне материала, определяемой шириной фронта УВР, что приводит к образованию «гладкой» поверхности откола. Подобный характер носит разрушение стальной подложки в зоне столкновения на ее поверхности двух направленных навстречу друг другу детонационных волн (ДВ) в слое взрывчатого вещества (ВВ), размещенном в контакте с этой подложкой.

В данной работе проведено численное моделирование работы взрывного резака для резки изнутри стальной трубы диаметром 762 мм и толщиной стенок 51 мм. Расчеты проводились в двух постановках. В первом случае использовалось двумерное приближение задачи. А во втором – трехмерное. В трехмерном случае рассмотрен в силу симметрии задачи сектор в 120 градусов, состоящий участка трубы и двух элементов взрывного резака. Также приведены результаты трехмерной задачи, где моделировалась нештатная работа взрывного резака с инициированием части ВВ.

Полученные результаты расчетов сравнивались с результатами экспериментов, проведенных в ИФВ РФЯЦ-ВНИИЭФ.

Численное моделирование выполнено с помощью методики ТИМ [6, 7], предназначенной для решения задач газодинамики, теплопроводности, магнитной гидродинамики с учетом диффузии магнитного поля, нестационарной упругопластичности с учетом разрушения материалов, детонации взрывчатых веществ на неструктурированных многоугольных/многогранных лагранжевых сетках. По методике ТИМ допускается расчет с применением смешанных ячеек, как задач с осевой симметрией в цилиндрической системе координат (двумерные задачи), так и полноценных трехмерных задач в декартовой системе координат. Трехмерный расчет проведен с использованием смешанных ячеек в автоматическом режиме счета без вмешательства пользователя и остановки счета. Применяемый подход счета основан на перестроении всей счетной области с переинтерполяцией величин на новую сетку. В результате перестроения на границе раздела веществ появляются смешанные ячейки, дальнейший счет ведется в лагранжевом приближении без допущения перетекания веществ между ячейками.

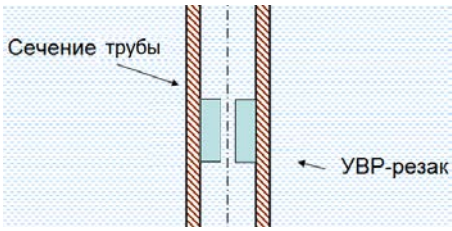
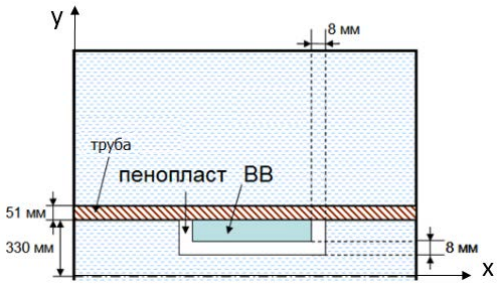


Рис. 1. Общий вид задачи

Общий вид задачи представлен на рис. 1.

Данная задача может быть сведена к двумерному случаю для численного моделирования в цилиндрической системе координат, рассматривая сечение половины геометрии с осью вращения OX. В данном случае элемент шашки будет представлен в виде тора. На рис. 2 схематично приведена двумерная геометрия задачи.



Вещества	Плотность, г/см^3
Железо (опора)	7.85
ВВ	1.6
Пенопласт	0.7

Рис. 2. Двумерная геометрия задачи и описание веществ

Размер ВВ составлял 70×14 мм. Уравнения состояния веществ были взяты в форме Ми-Грюнайзена. Для учета упругопластических свойств трубы использовалась модель Глушака [8] в сочетании с моделью разрушения «хрупкий откол», значение $P_{\text{отк}} = -1.4$ ГПа.

Устройство УРВ-резака для внутренней резки труб состоит из 6 элементов взрывного резака, расположенных последовательно с небольшим зазором. Поэтому для более точного численного моделирования необходимо использовать трехмерное приближение задачи. Также рассматривая трехмерную задачу появляется возможность изучения нештатной работы взрывного резака, связанной с разновременностью инициирования ВВ либо с отказом инициирования части ВВ.

Общий вид трехмерной геометрии приведен на рис. 3.

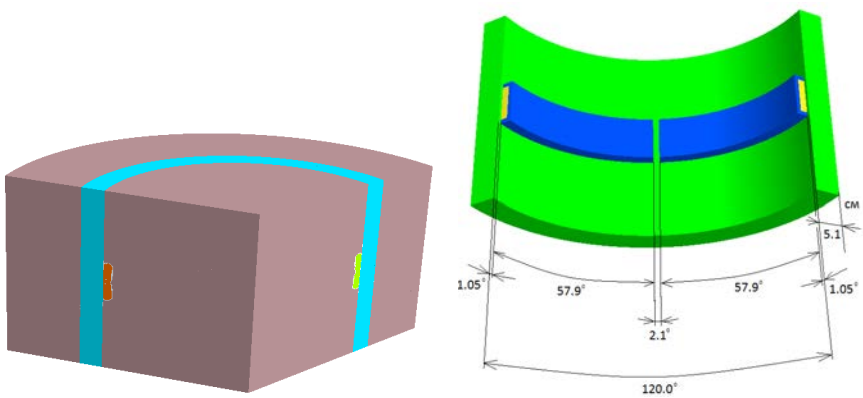


Рис. 3. Общий вид трехмерной геометрии

Счетная сетка расставлялась с учетом особенностей геометрии и выделения зон измельчения в районе образования трещины. Минимальный размер ячеек составлял ~0.01 см. Общее количество ячеек в двумерной задаче составляло около 1.2 млн, в трехмерной ~50 млн.

Расчет детонации проводился по модели Чепмена-Жуге с контролем над скоростью распространения фронта детонационной волны. Инициирование ВВ осуществлялось в двух нижних точках в двумерном случае и вдоль кривых (см. рис. 4) в трехмерном случае. В трехмерном случае рассмотрен расчет двух вариантов задачи: со штатным инициированием двух ВВ и вариант инициирования у одного из ВВ половины его части.

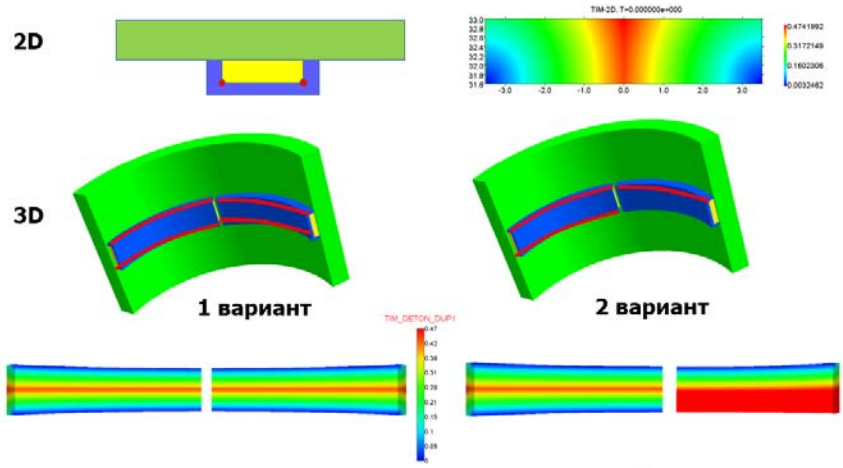


Рис. 4. Инициирование ВВ

Результаты расчетов приведены в виде распределения поля давлений (ГПа) и плотностей (г/см^3). В трехмерном случае приведены сечения плоскостями (см. рис. 5), проходящие через середину ВВ.

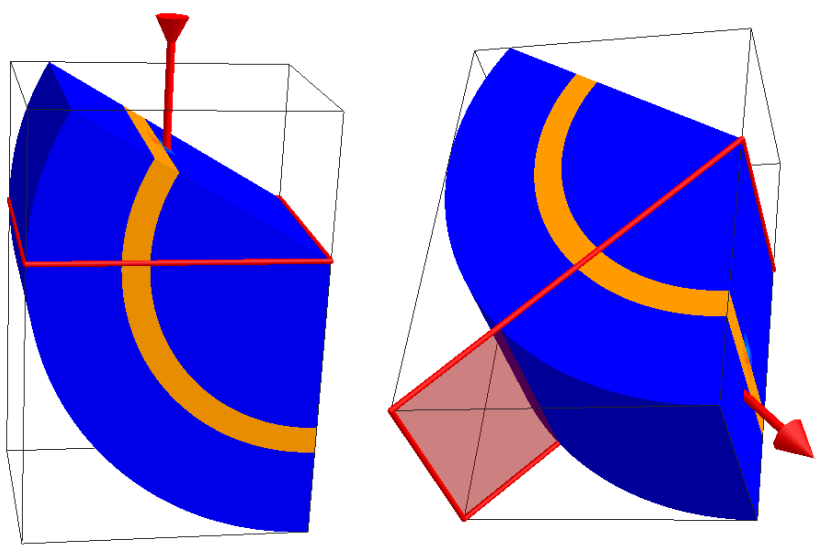


Рис. 5. Сечение плоскостью

На рис. 6 приведено распространение детонационной волны на момент времени 4.68 мкс. Максимальное давление, возникающее в трубе, на момент времени схождения детонационных волн, составляет ~70 ГПа.

Динамика зарождения откола в трубе для расчетов представлена на рис. 7–10 в виде распределений плотностей (г/см^3) на различные моменты времени.

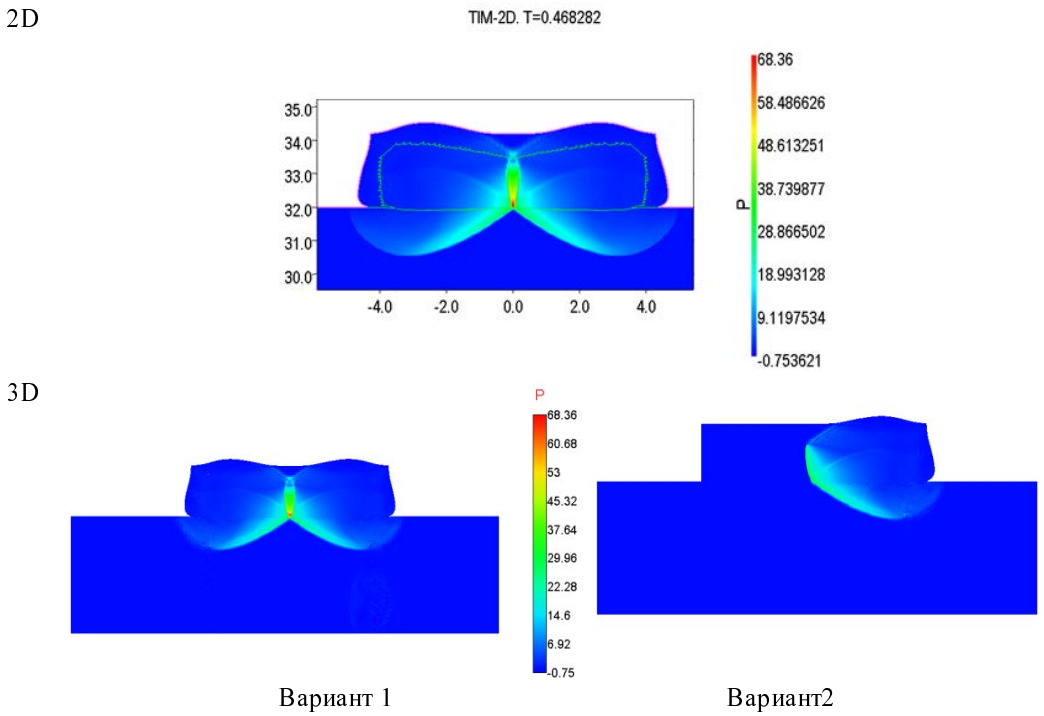


Рис. 6. Распространение детонационной волны, поле давлений (ГПа), момент времени 4.68 мкс

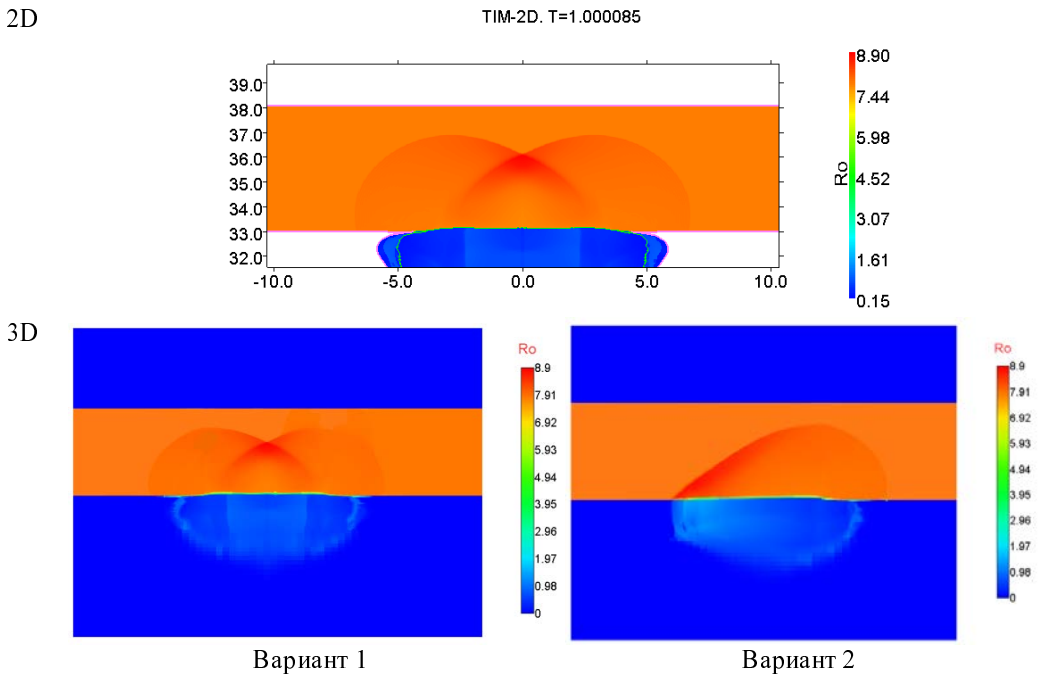


Рис. 7. Распределение поля плотности (г/см^3), момент времени 10 мкс

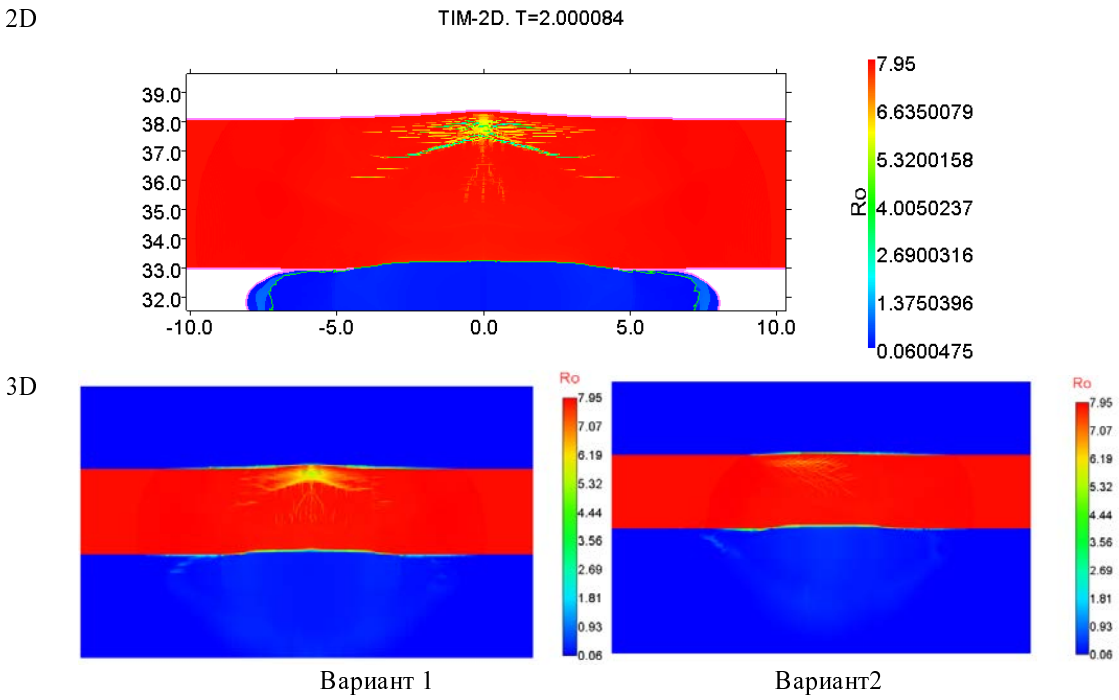


Рис. 8. Распределение поля плотности (г/см^3), момент времени 20 мкс

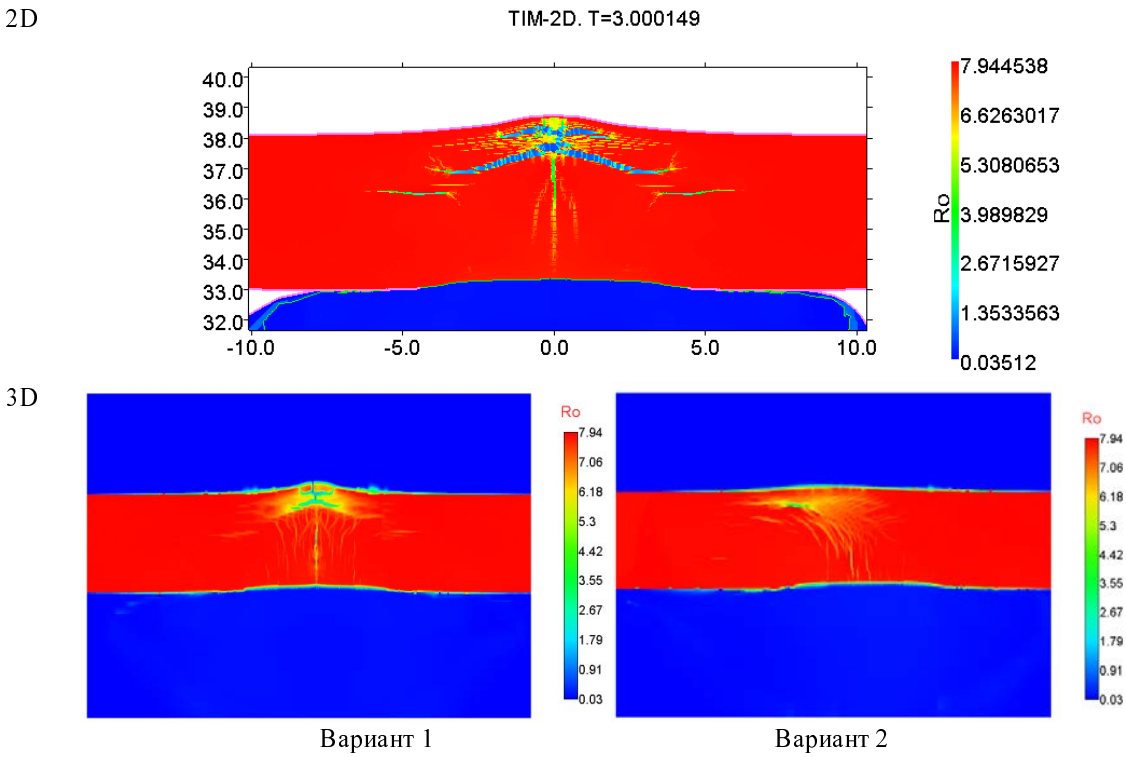
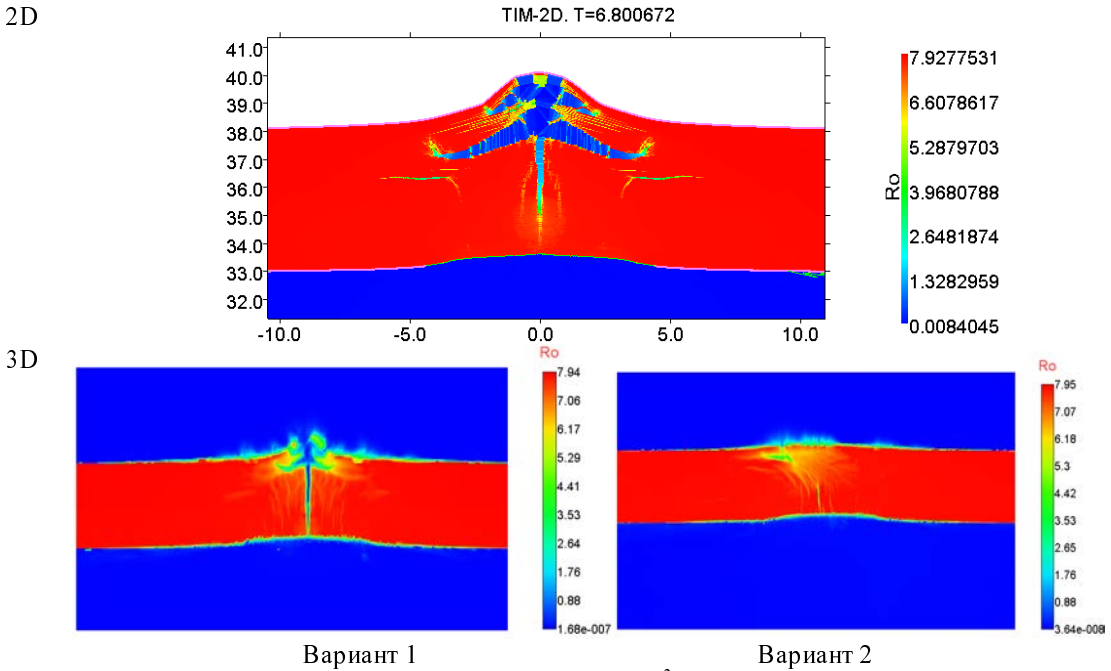


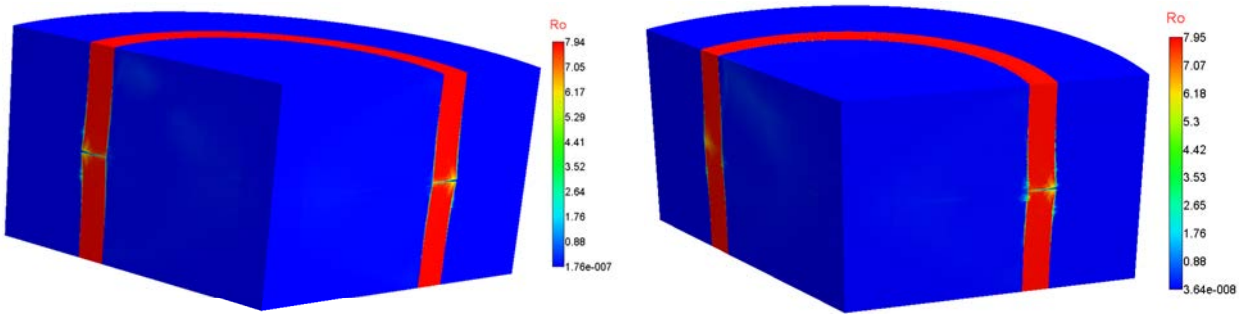
Рис. 9. Распределение поля плотности (г/см^3), момент времени 30 мкс



Вариант 1 Вариант 2

Рис. 10. Распределение поля плотности (г/см^3), момент времени 68 мкс

На рис. 11 приведен общий вид трехмерной геометрии на момент окончания счета.



Вариант 1 Вариант 2

Рис. 11. Общий вид трехмерной геометрии на момент окончания счета

Анализ проведенных расчетов показывает, что при штатной работе в двумерном и трехмерном случае наблюдается полный разрыв трубы (следовательно, моделирование можно проводить в упрощенной двумерной постановке). Моделирование нештатной ситуации инициированием только части одного из ВВ приводит к частичному разрушению трубы, полного разрыва не наблюдается, задача приобретает полностью трехмерный характер. Наличие смешанных ячеек не оказывает заметного влияния на характер разрушения трубы и формирование зоны гладкого откола.



Рис. 12. Эксперимент

Проведенный эксперимент в ИФВ показывает полное разрушение трубы, центральная часть фрагмента трубы в месте размещения взрывного резака разрушена на куски по диаметру. На месте проведения опыта обнаружены две крупные части разрезанной трубы, рис. 12.

Выводы

Проведенные расчеты показывают хорошее согласие двумерных и трехмерных расчетов с экспериментальными данными. Показан возможный вариант развития нештатной ситуации с отказом работы одной из части ВВ, в результате чего полного разделения трубы не наблюдается.

В результате проведенных трехмерных расчетов, был отработан подход, когда вся область считается в лагранжевом представлении с применением смешанных ячеек. Основная проблема при проведении подобного класса расчетов – это сильное искажение сетки в зоне тыльного откола. Введение оптимизации сетки во фрагментах или во всей математической области позволяет применять автоматическую коррекцию сетки. Используется глобальный пересчет величин на определенные этапы в задаче (по достижению критериев качества сетки, вырабатываемых в блоке локальной перестройки сетки в автоматическом режиме) и локальное исправление сетки с пересчетом величин (в небольших зонах задачи, где сетка подвержена максимальным деформациям). В результате перестроения на границе раздела веществ появляются смешанные ячейки. Между этапами глобального пересчета величин не допускается перетекание вещества из ячейки в ячейку, т. е. счет ведется в лагранжевом представлении. Данный подход позволяет решать задачи без ручного вмешательства пользователя в расчеты, что снижает риск влияния «человеческого фактора» на получаемое решение задач. Помимо автоматизации, за счет выделения зон измельчения счетной сетки с последующей ее динамической адаптацией под решение удается существенно экономить на общем количестве точек и, как следствие, занимать существенно меньшие вычислительные ресурсы, нежели при использовании равномерной расстановки сетки во всей области задачи. Проведенные расчеты показывают применимость методики ТИМ для проведения подобных классов расчетов с учетом смешанных ячеек.

Литература

1. Устройство для внутренней взрывной резки труб. Свидетельство на полезную модель № 7195 от 25.03.97. Батьков Ю. В., Новиков С. А. и др. БИ № 7, 1998.

2. Способ резки конструкций и генератор взрывной волны. Патент РФ № 2105946. БИ № 6. Батьков Ю. В., Новиков С. А, Синицына Л. М. 1998.

3. Новиков С. А. Взрывные технологии и конверсия // Вопросы атомной науки и техники. Серия: Импульсные реакторы и простые критические сборки. 1994. Вып. 2. С.40–47.

4. О возможности ударных волн разрежения. ЖЭТФ. Зельдович Я. Б. 1946. Вып. 4. С. 363–364.

5. Об ударных волнах разрежения в железе. ЖЭТФ. Иванов А. Г., Новиков С. А. 1961. Вып. 6. Т. 40. С. 1879–1882.

6. Соколов С. С., Панов А. И., Воропинов А. А., и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 2. С. 37–52.

7. Соколов С.С., Воропинов А.А., Новиков И.Г., и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29-43.

8. Глушак Б. Л., Новиков С. А., Батьков Ю. В. Определяющее уравнение Al и Mg для описания высокоскоростного деформирования УВ // Физика горения и взрыва. 1992. Т. 28, № 1.

2D AND 3D COMPUTATIONS OF THE SIMULATED OPERATION OF THE EXPLOSIVE CUTTER ON THE BASIS OF THE RAREFACTION WAVES TO CUT PIPES WITH IMPLEMENATION OF TIM-METHOD

L. Ya. Gabzetdinova, I. V. Zanegin, A. N. Katykov, I. G. Novikov, T. N. Polovnikova, A. V. Samodolov, I. V. Shiberin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

An approach to numerical simulation of the explosive cutter operation in 3D using Lagrangian TIM-method is under investigation. The work describes computational results for the operation of an explosive cutter to cut pipes from inside in 2D and 3D setup. The results are compared with each other and with experimental data. 3D computations were carried out using mixed cells that appear at the interfaces of different materials, 2D computations were performed without mixed cells with explicit moving contact boundaries. The goal of the work is polishing the approach to the performance of this class of computations in 3D case and the analysis of the influence of mixed cells on computational results.

Key words: 3D and 2D computations, numerical simulation, TIM-method, explosive cutter, cutting pipes from inside, mixed cells.

УДК 519.6
DOI 10.53403/9785951505071_2022_111

ПРОГРАММНОЕ СРЕДСТВО ИМПОРТА ВНЕШНИХ НАГРУЗОК В РАСЧЕТНУЮ МОДЕЛЬ ЛОГОС-ПРОЧНОСТЬ, РЕАЛИЗОВАННОЕ В РАМКАХ ПРЕПОСТПРОЦЕССОРА ПАКЕТА ПРОГРАММ «ЛОГОС»

А. В. Гордеев, В. Н. Дюпин, А. Ю. Еременко, Н. А. Мустаева, А. С. Санталов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе представлено описание программного средства импорта внешних нагрузок в расчетную модель «ЛОГОС-Прочность», реализованного в рамках препостпроцессора «ЛОГОС-Препост», входящего в состав пакета программ «ЛОГОС».

Ключевые слова: ЛОГОС, препостпроцессор, ЛОГОС-Препост, прочностная модель, импорт внешних нагрузок.

Доклад посвящен программному средству импорта внешних нагрузок в расчетную модель «ЛОГОС-Прочность», реализованному в рамках развития препостпроцессора «ЛОГОС-Препост» пакета программ «ЛОГОС».

Пакет программ «ЛОГОС» является отечественным многофункциональным пакетом программ, разработанным для инженерного анализа и суперкомпьютерного моделирования. Препостпроцессор «ЛОГОС-Препост» пакета программ «ЛОГОС» является программным комплексом для задания начальных, граничных условий расчетных задач и анализа результатов моделирования.

Программное средство задания внешних нагрузок решает задачу переноса внешних нагрузок на прочностную модель. Источником данных по внешним нагрузкам может являться как коммерческое программное обеспечение (КПО, например, PATRAN), так и программные модули пакета программ «ЛОГОС» (например, ЛОГОС-ТМП). Источники внешних нагрузок осуществляют расчет нагрузок, которые используются в гармоническом анализе в рамках программного модуля «ЛОГОС-ПА». Результаты гармонического анализа используются для расчета влияния гидродинамической модели на прочностную конструкцию. При расчете влияния гидродинамической модели на прочностную конструкцию осуществляется интерполяция сеточных данных на прочностную конструкцию. Результаты интерполяции формируют поле (физический объект, описываемый математическим скалярным или векторным полем). Сформированное поле используется в прочностном расчете.

При разработки программного средства импорта внешних нагрузок были сформулированы требования к программному средству:

– создание программного средства импорта внешних нагрузок;
– создание препроцессора программного средства для импорта сеточных данных гидродинамики;
– создание постпроцессора программного средства для интерполяции сеточных данных на прочностную модель;
– создание алгоритмов для визуализации импортируемых данных;
– интеграция внешних нагрузок в объектную модель «ЛОГОС-Прочность»;
– использование полей для задания прочностной модели.

При проектировании программного средства задания внешних нагрузок был проведен анализ функциональных возможностей коммерческого программного обеспечения «PATRAN» [1]. По ре-

зультату анализа была спроектирована архитектура программного средства и разработан план полного цикла процесса импорта внешних нагрузок.

На рис. 1 представлена схема взаимодействия программного средства с компонентами препостпроцессора «ЛОГОС-Препост».



Рис. 1. Схема взаимодействия программного средства импорта внешних нагрузок с компонентами препостпроцессора «ЛОГОС-Препост»

Программное средство импорта внешних нагрузок включает:

- плагин импорта данных;
- препроцессор;
- постпроцессор.

Препроцессор программного средства осуществляет ввод исходных данных для создания нового поля в объектной модели «ЛОГОС-Прочность» и визуализирует сеточные данные внешних файлов.

Постпроцессор программного средства осуществляет интерполяцию расчетных величин с внешних сеток на целевой набор данных, создает новый экземпляр поля в объектной модели «ЛОГОС-Прочность» и осуществляет визуализацию полученного поля.

Программное средство осуществляет взаимодействие с хранилищем внешних нагрузок, объектной моделью «ЛОГОС-Прочность» и визуализатором.

Взаимодействие программного средства со счетным модулем обеспечивается путем экспорта файлов модели «ЛОГОС-Прочность».

Полный цикл импорта внешних нагрузок включает этапы:

- создания прототипа нового экземпляра поля;
- выбора сеточных данных для наложения поля;
- импорта внешних сеточных данных;
- интерполяции сеточных данных;
- визуализации результатов интерполяции данных;
- создания нового экземпляра поля;
- задания полей на начальных и граничных условиях.

На начальном этапе препроцессор программного средства собирает данные для создания нового поля и передает плагину импорта данных полный путь до внешнего файла, который содержит сеточные данные нагрузки. Файлы внешних нагрузок формируются по результатам расчета смежных задач гидродинамики, аэродинамики и прочих методик.

Плагин импорта данных поддерживает методы для загрузки бинарных и текстовых форматов внешних данных:

– бинарный формат CGNS – формат хранения расчетных сеток и информации о нагрузке прочностной модели. Формат используется в коммерческом пакете «PATRAN»;

– бинарный формат EFR – формат хранения данных постобработки расчетной модели. Формат используется в программном комплексе Scientific View, который входит в пакет программ «ЛОГОС»;

– текстовый формат CSV – промежуточный формат хранения данных. Формат используется для обеспечения совместимости с расчетными модулями, которые не поддерживают форматы выходных данных CGNS и EFR.

Алгоритм импорта внешних данных заключается в определении числа временных шагов, загрузке узлов расчетной сетки из хранилища временных нагрузок и итерационной загрузке сеточных данных для заданного расчетного шага.

На рис. 2 представлена блок-схема алгоритма загрузки внешних данных.

Для оптимизации размещения сеточных данных в оперативной памяти в рамках программного средства был реализован алгоритм отложенной загрузки. Согласно реализованному алгоритму, на этапе пробега по временным шагам постпроцессор программного средства вызывает препроцессор и перезагружает сеточные данные.

Постпроцессор программного средства использует результаты препроцессора для формирования нового экземпляра поля в объектной модели «ЛОГОС-Прочность». Интеграция программного средства импорта внешних нагрузок в объектную модель «ЛОГОС-Прочность» осуществлялась путем регистрации нового элемента «Поля» в разделе «Функциональные зависимости» объектной модели «ЛОГОС-Прочность». Процесс регистрации нового элемента объектной модели сводится к описанию структуры объекта в конфигурационных файлах препостпроцессора «ЛОГОС-Препост» [2].

На рис. 3 представлена логическая структура данных программного средства.

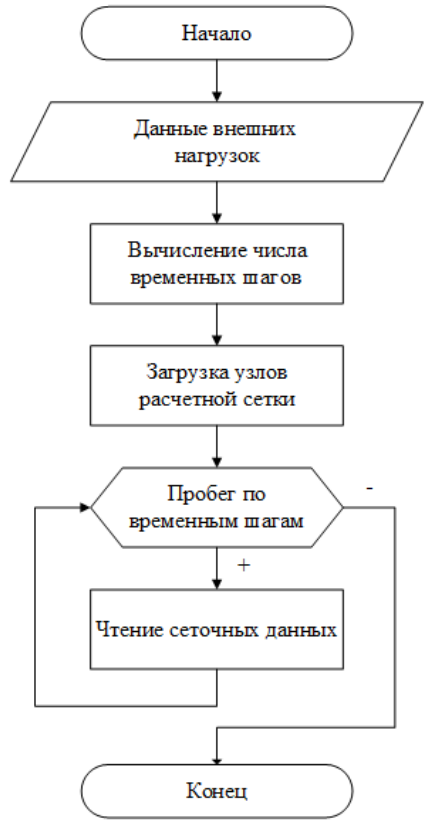


Рис. 2. Алгоритм импорта внешних нагрузок

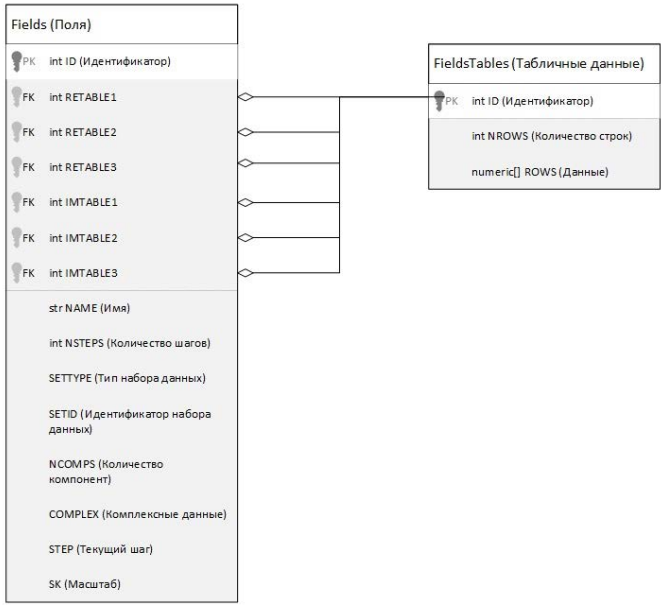


Рис. 3. Логическая структура данных программного средства

Структура Fields – описывает экземпляры полей:
– ID – идентификатор поля;
– NAME – название поля;
– SETTYPE – целевой набор данных (узлов, граней, элементов);
– SETID – идентификатор целевого набора данных;
– RETABLE1..3, IMTABLE1..3 – внешние ключи на данные поля;
– диагностические данные.

Структура FieldsTables – описывает экземпляры сеточных данных поля:
– ID – идентификатор набора данных;
– ROWS – массив сеточных данных.

Для оптимизации размещения сеточных данных в поле наиболее ресурсоемкие данные были перенесены в отдельный экземпляр объекта «Табличные данные». Связь данных между объектом «Поля» и «Табличные данные» осуществляется по внешним ключам «RETABLE1», «RETABLE2», «RETABLE3», «IMTABLE1», «IMTABLE2», «IMTABLE3».

Алгоритм расчета сеточных величин сводится к интерполяции внешних нагрузок на целевой набор данных поля. Интерполяция осуществляется методом «Максимальное расстояние» [3]. Метод максимального расстояния заключается в построении графа, объединяющего узлы внешнего набора данных с узлами целевого набора данных поля. Постпроцессор программного средства поддерживает 2 метода ограничения на построение ребер графа:

– ограничение на максимальное расстояние между узлами сеток;

– ограничение на максимальное количество узлов, объединенных с узлом внешней сетки.

Для оптимизации процесса интерполяции сеточных данных препроцессор осуществляет предварительный расчет весов *Wi*, где *i* – номер узла поля, d_j – расстояние от *j*-того узла внешней сетки до *i*-того узла поля (1).

$$W_i = \frac{1/d_i}{\sum_{j=1}^{N} 1/d_j}. \quad (1)$$

Расчет интерполяционного значения *P* для *i*-того узла поля вычисляется по формуле (2) как сумма произведения весов *Wj* на величину внешней нагрузки для *j*-того узла P_j [4].

$$P = \sum_{j=1}^{N} W_j P_j. \quad (2)$$

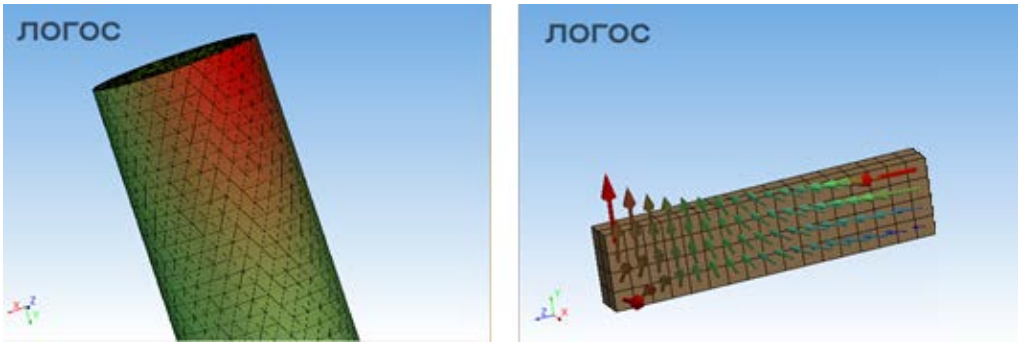
Для визуализации полученных полей постпроцессор программного средства передает визуализатору информацию о созданном поле и дополнительную информацию, содержащую параметры настройки отображения поля.

Модуль визуализации включает библиотеку алгоритмов отображения расчетных сеток и сеточных данных и поддерживает возможность послойного отображения данных. В рамках работ по созданию программного средства импорта внешних нагрузок в библиотеку алгоритмов визуализации были добавлены методы отображения:

– скалярных и векторных данных внешних расчетных сеток;
– полей с сеточными данными;
– графа связей между сеточными данными;
– раскраски сеточных данных.

На рис. 4,а показан пример отображения скалярного поля на фрагменте расчетной сетки цилиндрической формы. На рис. 4,б продемонстрирован результат визуализации векторного поля на фрагменте расчетной сетки прямоугольной формы.

Препостпроцессор «ЛОГОС-Препост» содержит ряд встроенных средств для реализации быстрой интеграции связей между данными объектной модели. Например, для поддержки полей в разделе объектной модели «Нагрузки | Механика | Давление» был добавлен новый элемент «Давление, заданное полем». Связь нового элемента с объектом «Поля» осуществляется с помощью встроенных средств выборки данных препостпроцессора «ЛОГОС-Препост».



а б

Рис. 4. Визуализация внешних нагрузок

С 2019 года программное средство стало активно использоваться в методических и производственных расчетах.

Литература

1. PATRAN User Guide [Электронный ресурс] – Режим доступа: https://simcompanion. mscsoftware.com/infocenter/index?page=content&id=DOC10145 (дата обращения: 13.01.2020).

2. Анищенко А. А., Дерюгин В. И., Дюпин В. Н., Иванов К. В., Санталов А. С., Санталова Е. Е. ПреПостПроцессор ЛОГОС-ПреПост. Архитектура уровня бизнес-логики, хранение, импорт и экспорт данных // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2014. Том, № 2. С. 78.

3. Шарый С. П. Курс вычислительных методов. – Новосибирск: Новосиб. гос. ун-т, 2019.

4. Зенков А. В. Численные методы: учеб. пособие. – Екатеринбург: Изд-во Урал. ун-та, 2016.

SOFTWARE TOOL TO IMPORT EXTERNAL LOADS INTO “LOGOS-SA” COMPUTATIONAL MODEL REALIZED WITHIN PREPOSTPROCESSOR OF “LOGOS” SOFTWARE PACKAGE

A. V. Gordeev, V. N. Dyupin, A. Yu. Eremenko, N. A. Mustaeva, A. S. Santalov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes a software tool of import external loads into the LOGOS-SA simulation model realized within prepostprocessor LOGOS Prepost, which is a part of “LOGOS” software package.

Key words: LOGOS, prepostprocessor, LOGOS Prepost, strength model, import of external loads.

УДК 519.6
DOI 10.53403/9785951505071_2022_116

БИБЛИОТЕКА УРС-ОФ ДЛЯ ВЕКТОРИЗОВАННЫХ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ТЕПЛОФИЗИЧЕСКИХ ФУНКЦИЙ НА ОБЩЕЙ ПАМЯТИ

Д. Г. Гордеев, Н. Н. Жильникова, Д. Н. Кидямкина, В. Г. Куделькин, М. В. Куликова, О. Н. Шумилина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе представлено описание структур данных, организации программ и вычислений в векторизованной версии библиотеки УРС-ОФ, позволяющей проводить параллельные вычисления теплофизических функций на общей памяти с использованием стандарта OpenMP. Дан краткий экскурс в историю создания библиотеки УРС-ОФ, описан порядок взаимодействия библиотеки УРС-ОФ с прикладной программой, приемы распараллеливания вычислений. Приведены примеры, иллюстрирующие особенности распараллеливания условных конструкций и итерационных циклов с использованием массивов масок, а также распараллеливание программ типа СуперУРС.

Ключевые слова: библиотека программ УРС-ОФ, распараллеливание на общей памяти, стандарт OpenMP, ширина векторного регистра, автовекторизация, условные конструкции, итерационные циклы, эффективность распараллеливания.

В Институте теоретической и математической физики (ИТМФ) РФЯЦ-ВНИИЭФ функционирует и развивается пакет УРС-ОФ, включающий в себя два раздела: библиотеку программ расчета теплофизических свойств (функций) веществ и систему информационного обеспечения [1]. Доклад посвящен библиотеке программ расчета теплофизических свойств веществ.

Библиотека УРС-ОФ представляет собой библиотеку подпрограмм нескольких типов: уравнение состояния, пробег фотонов, уравнение для расчета упругопластических свойств, электро- и теплопроводности, средних степеней ионизации. Она используется большинством программ численного решения задач математической физики, разработанных в ИТМФ. История создания пакета УРС-ОФ во ВНИИЭФ насчитывает более четырех десятилетий и подробно описана в работах [1–8]. Приведем основные этапы в развитии пакета:

- До 1983 – УРС-ОК (одномерный комплекс) программы на автокоде МАДЛЕН на БЭСМ-6 [2, 3];
- 1980 – УРС-ЕС-ОК программы на FORTRAN-IV на ЭВМ типа ЕС [4]-[6];
- Конец 80-х – УРС-ОФ. ОФ – трактуется как Общий Фонд;
- 1995 – поддержка различных компиляторов FORTRAN на многопроцессорных ЭВМ с универсальными процессорами и ПВМ с использованием общей памяти [7, 8];
- Начало 2000-х – начало перехода на FORTRAN-90, адаптация библиотеки для работы на распределенной памяти;
- 2009 – все программы переведены на стандарт FORTRAN-90 [1];
- 2012–2015 – версия библиотеки для работы на графических ускорителях [9, 10].

До 2010 года разработанные программы библиотеки УРС-ОФ были ориентированы на работу с универсальными процессорами. Использование процессоров с SIMD (Single Instruction Multiple Data) архитектурой вычислителей открыло новые возможности по ускорению расчетов теплофизических функций с применением параллелизма на уровне данных. В современных процессорах такие

вычисления проводятся в регистрах с поддержкой векторных операций. Адаптация существующих и разработка новых программ библиотеки УРС-ОФ для эффективного использования SIMD-архитектуры современных вычислителей – векторизация – стала актуальной задачей.

Одновременно велась разработка программ библиотеки УРС-ОФ для работы на графических ускорителях GPGPU [9, 10] (2012–2015 гг.). Результаты этой работы во многом определили структуры данных, программ и логику построения векторизованной библиотеки УРС-ОФ.

В докладе будут рассмотрены структуры данных, организация программ и вычислений в векторизованной версии библиотеки УРС-ОФ, позволяющей проводить параллельные вычисления теплофизических функций на общей памяти с использованием стандарта OpenMP.

На этапе расчета начальных данных прикладная программа вызывает функцию инициализации библиотеки УРС-ОФ, которая готовит служебную информацию, используемую на стадии регулярного счета в сервисных программах и расчетных модулях.

Традиционно расчет свойств веществ в прикладной программе производился в последовательном цикле по счетным ячейкам задачи вызовом программы-переключателя, которая обращалась к программе, реализующей физическую модель, заданную для каждого вещества. Предполагается, что данные о состоянии вещества для всех счетных ячеек, в которых надо рассчитать теплофизические свойства, располагаются в одномерных массивах.

На рис. 1 схематически изображен порядок взаимодействия прикладной программы с программами векторизованной версии библиотеки УРС-ОФ для расчета теплофизических функций. Стрелками на рис. 1 показан обмен данными с прикладной программой.

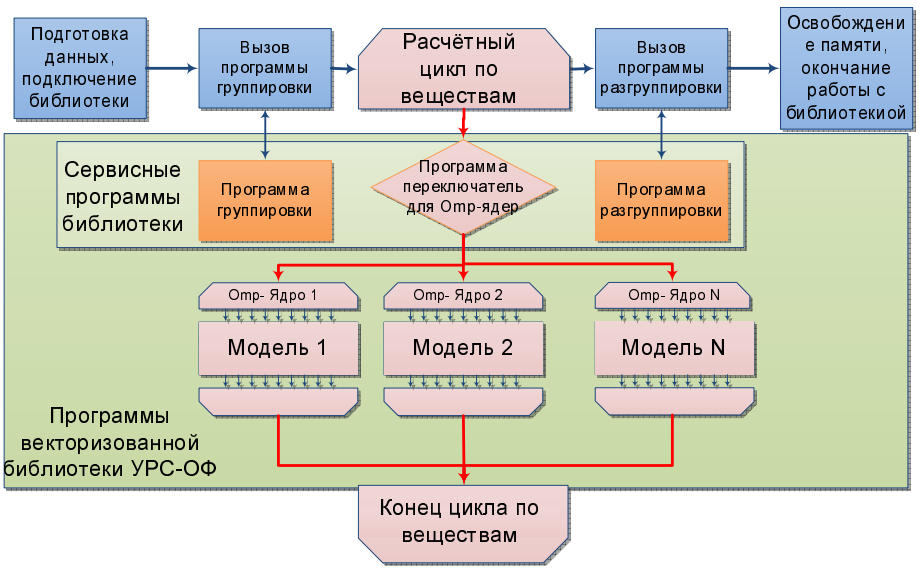


Рис. 1. Порядок взаимодействия прикладной программы с программами векторизованной версии библиотеки УРС-ОФ для расчета теплофизических функций

На первом этапе в прикладной программе проводится группировка исходных данных по веществам. Для этого из библиотеки вызывается программа группировки данных. Группировка выделяет данные для каждого вещества в отдельные непрерывные блоки памяти, их первоначальные положения запоминаются в служебном массиве для обратной расстановки после цикла вычислений. На рис. 2 схематично приведен процесс группировки.

На втором этапе происходят векторизованные параллельные вычисления теплофизических функций. Для их организации разработана специальная технология. Она включает в себя правила организации счетных модулей, данных и схему взаимодействия с прикладными программами, в которых используется библиотека УРС-ОФ. Приведем описание разработанной технологии.

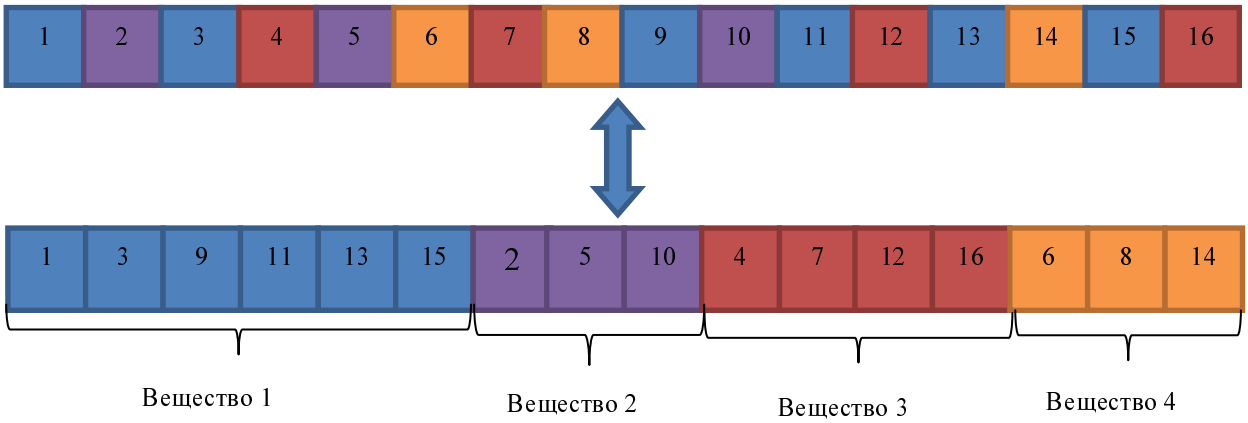


Рис. 2. Процесс группировки данных по веществам

Для обмена информацией с прикладными программами используются структуры данных, содержащие ссылки на массивы исходных данных и результатов. Используемые структуры аналогичны предложенным ранее [9, 10] при создании библиотеки для параллельных вычислений на GPGPU. На рис. 3 приведен фрагмент описания такой структуры, используемой при расчете термодинамических свойств.

```
! iS_OF – 4 или 8 байт, в зависимости от платформы
type urs_type_PA
   integer(kind = 4) L        ! номер заказа
   integer(kind = 4) num_prog  ! для определения имени программы в обработчике
   integer(kind = iS_OF) RO      ! адреса массивов плотностей
   integer(kind = iS_OF) T       ! адреса массивов температур …
   integer(kind = iS_OF) P       ! адреса массивов полного давления
   integer(kind = iS_OF) E       ! адреса массивов полной энергии
   integer(kind = iS_OF) DPDT  ! адреса массивов производных от P по T при ro-константа
   integer(kind = iS_OF) DEDT  ! адреса массивов производных от E по T при ro-константа
   integer(kind = iS_OF) DPDR  ! адреса массивов производных от P по ro при T-константа
   integer(kind = iS_OF) DEDR  ! адреса массивов производных от E по ro при T-константа
   integer(kind = iS_OF) CC      ! адреса массивов квадратов скорости звука
…
   real(kind=8) EPSA         ! константа абсолютной сходимости итераций в УРС
   real(kind=8) EPSO         ! константа относительной сходимости итераций в УРС
   real(kind=8) TF          ! “фоновая” температура

   integer(kind = iS_OF) ko    ! адреса массивов кодов ответа
 end type urs_type_PA
```

Рис. 3. Фрагмент описания структуры, используемой при расчете термодинамических свойств

В прикладной программе эти структуры заполняются адресами массивов с результатами и группированными исходными данными; организуется цикл по веществам, внутри которого происходит обращение к программе-переключателю. Программы-переключатели разработаны для универсальности обращения к расчетным модулям для конкретного вещества, реализующим определенную физическую модель. На рис. 4 представлена блок-схема программы-переключателя.

Omp-ядра обеспечивают разделение вычислений по потокам OpenMP для блоков памяти, соответствующих отдельному веществу. Они выполняют роли, аналогичные роли GPU-ядер, разработанных ранее при создании библиотеки для параллельных вычислений на GPGPU [9, 10].

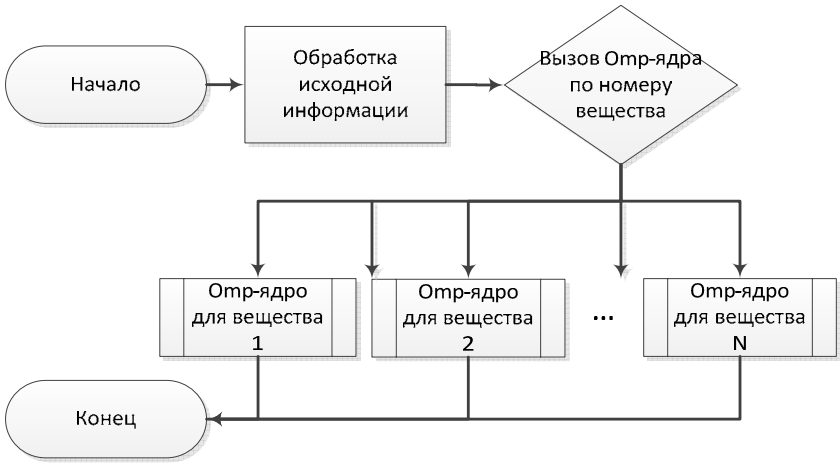


Рис. 4. Блок-схема программы-переключателя

Работа ведется с частью массивов, соответствующей веществу на текущем шаге (см. рис. 2 и 4). Рассмотрим процесс распараллеливания на общей памяти. На рис. 5 показана блок-схема omp-ядра.

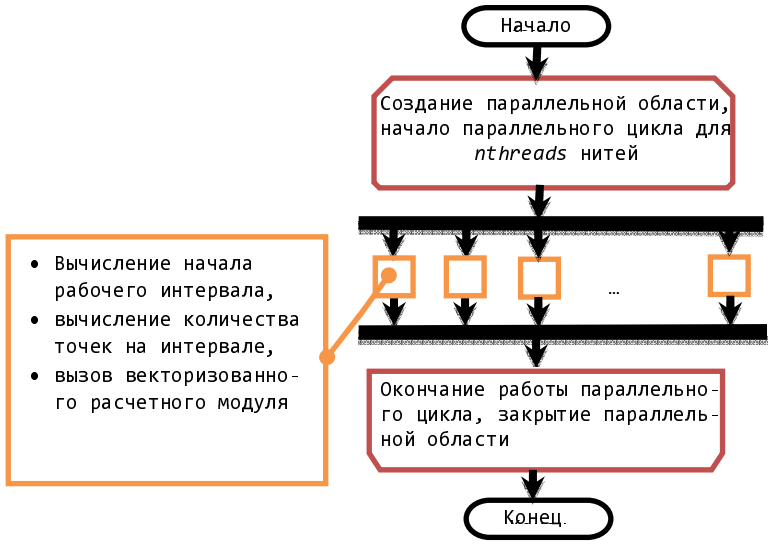


Рис. 5. Блок-схема omp-ядра

Пусть для i-го вещества есть N_i исходных точек. Для их вычисления организуется omp-параллельный цикл на nt нитей. Число omp-нитей определяется прикладной программой. Массивы данных разбиваются на интервалы по числу нитей. Для каждой нити вычисляются индексы начала расчетного интервала в общем массиве и число расчетных точек на нити. Нити с первой по предпоследнюю работают с N_i/nt точками, а последняя либо с N_i/nt, либо с меньшим остатком. После определения границ расчетного интервала вызывается векторизованный расчетный модуль теплофизических свойств.

Для того, чтобы автовекторизация, примененная компилятором к выражению, была эффективной, должно обеспечивается выполнение нескольких условий: адреса локальных массивов выровнены, данные расположены в памяти непрерывно, в векторизуемых циклах не должно быть зависимости по данным. Результаты работы компилятора по автовекторизации контролируются разработчиками программ по специальному отчету компилятора.

Для того, чтобы автовекторизация, примененная компилятором к выражению, была эффективной, должно обеспечивается выполнение нескольких условий: адреса локальных массивов выров-

нены, данные расположены в памяти непрерывно, в векторизуемых циклах не должно быть зависимости по данным. Результаты работы компилятора по автовекторизации контролируются разработчиками программ по специальному отчету компилятора.

Для автовекторизации средствами компилятора большинства вычислений в расчетных модулях, временные переменные объявляются как массивы фиксированного размера N_v, соответствующего ширине векторных регистров используемого вычислителя или кратностей. Вычисления производятся в цикле с шагом, соответствующим выбранному размеру. Внутри него в отдельных циклах организована передача данных из локальных массивов в массив прикладной программы и обратно. На рис. 6 показана блок-схема векторизованного модуля для вычисления теплофизических свойств.

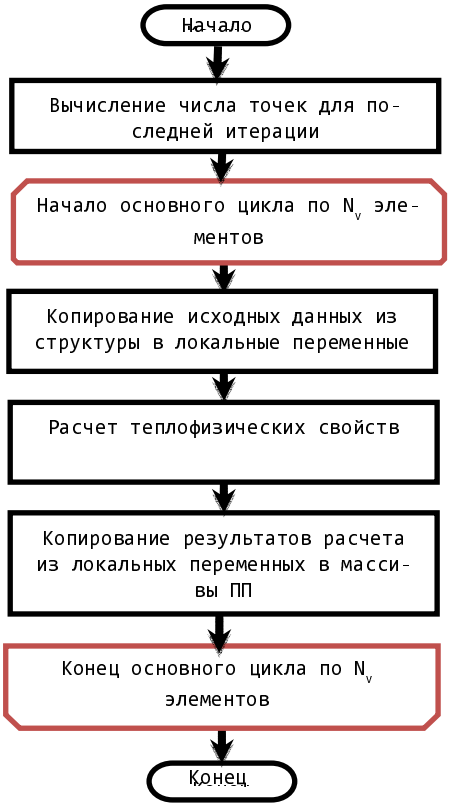


Рис. 6. Блок-схема векторизованного расчетного модуля

В результате анализа отчета компилятора по автовекторизации были определены конструкции языка и части алгоритмов неподдающихся автовекторизации – это циклы, содержащие логические конструкции, итерационные процедуры. Для них были разработаны специальные векторизуемые версии.

В качестве примера векторизации цикла с логическими конструкциями рассмотрим расчет по аппроксимации значений функции Дебая и ее первой производной [11].

В невекторизованном варианте при построении аппроксимации область определения функции была разбита на 5 участков. Результат разбиения приведен на рис. 7.

На участках 1,3–5 для вычисления функции Дебая и ее первой производной используются разные аналитические выражения, а на участке 2 используется сплайн-аппроксимация с равномерной расстановкой узлов. Для организации вычислений по участкам в скалярном режиме используются условные конструкции из пяти веток. На рис. 8 представлены блок-схема невекторизованной программы аппроксимации функции Дебая.

Первоначально при векторизации этой программы вещественные аргументы были заменены массивами размера N_v и организован цикл от 1 до N_v, внутри которого осуществлялась проверка попадания текущего элемента массива в один из участков и вычисление значения функции Дебая и ее первой производной. Анализ специального отчета компилятора показал, что такой цикл не автовекторизуется и не дает ускорения в расчете.

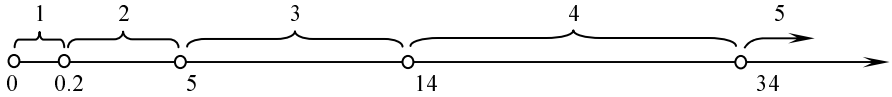


Рис. 7. Разбиение области определения функции Дебая

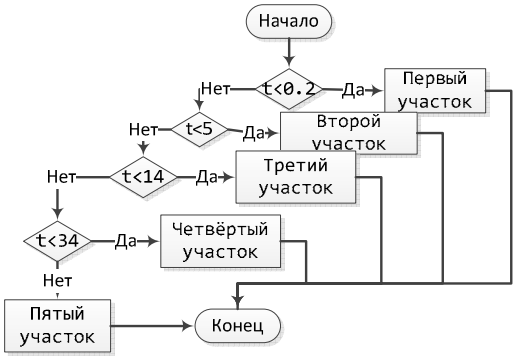


Рис. 8. Блок-схема невекторизованной программы аппроксимации функции Дебая

Для решения этой проблемы введен специальный массив-маска. Каждый элемент такого массива – это номер участка, к которому относится соответствующий элемент массива аргумента. В дальнейшем векторизованные вычисления производятся только для тех участков, номера которых были определены. При этом используется конструкция маскированного присваивания *where*, заменяющая условный оператор для массивов. Она осуществляет фильтрацию исходных данных и результатов вычислений, выбирая среди них элементы, удовлетворяющие заданному условию. Ниже на рис. 9 показан пример массива-аргумента, приведено выражение для вычисления массива-маски и приведен сам массив-маска, приведена схема работа векторизованной функции Дебая.

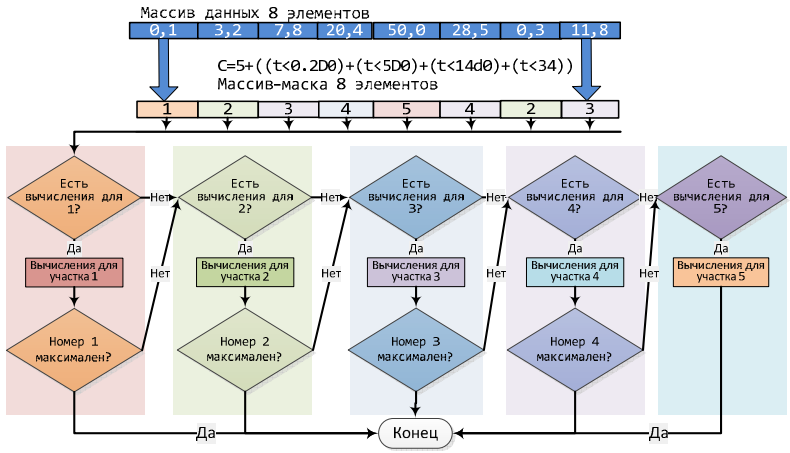


Рис. 9. Схема работы векторизованной функции Дебая[1]

Учет в программе особенностей решаемой задачи дает дополнительные ускорение и эффективность от ее векторизации. В случае функции Дебая при решении прикладных задач реализуются либо относительно низкие температуры, соответствующие большим значениям аргумента функции Дебая (отрезки 3–5 на рис. 7), либо очень высокие, соответствующие малым значениям аргумента (отрезки 1–2 на рис. 7). Поэтому дополнительные проверки на максимальное значение в массиве-маске позволяют избежать лишних вычислений и тем самым ускорить расчет (см. рис. 9).

К сложно векторизуемым конструкциям относятся и циклы, реализующие итерационные процедуры. На рис. 10 представлена блок-схема итерационного цикла.



Рис. 10. Блок-схема итерационного цикла

[1] Здесь принято, что выражения вида $(t < A)$ принимают значение –1, если условие выполнено, и 0, если условие не выполняется.

Нумерованными стрелками показаны два условия выхода из итерационного цикла: первый - если было достигнуто максимальное количество итераций; второй – если выполнилось условие сходимости итераций. В комментариях описаны действия в векторизованном итерационном цикле.

Как правило, число итераций, необходимое для определения искомой величины в разных областях фазовой диаграммы, различно между собой, т. е. для каждого элемента массива до достижения решения надо сделать различное число итерационных шагов. Вычисление специального массива-маски позволяет продолжать итерации только для тех элементов, решение для которых еще не найдено, тогда как элементы с найденным решением не изменяются. Это означает, что количество итераций всегда будет равно максимальному числу для заданного массива исходных данных. На рис. 11 приведен пример изменения значений массива-маски в итерационном цикле.

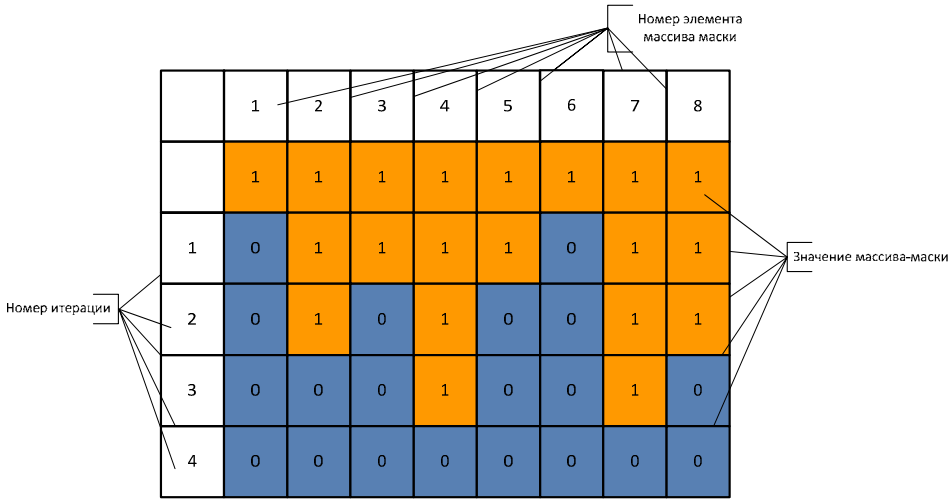


Рис. 11. Пример изменения значений элементов массива-маски в итерационном цикле

Общее число итераций на рис. 11 равно четырем. За одну итерацию найдены значения для первого и шестого элемента, за две итерации – для третьего и пятого, за три итерации – для второго и восьмого, за четыре – для четвертого и седьмого. Другими словами, если для одного элемента массива для нахождения решения достаточно двух итераций, а для другого – четыре, то для всего массива будет производиться четыре итерации.

Следует отметить, что ускорение от векторизации вычислений при такой организации итерационных процедур может быть получено, только если количество итераций для разных областей фазовой диаграммы отличаются между собой не слишком сильно. Для программ библиотеки УРС-ОФ это условие выполнено.

Помимо программ, реализующих алгоритмы расчета по моделям расчета теплофизических свойств веществ различной сложности, в библиотеке УРС-ОФ существует технология расчета теплофизических свойств под названием СуперУРС. Значение свойства рассчитывается как функция от свойств, вычисленных по нескольким моделям. То есть, отличия в организации вычислений векторизованных модулей СуперУРС от остальных модулей библиотеки УРС-ОФ, обусловлены необходимостью вызова различных векторизованных модулей в логических конструкциях и итерационных циклах.

Для векторизации СуперУРС применена та же технология, что и для обычных расчетных модулей. При вызове СуперУРС прикладная программа обращается к программе-переключателю, которая вызывает omp-ядро для СуперУРС.

Omp-ядро СуперУРС, аналогичное приведенному на рис. 5, вызывает векторизованный модуль СуперУРС.

В расчетных модулях вычисления внутри цикла одновременно производятся над группой данных размером N_v элементов (см. рис. 6), который равен или кратен ширине машинного регистра. Малое количество точек приводит к недозагруженности цикла и снижению ускорения. Поэтому для моделей СуперУРС необходимо подбирать размер группы N_{vSuper} таким образом, чтобы с обеспе-

чить кратность ширине машинного регистра внутри себя и внутри программы-модели и чтобы цикл внутри программы-модели был достаточно загружен для получения ускорения от векторизации.

Поскольку параллельная область уже создана в *omp*-ядре СуперУРС, для вызова векторизованных расчетных модулей из СуперУРС разработана программа-переключатель аналогичная приведенной на рис. 3, сразу вызывающая векторизованные модули. Для векторизации вычислений логических конструкций и итерационных циклов в СуперУРС, внутри которых вызываются векторизованные программы расчета термодинамических свойств веществ, также применялись массивы-маски.

Описанная в докладе технология позволила использовать для векторизации вычислений средства компилятора - автовекторизацию, адаптировать большое количество разных программ библиотеки УРС-ОФ к векторизованным вычислениям. В результате работы была создана библиотека УРС-ОФ для векторизованных параллельных вычислений теплофизических функций на общей памяти.

Все описанные в данном докладе векторизованные программы были разработаны на основе соответствующих скалярных аналогов. При разработке векторизованных программ проведено тестирование, обеспечившее надежность работы и повторяемость результатов расчетов: проверено соответствие расчетов по скалярной и векторизованной версиям программ, отсутствие ситуаций деления на ноль и переполнений, обеспечена безавостность работы программ при некорректной входной информации. При тестировании активно использовались возможности современных компиляторов.

По результатам тестирования при расчете на вычислителе с шириной вектора $N_v = 8$ получено сокращение времени расчета от двух до семи раз по сравнению со скалярной версией в зависимости от сложности моделей, реализованных программах. На рис. 12 приведен характерный график эффективности распараллеливания от числа omp-нитей.

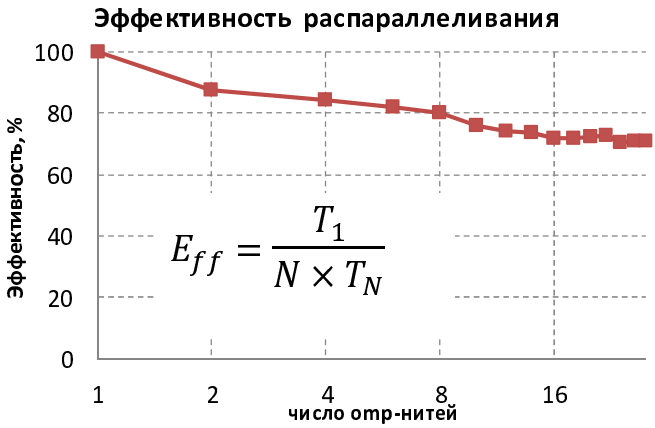


Рис. 12. Характерный график эффективности распараллеливания от числа omp-нитей: T_1 – время работы одной нити; N – число нитей; T_N – время работы N нитей

С увеличением числа нитей эффективность распараллеливания снижается до 70 %

Литература

1. Гордеев Д. Г., Голубкова Е. Ф., Гударенко Л. Ф., Куделькин В. Г., Сапронова О. В. Современное состояние пакета программ УРС-ОФ для расчета термодинамических и механических свойств веществ // XII Международная конференция «Супервычисления и математическое моделирование»: сб. науч. тр./под. ред. Р. М. Шагалиева. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2010. С. 115–118.

2. Забелина Т. В., Леонова Н. И., Сураева З. В., Тимонькина А. В. Организация программ уравнений состояния в одномерном комплексе // Вопросы атомной науки и техники. Сер. Методики и программы численного решения задач математической физики. 1979. Вып. 4(6). С. 13–15.

3. Софронов И. Д. Пакет программ для решения одномерных задач математической физики «Одномерный комплекс» // Вопросы атомной науки и техники. Сер. Методики и программы численного решения задач математической физики. 1978. Вып. 1(1). С. 3–6.

4. Воронов Е. Г., Каплунов М. И., Подвальный В. Г., Ребров С. В., Филатов В. И. Сервисная система Одномерного комплекса // Вопросы атомной науки и техники. Сер. Методики и программы численного решения задач математической физики. 1978. Вып. 2(2). С. 3–12.

5. Леонова Н. И., Софронов И. Д., Сураева З. В., Тимонькина А. В. Пакет программ расчета термодинамических функций уравнений состояния УРС-ЕС-ОК // Вопросы атомной науки и техники. Сер. Методики и программы численного решения задач математической физики. 1983. Вып. 2(13). С. 3–8.

6. Воронов Г. И., Каплунов М. И., Кленова В. И., Легоньков В. И., Леонова Н. И. Мурашкина В. А., Сапожников А. Т., Соколов В. П., Сураева З. В. Единая унифицированная система расчета термодинамических функций уравнений состояний // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1993. Вып. 1. С. 44–47.

7. Гударенко Л. Ф., Елисеев Г. М., Жеребцов В. А., Леонова Н. И., Сураева З. В., Тимонькина А. В. Обеспечение уравнениями состояния программ математического отделения ВНИИЭФ // Химическая физика. 1995. Т. 14, №1. С. 40–46.

8. Воронов Г. И., Горев И. В., Леонова Н. И., Осенкова Г. С., Сураева З. В. Программное обеспечение функционирования пакета УРС-ОФ // Вопросы атомной науки и техники. Сер. Методики и программы численного решения задач математической физики. 1999. Вып. 3. С. 56–58.

9. Гордеев Д. Г., Федоров А. А., Быков А. Н., Сизов Е. А., Куделькин В. Г., Жильникова Н. Н. Адаптация комплекса программ РАМЗЕС-КП для решения задач газовой динамики и теплопроводности на гибридных параллельных ЭВМ //XV Международная конференция «Супервычисления и математическое моделирование»: сб. науч. тр. / Под ред. Р. М. Шагалиева. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2015. С. 169–175.

10. Быков А. Н., Гордеев Д. Г., Куделькин В. Г., Сизов Е. А., Федоров А. А. Методика РАМЗЕС-КП на гибридных параллельных ЭВМ с графическими ускорителями // Вопросы атомной науки и техники. Сер.: Математическое моделирование физических процессов. 2017. Вып. 3. С. 70–76

11. Гордеев Д. Г., Ломайкин А. И. Аппроксимация функции Дебая $D_3(x)$ в области $0 \le x < \infty$ функцией класса C^2 // Вопросы атомной науки и техники. Сер.: Математическое моделирование физических процессов. 2008. Вып. 1. С. 42–50.

“URS-OF” LIBRARY FOR VECTORIZED PARALLEL COMPUTATIONS OF THERMOPHYSICAL FUNCTIONS USING SHARED MEMORY

D. G. Gordeev, N. N. Zhilnikova, D. N. Kidyamkina,V. G. Kudelkin, M. V. Kulikova, O. N. Shumilina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes structures of data, organization of programs and computations in vectorized versions of URS-OF package library that allow carrying out parallel computations of thermophysical functions using shared memory of OpenMP standard. A brief history of URS-OF development is provided, the interaction procedures of interaction between URS-OF with applied program is described, as well as techniques of computation parallelization. Some examples are provided to illustrate specific features of parallelization of symbolic structures and iterative cycles using arrays of masks and parallelization of programs of SuperURS type.

Key words: Library of URS-OF codes, parallelization on shared memory общей памяти, OpenMP standard, vector register width, auto-vectorization, symbolic structures, iterative cycles, parallelization efficiency.

УДК 004.4
DOI 10.53403/9785951505071_2022_125

ИСПОЛЬЗОВАНИЕ ИНТЕГРАТОРА МОДУЛЬНОЙ ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ ЛОГОС ДЛЯ КОНФИГУРИРОВАНИЯ ЗАДАЧ РАЗЛИЧНОГО ТИПА

Е. А. Губайдулина, А. А. Тюндина, О. Ю. Сурина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В состав модульной интеграционной платформы (ЛОГОС-МИП) входит графический интерфейс пользователя (Интегратор), обеспечивающий возможность конфигурирования и последующего решения задач, для чего, в общем случае, используются следующие основные этапы:

– конфигурирование одной или нескольких одиночных задач;

– настройка подключения к задачам дополнительных расчетных модулей;

– настройка связанного взаимодействия задач для организации мультидисциплинарного расчета;

– подготовка расчетных схем, включая настройку очередности этапов выполнения, конфигурирование параметров схемы;

– выполнение расчетной схемы;

– получение и анализ результатов.

В докладе представлены обобщенные алгоритмы конфигурирования задач следующих типов:

– связанная мультидисциплинарная задача;

– задача параметрических исследований;

– задача оптимизации.

Ключевые слова: Интегратор, ЛОГОС-МИП, конфигурирование задач, мультидисциплинарная задача, параметрические исследования, оптимизационная задача.

Общие принципы конфигурирования задач

Конфигурирование задачи в Интеграторе начинается с выбора расчетного модуля, который будет рассчитывать данную задачу. Расчетный модуль должен быть интегрирован в ЛОГОС-МИП – в Интегратор передается файл спецификации, в котором содержится информация о расчетном модуле (используемые препроцессор, постпроцессор, САПР и др.).

Процесс конфигурирования задачи может включать в себя как весь цикл подготовки задачи (от создания конструкторской и расчетной моделей до создания самой задачи на основе этих моделей), так и импорт ранее подготовленных моделей и задач.

Внешний вид дерева проектов Интегратора с подготовленной задачей в случае полного цикла подготовки задачи представлен на рис. 1, в случае импорта готовой задачи в проект – на рис. 2.

К созданной задаче есть возможность подключить дополнительные расчетные модули, используемые во время расчета. Так, к задаче можно подключить пользовательские функции, легковесные модули, программные адаптеры и т. д.

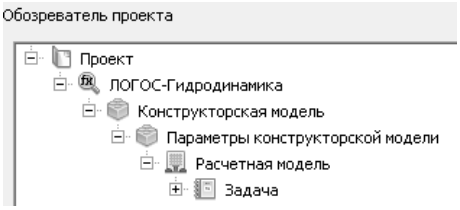


Рис. 1. Дерево объектов Интегратора с подготовленной одиночной задачей на основе конструкторской и расчетной моделей

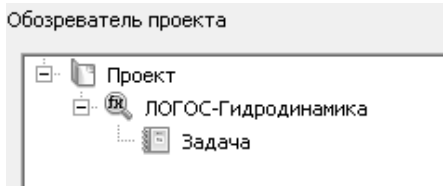


Рис. 2. Дерево объектов Интегратора с импортированной одиночной задачей

Для запуска задачи на расчет через Интегратор, необходимо сконфигурировать схему (сценарий) расчета. Сценарий расчета представляет собой последовательность выполнения необходимых для проведения расчета программных компонент, таких как САПР, препроцессор, расчетный модуль и другие. Вызов и логика работы с внешними компонентами в ЛОГОС-МИП обеспечивается с помощью Обработчика сценария и его внешних программных модулей, реализованных на языке программирования Python [1].

Интегратор имеет в своем составе схемный редактор, в котором каждый модуль Обработчика сценария имеет свой графический эквивалент в виде блока схемного редактора. Схемный редактор позволяет пользователю настраивать схему расчета задачи путем размещения отдельных блоков на сцене схемного редактора и соединения их друг с другом для настройки последовательности передачи управления, создавая тем самым схему проведения расчета. Пользователь может сам создать схему расчета в соответствии со спецификой проведения расчета определенной задачи, или воспользоваться готовыми схемами схемного редактора. Типовая схема локального расчета одиночной задачи представлена на рис. 3.

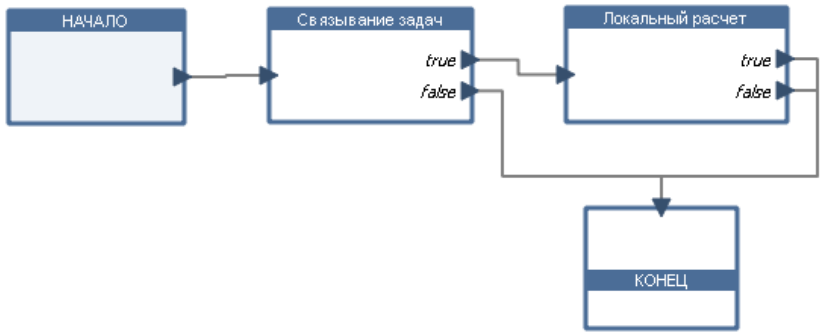


Рис. 3. Схема локального расчета одиночной задачи

Блок схемы расчета может иметь настраиваемые параметры, которые задаются пользователем в процессе конфигурирования схемы расчета. Пример конфигурации блока «Локальный расчет» представлен на рис. 4.

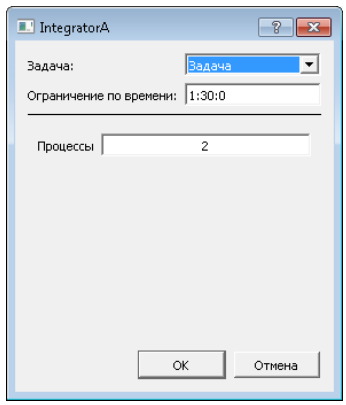


Рис. 4. Настраиваемые параметры блока «Локальный расчет»

После завершения конфигурирования схемы расчета, она может быть запущена. Интегратор отображает ход расчета, как показано на рис. 5, и статус проведения расчета в виде общей таблицы, как показано на рис. 6.

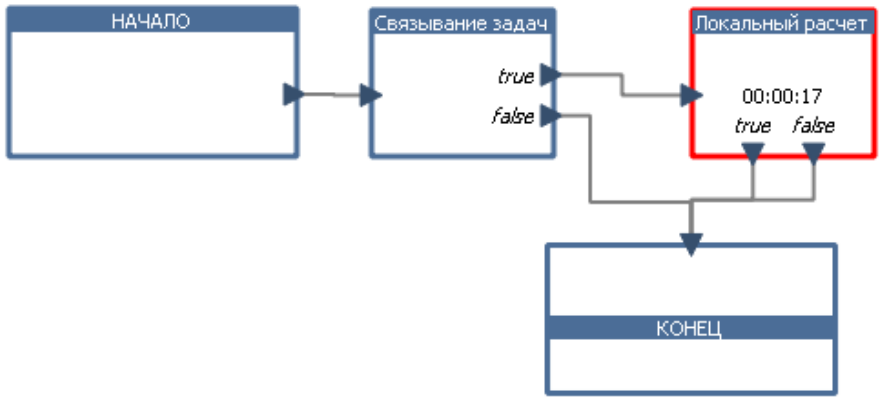


Рис. 5. Отображение хода расчета

Обозреватель заданий | Результаты

Задача	Статус	Запущена	Продолжительность
Схема расчета	Завершена	18.02.2020 16:33:04	00:00:28
Схема расчета_1	Выполняется	18.02.2020 16:40:44	00:00:04
Схема расчета_2	Аварийно завершена	18.02.2020 16:37:16	00:00:05
Схема расчета_3	Не запущена		

Рис. 6. Отображение статуса расчета

Интегратор предоставляет возможность просматривать табличные и текстовые результирующие файлы, строить графики, а также просматривать результаты расчета с помощью сторонних программных средств постобработки.

Подготовка задачи связанного расчета

Связанный расчет представляет собой расчет мультидисциплинарной задачи, в которой два (и более) расчетных модуля обмениваются между собой данными во время расчета. Обмен между расчетными модулями, в рамках модульной интеграционной платформы, проводится транспортным уровнем ЛОГОС−МИП, а настройка конфигурации обмена производится с помощью Интегратора ЛОГОС−МИП.

Подготовка задач связанного расчета выполняется в Интеграторе, путем объединения одиночных расчетных задач.

Подготовка связанного расчета состоит из следующих этапов:

– настройка общей схемы связанного расчета;

– связывание доступных обменных интерфейсов;

– конфигурирование транспортного уровня ЛОГОС-МИП.

Настройка общей схемы связанного расчета

Пользовательский интерфейс Интегратора при настройке общей схемы связанного расчета изображен на рис. 7.

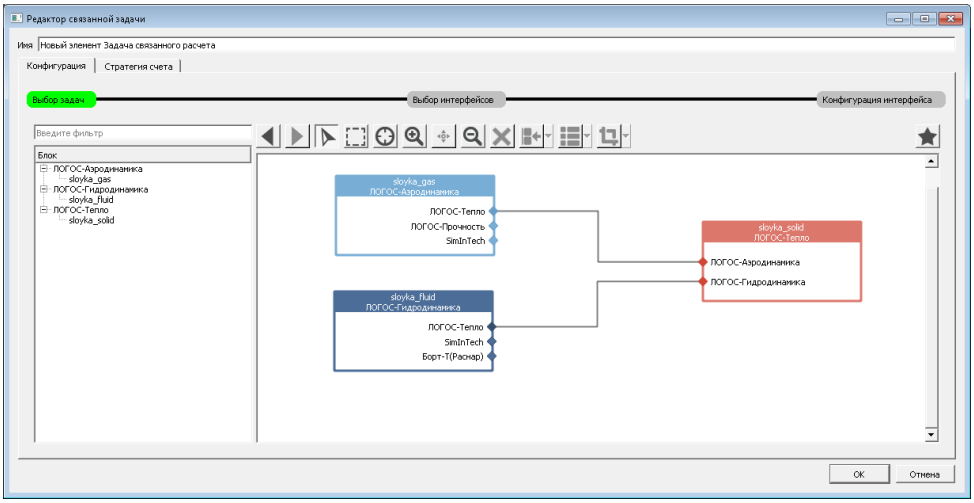


Рис. 7. Настройка общей схемы связанного расчета

В левой части окна расположен список всех подготовленных ранее в Интеграторе одиночных задач расчетных модулей. В правой части расположен схемный редактор, предназначенный для графического конфигурирования связей между задачами. Добавление одиночных задач в редактор осуществляется по технологии «drag-and-drop». Указание связей осуществляется графически в интерактивном режиме. Интегратор гарантирует добавление на схему только допустимых связей, т. е. позволяет связывать задачи только тех расчетных модулей, которые были интегрированы с возможностью проведения связанного расчета. Информация о возможных связях поступает в Интегратор из файлов спецификации. Переход к интерфейсу настройки каждой связи осуществляется по двойному нажатию левой кнопки мыши на линии требуемой связи.

Интегратор обеспечивает настройку связанного расчета мультидисциплинарной задачи для трех и более расчетных модулей.

Связывание доступных обменных интерфейсов

Настройка связей между расчетными модулями, участвующими в связанном расчете мультидисциплинарной задачи, осуществляется отдельно для каждой пары модулей. Пользовательский интерфейс Интегратора при связывании доступных обменных интерфейсов двух задач расчетных модулей представлен на рис. 8.

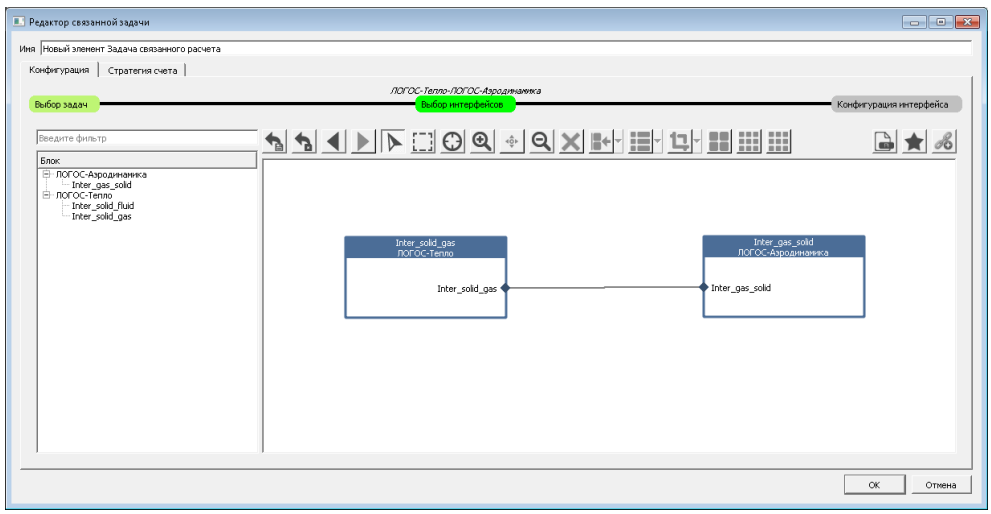


Рис. 8. Связывание доступных обменных интерфейсов для пары задач

Центральная часть окна содержит схемный редактор, в котором представлен перечень обменных интерфейсов, доступных для каждой из двух связываемых задач. Связывание требуемых интерфейсов осуществляется графически в интерактивном режиме. Дополнительно предусмотрена возможность автоматического связывания интерфейсов с одинаковыми именами, данное действие инициируется нажатием кнопки «Соединить автоматически». Наличие такой возможности видится удобным при необходимости связывания большого количества интерфейсов.

Типовым режимом использования Интегратора для настройки прохождения и обработки данных по каждому интерфейсу является использование заранее подготовленных шаблонов с описанной конфигурацией настройки, при этом требуемый шаблон может быть выбран из перечня доступных в выпадающем списке «Для всех интерфейсов использовать шаблон». В случае необходимости, настройка интерфейса может быть выполнена в ручном режиме. Переход к интерфейсу настройки осуществляется по двойному нажатию левой кнопки мыши на линии требуемой связи.

Настройка прохождения данных через интерфейс связи

Пользовательский интерфейс Интегратора при настройке прохождения данных через интерфейс связи представлен на рис. 9.

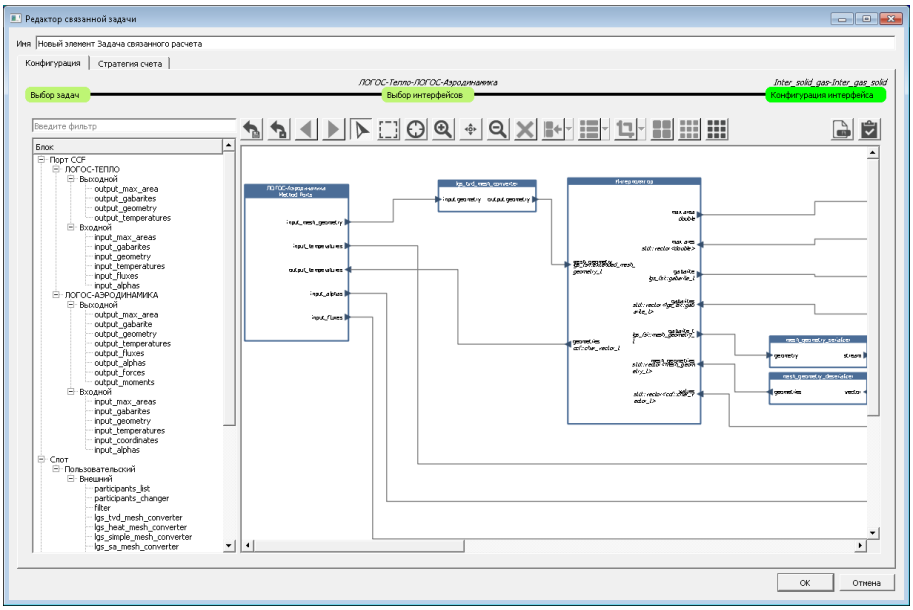


Рис. 9. Настройка прохождения данных через интерфейс связи

В центральной части окна расположен схемный редактор, предназначенный для графического конфигурирования схемы прохождения данных. Интегратор автоматически располагает на схеме доступные входные и выходные порты данных для каждой их задач, участвующей в настройке. Информация о доступных портах и типах их данных поступает в Интегратор из файлов спецификации расчетных модулей.

В левой части окна расположен список всех доступных обрабатывающих функций, которые могут быть использованы при составлении схемы. Добавление обрабатывающих функций в редактор осуществляется по технологии «drag-and-drop». Указание связей между элементами схемы осуществляется графически в интерактивном режиме.

Интегратор гарантирует добавление на схему только допустимых связей, т. е. могут быть соединены только те точки, у которых:

– полностью совпадает тип данных;

– является совместимым направление передачи данных (т. е. вход соединяется с выходом и наоборот).

При необходимости, в схемный редактор может быть загружен подготовленный ранее шаблон, подходящий для связи указанных расчетных модулей.

Подготовка задачи параметрических исследований

Задача параметрических исследований представляет собой цикл расчетов с перебором входных параметров и последующим формированием расчетной модели на основе полученных данных

В Интеграторе для настройки задачи параметрических исследований используется схемный редактор, вызываемый через команду контекстного меню «Свойства» объекта «Схема расчета». Настройка схемы параметрических исследований выполняется в 2 этапа:

– настройка общей схемы параметрических исследований (настроенная схема изображена на рис. 10);

– настройка единичного шага расчета модельной задачи, как показано на рис. 11.

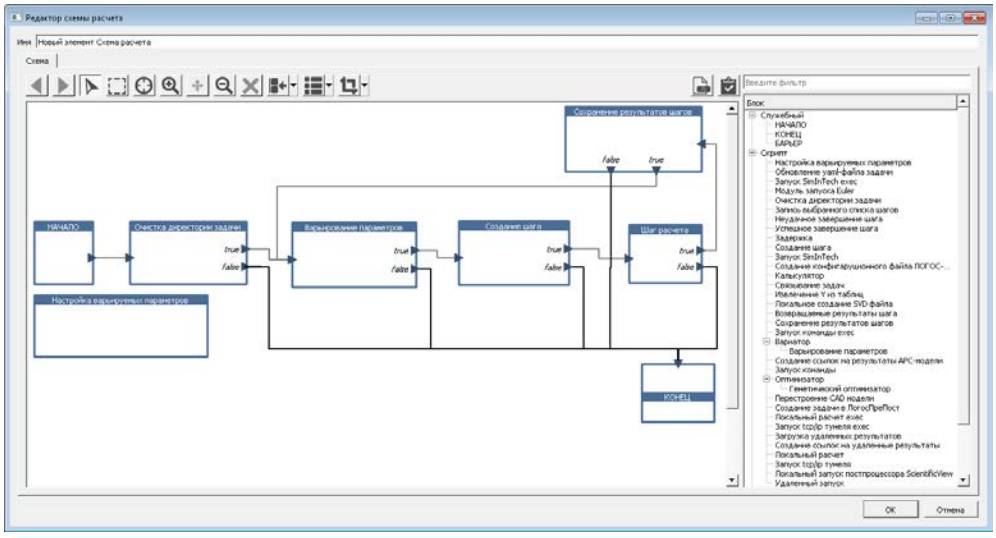


Рис. 10. Настройка схемы параметрических исследований

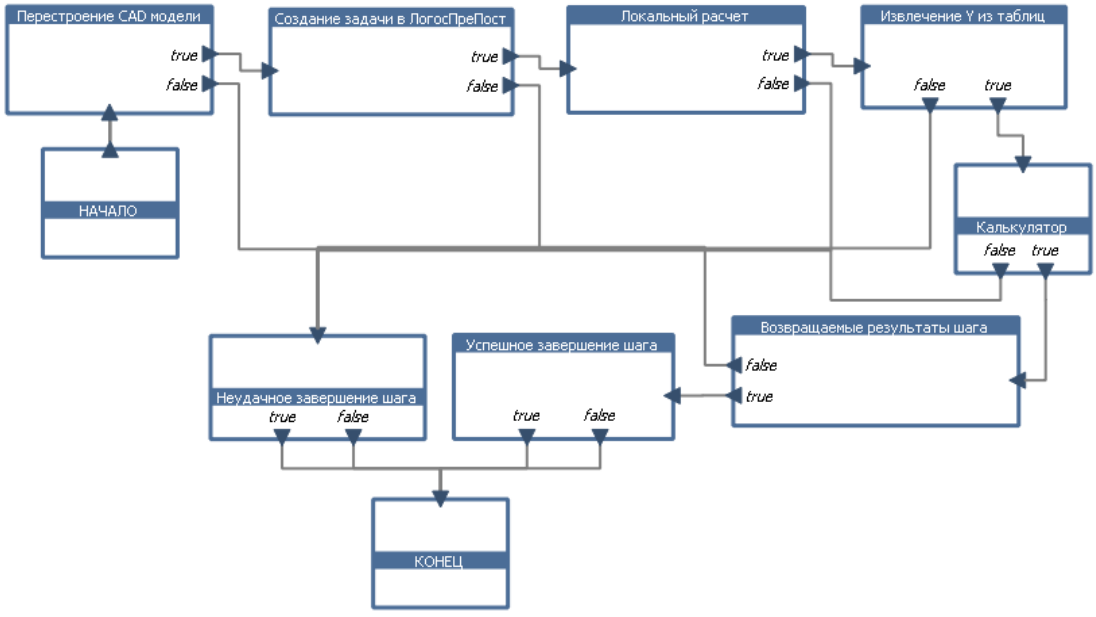


Рис. 11. Настройка единичного шага расчета модельной задачи

Настройка общей схемы параметрических исследований заключается в размещении на схеме требуемых шагов и указания очередности их выполнения. Для выполнения редактирования схемы используется панель команд редактора и контекстное меню блоков и связей.

Внешний вид редактора сценария изображен на рис. 10. В центральной части окна расположена общая схема выполнения сценария, на которой каждый прямоугольник соответствует одному шагу сценария, а стрелки определяют очередность выполнения шагов. В качестве шага сценария выступает подготовленный программный модуль обработчика сценария, выполняемый с помощью языка Python [1], либо другой («вложенный») сценарий.

Переход к редактированию вложенного сценария осуществляется по двойному нажатию левой клавиши мыши на прямоугольнике вложенного сценария. На схеме каждого сценария также должны быть представлены два служебных шага – «Начало» и «Конец», определяющие начальную и конечную точку выполнения сценария, соответственно.

В правой части окна расположен перечень доступных для выполнения в скрипте модулей, из которых формируется сценарий. Для размещения модулей в окне редактора используется механизм drag-and-drop.

Альтернативным способом подготовки сценария является выбор сценария из перечня заранее подготовленных шаблонов.

Для выполнения задачи параметрических исследований должны быть настроены следующие модули:

– модуль задания параметров варьирования;

– модуль варьирования параметров;

– модуль задания извлекаемых величин;

– модуль задания сохраняемых величин расчета.

В модуле задания параметров варьирования отображаются все доступные параметры варьирования для данной задачи. В данном модуле задаются значения перебираемых параметров. Задания параметров варьирования осуществляется в нескольких режимах - с использованием диапазона значений и шага, и в виде списка величин. Пример настроек модуля задания параметров варьирования представлен на рис. 12.

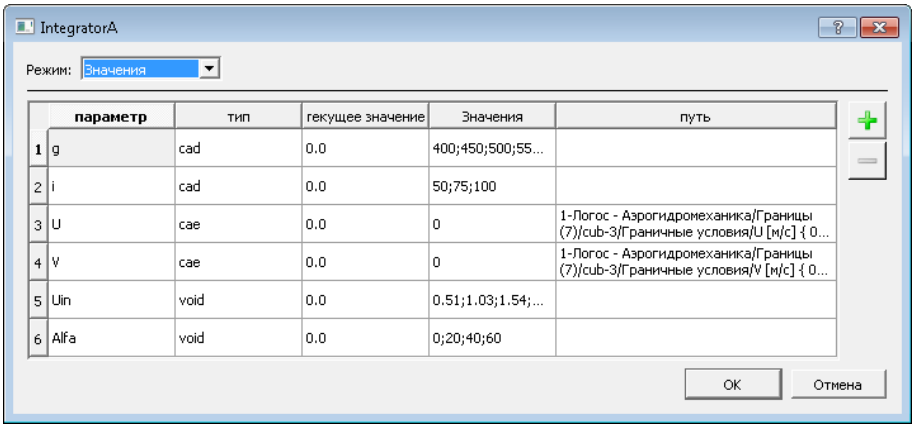


Рис. 12. Внешний вид редактора настройки варьируемых параметров разного типа для режима «Значения»

В модуле варьирования параметров имеется возможность выбрать метод формирования набора параметров при наличии нескольких списков варьируемых параметров. Поддерживается формирование комбинаций «каждый с каждым» и комбинаций из соответствующих элементов входных последовательностей. Например, для двух входных комбинаций (A1, A2) и (B1, B2) в первом случае получаются комбинации (A1, B1), (A1, B2), (A2, B1) и (A2, B2), а во втором (A1, B1) и (A2, B2).

В модуле задания извлекаемых величин создаются новые переменные, значения для которых извлекаются из результирующих файлов табличного типа. Внешний вид редактора модуля задания извлекаемых величин представлен на рис. 13.

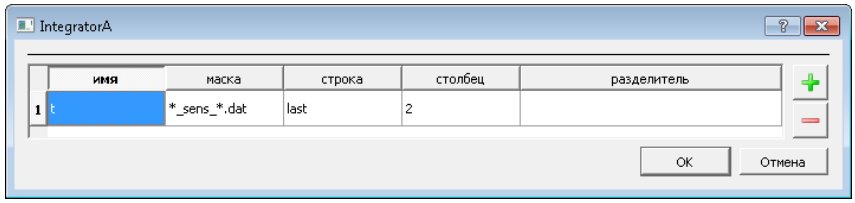


Рис. 13. Внешний вид редактора настройки извлекаемых величин

В модуле задания сохраняемых величин расчета задаются имена переменных, значения которых будут аккумулированы в единый сводный файл результатов параметрических исследований. На данном шаге используются переменные заданные на предыдущих шагах. Внешний вид редактора модуля задания сохраняемых величин представлен на рис. 14.

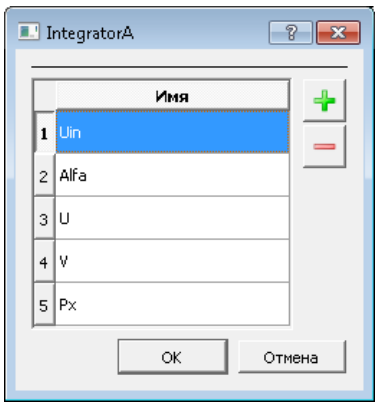


Рис. 14. Внешний вид редакторазадания сохраняемых величин

Также имеется возможность задания дополнительных настроек в модулях:

– задание времени перестройки расчетной модели;
– декомпозиция задачи;
– задание формул пересчета параметров;
– время счета задачи и т. п.

Подготовка задачи оптимизации

Задача оптимизации в рамках ЛОГОС-МИП представляет собой цикл расчетов с перебором входных параметров и последующим формированием расчетной модели на основе полученных данных с целью приведения значений целевых параметров к определенным границам.

Подготовка задачи оптимизации аналогична подготовке задачи параметрических исследований. Настройка общей схемы задачи оптимизации отличается от общей схемы параметрических исследований тем, что вместо модуля варьирования параметров используется модуль оптимизатора. Общая схема задачи оптимизации представлена на рис. 15. Настройка единичного расчета модельной задачи совпадает с настройками единичного расчета параметрических исследований и представлена на рис. 11.

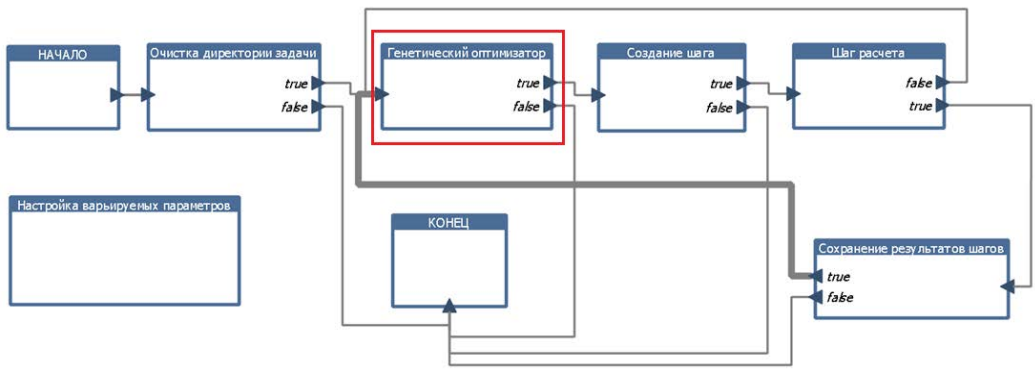


Рис. 15. Настройка схемы задачи оптимизации

Для выполнения задачи оптимизации должны быть настроены следующие модули:

– модуль задания параметров оптимизатора;
– модуль задания параметров варьирования;
– модуль задания извлекаемых величин;
– модуль задания формул пересчета;

– модуль задания сохраняемых величин шага расчета;

– модуль задания сохраняемых величин.

В модуле задания параметров оптимизатора задаются ограничения оптимизатора – количество функций, сводящихся к минимуму (nf), количество ограничивающих неравенств(nh), максимальное количество вариантов перебора (budget) и максимальное количество попыток генерирования наборов изменяемых параметров в случае получения неудовлетворительных результатов (limit). Внешний вид редактора модуля задания параметров оптимизатора представлен на рис. 16.

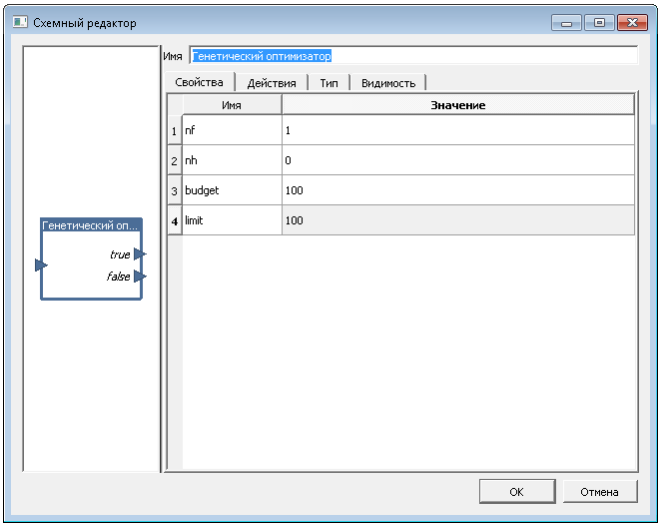


Рис. 16. Внешний вид задания параметров оптимизатора

В модуле задания формул пересчета задаются формулы зависимостей от параметров, изменяемых в процессе расчета. Внешний вид редактора модуля задания формул пересчета представлен на рис. 17.

В модуле сохраняемых величин шага расчета помимо переменных, значения которых будут аккумулированы в единый сводный файл результатов задачи оптимизации, задаются переменные, значения которых в дальнейшем будут передаваться в оптимизатор для их обработки. Такие переменные задаются с помощью указания типа «objective». Внешний вид редактора модуля сохраняемых величин расчета представлен на рис. 18.

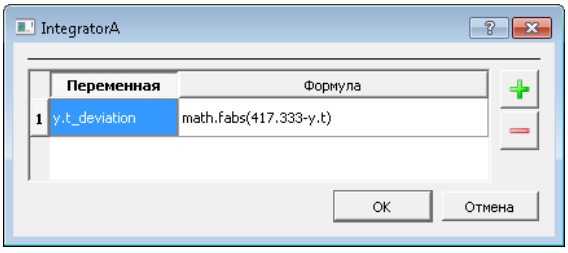


Рис. 17. Внешний вид задания формул пересчета

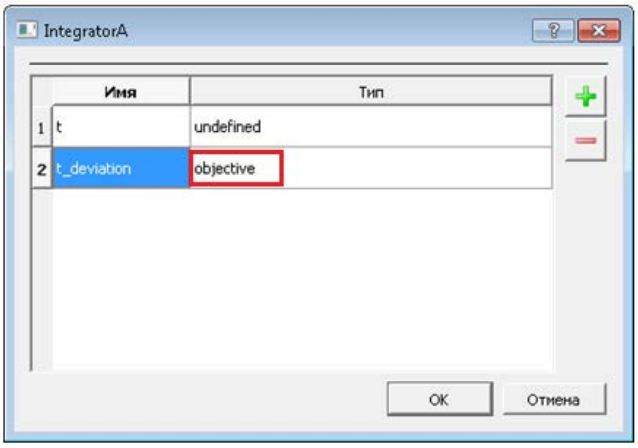


Рис. 18. Внешний вид сохранения величин шага расчета

Модули задания параметров варьирования, извлекаемых величин и задания сохраняемых величин настраиваются таким же образом, как и при параметрических исследованиях.

Заключение

В докладе были рассмотрены основные принципы конфигурирования задач в Интеграторе ЛОГОС-МИП. Подробно были рассмотрены принципы подготовки задач связанного мультидисциплинарного расчета, задач параметрических исследований и оптимизационных задач. В рамках конфигурирования задач графический интерфейс позволяет:

– применять для проведения расчетов сторонние продукты численного моделирования;
– использовать внешние САПР и программы визуализации;
– выполнять решение задач на различных платформах;
– подключать дополнительные расчетные модули для расчета задачи;
– выполнять решение мультидисциплинарной задачи;
– выполнять решение задач параметрических и оптимизационных исследований.

Литература

1.Язык программирования Python [Электронный ресурс]: Python Software Foundation. – Virginia, USA, [2001-2017] – Режим доступа: https://docs.python.org/3/index.html.

USE OF INTEGRATOR OF THE LOGOS MODULAR INTEGRATION PLATFORM TO CONFIGURE TASKS OF VARIOUS TYPES

E. A. Gubaydulina, A. A. Tyundina, O. Yu. Surina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The modular integration platform LOGOS-MIP includes the graphics user interface (Integrator) that makes it possible to configure and further fulfil tasks and this process consists, in general, of the following main stages:

- configuring one, or several single tasks;
- setting the connection of additional computational modules to fulfil these tasks;
- setting the coupled interaction of tasks to arrange the multiphysics computations;
- preparing computational schemes with setting the order of execution stages, configuring the scheme parameters;
- executing the computational scheme;
- obtaining and analyzing results.

The paper presents the generalized algorithms of configuring tasks of the following types:

- a coupled multiphysics task;
- a parametric study task;
- an optimization task.

Key words: Integrator, LOGOS-MIP, configuring a task, multiphysics task, parametric study, optimization task.

УДК 519.683.8
DOI 10.53403/9785951505071_2022_135

ЭФФЕКТИВНОЕ МЕТАПРОГРАММИРОВАНИЕ АЛГОРИТМОВ МЕТОДА КОНЕЧНЫХ ЭЛЕМЕНТОВ СРЕДСТВАМИ ЯЗЫКА C++17

А. М. Гурин, А. Н. Байкин

Институт гидродинамики им. М. А. Лаврентьева СО РАН, Новосибирск

Работа посвящена применению функционального и метапрограмирования для разработки конечно-элементного решателя. Возможности современного C++ позволяют реализовать метод конечных элементов непосредственно через работу с функциями при помощи метафункций. Такой метод обобщает сборку локальной матрицы из слабой формы уравнения. Подход протестирован на решении уравнения Пуассона и выполнено сравнение с известными открытыми кодами.

Ключевые слова: метод конечных элементов, функциональное программирование, метапрограммирование.

Введение

Метод конечных элементов (МКЭ) позволяет дискретизовывать системы уравнений в частных производных на расчетных сетках, состоящих из элементов - малых объемов, для которых записан некоторый интерполирующий полином. Для уравнений записанных в слабой постановке возможно эффективно автоматизировать построение решателя, так как алгоритм перехода от уравнений в частных производных к дискретизации в виде системы линейных алгебраических уравнений (СЛАУ) достаточно прямолинеен. Тем не менее, реализация алгоритма в виде компьютерной программы представляет значительную сложность, так как математический язык плохо согласуется с языком программирования. Существуют известные программные решения реализующие МКЭ в обобщенном виде – deal.ii, FEniCS, FreeFEM++ [1], Comsol и множество других. Перспективный метод разработки таких обобщенных кодов это метапрограммирование. Оно позволяет избежать падения производительности, связанного с динамическим полиморфизмом, так как некоторая часть кода компилируется специально под решаемую задачу. В таком подходе реализованы deal.ii и FeniCS [2]. В лаборатории авторов был реализован МКЭ код FEMEngine с использованием шаблонного метапрограммирования на C++17. Такой подход позволил реализовать МКЭ напрямую в терминах функций и операций над ними.

Алгоритм МКЭ

МКЭ легко проиллюстрировать на решении уравнения Лапласа,

$$\Delta T = 0 \tag{1}$$

Для применения метода уравнение записывается в слабой постановке,

$$\int_{\Omega} \nabla T \cdot \nabla \psi d\Omega = 0 \tag{2}$$

Неизвестная функция $T(X)$ выражается в виде интерполяционного полинома,

$$T(X) = \sum_{i=1}^{N} T_i \phi_i \tag{3}$$

Теперь подставив $T(X)$ в слабую постановку и выбрав интерполяционные функции ϕ_i в качестве тестовых ψ можно получить систему линейных уравнений

$$\sum_{i=1}^{N} T_i \int \nabla\phi_i \cdot \nabla\phi_j d\Omega - 0, j - 1..N \tag{4}$$

$$\sum_{i=1}^{N} T_i M_{ij} = 0, j = 1..N \tag{5}$$

$$M_{ij} = \int \nabla\phi_i \cdot \nabla\phi_j d\Omega \tag{6}$$

Решение системы (5) является решением исходного уравнения (1). Алгоритм решения состоит из нескольких блоков. Сначала требуется выбрать или сконструировать функцию для сборки матрицы (6), аппроксимирующей слабую постановку (2). Для вычисления элементов матрицы требуется интегрировать интерполяционные функции на элементе, который представляет собой некоторую геометрическую фигуру. Для удобства интегрирование проводится численной квадратурой на фиксированном, геометрически простом (каноническом) элементе, к которому производится преобразование координат с реального элемента.

Умножение функций

Лямбда функции вместе с контейнером std::tuple могут быть использованы для хранения интерполяционных функций элемента как показано на рис. 1. Контейнер std::tuple допускает оптимизацию подстановкой. Эта оптимизация является ключевой для всего алгоритма, так как позволяет полностью избавиться от большого дерева вызовов метафункций. Количество и типы аргументов лямбда функции в современном C++ могут быть определены на стадии компиляции.

```
constexpr auto phi1 = [](
  std::tuple<double, double> r
  ){
    auto[xi, eta] = r;
    return -eta - xi + 1.0;
  }
...
std::tuple phi{phi1, phi2, phi3};
```

Рис. 1. Интерполяционные функции

Благодаря этому можно реализовать метафункцию показанную на рис. 2. Эта функция принимает две функции f1 и f2 с одинаковыми аргументами. Класс свойств функции function_traits узнает аргументы этих функций и, с использованием этой информации, конструируется третья функция, являющаяся произведением первых двух. Аналогичным способом можно реализовать метафункции для различных операций с двумя аргументами и даже общую функцию, которая принимает две функции и операцию над ними.

```
template<class F1,
         class F2,
         class ... Args>
auto multiply( F1 f1,
               F2 f2,
               std::tuple<Args...> ) {
  return [=]( Args... args ){
      return f1(args...) * f2(args...); };
}

template<class F1,
         class F2>
auto multiply( F1 f1, F2 f2) {
  return multiply(
    f1,
    f2,
    typename function_traits
    <decltype(&F1::operator())>::args{} );
}
```

Рис. 2. Умножение функций

Арифметические операции над функциями лежат в основе алгоритма. Следующая метафункция, показанная на рис. 3, это декартово произведение наборов функций, которое требуется для вычисления локальной матрицы (6). Оператор многоточие, разворачивающий блоки типов, может быть применен не только непосредственно сразу после блока, но и в других местах, имеющих семантический смысл. Например, многоточие в аргументах функций f1 и f2 на рис. 2 разворачивает аргументы, с которыми будет вызвана функция. Многоточие, примененное снаружи вызова функции сгенерирует множество вызовов функции с разными аргументами из блока. Такое использование многоточия показано на рис. 3.

```
template<class ...Args1,
         class ...Args2,
         size_t ...Is>
auto tensorProd(std::tuple<Args1...> t1,
                std::tuple<Args2...> t2,
                std::index_sequence<Is...>)
{
  return std::make_tuple(
    tupleBinaryOp(
      t1,
      std::get<Is>(t2),
      std::multiplies{}
    ) ... );
}

template<class ...Args1,
         class ...Args2>
auto tensorProd(std::tuple<Args1...> t1,
                std::tuple<Args2...> t2)
{
  return tensorProd(
    t1,
    t2,
    std::make_index_sequence
      <sizeof...(Args2)>{} );
}
```

Рис. 3. Декартово произведение функций

Интегрирование функции на элементе

Интегралы вида (6) обычно вычисляются при помощи квадратуры на каноническом элементе и преобразования координат. Интеграл на произвольном элементе выглядит следующим образом

$$\int_{\Omega_k} f\, d\Omega - \sum_{i=1}^{M} f(\vec{r_i}) \cdot w_i |J_k|, \tag{7}$$

где $\vec{r_i}$, w_i — это узлы и веса квадратуры и $|J_k|$ — Якобиан преобразования координат от канонического элемента к реальному элементу Ω_k в вычислительной области. Интегрирование, так же как умножение и декартово произведение, выполняется в функциональном стиле. Метафункция интегрирования принимает функцию и квадратуру и возвращает функцию, которая проинтегрирована неявно. Такая функция будет возвращать значение интеграла, принимая Якобиан. Также применяется метод частичного выполнения функции. В процессе интегрирования выполняется подстановка аргументов-координат и добавляется аргумент-матрица Якоби. В случае интегрирования производных функции, преобразование координат выглядит следующим образом

$$\frac{\partial \phi_i}{\partial x} = \frac{\partial \phi_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \phi_i}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \phi_i}{\partial \zeta}\frac{\partial \zeta}{\partial x}, \tag{8}$$

где ξ, η, ζ — координаты в системе канонического элемента. Такое преобразование координат также выполняется в функциональном стиле с изменением сигнатуры функции.

Сборка матрицы

Вышеописанные метафункции используются для создания функции, генерирующей локальную матрицу. Например, для вычисления локальной матрицы из выражения

$$\int_{\Omega_k} \phi_i \phi_j d\Omega, i = 1, \dots, N, j = 1, \dots, N, \tag{9}$$

на треугольном элементе первого порядка выполняются следующие операции. Интерполяция на элементе задана тремя функциями,

$$\phi = |\phi_1, \phi_2, \phi_3|. \tag{10}$$

Выполняется декартово произведение вектора ϕ состоящего из функций самого на себя. Затем применяется преобразование координат и интегрирование. Полученная функция используется в цикле по элементам для вычисления локальных матриц и сборке их в глобальную.

Эффективность метода

Важная особенность примененной методологии это обобщение МКЭ. Одна и та же реализация вышеописанных функций используется для 1D, 2D, 3D элементов любого порядка. Такой подход менее подвержен ошибкам, так как большая их часть обнаруживается на стадии компиляции.

C++ сущности, такие как tuple и лямбда функции, примененные для реализации алгоритма, хорошо оптимизируются компилятором. Когда функция для сборки локальной матрицы генерируется метафункцией, компилятор удаляет огромное дерево вызовов к лямбдам, tuple и промежуточным метафункциям и оптимизирует окончательный код. Эффективность оптимизации была протестирована на простом интеграле для 1D случая

$$\int_{\Omega_k} T\, \phi_i \phi_j d\Omega \qquad i = 1,2; j = 1,2. \tag{11}$$

Значения коэффициента T вводятся юзером через стандартный поток ввода std::cin. Это важно, поскольку если значения T будут непосредственно внесены в код программы, тогда компилятор

вычисляет все значения матрицы на стадии компиляции и удаляет весь машинный код кроме вывода в терминал.

```
1    call sym std::istream
2    movsd xmm3, qword [rsp + 8]
3    movsd xmm1, qword [rsp + 0x10]
4    movsd xmm4, qword [0x00000db0]
5    movapd xmm5, xmm3
6    movsd xmm0, qword [0x00000da8]
7    movapd xmm2, xmm1
8    mulsd xmm5, xmm4
9    mulsd xmm1, xmm4
10   mulsd xmm2, xmm0
11   mulsd xmm0, xmm3
12   addsd xmm2, xmm5
13   addsd xmm0, xmm1
14   mulsd xmm2, qword [0x00000db8]
15   mulsd xmm0, qword [0x00000dc0]
16   addsd xmm0, xmm2
17   call sym std::ostream
```

Рис. 4. Дизассемблированный код

Тестовый код был скомпилирован компилятором gcc 7.4 с флагом оптимизации “-Ofast” и получившийся двоичный файл был дизассемблирован программой radare2. На рис. 4 показана часть дизассемблированного кода функции main. Этот код выполняется между вводом и выводом в терминал и состоит из низкоуровневых операций. Такой код был сгенерирован вложенными вызовами метафункций декартового произведения, интегрирования, интерполирования коэффициента и преобразования координат и итоговый машинный код состоит только из низкоуровневой арифметики. Таким образом, компилятор удалил все вложенные вызовы метафункций и выполнил оптимизацию кода.

Тест на решении уравнения Пуассона

Чтобы продемонстрировать применимость этого метода для решения задач на реальных сетках, решение уравнения Пуассона (14) выполняется на FEMEngine и сравнивается с широко известными открытыми исходными кодами FEniCS и FreeFEM++. Выражение в правой части уравнения (12) выбрано таким образом, чтобы формула (15) описывала точное решение. Граничные условия определяются выражением (16).

$$f(x,y,z) = 12(x^2 + y^2 + z^2), \tag{12}$$

$$\Delta T = f(x,y,z), \tag{13}$$

$$\int_\Omega \nabla T \cdot \nabla \psi d\Omega = -\int_\Omega f(x,y,z)\psi \, d\Omega, \tag{14}$$

$$T(x,y) = x^4 + y^4 + z^4, \tag{15}$$

$$T(x,y)|_{\partial\Omega} = x^4 + y^4 + z^4. \tag{16}$$

Для тестирования использовалась 1x1x1 кубическая геометрия разбитая на тетраэдрическую сетку из 256000 ячеек. Вычисления производились на процессоре Intel i7-6700K. FEMEngine вычисляет глобальную матрицу за 1.07 с, тогда как FreeFem++ затрачивает 8.32 с. Пакет FEniCS использует компиляцию во время выполнения, поэтому первый запуск расчета затрачивает 3.41 с на сборку матрицы, а последующие только 0.21 с. Численные результаты, полученные с помощью FEMEngine и FreeFEM++ на одной и той же сетке, полностью совпадают, если применяемая квадратура и решатель линейной системы идентичны. Решение FEniCS отличается на 0,03 %, как и ожидалось, поскольку используется другой решатель линейной системы. Численное решение показано на рис. 5.

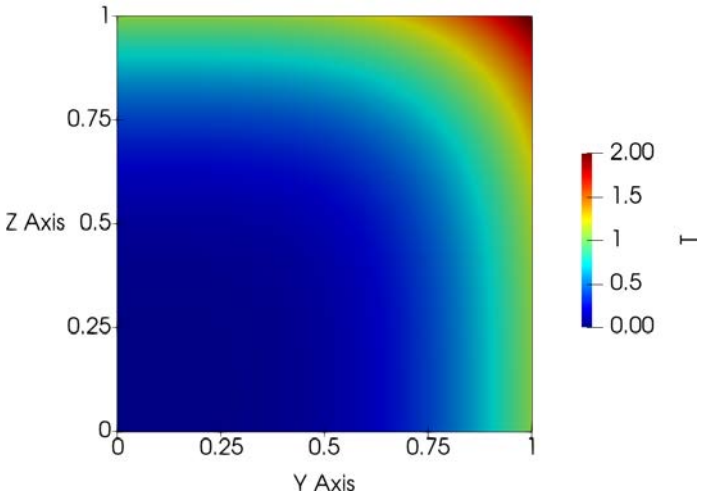


Рис. 5. Решение уравнения Пуассона в плоскости YZ, $x = 0.5$

FreeFEM++ анализирует файл “.edp”, содержащий постановку задачи и собирает матрицу во время выполнения. Коды, которые могут решать широкий класс задач, определяемых входными данными, полученными во время выполнения, обычно полагаются на затратный по времени динамический полиморфизм и сложную логику построенную на ветвлениях. FEniCS использует метод компиляции линковки во время выполнения вместе с генерацией кода C на Python. Оптимизированный код для решения задачи компилируется в динамическую библиотеку во время выполнения программы и загружается программой на ходу. Этот метод чрезвычайно сложно реализовать и поддерживать, но его производительность и универсальность являются одними из самых высоких. FEniCS собирает матрицу в 40 раз быстрее, чем FreeFEM++ для этой задачи. FEMEngine основан на шаблонах C++, поэтому программа компилируется для одной решаемой задачи. FEMEngine собирает матрицу примерно в 8 раз быстрее, чем FreeFEM++, и в 5 раз медленнее чем операции сборки матрицы FEniCS. Шаблонный код для генерации локальной матрицы жесткости основан на оптимизации, обеспечиваемой компилятором C++, которая уже достаточно эффективна и ее трудно улучшить.

Выводы

Разработана библиотека FEMEngine на основе шаблонного метапрограммирования C++ для конечно-элементного анализа.

Шаблонное метапрограммирование наряду с функциональным подходом имеет большой потенциал для разработки кода конечных элементов. Эти методы программирования позволяют писать надежный, универсальный и эффективный код.

Проведено сравнение FEMEngine и FreeFEM++ и достигнуто 8-кратное преимущество по времени построения глобальной матрицы. Сравнение с кодом FEniCS показывает, что существует потенциал для оптимизации узких мест текущего алгоритма сборки матрицы.

Литература

1. Hecht F. New development in freefem++ // J. Numer. Math. 2012. Vol. 20, № 3–4. P. 251–265.

2. Logg A., Mardal K. A., Wells G. N. Automated Solution of Differential Equations by the Finite Element Method. – Springer, 2012.

EFFECTIVE METAPROGRAMMING OF FEM ALGORITHMS ON C++17

A. M. Gurin, A. N. Baykin

Lavrentiev Institute of Hydrodynamics SB RAS, Novosibirsk

The paper is devoted to the use of functional and metaprogramming for the development of a finite element solver. The capabilities of modern C++ allow you to implement finite element method straightforwardly through metafunctions acting on functions. This method generalizes the assembly of a local matrix from the weak form of the equation. The approach is tested on solving the Poisson equation and compared with known open-source codes.

Key words: finite element method, functional programming, metaprogramming.

УДК 004.4
DOI 10.53403/9785951505071_2022_142

ПАКЕТ ПРОГРАММ ЛОГОС. ИСПОЛЬЗОВАНИЕ МЕТОДА СГЛАЖЕННЫХ ЧАСТИЦ ПРИ РЕШЕНИИ ЗАДАЧ ВЫСОКОСКОРОСТНОГО ДИНАМИЧЕСКОГО ДЕФОРМИРОВАНИЯ

Д. Ю. Дьянов, А. В. Казанцев, А. А. Челаков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Доклад посвящен описанию возможностей, реализованных в рамках SPH-методики пакета программ ЛОГОС в период 2015–2019 гг. В частности, приводится описание модели откола, граничного условия «SPH плоскость симметрии» и условия совместности деформации («склейка») между SPH-частицами и лагранжевой конечно-элементной сеткой, кроме того описаны алгоритмы для проведения расчетов осесимметричных течений методом SPH в двумерной постановке. Работоспособность и эффективность функциональных возможностей, представленных в рамках данного доклада, демонстрируется на решении ряда верификационных задач.

Ключевые слова: пакет программ ЛОГОС, программный модуль ЛОГОС-ПА, динамическое деформирование, метод конечных элементов, метод сглаженных частиц, контактное взаимодействие, «склейка».

Введение

В ФГУП «РФЯЦ-ВНИИЭФ» ведется разработка расчетно-математического пакета программ ЛОГОС [1], предназначенного для инженерного анализа и суперкомпьютерного моделирования в интересах высокотехнологичных отраслей промышленности. Одной из составных частей пакета программ ЛОГОС является программный модуль для моделирования НДС конструкций при динамических, статических, вибрационных нагрузках (далее – ЛОГОС-ПА) [2].

Одним из условий успешного внедрения пакета программ ЛОГОС в полный цикл разработки изделий на предприятиях является наличие широкого набора функциональных возможностей для моделирования задач динамического деформирования. Базовом способом численного решения указанного класса задач в ЛОГОС-ПА является использование лагранжевого подхода на основе метода конечных элементов для аппроксимации по пространству при явной конечно-разностной аппроксимации по времени [3–4].

При моделировании задач динамического деформирования часто встречаются ситуации, когда возникают области больших деформаций и разрушений с разлетом вещества, как, например, при высокоскоростном соударении, а также ином интенсивном нагружении тел. Одним из способов расчета подобных задач является использование бессеточного лагранжевого метода SPH [5]. Основной идеей данного метода является моделирование сплошной среды набором частиц постоянной массы (далее SPH-частицы).

SPH-подход был внедрен в ЛОГОС-ПА в 2013 году. Особенности реализации данного метода в ЛОГОС-ПА приведены в [6], где описана численная аппроксимация, алгоритмы распараллеливания на распределенной памяти, а также алгоритмы проведения расчетов связанного деформирования лагранжевых конечно-элементных сеток и подобластей SPH-частиц на основе расчета контактного взаимодействия между ними.

Данный доклад посвящен описанию развития SPH-методики пакета программ ЛОГОС в период 2015–2019 гг. Целью развития было повышение точности моделирования высокоскоростного (до 10 км/с) удара твердых тел и реализация ряда методов для сокращения вычислительных затрат. Для повышения точности реализована описанная в докладе модель откола, работоспособность которой демонстрируется на решении двух верификационных задач. Кроме того, на одной из указанных задач в докладе показана эффективность ряда реализованных методов, предназначенных для сокращения вычислительных затрат. Данные методы включают в себя: граничное условие совместности деформации между SPH-частицами и лагранжевой сеткой («склейка»), граничное условие «SPH плоскость симметрии», а также алгоритмы для проведения расчетов осесимметричных течений методом SPH в двумерной постановке.

1. Модель откола

При численном моделировании высокоскоростного (до 10 км/с) удара твердых тел важен учет эффектов, связанных с разрушением материалов. В методике SPH для моделирования разрушения реализована модель откола, согласно которой зануляется девиатор тензора напряжений и давление при достижении критерия [7]. В дальнейшем разрушенные SPH-частицы считаются в газодинамическом приближении (т. е. без учета девиатора тензора напряжений) с занулением давления в случае растяжения. Общая схема расчета давления разрушенных SPH-частиц выглядит следующим образом:

$$P = \begin{cases} P \text{ при } P > 0; \\ 0 \text{ при } P \leq 0. \end{cases} \quad (1)$$

Задача о соударении стального шарика со стальной преградой с начальной скоростью 923 м/с

В качестве первого тестового примера рассматривается задача о моделировании воздействия сферическим ударником по плоской преграде. Ударник и преграда выполнены из стали. На рис. 1 приведена начальная геометрия задачи.

Начальная скорость V ударника составляет 923 м/с и направлена по нормали к преграде. Диаметр d ударника равен 7.1 мм. Преграда представляет собой диск с диаметром $D = 150$ мм и толщиной $h = 6$ мм. Боковая поверхность диска жестко закреплена.

Моделирование проводится методом SPH в трехмерной постановке на последовательно сгущающихся разбиениях расчетной области. В табл. 1 приведены параметры дискретизации.

На рис. 2 показана начальная расстановка SPH-частиц ударника и преграды в области зоны удара (вид в плоскости XOY).

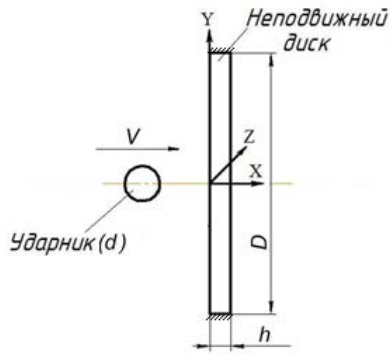
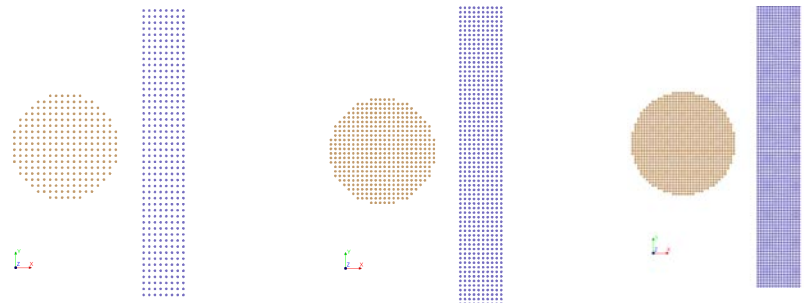


Рис. 1. Начальная геометрия

Таблица 1

Параметры дискретизации расчетной области

Шаг разбиения Δ, мм	0.8	0.6	0.4
Количество SPH-частиц ударника	3112	7208	24464
Количество SPH-частиц преграды	222144	490800	1656915



Δ = 0.8 мм Δ = 0.6 мм Δ = 0.4 мм

Рис. 2. Начальная расстановка частиц в зоне удара (вид в плоскости XOY)

Для описания сдвиговой прочности материала в расчете использовалась модель Джонсона-Кука. В качестве параметра, определяющего разрушение, было принято первое главное напряжение. В табл. 2 приведены физико-механические параметры модели материала.

Таблица 2

Физико-механические параметры модели материала

Параметр	Значение
начальная плотность ρ, кг/м3	7850
модуль сдвига *G*, Гпа	81.8
начальный предел текучести *A*, МПа	350
параметр деформационного упрочнения *B*, МПа	275
параметр деформационного упрочнения *N*	0.36
параметр динамического упрочнения *C*	0.022
параметр термического разупрочнения *M*	1
температура плавления TM, K	1811
первое главное напряжения разрушения SIGP1, Гпа	2.25

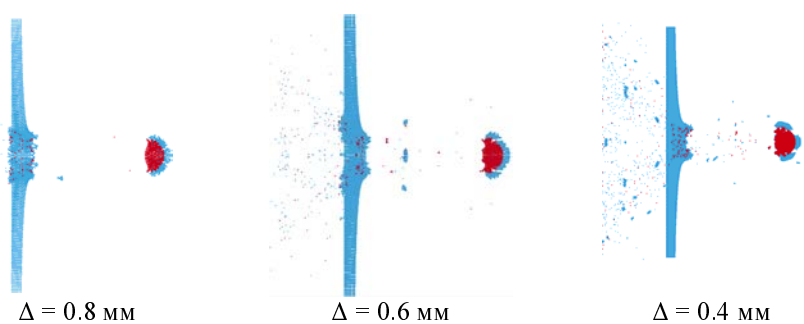
Для вычисления давления и температуры использовался УРС Ми-Грюнайзена. В табл. 3 приведены параметры УРС.

Таблица 3

Параметры УРС

Параметр	Значение
коэффициент Грюнайзена GAMMA0	2.17
начальная скорость звука *C*1, м/с	4596
констант *S*1	1.49
нормальная температура Tref, K	298
удельная теплоемкость *CV*, Дж / (кг·K)	452

На рис. 3 приведено полученное численное решение. Для наглядности результаты приводятся для половины геометрии.



Δ = 0.8 мм Δ = 0.6 мм Δ = 0.4 мм

Рис. 3. Результаты расчета

На рис. 4 приведена зависимость запреградной скорости от дискретизации, полученная в расчетах по ЛОГОС.

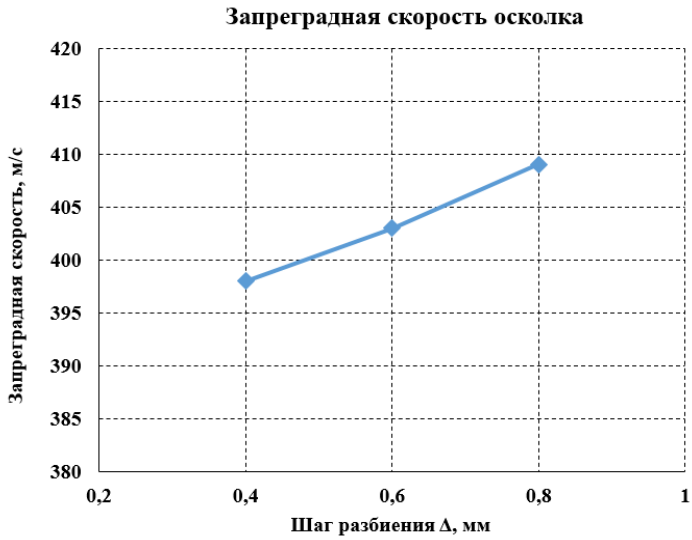


Рис. 4. Зависимость запреградной скорости от дискретизации

Приведенная зависимость имеет характер, близкий к линейному. На основе чего можно сделать вывод, что порядок сходимости численного решения близок к первому.

Задача о соударении алюминиевого шарика с алюминиевой преградой с начальной скоростью 7400 м/с

В качестве второго тестового примера рассматривается задача о моделировании воздействия алюминиевого сферического ударника по плоской алюминиевой преграде. Постановка задачи описана в [8]. Начальная скорость ударника составляет 7400 м/с по нормали к преграде. Диаметр ударника – 3.2 мм. Преграда представляет собой прямоугольный параллелепипед с габаритами 80×80 мм и толщиной 14 мм.

Расчет проводится методом SPH в трехмерной постановке. Использовалось решетчатое разбиение расчетной области с шагом 0.4 мм. Таким образом расчетная модель состоит из 1400280 SPH-частиц (из них 280 – шарик, 1400000 – преграда). На рис. 5 показана начальная расстановка SPH-частиц ударника и преграды в области зоны удара (вид в плоскости XOY).

Для описания сдвиговой прочности материала в расчете использовалась модель Джонсона-Кука. В качестве параметра, определяющего разрушение, было принято первое главное напряжение. В табл. 4 приведены физико-механические параметры модели материала.

Для вычисления давления и температуры использовался УРС Ми-Грюнайзена. В табл. 5 приведены параметры УРС.

Расчеты проводились в двух постановках: с учетом и без учета модели откола. На рис. 6 приведены результаты расчетов в сравнении с экспериментальными данными [8]. Для наглядности результаты показаны для двух слоев SPH-частиц в области плоскости симметрии.

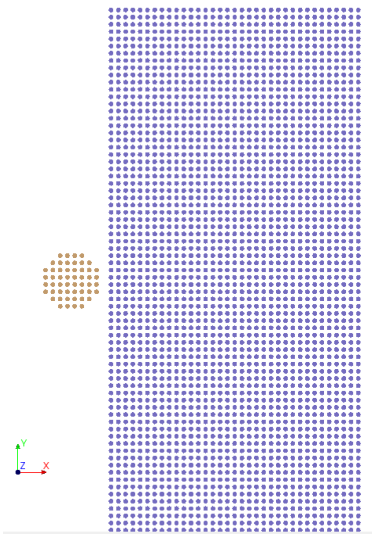


Рис. 5. Начальная расстановка SPH-частиц в зоне удара (вид в плоскости XOY)

Таблица 4

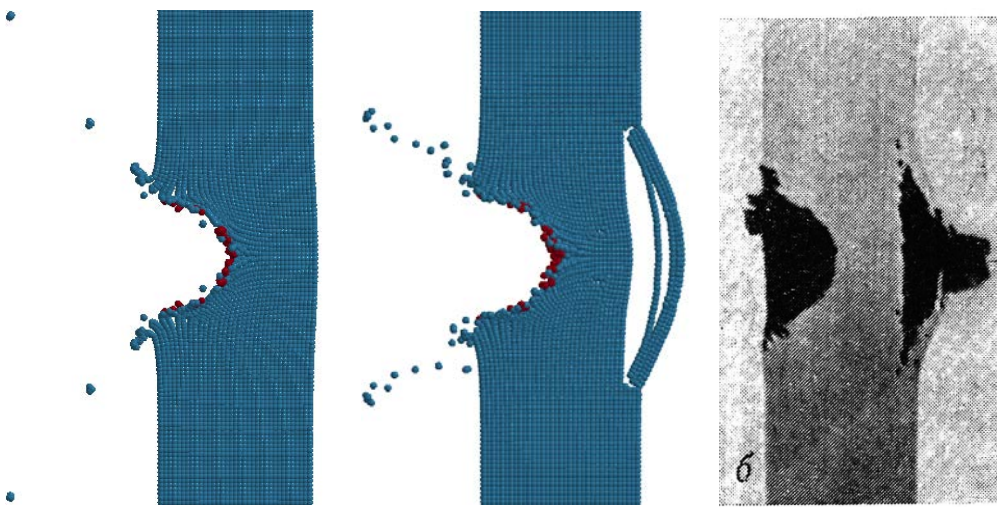
Физико-механические параметры модели материала

Параметр	Значение
начальная плотность ρ, кг/м3	2780
модуль сдвига G, Гпа	27.6
начальный предел текучести A, МПа	330
параметр деформационного упрочнения B, МПа	426
параметр деформационного упрочнения N	0.34
параметр динамического упрочнения C	0.015
параметр термического разупрочнения M	1
температура плавления TM, K	910
первое главное напряжения разрушения SIGP1, Гпа	1.1

Таблица 5

Параметры УРС

Параметр	Значение
коэффициент Грюнайзена GAMMA0	2
начальная скорость звука C1, м/с	5328
констант S1	1.338
нормальная температура Tref, K	298
удельная теплоемкость CV, Дж /(кг·K)	875



без учета модели откола
глубина кратера 6.5 мм

с учетом модели откола
глубина кратера 6.6 мм

эксперимент -
глубина кратера 6.9 мм

Рис. 6. Сравнение результатов расчетов с экспериментом (результаты расчетов показаны для двух слоев SPH-частиц в области плоскости симметрии)

Из приведенных результатов видно, что расчет без использования модели откола неплохо согласуется с экспериментом лишь по глубине кратера. А расчет, использующий модель откола, хорошо описывает эксперимент как с точки зрения глубины кратера, так и с точки зрения откола тыльной стороны преграды.

2. Граничное условие «склейка» SPH с лагранжевой сеткой

Для моделирования совместности деформации несогласованных по счетной сетке частей модели в лагранжевой конечно-элементной методике ЛОГОС-ПА доступно граничное условие «склейка». При подготовке расчетной модели задаются пары «склеенных» поверхностей - подчиненная и главная.

Вычислительный алгоритм накладывает связь между сеточными узлами подчиненной поверхности и гранями главной поверхности.

Граничное условие «склейка» между SPH-частицами и лагранжевой сеткой целесообразно использовать, например, когда область преграды вне зоны воздействия на нее ударника деформируется незначительно. Малодеформируемую область преграды можно моделировать в конечно-элементном приближении с менее подробной дискретизацией. Для согласованного деформирования используется условие «склейка» на границе SPH-частиц и лагранжевой сетки. Тем самым значительно сокращается количество SPH-частиц в расчете, а, следовательно, и вычислительные затраты.

Базовым методом ЛОГОС-ПА для расчета «склейки» является метод кинематических связей при подходе «узел-грань» [7]. Далее приведено общее описание расчета «склейки» в ЛОГОС-ПА.

Используя параметрическое представление, определяется положение подчиненного узла на главной грани, как показано на рис. 7. Делается это один раз – при инициализации задачи.

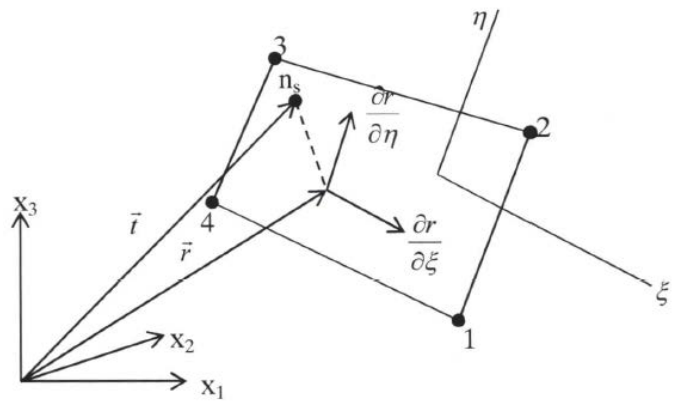


Рис. 7. Проекция подчиненного узла на главную грань

Согласно методу кинематических связей для согласованного движения пары «узел-грань» на каждом счетном шаге выполняется два этапа:

– интерполяция узловых сил f_S и массы M_S подчиненных узлов в узлы главной грани

$$\Delta f_M^i = \varphi_i(\xi_c,\eta_c) f_S, \quad (2)$$

$$\Delta M_M^i = \varphi_i(\xi_c,\eta_c) M_S; \quad (3)$$

– интерполяция насчитанных скоростей v_M узлов главной грани в подчиненные узлы

$$v_S^i - \sum_j \varphi_j(\xi_c,\eta_c) v_M^j, \quad (4)$$

где i, j – индексы узлов главной грани, (ξ_c,η_c) – локальные координаты узла на гране, φ_i – функции формы грани.

Для расчета согласованного деформирования лагранжевой сетки и области SPH проведена работа по адаптации процедур определения информации о связываемых парах «узел-грань», а также вычислительных процедур метода кинематический связей.

Отдельно стоит отметить распараллеливание алгоритмов «склейки» при счете на распределенной памяти. В ЛОГОС-ПА используется подход, согласно которому для каждой связанной пары «узел-грань» главная грань лежит на «своем» mpi процессе, а подчиненный узел – на любом. В случае, если подчиненный узел лежит не том же mpi процессе, что и главная грань, выполняются процедуры синхронизации необходимых величин. Для метода кинематических связей перед процедурой интерполяции «подчиненный узел -> узлы главной грани» выполняется процедура синхронизации узловых сил и массы. А перед интерполяцией «узлы главной грани -> подчиненный узел» вычисленные скорости подчиненных узлов отсылаются на те mpi процессы, на которых они лежат. В случае, когда подчиненный узел является SPH-частицей, дело усложняется тем, что она может «перелетать» с одного mpi процесса на другой. В связи с этим, в структуру данных «узел-грань» добавлен признак о том, что подчиненный узел является SPH-частицей, а также номер процесса, на

котором лежит подчиненный узел. Кроме этого, в структуру SPH-частиц добавлено два поля для хранения номера в массиве «узел-грань» и номера mpi процесса, на котором лежит главная грань. С использованием указанной информации реализованы две дополнительные процедуры, выполняющие синхронизацию необходимых величин между mpi процессами, когда подчиненный узел является SPH-частицей.

Тестовый пример на «склейку» SPH с лагранжевой сеткой

Эффективность данного метода демонстрируется на примере рассмотренной ранее задачи о соударении алюминиевого ударника с алюминиевой преградой с начальной скоростью 7400 м/с. Ударник и область преграды размером 40×40 мм моделируются SPH-частицами, остальная часть преграды – в лагранжевом конечно-элементном приближении. Пространственная дискретизация и области SPH, и сеточной области выполнялась с шагом 0.4 мм. На рис. 8 приведено изображение вида расчетной модели.

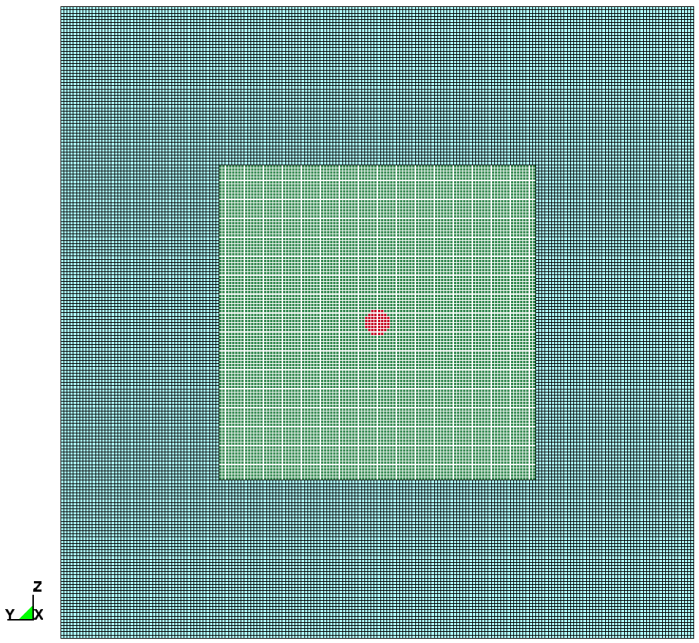


Рис. 8. Расчетная модель (вид в плоскости YOZ)

Параметры пространственной дискретизации приведены в табл. 6.

Таблица 6

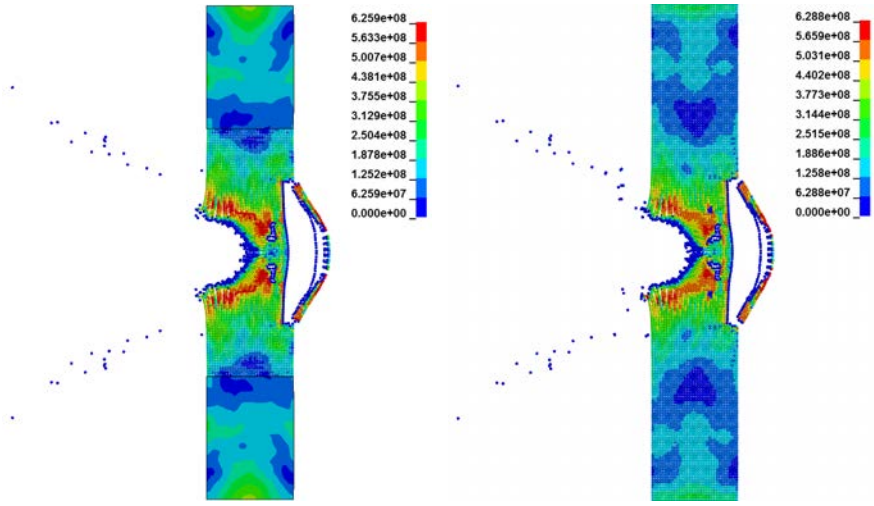
Параметры дискретизации

	SPH-частицы	8-ми узловой гексаэдр
Ударник	280	–
Преграда	350000	1050000

На границе сетки и области SPH-частиц задается условие «склейка» для моделирования их согласованного деформирования.

На рис. 9 приведены полученные результаты в сравнении с аналогичным расчетом в чистой SPH постановке. Для наглядности результаты показаны для двух слоев SPH-частиц в области плоскости симметрии. Кроме того, отображено поле напряжений по Мизесу (шкала приведена в Па).

Из приведенных результатов видно, что связанная постановка практически идеально повторяет результат чистой SPH задачи. В табл. 7 приводится сравнение времен счета и эффективности распараллеливания задачи в связанной и чистой SPH постановках с использованием разного числа mpi процессов.



связанная постановка чистая SPH постановка

Рис. 9. Сравнение результатов расчетов связанной и чистой SPH постановок

Таблица 7

Сравнение времен счета и эффективности распараллеливания

Количество ядер (mpi)	Связанная постановка		Чистая SPH постановка	
	Время счета, с	Эффективность, %	Время счета, с	Эффективность, %
24	678	68	2165	70
48	366	63	1145	66
96	197	58	605	63

3. Граничное условие «SPH плоскость симметрии»

Зачастую при моделировании задач, имеющих симметрию относительно какой-либо плоскости, с целью сокращения вычислительных затрат используют только половину геометрии с заданием условия симметрии. В случае лагранжевой конечно-элементной методики для задания условия симметрии достаточно ограничить перемещения узлов, лежащих в плоскости симметрии. В случае же SPH-частиц этого делать недостаточно по двум причинам. Первая причина состоит в том, что SPH-частицы перемешиваются в процессе счета, и, если закрепить перемещения для SPH-частиц на плоскости симметрии, то соседние к ним вполне могут проникнуть за плоскость. Во-вторых, у SPH-частиц, находящихся рядом с плоскостью симметрии, не хватает соседей, что ведет к занижению в них давления. В связи с этим, реализовано граничное условие «SPH плоскость симметрии» [7]. Далее описана суть алгоритма, реализованного в ЛОГОС-ПА.

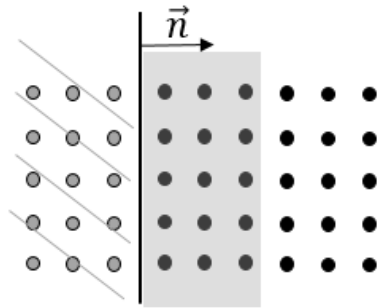


Рис. 10. Виртуальные SPH-частицы (заштрихованная область)

На каждом счетном шаге для исходных SPH-частиц, расстояние от которых до плоскости симметрии не более $2h$, строятся виртуальные. Виртуальная SPH-частица получаются с помощью операции зеркального отражения исходной относительно плоскости симметрии, задаваемой по точке и вектору нормали $\vec{n}$, как показано на рис. 10.

В процессе численной аппроксимации рассматриваются только исходные SPH-частицы, а виртуальные используются в качестве соседей.

Масса, давление и плотность виртуальных SPH-частиц равны значениям этих параметров в исходных. Координаты и скорость вычисляются зеркальным отображением значений в исходных SPH-частицах относительно плоскости симметрии:

$$r_V^\alpha = r^\alpha - 2dn^\alpha, \quad u_V^\alpha = u^\alpha - 2(u,n)n^\alpha, \tag{5}$$

где d – расстояние от исходной SPH-частицы до плоскости симметрии.

Тензор напряжений σ_V виртуальной SPH-частицы получается путем преобразования $\sigma_V = M^T \sigma M$, где σ – тензор напряжений исходной, M – матрица зеркального отражения в декартовой системе координат относительно плоскости с вектором единичной нормали $\vec{n}(A, B, C)$:

$$M = \begin{pmatrix} -A^2 + B^2 + C^2 & -2AB & -2AC \\ -2AB & A^2 - B^2 + C^2 & -2BC \\ -2AC & -2BC & A^2 + B^2 - C^2 \end{pmatrix}. \tag{6}$$

Тестовый пример на использование «SPH плоскости симметрии»

Эффективность данного метода демонстрируется также на примере рассмотренной ранее задачи о соударении алюминиевого ударника с алюминиевой преградой с начальной скоростью 7400 м/с, но моделирование выполняется с использованием половины геометрии. На рис. 11 изображена расчетная модель.

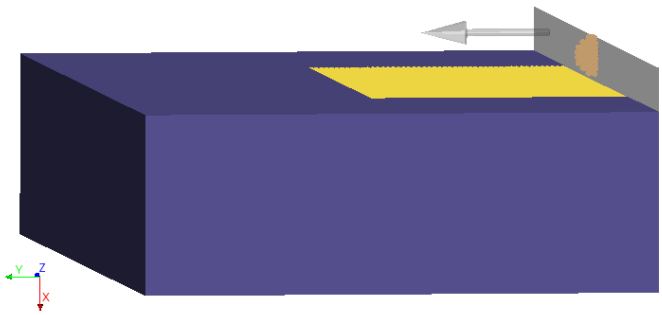


Рис. 11. Расчетная модель

Для моделирования симметрии задается граничное условие «SPH плоскость симметрии», изображенное на рис. 11. Для сеточной модели симметрия задается ограничением перемещений вдоль оси Oy для узлов сетки, находящихся в плоскости симметрии. Как и в предыдущем примере, на границе сетки и области SPH-частиц задается условие «склейка» для моделирования их согласованного деформирования.

На рис. 12 приведены результаты расчета на один из моментов времени для двух видов с отображением поля напряжений по Мизесу.

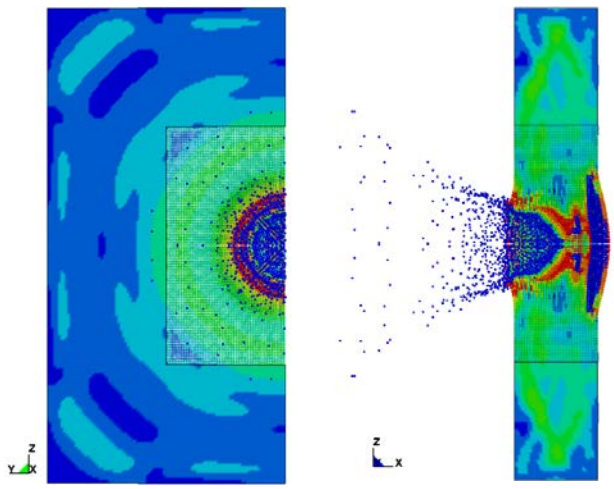


Рис. 12. Результаты расчета

Видно, что полученный результат симметричен относительно заданной плоскости симметрии, при этом он повторяет результат расчета полной модели. С точки зрения вычислительных затрат, на проведение расчетов необходимо примерно в 1.9 раз меньше времени по сравнению с аналогичными расчетами полной модели.

4. Алгоритм для расчета осесимметричных течений

Одним из востребованных способов проведения расчетов течений, имеющих осевую симметрию, является двумерное приближение. Существует несколько подходов к решению данной задачи методом SPH [9–10]. Для реализации в ЛОГОС-ПА взят за основу алгоритм, выполняющий переход от двумерной к решению трехмерной задачи, путем поворота исходных SPH-частиц относительно оси симметрии. Далее приведено более подробное описание реализованного алгоритма.

На каждом счетном шаге для исходных SPH-частиц строятся «дублеры», которые получаются с помощью операции поворота вокруг оси симметрии. Их количество n выбирается, исходя из того, что расстояние между ними не должно превышать h, и определяется выражением

$$n = \frac{\pi}{arcsin(\frac{h}{2r})}. \quad (7)$$

где r – расстояние от исходной частицы до оси симметрии.

В процессе численной аппроксимации рассматриваются только исходные SPH-частицы, а «дублеры» используются в качестве соседей. Для исходных SPH-частиц, расстояние от которых до оси не более $2h$, «дублеры» используются со всей окружности, а для всех остальных – только две (по одной справа и слева от исходной). На рис. 13 приведена общая схема построения «дублеров».

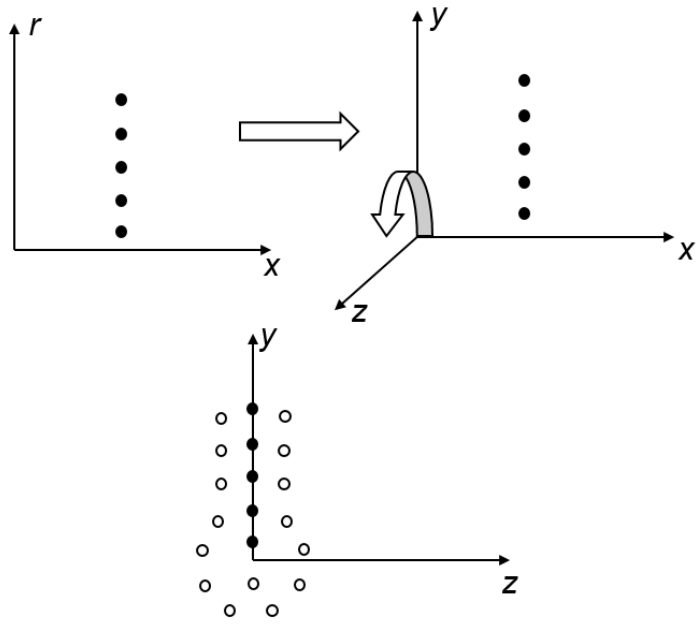


Рис. 13. Общая схема построения «дублеров» (закрашенные точки – исходные SPH-частицы, незакрашенные – «дублеры»)

Массы SPH-частиц определяются на каждом счетном шаге. Вычисление происходит исходя из сохранения массы тора, полученного вращением окружности относительно оси симметрии (радиус окружности равен $h/2$, а центр расположен в первоначальном положении исходной SPH частицы).

Масса, давление и плотность «дублеров» равны значениям этих параметров в исходных SPH-частицах. Координаты и скорость вычисляются поворотом значений исходных SPH-частиц на угол φ против часовой стрелки вокруг оси симметрии:

$$x_\varphi = x, y_\varphi = r\cos(\varphi), z_\varphi = r\sin(\varphi)$$
$$u_\varphi^x = u^x, u_\varphi^y = u^y\cos(\varphi), u_\varphi^y = -u^y\sin(\varphi) \quad (8)$$

Тензор напряжений υ_φ «дублеров» получается путем преобразования $\sigma_\varphi = A^T\sigma A$, где σ – тензор напряжений исходной SPH-частицы, A – матрица поворота в декартовой системе координат вокруг оси Ox (т.е. вокруг оси симметрии) на угол φ против часовой стрелки:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix}. \quad (9)$$

Тестовый пример на алгоритм расчета осесимметричных течений

Работоспособность данного метода демонстрируется также на примере рассмотренной ранее задачи о соударении алюминиевого ударника с алюминиевой преградой с начальной скоростью 7400 м/с, но моделирование выполняется в двумерной осесимметричной постановке. Дискретизация расчетной области выполнялась, как и в трехмерных расчетах, с шагом 0.4 мм. Таким образом расчетная модель состояла из 3526 SPH-частиц (из них 26 – шарик, 3500 - преграда). Начальная расстановка SPH-частиц приведена на рис. 14.

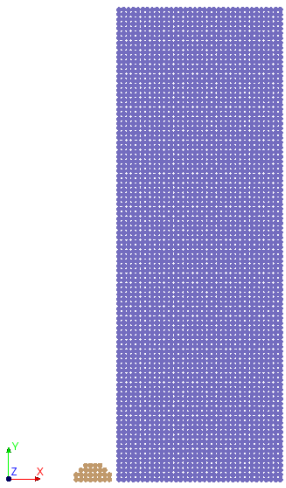
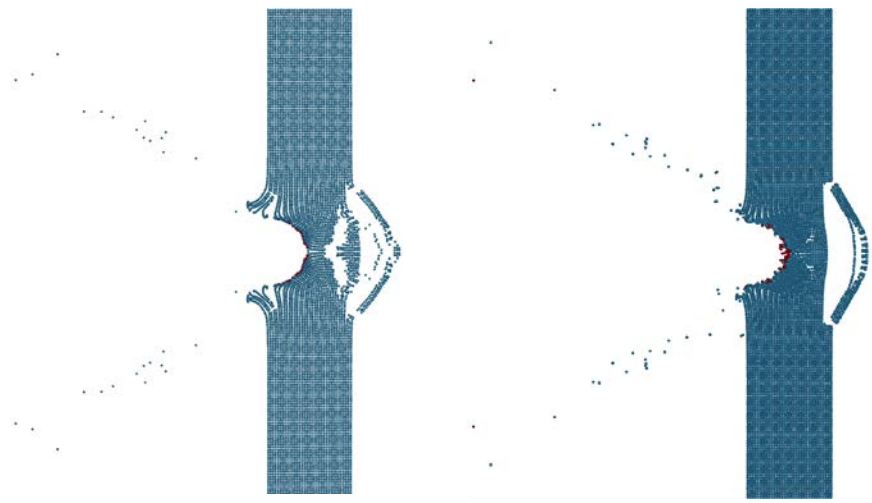


Рис. 14. Начальная расстановка SPH-частиц

На рис. 15 приведены результаты двумерного расчета в сравнении с аналогичным трехмерным. Для наглядности результаты двумерных расчетов отражены зеркально относительно плоскости XOZ, а трехмерные показаны для двух слоев SPH-частиц в области плоскости симметрии.

Из приведенных результатов видно, что в двумерном расчете, как и в аналогичном трехмерном, получено отсутствие пробития и откол тыльной стороны преграды. Но стоит отметить, что двумерный расчет дает чуть более завышенные параметры разрушения преграды. С точки зрения вычислительных затрат, двумерный расчет выполнился менее чем за полминуты в скалярном режиме на персональном компьютере. В табл. 7 приведены вычислительные затраты для аналогичных трехмерных задач, где самый быстрый расчет чистой SPH постановки выполнен за 10 минут с использованием 96 mpi процессов вычислительного кластера.



двумерная постановка трехмерная постановка

Рис. 15. Сравнение результатов расчета задачи в двумерной и трехмерной постановках

Заключение

Методы и алгоритмы, описанные в рамках данной работы, предназначены для повышения точности моделирования, а также сокращения вычислительных затрат при моделировании задач динамического деформирования в программном модуле ЛОГОС-ПА с использованием метода SPH.

В работе приведено описание реализованной модели откола, а также граничных условий «склейка» и «SPH плоскость симметрии». Описаны реализованные алгоритмы для проведения расчетов осесимметричных течений методом SPH в двумерной постановке. Приведенные результаты расчета ряда верификационных задач подтверждают работоспособность и эффективность реализованных возможностей.

Литература

1. Дерюгин Ю. Н., Козелков А. С., Спиридонов В. Ф., Циберев К. В., Шагалиев Р. М. Многофункциональный высокопараллельный пакет программ ЛОГОС для решения задач тепломассопереноса и прочности // С.-Петербургский науч. форум «Наука и общество»: Сб. тез. докл., С.-Пб, 2012. С. 102.

2. Авдеев П. А., Александрова О. Л., Артемова Е. О. и др. Обзор возможностей моделирования задач прочности с использованием пакета программ ЛОГОС // Супервычисления и математическое моделирование: Труды XV международного семинара, Саров, 13–17 октября, 2014. С. 9–17.

3. Дьянов Д. Ю., Казанцев А. В., Циберев К. В. и др. Пакет программ ЛОГОС. Функциональные возможности для решения динамических задач прочности // Супервычисления и математическое моделирование: Труды XVII международного семинара, – Саров, 15–19 октября, 2018. С. 210–219.

4. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В. и др. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 2. С. 3–13.

5. Gingold R. A., Monagham J. J. Smoothed Particle Hydrodynamics: Theory and application to non-spherical stars // Month. Not. Royal Astr. Soc. 1977. P. 375–389.

6. Дьянов Д. Ю., Казанцев А. В., Морозов С. В. и др. Пакет программ ЛОГОС. Функциональные возможности для решения связанных задач прочности и гидродинамики сглаженных частиц // Супервычисления и математическое моделирование: Труды XV международного семинара, Саров, 13–17 октября, 2014. С. 212–220.

7. Hallquist John O. LS-DYNA Theory Manual. - Livermore Software Technology Corporation, 2015. http:/www.lstc.com/download/manuals.

8. Геринг Дж. Высокоскоростной удар с инженерной точки зрения. Высокоскоростные ударные явления. – М.: МИР, 1973.

9. Башуров В. В., Иоилев А. Г. Алгоритм метода сглаженных частиц для моделирования осесимметричных упругопластических течений // Всероссийская конференция «Теоретические основы и конструирование численных алгоритмов решения задач математической физики»: Сб. докл., Новосибирск, сентябрь, 2003.

10. Зубов А. Д., Лебедев А. М. Метод сглаженных частиц SPH для расчетов газодинамических задач со сферической и цилиндрической симметриями // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2009. Вып. 1. С. 19–28.

"LOGOS" SOFTWARE PACKAGE. APPLICATION OF THE SMOOTHED-PARTICLE HYDRODYNAMICS METHOD FOR HIGH-STRAIN-RATE DYNAMIC DEFORMATION SIMULATIONS

D. Yu. Dyanov, A. V. Kazantsev, A. A. Chelakov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents the capabilities implemented based on the SPH method in the LOGOS software package in 2015-2019. In particular, it describes the spall model, the "SPH plane of symmetry" boundary condition and the strain compatibility (merging) conditions for SPH particles and Lagrange finite-element meshes, and specifies algorithms for two-dimensional axisymmetric flow calculations by the SPH method. A number of verification cases are given to demonstrate the performance and efficiency of the functional capabilities presented in the paper.

Key words: LOGOS software package, LOGOS-SA software module, dynamic deformation, finite-element method, smoothed-particle hydrodynamics, contact interaction, merging

УДК 004.4
DOI 10.53403/9785951505071_2022_155

ПАКЕТ ПРОГРАММ ЛОГОС. МЕТОДЫ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ КОМПОЗИТНЫХ МАТЕРИАЛОВ ПРИ ДИНАМИЧЕСКОМ НАГРУЖЕНИИ

Д. Ю. Дьянов, М. В. Медведкина, Е. И. Наумова, Е. В. Шувалова

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе представлены методы моделирования композитных материалов, доступные в пакете программ ЛОГОС при численном моделировании процесса динамического деформирования композитных конструкций. Приведено описание некоторых моделей композитного материала, а именно моделей многослойного композитного материала с учетом разрушения. Модель многослойного композитного материала основана на представлении композита как пакета слоев, с различными углами укладки друг относительно друга и использовании теории оболочек.

Для демонстрации работоспособности реализованного функционала в докладе представлены результаты численных расчетов ряда постановок конструкций с композитными материалами, проведенных с использованием пакета программ ЛОГОС.

Ключевые слова: пакет программ ЛОГОС, программный модуль ЛОГОС-ПА, динамическое деформирование, метод конечных элементов, композитные материалы.

Введение

В настоящее время во ФГУП «РФЯЦ-ВНИИЭФ» разрабатывается пакет программ ЛОГОС [1–2] для решения широкого круга задач, в том числе механики деформируемого твердого тела на многопроцессорных вычислительных ресурсах. Программный модуль (ПМ) пакета программ ЛОГОС для моделирования задач прочности (ЛОГОС-ПА) обеспечивает решение задач статической, динамической и вибрационной прочности. Для успешного внедрения пакета программ ЛОГОС в полный цикл разработки изделий на предприятиях оборонно-промышленного комплекса и высокотехнологичных предприятий гражданской отрасли необходима программная реализация современных моделей поведения конструкционных материалов, в частности композитных материалов.

Композитные материалы, или композиты, состоят из двух и более компонент, существенно отличающихся по физико-механическим характеристикам. В их основе лежит уникальная по простоте идея армирования, когда соединяют «полярные» по свойствам материалы – податливую матрицу и жесткую и прочную арматуру. Армирующие элементы обычно обеспечивают необходимые механические характеристики материала (прочность, жесткость), а матрица (или связующее) обеспечивает совместную работу армирующих элементов и защиту их от механических повреждений. Кроме того, соединение армирующих элементов и матрицы дает эффект синергизма, связанный с появлением у композиции свойств, не характерных для изолированных исходных компонентов. Композитные материалы [3–5] находят широкое применение в инженерной практике, так как позволяют создавать конструкции с уникальными весовыми, прочностными и диссипативными характеристиками, которых практически невозможно достичь использованием традиционных конструкционных материалов.

Актуальность использования композитных материалов вызвала появление новых классов задач в области инженерного анализа поведения конструкций. Как следствие этого, возникла необходи-

мость в реализации моделей, позволяющих моделировать поведение композитных материалов, в том числе с учетом разрушения.

В докладе представлены методы моделирования композитных материалов, доступные в ПМ ЛОГОС-ПА при численном моделировании процесса динамического деформирования композитных конструкций. Приведено описание некоторых моделей композитных материалов, а именно моделей многослойного композитного материала с учетом разрушения.

Для демонстрации работоспособности и эффективности реализованного функционала в докладе представлены результаты численных расчетов ряда постановок конструкций с композитными материалами, проведенных с использованием пакета программ ЛОГОС.

За предоставление тестовых задач, помощь в проведении расчетов и анализе результатов авторы выражают благодарность Бухареву Ю. Н. (СарФТИ НИЯУ МИФИ), Рябову А. А., Речкину В. Н., Маслову Е. Е., Романову В. И. (ФГУП «РФЯЦ-ВНИИЭФ»).

Классификация композитов

Поскольку композиты состоят из разных компонентов, количество их возможных сочетаний очень велико. В зависимости от материала матрицы выделяют композиты с металлической, керамической или полимерной матрицей. В зависимости от вида армирующего компонента композиты могут быть разделены на три основные группы: дисперсно-упрочненные, волокнистые, слоистые [6].

Дисперсно-упрочненные материалы – композитные материалы, в связующий компонент которых (матрицу) включены армирующие элементы в виде специально вводимых частиц. Моделирование подобных материалов, как правило, осуществляется с помощью изотропных моделей.

У волокнистых композитов матрица армирована высокопрочными волокнами, проволокой, нитевидными кристаллами. Моделирование подобных материалов усложняется их анизотропией, зависящей от направления волокон, порядка и места их расположения.

Слоистые композиты аналогичны волокнистым, только армирование проводится слоями.

Методы моделирования волокнистых и слоистых композитов

В инженерной практике используются разные методы моделирования волокнистых и слоистых композитов [7]. В ЛОГОС-ПА реализованы следующие методы:

– прямой метод моделирования волокон и слоев посредством объемных, оболочечных и балочных элементов. Данный подход напрямую моделирует все волокна и слои композитного материала, но требует больших вычислительных и трудозатрат. Моделирование композитов данным методом возможно с использованием моделей, описанных в работах [8, 9];

– метод, основанный на усредненных характеристиках (с использованием ортотропных и анизотропных моделей для оболочечных и объемных элементов [9], а также с использованием моделей композитных материалов для объемных элементов [9]);

– метод, основанный на модели многослойного материала на основе модели ортотропного материала. Примером данной модели является модель, представленная в докладе и описанная далее.

Модель многослойного композитного материала

Модель многослойного композитного материала основана на представлении композита как пакета слоев, с различными углами укладки друг относительно друга и использовании теории оболочек (рис. 1). Каждый слой представляет собой ориентированный материал, армированный параллельными волокнами (рис. 2). Для каждого из слоев можно задать модель материала, толщину слоя и угол направления волокон.

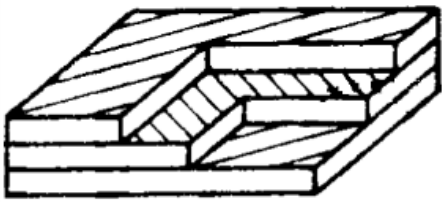


Рис. 1. Пример многослойного композита

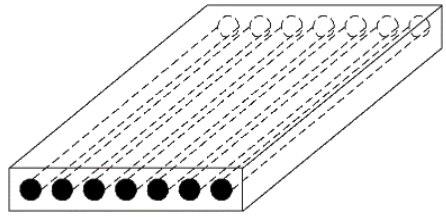


Рис. 2. Схема слоя, армированного непрерывными волокнами

Упругие и прочностные характеристики слоя существенно различаются по взаимно перпендикулярным направлениям. Модель композитного материала основана на модели ортотропного материала. Главные оси ортотропии направлены вдоль и поперек волокон. В продольном направлении свойства композитного материала определяются свойствами волокна, в поперечном – свойствами матрицы. Материал слоя характеризуется [10]:

1) упругими характеристиками:

– модулями упругости E_a и E_b вдоль и поперек волокон;

– модулем сдвига G_{ab} в плоскости волокон;

– коэффициентом Пуассона v_{ab} в поперечном направлении;

2) прочностными характеристиками:

– разрушающими напряжениями $\sigma_a^{\text{р}}$ и $\sigma_a^{\text{сж}}$ при растяжении и сжатии вдоль волокон;

– разрушающими напряжениями $\sigma_b^{\text{р}}$ и $\sigma_b^{\text{сж}}$ при растяжении и сжатии поперек волокон;

– разрушающими напряжениями σ_{ab}^{*} при сдвиге в плоскости волокон.

Эти характеристики определяются экспериментально и содержатся в базе данных материалов. Диаграммы деформирования всех ориентированных композитов при растяжении и сжатии вдоль волокон в первом приближении можно считать линейными вплоть до разрушения материала [10].

Каждый слой характеризуется также системой координат материала. В ЛОГОС-ПА реализована поддержка различных систем координат (рис. 3).

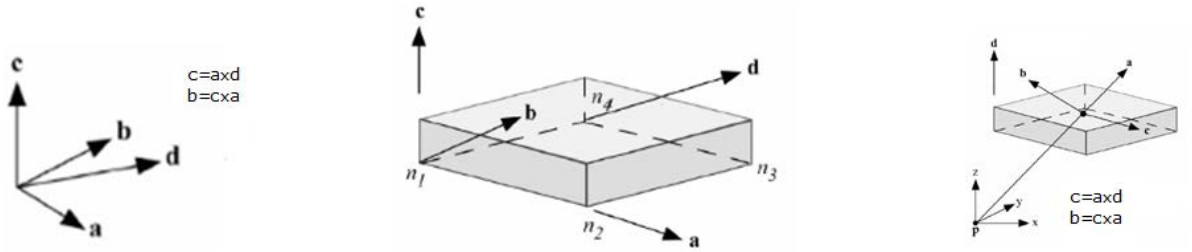


Рис. 3. Примеры материальных систем координат

Модель композитного материала включает в себя модель упругого деформирования однонаправленного слоя композитного материала и модель разрушения. Связь между тензорами напряжений и деформаций устанавливается на основе закона Гука для ортотропных материалов. Модель разрушения включает в себя различные критерии.

Модель упругого деформирования однонаправленного слоя композитного материала на полимерной основе

Рассмотрим модель упругого деформирования однонаправленного слоя композитного материала. Пусть слой, отнесенный к координатам *a*, *b*, *c*, связан с направлением армирования. Введем ортогональные координаты α, β, γ и предположим, что ось a армированного слоя составляет с осью α угол φ (рис. 4).

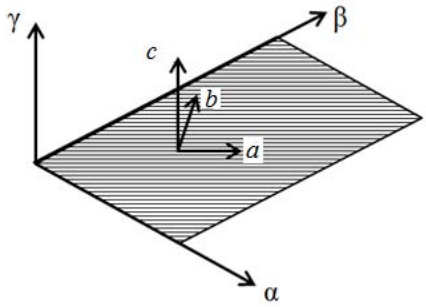


Рис. 4. Элемент однонаправленного слоя композита

Соотношения, связывающие напряжения и деформации в системах координат *a, b, c* и α, β, γ, имеют следующий вид:

$$\sigma_L = q^t \sigma q;$$
$$\varepsilon_L = q^t \varepsilon q.$$

Тензоры напряжений и деформаций σ и ε относятся к системе координат α, β, γ. Тензоры напряжений и деформаций с индексом *L* (σ_L и ε_L) относятся к системе координат *a*, *b*, *c*, связанной с направлением волокон.

Для рассматриваемой модели матрица преобразований *q* имеет следующий вид:

$$q = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Компоненты тензора напряжений в системе координат, связанной с направлением волокон (*a*, *b*, *c*), определяются как $\sigma_L^{n+1} = \sigma_L^n + \Delta\sigma_L$, где $\Delta\sigma_L$ – приращение компонент тензора напряжений.

Приращение компонент тензора напряжений в системе координат, связанной с направлением волокон (*a*, *b*, *c*), определяется согласно закону Гука:

$$\Delta\sigma_a = \overline{E_a}(\varepsilon_a + \upsilon_{ba}\varepsilon_b);$$
$$\Delta\sigma_b = \overline{E_b}(\varepsilon_b + \upsilon_{ab}\varepsilon_a);$$
$$\Delta\tau_{ab} = G_{ab}\varepsilon_{ab};$$
$$\Delta\tau_{ac} = G_{ac}\varepsilon_{ac};$$
$$\Delta\tau_{bc} = G_{bc}\varepsilon_{bc};$$
$$\overline{E_a} = \frac{E_a}{1-\nu_{ab}\nu_{ba}};$$
$$\overline{E_b} = \frac{E_b}{1-\nu_{ab}\nu_{ba}},$$

где ε_a, ε_b, $\varepsilon_{ab}, \varepsilon_{ac}, \varepsilon_{bc}$ – компоненты тензора деформаций в направлениях *a*, *b* и плоскостях *ab*, *ac* и *bc* соответственно.

Здесь имеет место условие симметрии упругих постоянных $E_a\upsilon_{bc} = E_b\upsilon_{ab}$.

Модель разрушения композитного материала

Модель разрушения композитного материала включает следующие критерии [8]:

– критерий растяжения матрицы $\sigma_b > 0$: $F_b^p = \left(\frac{\sigma_b}{\sigma_b^p}\right)^2 + \bar{\tau}$. Если $F_b^p > 1$, то константы материала E_b, G_{ab}, ν_{ba} и ν_{ab} полагаются равными нулю.

– критерий сжатия матрицы $\sigma_b < 0$: $F_b^{сж} = \left(\frac{\sigma_b}{2\sigma_{ab}^*}\right)^2 + \left[\left(\frac{\sigma_b^{сж}}{2\sigma_{ab}^*}\right)^2 - 1\right]\frac{\sigma_b}{\sigma_b^{сж}} + \bar{\tau}$. Если $F_b^{сж} > 1$, то константы материала E_b, ν_{ba} и ν_{ab} полагаются равными нулю.

– критерий растяжения вдоль волокон $\sigma_a > 0$: $F_a^p = \left(\frac{\sigma_a}{\sigma_a^p}\right)^2 + \bar{\tau}$. Если $F_a^p > 1$, то константы материала E_a, E_b, G_{ab}, ν_{ab} и ν_{ba} полагаются равными нулю.

В представленных критериях $\bar{\tau}$ – сдвиговый член, $\bar{\tau} = \frac{\frac{\tau_{ab}^2}{2G_{ab}} + \frac{3}{4}\alpha\tau_{ab}^4}{\frac{\sigma_{ab}^{*2}}{2G_{ab}} + \frac{3}{4}\alpha\sigma_{ab}^{*4}}$; α – параметр сдвигового напряжения для нелинейного члена.

Если для элемента в одном из слоев выполняется один из вышеперечисленных критериев разрушения, то напряжения в этом слое рассчитываются с учетом новых значений констант. Элемент считается разрушенным, если критерии разрушения выполнены во всех слоях, либо если выполнен критерий растяжения вдоль волокон.

Улучшенная модель композитного материала

Улучшенная модель композитного материала отличается от описанной выше модели только моделью разрушения. Данная модель включает следующие критерии разрушения:

– критерий растяжения матрицы $\sigma_b > 0$: $F_b^{\text{р}} = \left(\frac{\sigma_b}{\sigma_b^{\text{р}}}\right)^2 + \left(\frac{\sigma_{ab}}{\sigma_{ab}^{\text{р}}}\right)^2$. Если $F_b^{\text{р}} > 1$, то константы материала E_b, G_{ab}, v_{ba} и v_{ab} полагаются равными нулю.

– критерий сжатия матрицы $\sigma_b < 0$: $F_b^{\text{сж}} = \left(\frac{\sigma_b}{2\sigma_{ab}^{\text{р}}}\right)^2 + \left[\left(\frac{\sigma_b^{\text{сж}}}{2\sigma_{ab}^{\text{р}}}\right)^2 - 1\right]\frac{\sigma_b}{\sigma_b^{\text{сж}}} + \left(\frac{\sigma_{ab}}{\sigma_{ab}^{\text{р}}}\right)^2$. Если $F_b^{\text{сж}} > 1$, то константы материала E_b, v_{ba} и v_{ab} полагаются равными нулю.

критерий растяжения вдоль волокон $\sigma_a > 0$: $F_a^{\text{р}} = \left(\frac{\sigma_a}{\sigma_a^{\text{р}}}\right)^2 + \beta\left(\frac{\sigma_{ab}}{\sigma_{ab}^{\text{р}}}\right)$. Если $F_a^{\text{р}} > 1$, то константы материала v_{ba} и v_{ab} полагаются равными нулю.

критерий сжатия вдоль волокон $\sigma_a < 0$: $F_a^{\text{сж}} = \left(\frac{\sigma_a}{\sigma_a^{\text{сж}}}\right)^2$. Если $F_a^{\text{сж}} > 1$, то константы материала E_a, v_{ba} и v_{ab} полагаются равными нулю.

В представленных критериях β – весовой множитель сдвига при растяжении волокон.

Если для элемента в одном из слоев выполняется один из вышеперечисленных критериев разрушения, то напряжения в этом слое пересчитываются с учетом новых значений констант. Элемент считается разрушенным, если:

– критерии разрушения выполнены во всех слоях;

– если значение максимальной деформации при растяжении вдоль волокон dfailt>0 и значение хотя бы одной из компонент тензора деформаций превышает значение соответствующей величины: максимальной деформации при растяжении или сжатии матрицы, максимальной деформация при сдвиге, максимальной деформация при растяжении вдоль волокон, максимальной деформация при сжатии волокна (параметры модели);

– если значение максимальной деформации при растяжении вдоль волокон dfailt=0 и выполнен критерий растяжения вдоль волокон (после выполнения данного критерия происходит постепенное повреждение элемента с последующим его удалением).

В улучшенной модели разрушения имеется возможность использовать дополнительно:

– коэффициент разупрочнения для предела прочности волокна при растяжении ($fbrt$) после разрушения матрицы при сжатии. Если $F_b^{\text{сж}} > 1$, то $\sigma_a^{\text{р}} = \sigma_a^{\text{р}} * fbrt$.

– коэффициент уменьшения прочности волокна при сжатии (Y_{cfac}) после разрушения матрицы при сжатии. Если $F_b^{\text{сж}} > 1$ то $\sigma_a^{\text{сж}} = \sigma_b^{\text{сж}} * Y_{cfac}$.

Тестовые расчеты

Нагружение многослойной пластины

Цель данного теста – проверить поведение материала до момента разрушения. До начала разрушения композитный материал ведет себя как упругий. В представленной задаче угол направления волокон во всех слоях 0°, кроме того заданы одинаковые свойства материала в продольном и поперечном направлении, поэтому результаты расчета должны быть аналогичны результатам расчета, в котором каждый из слоев описывается с помощью модели изотропного упругого материала.

Постановка задачи приведена в [11]. Рассматривается задача о нагружении многослойной пластины. На рис. 5 схематично изображено расположение слоев задачи.

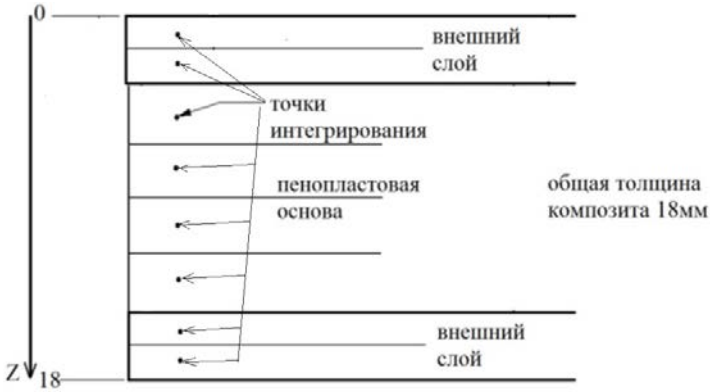


Рис. 5. Схематическая геометрия задачи

Задача представляет собой прямоугольную область 150×5 мм, толщина оболочки 18 мм. Счетная область разбивалась следующим образом: задавалось 30 элементов по оси X и 1 элемент по оси Y. В расчете использовался оболочечный элемент с 8-ю точками интегрирования по толщине и сдвиговым множителем равным 1. Толщина каждого из слоев 2.25 мм.

Для моделирования свойств материалов использовалась модель композитного материала. Физико-механические характеристики материалов следующие.

Внешний слой: начальная плотность ρ_0 = 2.7e-06 кг/мм^3, модули Юнга $E_a = E_b$ = 73.4 ГПа, коэффициент Пуассона ν_{ba} = 0.32, модули сдвига $G_{ab} = G_{bc} = G_{ca}$ = 27.8 ГПа.

Внутренний слой: начальная плотность ρ_0 = 6.3e-07 кг/мм3, модули Юнга $E_a = E_b$ = 0.286 ГПа, коэффициент Пуассона ν_{ba} = 0.3, модули сдвига $G_{ab} = G_{bc} = G_{ca}$ = 0.11 ГПа.

Левый конец оболочки жестко закреплен. Верхняя и нижняя границы – свободные поверхности.

К поверхности оболочки приложено давление, возрастающее от 0 в начальный момент времени, до 0.015 ГПа к моменту окончания расчета t = 5 мс.

На рис. 6 приведена зависимость смещения по оси Z правой границы от времени. Результаты расчета с моделью композитного материала сравниваются с результатами расчета с моделью изотропного упругого материала. Разрушения в данной задаче не происходит. Как видно из рис. 6, результаты расчетов с моделью композитного материала согласуются с результатами расчета с моделью изотропного упругого материала.

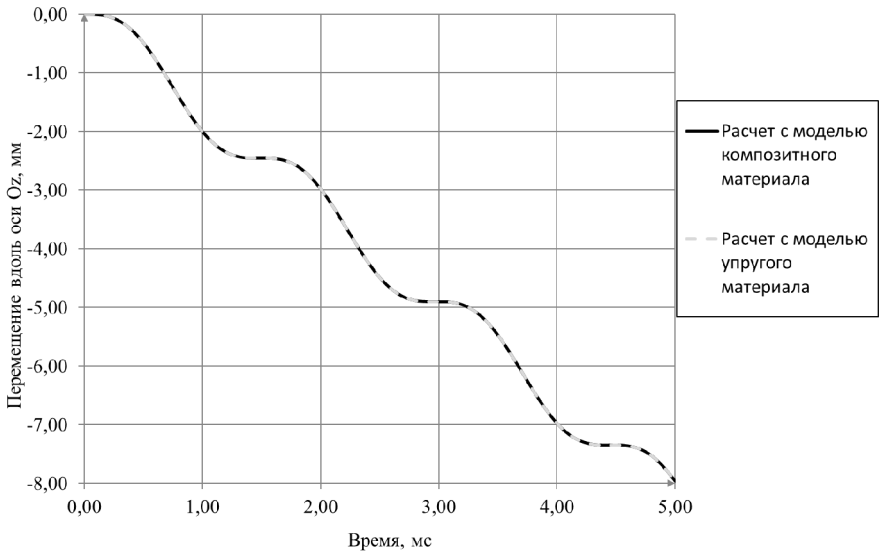


Рис. 6. Зависимость смещения по оси Z правой границы от времени

Растяжение вдоль волокон

Цель следующих тестов – проверить работоспособность реализованных алгоритмов расчета напряжений и критериев разрушения, а также принципов моделирования поведения поврежденного материала.

Постановка задачи взята из работы [12]. На рис. 7 показана геометрия тестовой задачи, представляющая собой квадратный элемент 0.01×0.01 см. В расчете использовался оболочечный элемент с одной точкой интегрирования по толщине и сдвиговым множителем, равным 1.

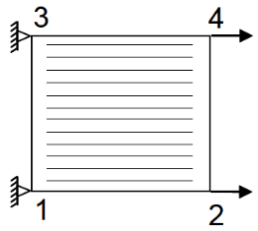


Рис. 7. Начальная геометрия задачи о растяжении вдоль волокон, закрепленные узлы и приложенная нагрузка

Для моделирования свойств материала использовалась модель разрушения композитного материала. Физико-механические характеристики материала следующие: начальная плотность ρ_0 = 1.65 г/см^3, модули Юнга E_a = 80 ГПа, E_b = 15 ГПа, коэффициент Пуассона ν_{ba} = 0.3, модуль сдвига $G_{ab} = G_{ca} = G_{bc}$ = 11.5 ГПа. Параметры разрушения: σ_a^p = 0.8 ГПа.

Задавались следующие граничные условия (рис. 7): один конец элемента жестко закреплен. На другом конце задана постоянная скорость вдоль оси 0*x* 0.01 км/с.

На рис. 8 приведены результаты, полученные по ЛОГОС-ПА, в сравнении с результатами из работы [12]. Элемент растягивается в направлении волокон, пока напряжение в направлении волокон не достигнет максимального значения σ_a^p=0.8 ГПа. После этого элемент разрушается.

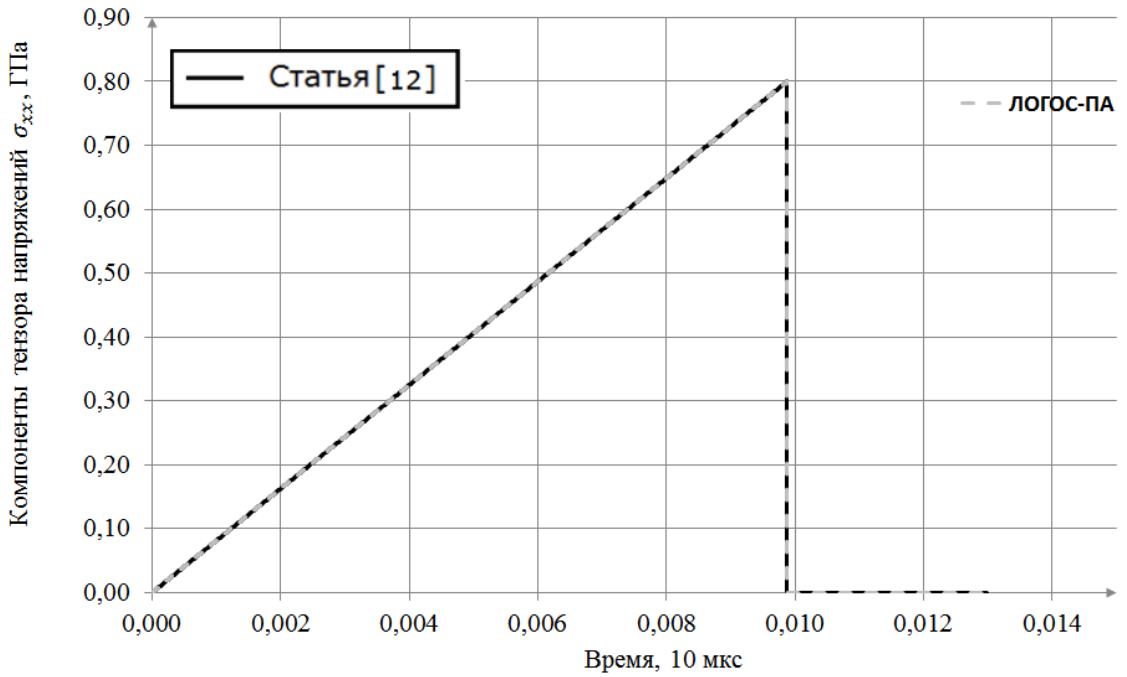


Рис. 8 Зависимость напряжений от времени в случае растяжения вдоль волокон

Растяжение вдоль волокон и сжатие матрицы

Постановка задачи взята из работы [12]. На рис. 9 показана геометрия тестовой задачи, представляющая собой квадратный элемент 0.01×0.01 см. В расчете использовался оболочечный элемент с одной точкой интегрирования по толщине и сдвиговым множителем, равным 1.

Для моделирования свойств материала использовалась улучшенная модель композитного материала. Физико-механические характеристики материала следующие: начальная плотность ρ_0 = = 1.65 г/см^3, модули Юнга E_a = 80 ГПа, E_b = 15 ГПа, коэффициент Пуассона ν_{ba} = 0.3, модули сдвига $G_{ab} = G_{ca} = G_{bc}$ = 11.5 ГПа. Параметры разрушения: σ_a^p= 0.8 ГПа, $\sigma_a^{сж}$= 0.5 ГПа, σ_b^p= 0.045 ГПа, $\sigma_b^{сж}$= 0.08 ГПа, σ_{ab}^*= 0.1 ГПа, $fbrt$ =0.5.

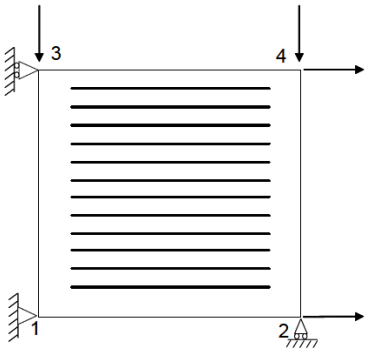


Рис. 9. Начальная геометрия задачи о растяжении вдоль волокон и сжатии матрицы, закрепленные узлы и приложенная нагрузка

Элемент растягивался вдоль направления волокон с постоянной скоростью 0.01 км/с и сжимался поперек направления волокон с постоянной скоростью 0.03 км/с.

На рис. 10 приведены результаты, полученные по ЛОГОС-ПА, в сравнении с результатами из работы [12]. Сначала выполняется критерий сжатия матрицы на момент времени t = 0.02 мкс. Далее, так как задан коэффициент разупрочнения для предела прочности волокна при растяжении $fbrt$, пересчитывается величина $\sigma_a^p = \sigma_a^p * fbrt = 0.4$ ГПа. Элемент растягивается в направлении волокон, пока напряжение в направлении волокон не достигнет значения σ_a^p = 0.4 ГПа. После этого происходит постепенное повреждение элемента, с последующим его удалением.

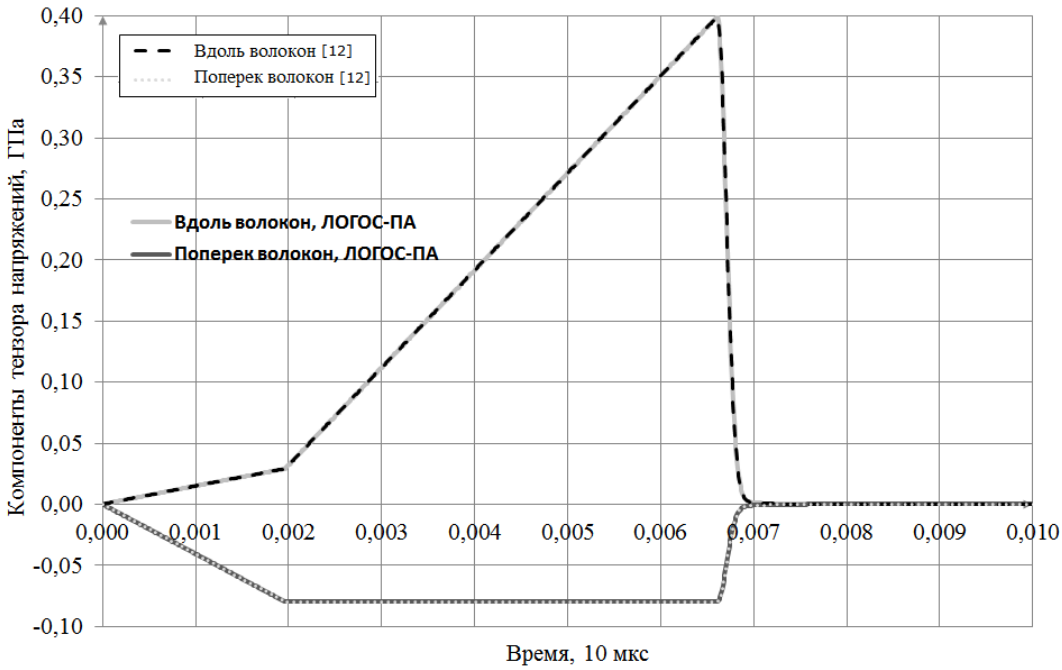


Рис. 10. Зависимость напряжений от времени в случае растяжения вдоль волокон и сжатия матрицы

Задача о соударении сферического ударника с композитной пластиной

Постановка задачи взята из работы [13]. Начальная геометрия задачи приведена на рис. 11 и представляет собой пластину и сферический ударник. Пластина размером 125×125 мм является многослойной и состоит из оболочечных элементов. Толщина пластины 2.28 мм. Пластина состоит из 16 слоев с последовательностью укладки $[45°/0°/-45°/90°]_{2s}$, толщина каждого из слоев 0.1425 мм. Диаметр сферического ударника 12.7 мм, начальная скорость ударника 1.482 м/с. Ударник состоит из объемных элементов.

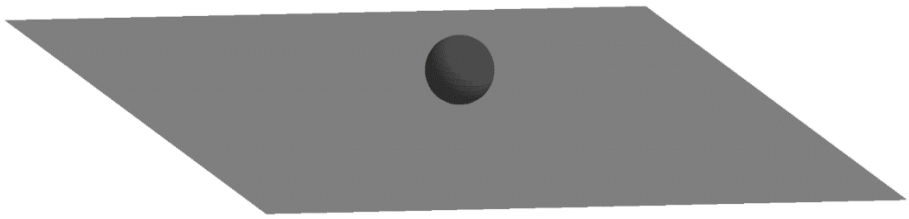


Рис. 11. Начальная геометрия задачи

Счетная область разбивалась следующим образом: в пластине задавалось 63 элемента по оси X и 63 элемента по оси Y (3969 элементов). Ударник состоит из 7000 объемных элементов.

Для моделирования свойств материала ударника использовалась упругопластическая модель с кинематическим упрочнением [8]. Для моделирования свойств материала пластины использовалась улучшенная модель композитного материала. Физико-механические характеристики материалов взяты из работы [13].

Граничные условия задавались следующим образом: на левой и правой границах пластины (рис. 11), ограничивались все поступательные и вращательные степени свободы, на верхней и нижней границах запрещались только вращения вокруг оси 0Y.

Проводилось два варианта расчета: в первом случае пластина предварительно не нагружена, во втором случае пластина предварительно нагружена растягивающей силой.

Моделирование предварительного нагружения пластины проводилось следующим образом: сначала рассчитывались деформированное состояние пластины под действием давления, возрастающего от 0 в начальный момент времени, до 0.525536 ГПа при t = 3 мкс. Давление прикладывалось к левой и правой границам пластины. Полученные значения перемещений и напряжений использовались в качестве начальных условий для решения задачи о соударении ударника с пластиной.

На рис. 12 приведено поле смещений по оси 0z на момент времени t = 2.5 мкс в расчете без предварительного нагружения пластины. На рис. 13 и 14 приведены зависимости силы контакта от времени в зоне соударения пластины с ударником, полученные экспериментально [13] и по ЛОГОС-ПА, без предварительного нагружения и в случае с предварительным нагружением пластины растягивающей силой соответственно.

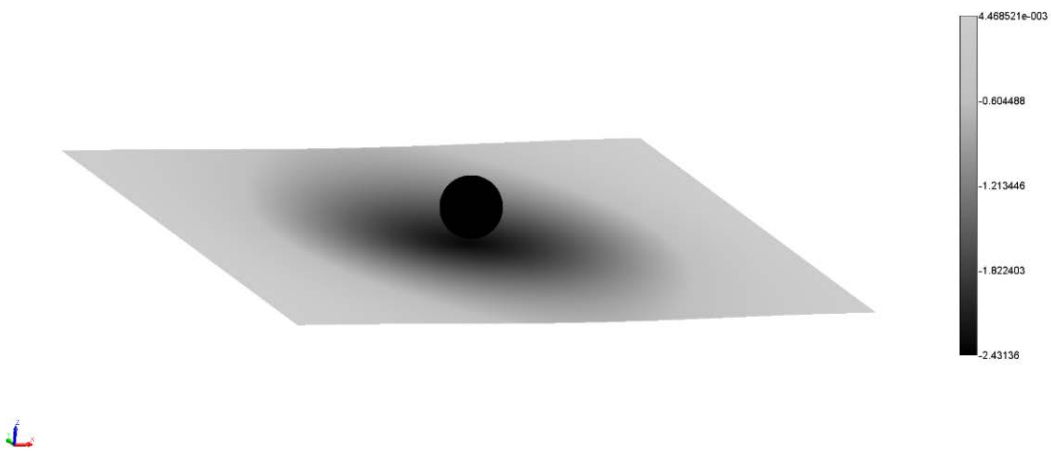


Рис. 12. Поле смещений по оси 0z на момент времени t = 2.5 мкс

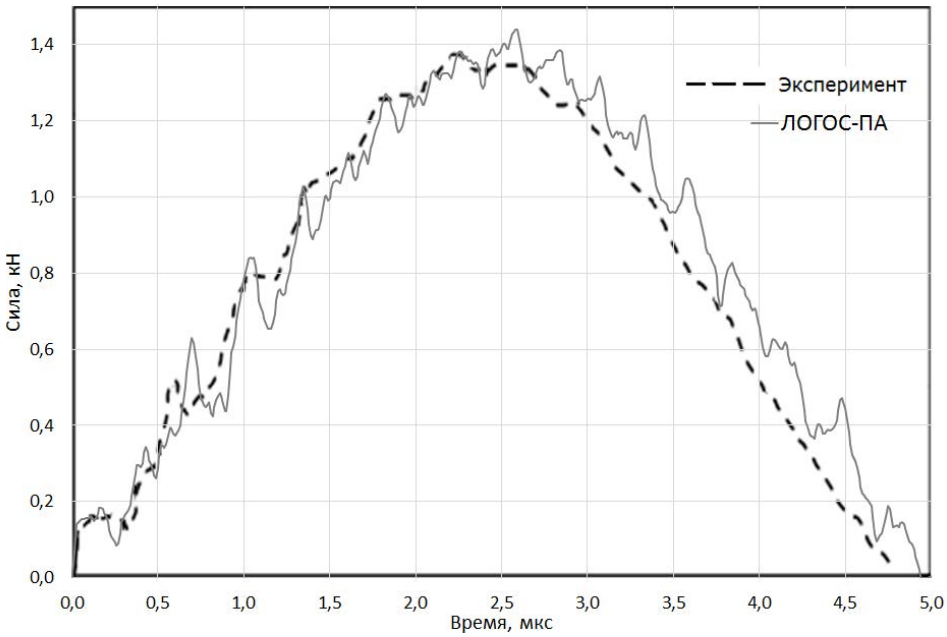


Рис. 13. Зависимость силы контакта без предварительного нагружения

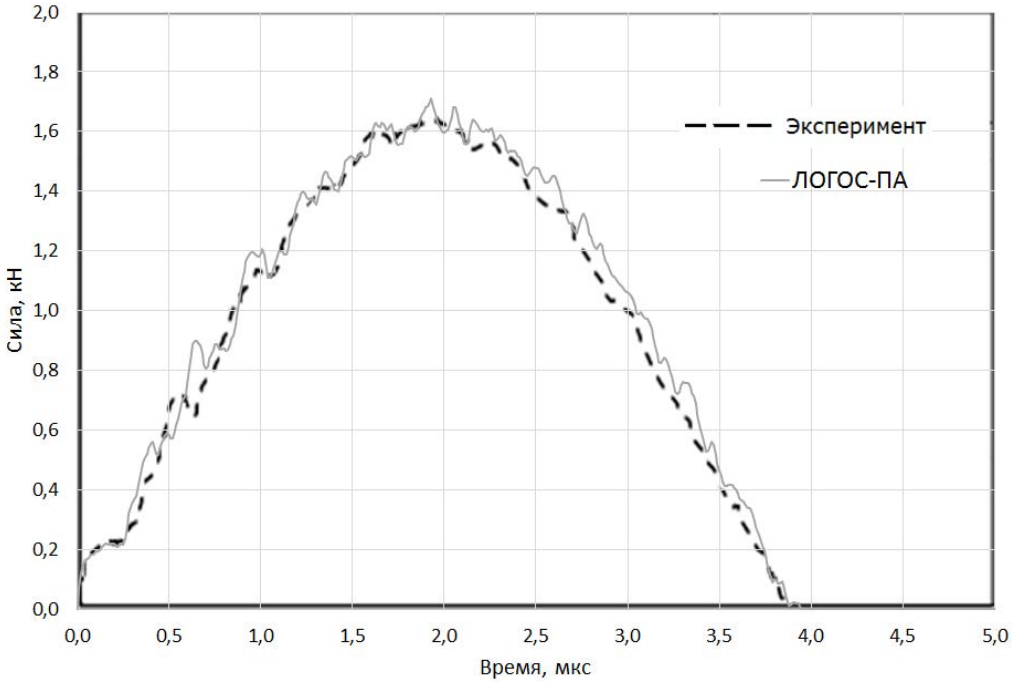


Рис. 14. Зависимость силы контакта от времени в случае с предварительным нагружением пластины растягивающей силой

Из рис. 13 и 14 видно, что результаты расчетов по ЛОГОС-ПА согласуются с результатами экспериментов.

Задача о соударении полусферического ударника с композитной пластиной

Постановка задачи взята из работы [14]. Рассматривается задача о деформировании композитной пластины при взаимодействии с полусферическим ударником. Образец представляет собой прямоугольную композитную пластину с размерами 150×100 мм и последовательностью укладки

слоев $[45_4/0_4/-45_4/90_4]_s$, помещенную на плоскую опору с размерами 300×200 мм и прямоугольным вырезом 112×75 мм. Пластина зафиксирована с помощью направляющих штифтов и резиновых зажимов (рис. 15).

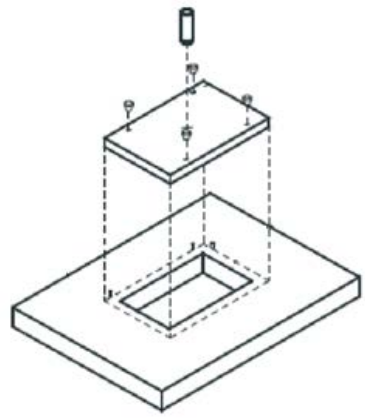


Рис. 15. Расположение пластины и опоры в образце

Начальная геометрия задачи приведена на рис. 16. Все детали моделируются с помощью элементов сплошной среды. Пластина является многослойной и состоит 32 слоев с последовательностью укладки $[45_4/0_4/-45_4/90_4]_s$. Толщина пластины 8 мм. Характерный размер конечного элемента, используемый для моделирования образца, составляет ~1.6×1.6 мм и уменьшается в зоне контакта с ударником до ~0.4×0.4 мм (480000 элементов). Опора состоит из ~6500 элементов. Диаметр наконечника полусферического ударника 8 мм, длина 58 мм. Ударник состоит из ~46400 элементов. Рассматриваются два случая подлета ударника к композитному образцу со скоростями $V_0 = 3.38$, 2.78 м/с.

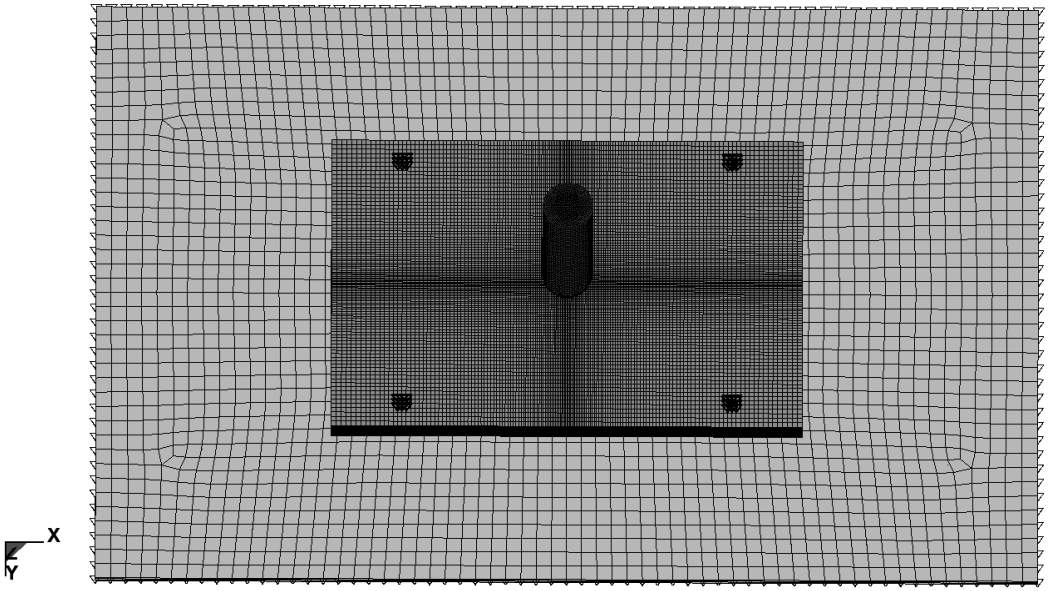


Рис. 16. Начальная геометрия задачи о соударении полусферического ударника с композитной пластиной, закрепленные узлы

Опора жестко закреплена, также закреплены узлы в зонах расположения направляющих штифтов (рис. 16).

Для моделирования свойств материала ударника использовалась упругопластическая модель с кинематическим упрочнением [8]. Для моделирования свойств материала пластины использовалась модель композитного материала. Для моделирования свойств материала опоры использовалась модель твердого материала. Физико-механические характеристики материалов взяты из работы [14].

На рис. 17–18 приведена зависимость силы контакта от времени в зоне соударения пластины с ударником, полученная экспериментально [14] и по ЛОГОС-ПА для разных начальных скоростей ударника. Как видно из рисунков, результаты расчетов и эксперимента качественно согласуются друг с другом.

В таблице приведены максимальные значения контактной силы для всех расчетных случаев в сравнении с результатами эксперимента [14]. Как видно из таблицы, наибольшее отличие от эксперимента не превышает 2 %.

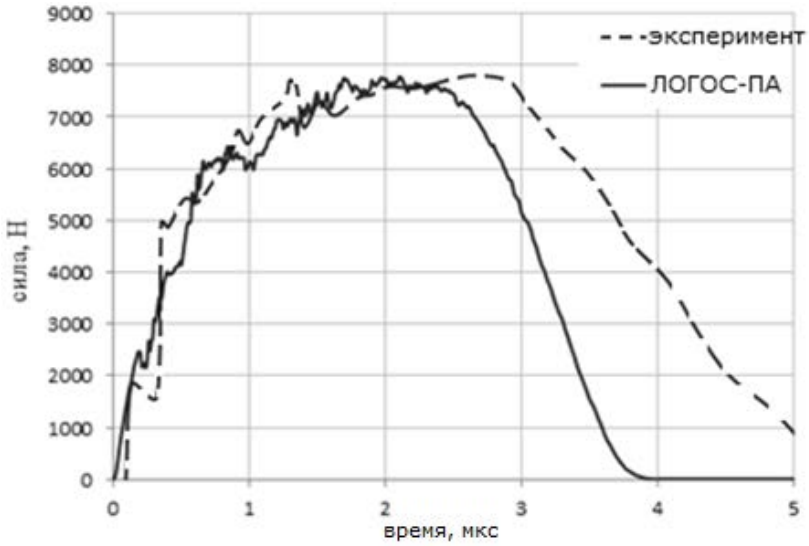


Рис. 17. Зависимость силы контакта от времени для начальной скорости ударника $V_0 = 3.38$ м/с

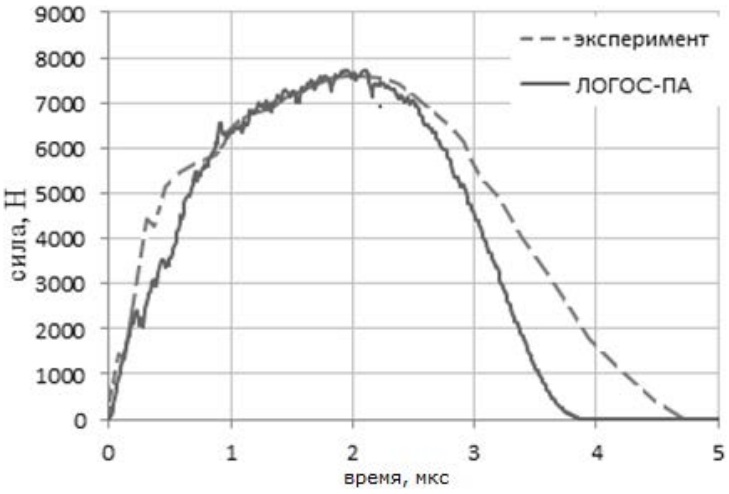


Рис. 18. Зависимость силы контакта от времени для начальной скорости ударника $V_0 = 2.78$ м/с

Таблица 1

Максимальные значения контактной силы

Начальная скорость V_0, м/с	Результаты эксперимента [13]	ЛОГОС-ПА	Δ, %
3.38	7800	7780	0.25
2.78	7600	7730	1.7

Задача о пробитии сферическим ударником пластины

Постановка задачи взята из работы [15]. Рассматривается задача о пробитии композитной пластины с размером 100×50 мм сферическим ударником диаметром 6.35 мм.

Начальная геометрия задачи приведена на рис. 19. Все детали моделируются с помощью элементов сплошной среды. Пластина является многослойной и состоит 6 слоев с последовательностью укладки $[0°/90°]_s$. Толщина пластины 1.95 мм. Характерный размер конечного элемента, используемый для моделирования пластины, составляет ~0.4×0.4 мм (187500 элементов). Ударник

состоит из ~40000 элементов. Размер элемента на поверхности шарика близок к размеру элемента на поверхности пластины. Начальная скорость ударника 455 м/с.

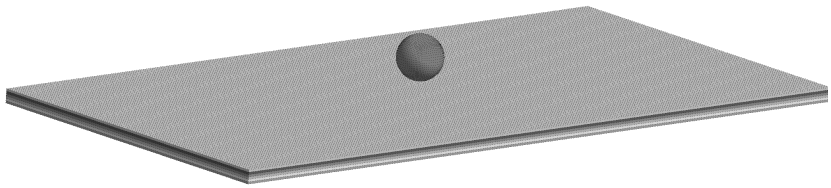


Рис. 19. Начальная геометрия задачи о пробитии сферическим ударником композитной пластины

Левый и правый края пластины жестко закреплены (рис. 19).

Для моделирования свойств материала ударника использовалась упрощенная модель Джонсона–Кука [8]. Для моделирования свойств материала пластины использовалась модель композитного материала. Физико-механические характеристики материалов взяты из работы [15].

На рис. 20 и 21 приведена картина деформирования и разрушения ударника на моменты времени t = 0.01 мкс и t = 0.02 мкс. Как видно из рисунков, происходит пробитие пластины ударником. В результате эксперимента также было получено пробитие ударником пластины, остаточная скорость ударника составила V_r = 370 м/с [14]. Остаточная скорость ударника в расчете по ЛОГОС-ПА составил а V_r =386 м/с, отличие от экспериментальных данных не превышает 4.5 %.

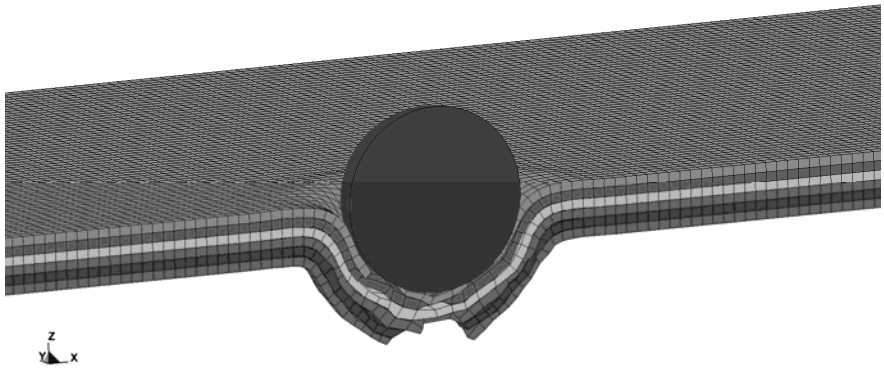


Рис. 20. Деформирование и разрушение пластины на момент времени t = 0.01 мкс (фрагмент, сечение плоскостью Y = 25 мм)

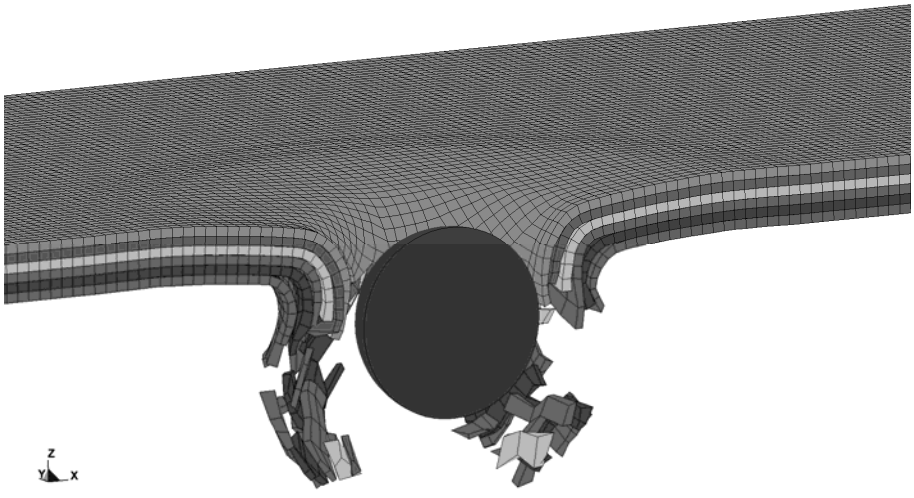


Рис. 21. Деформирование и разрушение пластины на момент времени t = 0.02 мкс (фрагмент, сечение плоскостью Y = 25 мм)

Заключение

В работе представлены методы моделирования композитных материалов, доступные в ПМ ЛОГОС-ПА при численном моделировании процесса динамического деформирования композитных конструкций. Приведено описание некоторых моделей композитных материалов, а именно моделей многослойного композитного материала с учетом разрушения. Модель многослойного композитного материала основана на представлении композита как пакета слоев, с различными углами укладки друг относительно друга и использовании теории оболочек.

В работе представлены результаты численных расчетов ряда постановок конструкций с композитными материалами, проведенных с использованием пакета программ ЛОГОС. Приведенные результаты подтверждают работоспособность реализованного функционала для моделирования конструкций из композитных материалов при решении динамических задач прочности.

Литература

1. Циберев К. В., Авдеев П. А., Артамонов М. В., Борляев В. В., Величко С. В., Волков А. Ю., Володина Н. А., Дьянов Д. Ю., Корсакова Е. И., Косарим С. С., Кулыгина О. Н., Мышкина И. Ю., Наумов А. О., Приставш М. М., Резвова Т. В., Резяпов А. А., Родионов А. В., Симонов Г. П., Спиридонов В. Ф., Стародубов С. В., Тарадай И. Ю., Филимонкин Е. А., Челаков А. А., Шувалова Е. В., и др. Пакет программ ЛОГОС. Обзор текущих возможностей для решения задач прочности // XIV международный семинар «Супервычисления и математическое моделирование»: Сборник научных трудов. Саров, 2013. С. 599–609.

2. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В., Наумова Е. И., Борляев В. В., Стародубов С. В., Шувалова Е. В., Медведкина М. В., Артемова Е. О., Челаков А. А., Казанцев А. В., Рябов А. А., Романов В. И., Куканов С. С. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. 2018. Вып. 1. С. 3–13.

3. Васильев В. В. Механика конструкций из композиционных материалов. – М.: Машиностроение, 1988.

4. Абросимов Н. А. Методика построения разрешающей системы уравнений динамического деформирования композитных элементов конструкций. Учебно-методическое пособие. – Н. Новгород: 2010.

5. Протасов В. Д. Механика в машиноведении композитных конструкций // Механика композитных материалов. 1987. С. 490–493.

6. Реслер И., Хардерс Х., Бекер М. Механическое поведение конструкционных материалов. Пер. с нем. Учебное пособие. – Долгопрудный: Издательский дом «Интеллект», 2011.

7. Скворцов Ю. В. Механика композиционных материалов. Конспект лекций по дисциплине. Самара: 2013.

8. Дьянов Д. Ю., Корсакова Е. И., Симонов Г. П., Циберев К. В., Шувалова Е. В. Результаты верификации моделей упругопластического деформирования и разрушения, реализованных в пакете программ ЛЭГАК-ДК // XII международный семинар «Супервычисления и математическое моделирование»: Сборник научных трудов. Саров, 2011. С. 163–173.

9. LS-DYNA theoretical manual [Electronic resource] / Livermore Software Technology Corporation. – Hallquist J.O., 2006. – Mode of Access: http:/www.lstc.com/download/manuals.

10. Композиционные материалы: справочник / Под ред. В. В. Васильева и Ю. М. Тарнопольского. – М.: Машиностроение, 1990.

11. [Electronic resource]. Mode of Access: https://www.dynasupport.com/howtos/material/composite-models.

12. Xiahua Zheng. Nonlinear strain rate dependent composite model for explicit finite element analysis: dissertation… PhD dissertation, University of Akron – Akron, Ohio, 2006.

13. Choi I. H., Kim I. G., Ahn S. M., Yeom C. H. Analytical and Experimental Studies on the Low-Velocity Impact Response and Damage of Composite Laminates under in-Plane Loads with Structural Damping Effects // Compos. Sci. Technol. 2010. Vol. 70 (10). P. 1513–1522.

14. Жихарев М. В. Влияние предварительного нагружения на баллистические свойства стеклопластиковых панелей // Вестник ЮУрГУ. Серия: «Машиностроение». Челябинск. 2017. Т. 17, № 4. С. 82–90.

15. Perillo G., Vedvik N. P., Echtermeyer A. T. Numerical analyses of low velocity impacts on composite. Advanced modelling techniques. SIMULIA Community Conference. NTNU, 2012.

"LOGOS" SOFTWARE PACKAGE. IMPLEMENTATION OF APPROACHES AND ALGORITHMS TO SIMULATE THE BEHAVIOR OF COMPOSITE MATERIALS

D. Yu. Dyanov, M. V. Medvedkina, E. I. Naumova, E. V. Shuvalova

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents the methods offered by the LOGOS software package for numerical simulations of dynamic deformation of composite structures. Some composite models are described, in particular, a multi-layer composite model including damage effects. The multi-layer composite model represents composites as a package of layers stacked at different angles relative to each other and uses the shell theory.

To demonstrate the performance of the implemented functions, the paper reports the results of numerical simulations of a number of structures with composites using the LOGOS software package.

Key words: LOGOS software package, LOGOS-SA software module, dynamic deformation, finite-element method, composite materials.

УДК 519.6
DOI 10.53403/9785951505071_2022_170

АЛГОРИТМЫ АВТОМАТИЧЕСКОГО ИСПРАВЛЕНИЯ ПЕРЕСЕЧЕНИЙ, РЕАЛИЗОВАННЫЕ В ГЕНЕРАТОРЕ ПОВЕРХНОСТНЫХ ТРЕУГОЛЬНЫХ СЕТОК В ПАКЕТЕ ПРОГРАММ «ЛОГОС»

Е. О. Евстифеева, Д. М. Панкратов, Т. В. Цалко

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе приведено описание реализованных алгоритмов для исправления пересечений и оптимизации поверхностной треугольной сетки в автоматическом режиме. Приведены результаты программной реализации алгоритмов, примеры.

Ключевые слова: генераторы сеток, препостпроцессор, пакет программ «ЛОГОС», генератор поверхностных треугольных сеток, пересечения треугольников.

Введение

В настоящее время в рамках работ по пакету программ «ЛОГОС» проводится развитие генератора поверхностной треугольной сетки, который совместно с генератором объемной сетки [1] проводит построение расчетной сеточной модели для задач аэрогидродинамики.

Построенная поверхностная сетка является исходными данными для генератора объемной сетки, который в свою очередь предъявляет требования к ее качеству. При наличии дефектов в поверхностной сетке не может быть построена корректная объемная сетка. Так, одной из критических ошибок поверхностной сетки, при наличии которой объемный генератор не будет запущен, является пересечение треугольников. В данной работе представлены автоматические способы устранения данной ошибки.

Пересечения в сетке могут появиться из различных источников. Во-первых, исходная модель уже может содержать критические ошибки, полученные в ходе построения тесселяции[1] CAD[2]-модели. Во-вторых, при задании размеров построения, не соответствующих габаритам модели или размерам ее особенностей (искривления, кромки, близкие поверхности и т. д.), ошибки могут появиться и в ходе построения сетки, как показано на рис. 1. Поэтому важно исправлять пересечения в автоматическом режиме как на начальном этапе, так и после перестроения сетки с заданным размером.

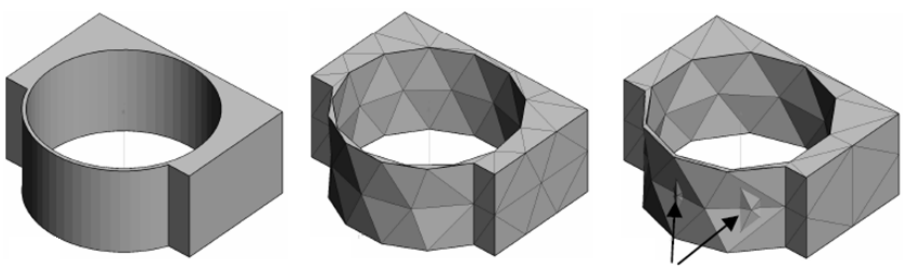


Рис. 1. Слева: исходная деталь; по центру: построенная сетка с подходящим размером; справа: сетка, построенная с неподходящим размером (пересечение близких поверхностей)

[1] Тесселяция – поверхностная сетка низкого качества (состоящая из узких вытянутых треугольников), полученная по модели в параметрическом представлении.

[2] CAD (Computer Aided Design) – система автоматизированного проектирования.

Алгоритмы исправления пересечений можно разделить на две группы: менее агрессивные алгоритмы, использующие локальные модификации над элементами сетки, и более агрессивные алгоритмы, применяющие удаление области треугольников с пересечениями с последующим заполнением образовавшегося отверстия новыми треугольниками.

Следует отметить, что разработанный инструмент для исправления пересечений нацелен на решение данной проблемы в локальных областях поверхностной сетки. В случае, когда исходная сетка представляет собой глобально несогласованную модель с большим количеством ошибок, таких как пересечения (особенно в зоне характерных кривых), несвязанные части или зазоры (отверстия), следует применять другие средства исправления модели [2].

В работе описаны реализованные структуры и алгоритмы для исправления пересечений, а также дополнительные алгоритмы для оптимизации полученного результата, в заключении приведены результаты с примерами.

Поиск пересечений треугольников

Для поиска пересекающихся треугольников в библиотеке генератора поверхностных треугольных сеток используется декомпозиция пространства модели при помощи структуры, известной как *k*-мерное дерево. Под пространством модели понимается габаритная коробка, ориентированная по координатным плоскостям и содержащая внутри себя все сеточные элементы. Результатом такой декомпозиции является набор прямоугольных параллелепипедов с гранями, параллельными координатным плоскостям. Эти блоки покрывают все пространство модели. Пример исходной треугольной сетки, взятой из открытого источника [3], и ее разбиения на блоки показан на рис. 2.

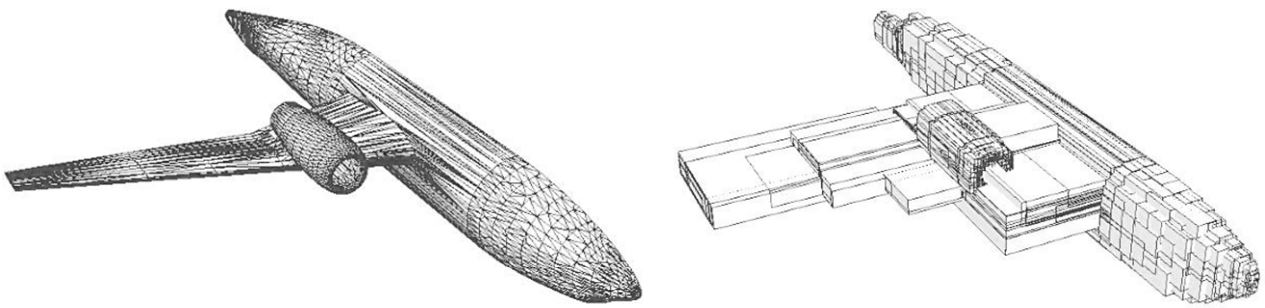


Рис. 2. Слева: пример треугольной сетки; справа: *k*-мерное дерево для сетки слева

К каждому из блоков приписаны все треугольники, находящиеся внутри этого блока или пересекающие его грани. Таким образом, пересекающиеся треугольники находятся в одном из блоков. Такое разделение пространства позволяет избежать глобального поиска пересечений треугольников «каждого с каждым», имеющим квадратичную сложность, и ограничиться проверкой треугольников только внутри блоков.

После того, как список всех пересекающихся треугольников был сформирован, запускается алгоритм исправления. Реализован он следующим образом: сначала выполняются локальные модификации пересекающихся элементов сетки, затем, если остались пересечения, запускается более агрессивное исправление с удалением проблемной области и заполнением ее новыми треугольниками

Первый этап исправления пересечений с помощью локальных модификаций

Первый подход с использованием локальных операций над треугольниками эффективно удаляет пересечения, полученные после построения тесселяции по CAD-модели с дефектами или после построения поверхностной сетки для модели с близкими поверхностями при заданном грубом разме-

ре[3], восстанавливая корректность поверхностной сетки. Реализованный метод принимает на вход треугольную сетку и итерационно применяет следующие локальные операции: переброс ребра между смежными треугольниками, смещение вершины или смещение треугольника [4], как показано на рис. 3.

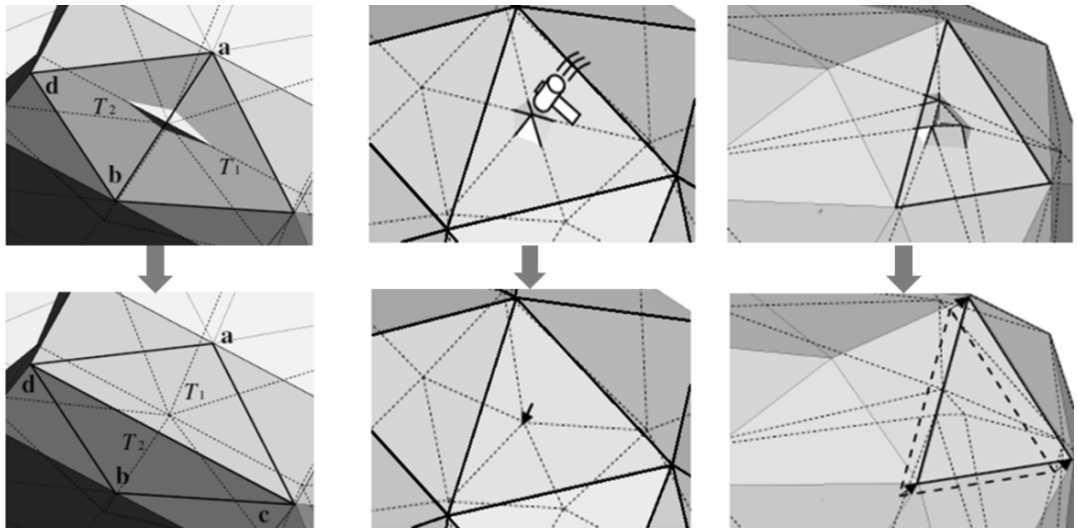


Рис. 3. Результат исправления пересечения с помощью локальных операций: переброс ребра между смежными треугольниками (слева), смещение вершины (по центру) и смещение треугольника (справа)

Каждая из этих операций меняет координаты вершин, инцидентных треугольникам с пересечениями, исходя из информации о соседстве и локации пересечения.

Несмотря на то, что локальные операции являются достаточно эффективным инструментом для сокращения количества пересечений, они не всегда могут гарантировать полное их устранение. В худшем случае они могут сгенерировать новые пересечения. Поэтому предложенный метод на каждой итерации должен удовлетворять двум критериям:

1. Количество пересечений снизилось.
2. Не образовалось новых треугольников с пересечениями.

Операция смещения вершины

Локальная операция смещения вершины сдвигает узел сетки и тем самым модифицирует инцидентные ей ребра, «протыкающие» другой треугольник, освобождая их от пересечений.

Исходя из того, что перемещаемая вершина может принадлежать нескольким ребрам, пересекающим треугольник, ее новое положение вычисляется как суперпозиция нескольких точек, пример представлен на рис. 4. Пусть ребро ab_n является пересекающим треугольник ребром ($n = 1,..,N$), инцидентным вершине *a*. Тогда перемещаемая вершина n вычисляется следующим образом:

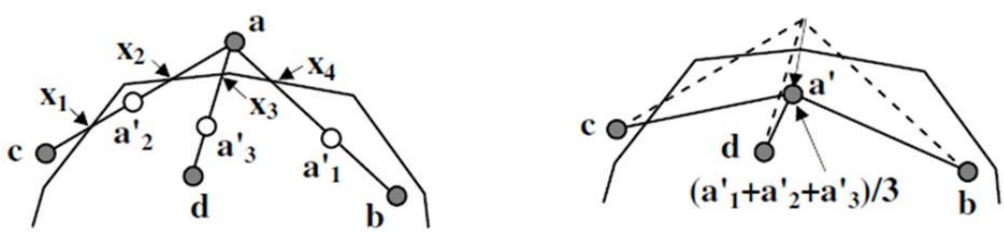


Рис. 4. Пример схематичного применения операции смещения узла

[3] Грубый размер – чрезмерно крупный размер треугольника. Построенная с таким параметром сетка сильно отклоняется от исходной поверхности.

а) если ребро ab_n пересекается с одним треугольником в точке *x*, то точка сдвига a_n' вычисляется по формуле (1):

$$u_n' = \frac{x+b_n}{2}, \tag{1}$$

б) если ребро ab_n пересекается с несколькими треугольниками в точках $x_1, x_2, .., x_k$, и x_1 и x_2 являются двумя точками пересечения, ближайшими к вершине *a*, то точка сдвига a_n' вычисляется по формуле (2):

$$a_n' = \frac{x_1+x_2}{2}, \tag{2}$$

Предложенный метод сдвигает вершину *a* в среднюю позицию найденных точек a_n' по формуле (3), если операция сдвига удовлетворяет критерию, описанному в предыдущем пункте.

$$a' = \sum a_n' / n. \tag{3}$$

Операция смещения треугольника

Операция смещения треугольника сдвигает треугольник по направлению его нормали. Рассмотрим схематично: предположим, что треугольник T_1 с вершинами t_1, t_2, t_3 пересекается с одним или более ребрами, и вершины $p_1, p_2, .., p_n$ принадлежат этим ребрам. Вектор n_1 – вектор единичной нормали для T_1 и ε – точность, представляющая собой расстояние между треугольником и вершиной, при котором пересечение не диагностируется.

Минимум и максимум знаковых расстояний от вершин $p_i,\ i = 1, .., n$ до T_1 обозначим как d_{min} и d_{max} соответственно и будем вычислять по формулам (4) и (5) соответственно.

$$d_{min} = min((p_i - t_1) \cdot n_1), \tag{4}$$
$$d_{max} = max((p_i - t_1) \cdot n_1), \tag{5}$$

Предложенный метод перемещает треугольник T_1 на две возможные позиции с его помощью переноса от начального положения по формулам (6) и (7).

$$n_1(d_{max} + \varepsilon). \tag{6}$$
$$n_1(d_{min} - \varepsilon). \tag{7}$$

Выбор значения ε, которое используется для вычисления величины смещения, зависит от минимального размера элементов исходной сетки и заданных размеров элементов, и представлено формулой (8).

$$\varepsilon = 0.05 \cdot min(L_{min}, Metric_{min}), \tag{8}$$

где L_{min} – минимальный размер ребра исходной сетки; $Metric_{min}$ – минимальный размер из заданных пользователем размеров.

Если критерию удовлетворяет лишь одна из найденных позиций, тогда треугольник T_1 смещается на эту позицию. Если критерию удовлетворяют обе найденные позиции, то выбирается та из них, при сдвиге в которую происходит наименьшее изменение объема модели. Пример применения операции смещения треугольника приведен на рис. 5.

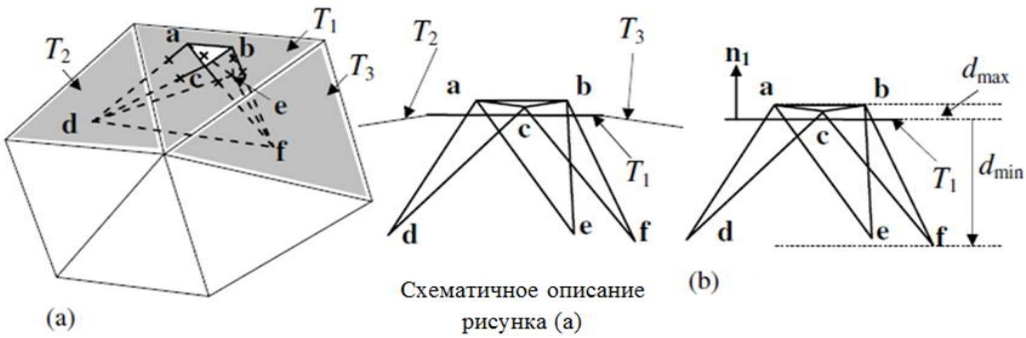


Рис. 5. Пример схематичного применения операции смещения треугольника

Выбор локальной операции

Результат итерационного алгоритма исправления пересечений с помощью локальных операций зависит от порядка их выполнения. Если операции были применены в неподходящем порядке, то они могут либо сильно исказить сетку, либо просто не добиться сокращения количества пересечений.

Поэтому до выбора операции на каждом шаге производится предварительная оценка изменения объема модели, вызванного применением этой операции, и выбирается та операция, которая влечет наименьшее изменение объема.

Изменение объема после переброса ребра между смежными треугольниками вычисляется как объем тетраэдра, как представлено на рис. 6.

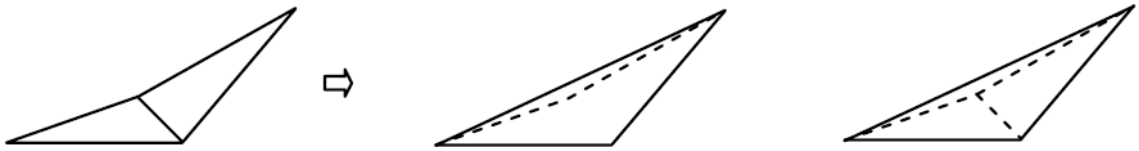


Рис. 6. Изменение объема в результате применения операции переброса ребра смежных треугольников

Изменение объема, вызванное смещением вершины, вычисляется как сумма объемов тетраэдров, как представлено на рис. 7.

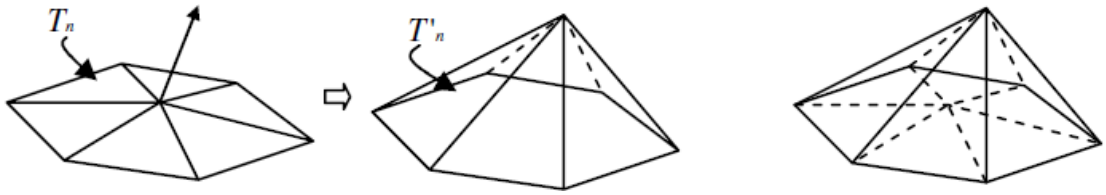


Рис. 7. Изменение объема в результате применения операции смещения вершины

Изменение объема, вызванное смещением треугольника, вычисляется как сумма объемов тетраэдров, пирамид с четырехугольным основанием и объема треугольной призмы, как представлено на рис. 8.

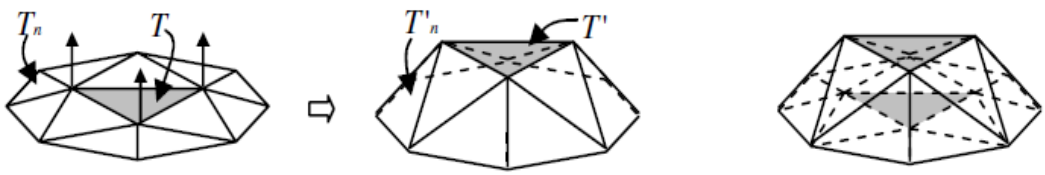


Рис. 8. Изменение объема в результате применения операции смещения треугольника

По результатам проведенных экспериментов максимальное количество итераций ограничено пятнадцатью, а также было установлено, что даже при большом количестве пересекающихся треугольников образуется относительно небольшое количество проблемных областей, и часто при исправлении одного пересечения «распутывается» целая область пересекающихся треугольников.

Пример работы алгоритма на первом этапе

Пример исправления пересечения типа «прокол» с помощью локальной операции смещения вершины представлен на рис. 9.

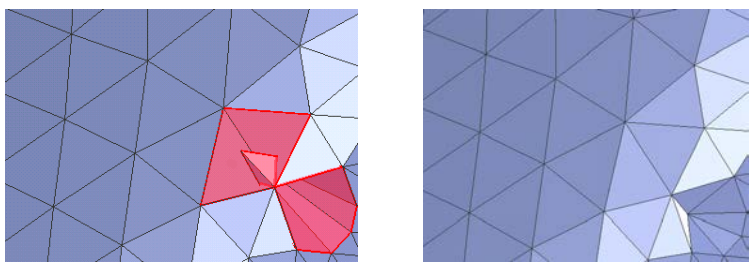


Рис. 9. Пересечение типа «прокол» (слева) исправлено с помощью операции смещения вершины (справа)

Второй этап исправления пересечений

После локальных операций в сетке все еще могут присутствовать пересечения треугольников. В этом случае используется более агрессивный подход для их исправления, который заключается в итерационном повторении следующих шагов:

1) наращивание проблемной области за счет добавления треугольников окружения (пример для трех итераций наращивания приведен на рис. 10, порядок кольца соседей соответствует номеру итерации);

2) удаление из сетки треугольников, входящих в область с шага 1, формирование контура из свободных ребер вокруг образованного отверстия (пример удаленной области с выделенным контуром приведен на рис. 11 слева сверху);

3) заполнение образованного отверстия новыми треугольниками с использованием только вершин контура с шага 2;

4) адаптивное измельчение треугольников в областях, построенных на шаге 3;

5) сглаживание сетки в областях, построенных на шаге 4;

6) для каждой области проводятся проверки на недопустимое изменение площади поверхности, на количество пересечений, а также на появление новых пересечений. Если все проверки прошли успешно, то применяем к сетке в области изменения, иначе возвращаем исходные треугольники.

7) если в сетке не осталось пересечений или достигнуто максимальное количество итераций, то завершаем алгоритм, иначе переходим на шаг 1.

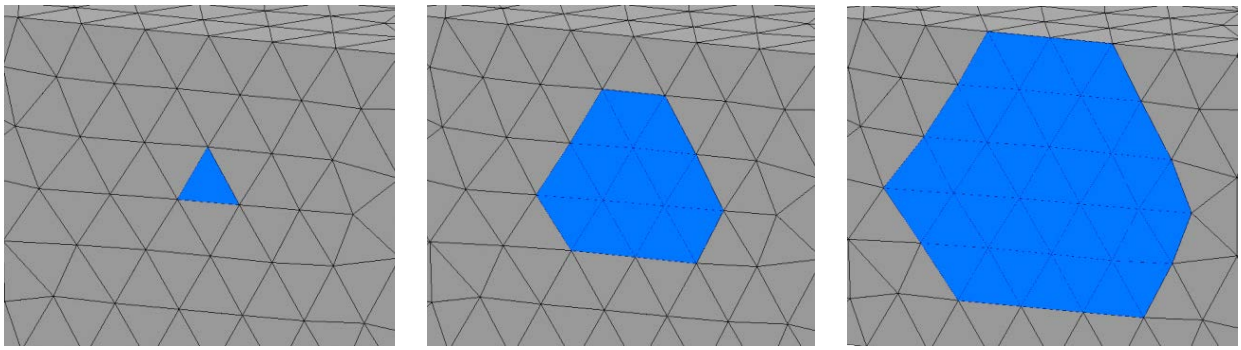


Рис. 10. (слева) Кольцо соседей нулевого порядка (выделенный треугольник); (по центру) кольцо соседей первого порядка; (справа) кольцо соседей второго порядка

Алгоритм затягивания открытого контура

На шаге 3 затягиваются односвязные контуры без самопересечений и самокасаний. Таким образом, затягивание запускается на каждой из найденных замкнутых цепочек открытых ребер. Одним из наиболее простых и эффективных алгоритмов закрытия контуров является метод «Ear cutting» [5, 6].

Изначально все вершины помечаются как активные. На каждой итерации производится поиск наиболее подходящей для построения треугольника тройки соседних вершин контура V_{i-1}, V_i, V_{i+1}. Критерием является минимальная величина угла α_i в вершине $\vec{V}_i$. Угол α_i вычисляется между ребрами (V_{i-1}, V_i) и (V_i, V_{i+1}). После вставки треугольника с вершинами V_{i-1}, V_i, V_{i+1} вершина V_i становится неактивной. Построение продолжается, пока в контуре есть более двух активных вершин. Схематичная работа алгоритма приведена на рис. 11.

В результате работы описанного выше алгоритма получается корректная сетка без пересечений, однако, при использовании его на трехмерных контурах могут возникать проблемы, выраженные в «подгибах» и треугольниках низкого качества. Для повышения качества получаемой сетки проводится этап оптимизации путем переброса диагоналей смежных треугольников, если в результате этого минимизируется суммарная площадь пары треугольников.

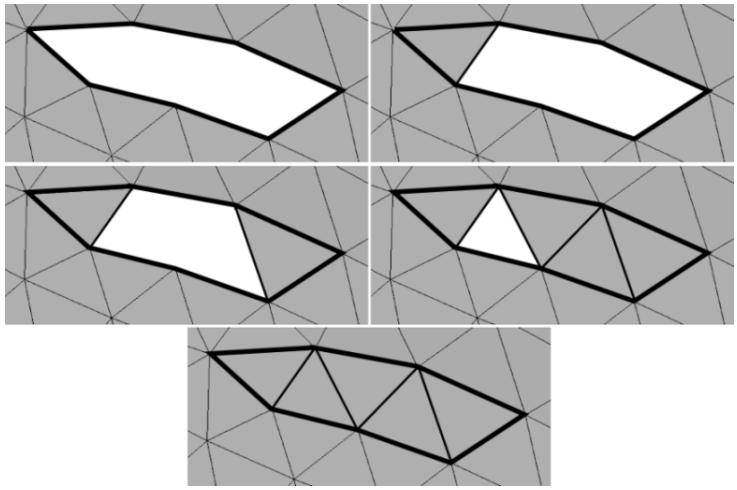


Рис. 11. Этапы работы алгоритма затягивания открытого контура

Вследствие того, что образованный свободный контур на шаге 3 затягивается без добавления новых вершин, может произойти значительное изменение кривизны поверхности в измененной области. Для восстановления формы поверхности были реализованы алгоритмы адаптивного измельчения и сглаживания сетки в области (шаги 4 и 5 алгоритма), описанные далее.

Адаптивное измельчение сетки

После выполнения итерации описанного выше алгоритма размеры новых треугольников в большинстве случаев не будут соответствовать размерам в окрестности измененной области. Для того чтобы размеры в затянутой области и окружающей сетке совпадали, был добавлен алгоритм адаптивного измельчения сетки [7].

Идея алгоритма заключается в том, чтобы вычислить значения длин ребер окружения, инцидентных вершинам, образующим контур измененной области, и применить данные величины для треугольников новой области. Для уменьшения длины ребер треугольники области разбиваются путем добавления новой вершины, и затем выполняется операция релаксации внутренних ребер области для получения сетки Делоне. Под релаксацией ребра двух смежных треугольников, инцидентных данному ребру, подразумевается проверка того, что каждая из двух не совпадающих вершин данных треугольников лежит вне описанной окружности смежного треугольника (рис. 12), и если условие не выполнено, перебросить ребро, как показано на рис. 13.

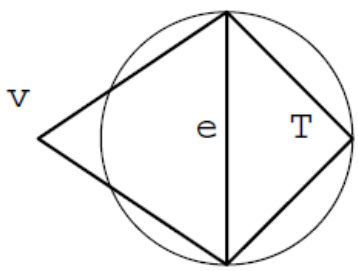


Рис. 12. Пример смежных треугольников, для которых не требуется релаксации общего ребра e

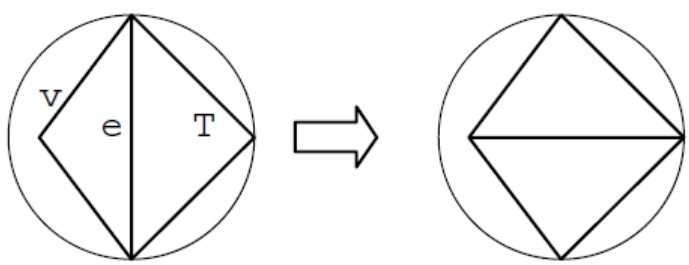


Рис. 13. Пример смежных треугольников, для которых требуется релаксация общего ребра e

Описание алгоритма:

1. Для каждой вершины контура v_i, вычисляется масштабирующий коэффициент $\sigma(v_i)$, как средняя длина ребер, инцидентных v_i в сетке окружения. Выполняется инициализация внутренней сетки области начальными треугольниками после алгоритма затягивания.

2. Для каждого треугольника (v_i, v_j, v_k) в сетке области вычисляется барицентр v_c и соответствующий масштабирующий коэффициент:

$$\sigma(v_c) = \frac{(\sigma(v_i)+\sigma(v_j)+\sigma(v_k))}{3}. \quad (9)$$

3. Для $m = i, j, k$ если $\alpha\|v_c - v_m\| > \sigma(v_c)$ и $\alpha\|v_c - v_m\| > \sigma(v_m)$, то треугольник (v_i, v_j, v_k) заменяется треугольниками (v_c, v_j, v_k), (v_i, v_c, v_k) и (v_i, v_j, v_c) в сетке области и выполняется релаксация ребер (v_i, v_j), (v_i, v_k) и (v_j, v_k).

Если на шаге 2 не было образовано новых треугольников, сетка области построена.

Релаксация всех внутренних ребер области.

Если на шаге 4 не было переброшено ни одного ребра, переходим на шаг 2. Иначе переходим на шаг 4.

По результатам проведенных испытаний значение коэффициента плотности распределения α было выбрано равным $\sqrt{6}$, так как размеры треугольников вне и внутри области близки при использовании данного значения.

Сглаживание сетки

После алгоритма адаптивного измельчения треугольников в перестроенных областях запускается сглаживание для получения гладкой поверхности, согласованной с сеткой окружения. Процесс сглаживания основан на решении задачи квадратичной минимизации [8]. Минимизация квадратичного функционала достигается решением системы линейных уравнений, составленной для всех внутренних вершин области без вершин контура, методом сопряженных градиентов.

В качестве минимизируемого функционала было выбрано уравнение Эйлера–Лагранжа (10), использующее оператор Лапласа 1 и 2-го порядка:

$$-\Delta_S d + \Delta_S^2 d = 0, \; d: S \to R^3. \quad (10)$$

Для кусочно-линейной функции получения координат вершин $f: S \to R^3$, определенной на треугольной сетке S, дискретный оператор Лапласа-Бельтрами в вершине v_i вычисляется по формуле (11):

$$\Delta_S f(v_i) = w_i \sum_{v_j \in N_1(v_i)} w_{ij} (f(v_j) - f(v_i)), \quad (11)$$

где $v_j \in N_1(v_i)$ – кольцо соседей первого порядка для вершины v_i (рис. 14).

Лапласиан более высокого порядка вычисляется рекурсивно, как представлено в формулах (12–13):

$$\Delta_S^k f(v_i) = w_i \sum_{v_j \in N_1(v_i)} w_{ij} (\Delta_S^{k-1} f(v_j) \quad \Delta_S^{k-1} f(v_i)), \quad (12)$$

$$\Delta_S^0 f(v_i) = f(v_j). \quad (13)$$

Дискретизация зависит от выбранных весов для каждой вершины и ребра области: w_i и $w_{ij} = w_{ji}$ соответственно. В работе было рассмотрено две вариации весов: котангенсальные (14) и масштабирующие (15):

$$w_i = \frac{1}{A_i}, w_{ij} = \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}), \quad (14)$$

где α_{ij} и β_{ij} – два угла, противоположные ребру (v_i, v_j), а A_i – площадь Вороного для вершины v_i, которая определяется как площадь поверхности, образованная соединением центров инцидентных ребер с центрами описанных окружностей (для остроугольных треугольников) или центрами противоположных ребер (для тупоугольных треугольников), как показано на рис. 12.

$$w_i = \frac{1}{\sum_j w(v_i, v_j)}, w_{ij} = \frac{1}{\|v_i - v_j\|}. \quad (15)$$

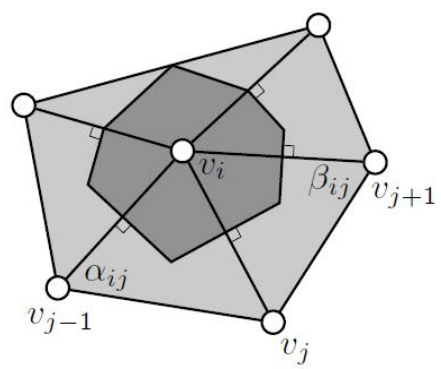


Рис. 14. Углы α_{ij} и β_{ij} и площадь Вороного A_i (темно-серая область), используемые для дискретизации оператора Лапласа–Бельтрами Δ_S в вершине v_i в уравнениях (10) и (11)

Сглаживание по формуле (15) оказалось предпочтительнее для применения на всех используемых в генераторе видах сеток (тесселяция и перестроенная поверхностная), так как использование формулы (14) менее устойчиво к наличию в сетке вырожденных элементов низкого качества (узкие вытянутые треугольники).

Дополнительные проверки различия между исправленной и исходной сеткой

Максимальное количество итераций алгоритма исправления пересечений на втором этапе ограничено тремя, так как при расширении кольца соседей увеличивается область сетки, которая будет изменена, и после затягивания области новыми треугольниками может произойти сильное искажение поверхности. Например, если исходная сетка представляет собой тесселяцию, даже первое кольцо соседей может образовывать область, масштаб которой будет сопоставим с масштабами целой модели.

После успешного выполнения алгоритма исправления производится дополнительная проверка на количество несвязанных областей. Если количество несвязанных областей увеличилось в результате работы алгоритма и их размеры (по суммарной площади и по количеству треугольников) малы относительно исходной модели, то такие области удаляются из сетки.

Пример работы алгоритмов на втором этапе

На рис. 15–16 приведен пример работы алгоритма: на рис. 15 слева изображена часть конструкции летательного аппарата с обнаруженными на кромке мотогондолы пересечениями, выделенными красным цветом, приближение выделенной рамкой области приведено на правой части рисунка.

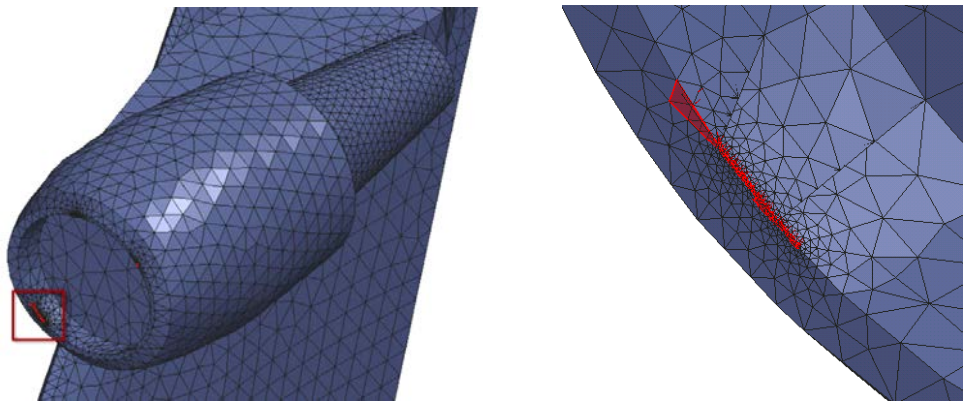


Рис. 15. Пример части конструкции с пересечением на кромке (слева); приближенная часть выделенной области с левой части рисунка (справа)

На рис. 16 слева изображена область после работы алгоритма удаления треугольников и затягивания контура новыми треугольниками. На центральной части рис. 16 изображен результат работы алгоритма адаптивного измельчения, который заметно уменьшает размеры новых треугольников. А на правой части рис. 16 приведен тот же фрагмент проблемной области после алгоритма сглаживания, который притягивает новые вершины сетки ближе к исходной поверхности.

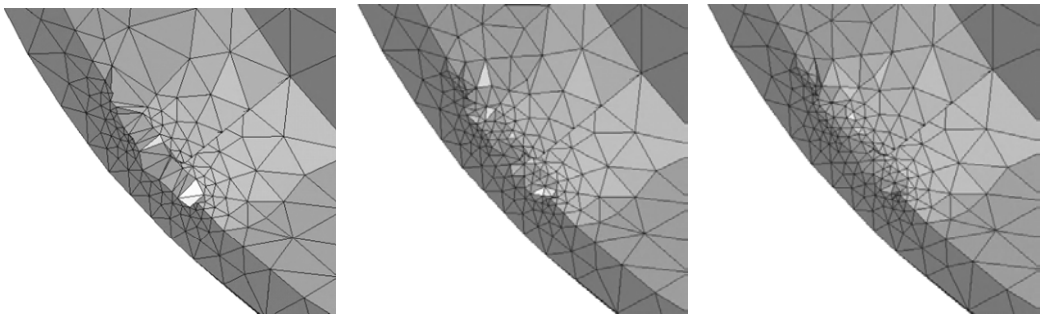


Рис. 16. Пример последовательного применения алгоритмов затягивания (слева), адаптивного измельчения (по центру) и сглаживания (справа)

Статистика исправлений

Тестирование реализованного алгоритма проводилось на базе задач различной сложности (более 100 тестов). В табл. 1 приведены результаты работы алгоритма для выборки сеток с различным типом и количеством пересечений. Алгоритм подтвердил свою эффективность в исправлении пересечений в рамках использования в генераторе поверхностной сетки. Однако, не во всех случаях получается достичь полного устранения пересечений, как показано на примере «Тест_6», который представляет собой сетку низкого качества с большим количеством свободных ребер и наложений треугольников помимо пересечений. В данном случае необходимо использовать другие средства для глобального исправления сетки.

Таблица 1

Количество областей с пересечениями в сетке до и после алгоритма исправления пересечений

Модель	Количество областей с пересечениями до исправления	Количество областей с пересечениями после исправления
Тест_1	21	0
Тест_2	17	0
Тест_3	14	0
Тест_4	12	0
Тест_5	19	0
Тест_6	110	1
Тест_7	9	0
Тест_8	36	0
Тест_9	371	0
Тест_10	4	0
Тест_11	8	0
Тест_12	18	0
Тест_13	88	0
Тест_14	10	0
Тест_15	12	0

Заключение

В настоящий момент реализованный алгоритм исправления пересечений треугольной сетки применяется в сквозной цепочке построения сеточных моделей для решения задач аэрогидродинамики в пакете программ «ЛОГОС». Использование алгоритма на практических задачах позволяет в автоматическом режиме исправлять такие критические ошибки поверхностной сетки, как пересечения, которые ранее не позволяли в автоматическом режиме проводить построение расчетной сетки для проведения численного моделирования.

Литература

1. Смолкина Д. Н., Борисенко О. Н., Черенкова М. В., Гиниятуллина А. Г., Кузьменко М. В., Чухманов Н. В, Потехина Е. В., Попова Н. В., Турусов М. Р. Автоматический генератор неструктурированных многогранных сеток в препроцессоре пакета программ «ЛОГОС» // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып.2. С. 25–39.

2. Голомидов Ф. О., Никитин В. А., Панов А. И., Соколов С. С., Шурыгин А. В. Генератор замкнутой поверхностной триангуляционной сетки в комплексе программ ЛОГОС // Всероссийская молодежная научно-инновационная школа «Математика и математическое моделирование», г. Саров, 2016.

3. 3rd AIAA CFD Drag Prediction Workshop, San Francisco, June 2006 // http://aiaa-dpw.larc.nasa.gov/Workshop3/workshop3.html.

4. Yamakawa S. and Shimada K. Removing Self Intersections of a Triangular Mesh by Edge Swapping, Edge Hammering, and Face Lifting. // The Department of Mechanical Engineering, Carnegie Mellon University, 2009.

5. Attene M. A lightweight approach to repairing digitized polygon meshes // The Visual Computer, Nov. 2010.

6. Michal Varnuska, Jindrich Parus, Ivana Kilingerova: Simple holes triangulation in surface reconstruction // Proceedings of ALGORITMY, 2005. P. 280–289.

7. Peter Liepa. Filling Holes in Meshes // Eurographics Symposium on Geometry Processing, 2003.

8. Mario Botsch, Olga Sorkine. On Linear Variational Surface Deformation Methods // IEEE Transactions on Visualization and Computer Graphics, 2008. P. 213–230.

ALGORITHMS FOR AUTOMATED REPAIRING OF INTERSECTIONS IMPLEMENTED IN SURFACE TRIANGULAR MESH GENERATOR IN "LOGOS" SOFTWARE PACKAGE

E. O. Evstifeeva, D. M. Pankratov, T. V. Tsalko

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes realized algorithms to repair intersections and to optimize surface triangular mesh in automated mode. It provides the results of the program realization of algorithms and the examples.

Key words: mesh generators, prepostprocessor, LOGOS software package, generator of surface triangular meshes, intersections of triangles.

УДК 004.02+004.04+004.08
DOI 10.53403/9785951505071_2022_181

СПЕЦИАЛИЗИРОВАННЫЕ БОЛЬШИЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ ДЛЯ РЕАЛИЗАЦИИ НЕЙРОСЕТЕВЫХ ВЫВОДОВ

Г. С. Елизаров, В. Н. Конотопцев, В. В. Корнеев

Научно-исследовательский институт «Квант», Москва

Представлены результаты экспериментального проектирования СБИС для реализации вывода сверточной нейросети SqueezeNet. Приведено сравнение прогнозируемых параметров этой СБИС с различными аппаратными платформами по производительности и производительности на ватт на тестовом наборе данных ImageNet.

Ключевые слова: сверточные нейросети, СБИС реализации вывода нейросети.

Для решения ряда важных научно-технических задач необходимо создание высокопроизводительных систем, использующих нейросетевые алгоритмы распознавания объектов и фона, навигации в пространстве. Ведущие университеты мира и корпорации, в том числе IBM и Intel, занимаются в научных и коммерческих целях разработкой эффективных нейросетевых парадигм и аппаратных архитектур для обучения и реализации выводов нейросетей. Высокая вычислительная сложность актуальных нейросетей привела к разработке библиотек для ускорения работы сетей на графических процессорах. Однако для высокопроизводительных моделей графических процессоров требуется мощность электропитания от 100 Вт, что делает проблемным их использование в автономных системах.

Для сокращения энергопотребления в процессе распознавания предлагается отделить процесс обучения и создавать нейросети, в которые сформированные ранее обучающие коэффициенты загружаются в виде максимально сокращенных массивов данных. Дополнительно сокращается трудоемкость отдельных операций путем замены операций с плавающей точкой на операции с фиксированной точкой.

В рамках НИР «Газель» (2015–2018 годы) по заказу Фонда перспективных исследований РФ была разработана технология быстрого проектирования СБИС. В качестве доказательства достижения поставленной цели был разработан проект специализированной СБИС Квант SN-2/2836, реализующей сверточную нейросеть (СНС) SqueezeNet [1]. Эта нейросеть выбрана в связи с тем, что в ней размер модели уменьшен приблизительно в 50 раз по сравнению с AlexNet, (AlexNet – 240 Мбайт [2]) при сохранении точности распознавания на том же уровне и применимости для решения разнообразных задач.

В СБИС реализованы все слои СНС SqueezeNet за исключением заключительного слоя softmax, вычислительная сложность которого невелика. Вычисления в СБИС Квант SN-2/2836 проводятся в формате чисел с фиксированной запятой, под целую часть отведено 13, под дробную – 11 разрядов. Точность вычислений при этом не хуже эталонного варианта реализации SqueezeNet с использованием арифметики с плавающей точкой. Вычислительная часть СБИС включает 2 синхронных конвейерных канала с общим управлением и общей памятью коэффициентов. Одновременно в обработке находятся до 4 кадров изображений (по два в каждом канале обработки) размером 224·224 пикселей. Полная обработка 2 кадров в СБИС занимает приблизительно $3{,}7 \cdot 10^6$ тактов работы. За каждый такт в каждом канале обработки выполняется 500 арифметических операций (умножение и сложение), всего в СБИС – 1 000 операций за такт.

Проект СБИС Квант SN-2/2836 был полностью промоделирован на ПЛИС Xilinx VC709 (XC7VX690T).

При изготовлении по технологии 28 нм на кристалле площадью 36 мм^2 прогнозируются следующие характеристики СБИС:

– частота работы – 1 200 МГц;

– мощность электропитания – от 12 до 14 Вт;

– производительность в числе кадров в секунду – 648 кадр/с;

– производительность в числе арифметических операций: 10^{12} оп/с.

В табл. 1 приведено сравнение эффективности реализации СНС SqueezeNet на различных аппаратных платформах на тестовом наборе данных ImageNet.

Приведенные данные демонстрируют высокую энергоэффективность прямой аппаратной реализации на СБИС этапа вывода (распознавания) для сверточных нейросетей SqueezeNet, а, также, возможность широкого масштабирования таких СБИС. Проектирование было осуществлено в директивный срок, предусмотренный разработанной в НИР «Газель» технологией.

Приведенные данные демонстрируют высокую энергоэффективность прямой аппаратной реализации на СБИС этапа вывода (распознавания) для сверточных нейросетей SqueezeNet – на обработку одного кадра затрачивается более чем на порядок меньше энергии, чем при использовании графических процессоров и более чем на два порядка меньше, чем в процессорах общего назначения.

Разработанная архитектура специализированной СБИС допускает широкое масштабирование вычислительных ядер, что позволяет обеспечивать требуемый уровень производительности для конкретных применений.

Проектирование СБИС Квант SN-2/2836а было осуществлено в директивный срок, предусмотренный разработанной в НИР «Газель» технологией.

Как результат активно протекающего процесса развития архитектуры нейронных сетей к настоящему времени разработаны нейросети, позволяющие еще более повысить достигнутые в проекте специализированного вычислителя Квант SN-2/2836 характеристики реализации нейросетевых выводов.

Среди направлений повышения производительности и энергоэффективности нейросетевых выводов [3] интенсивно развиваются бинарные нейросети с весами нейронов 1 или минус 1 и функциями активации с бинарными или целыми (int 32) значениями [4]. В бинарных сетях не используется энергозатратное умножение с плавающей точкой, существенно сокращается объем памяти для хранения весов.

Таблица 1

Эффективность реализации вывода СНС SqueezeNet на различных аппаратных платформах

Тип СБИС	CPU	GPU	SoC	FPGA	ASIC
Модель СБИС	Intel 7700k	Nvidia GTX 980Ti	Qualcomm Snapdragon 821	Xilinx VC709 (XC7VX690T)	Квант SN-2/2836
Технология, нм	14	28	14	28	28
Частота, ГГц	4,2/4,5	1/1,076	2,4	0,11	1,2
Вычислительные ядра, шт.	4	2816	4	2	2
Производительность, кадр/с	19,9	468,1	5,7	274,0	648,0
Мощность, Вт	104,5	203,5	4,6	27,7	14,0
Энергоэффективность, кадр/Дж	0,19	2,3	1,24	9,89	46,3

В табл. 2 приведено сравнение точности Top-1 верного распознавания нейросетей MPT-1/1 и MPT-1/32 с бинарными весами, сформированных и обученных по алгоритмам [4], и нейросетей с наилучшей достигнутой точностью с 32-разрядными весами на двух тестовых наборах данных CIFAR-10 и ImageNet.

Из табл. 2 видно, что объем памяти для хранения бинарных нейросетей меньше, чем требуется для нейросетей с 32-разрядными весами, но не в 32 раза, так как бинарная нейросеть может иметь большее число слоев и нейронов в слоях. Но отсутствие операций с плавающей точкой, делает вывод этих сетей более производительным и менее энергозатратным.

Таблица 2

Сравнение точности бинарных нейросетей МРТ и нейросетей с наилучшей достигнутой точностью с 32-разрядными весами на двух тестовых наборах данных CIFAR-10 и ImageNet

Нейросеть – разрядность весов/функции активации, тестовый набор	Память, Мбайт	Top-1, %
VGG-Small – 32/32, CIFAR-10	4,6	93,6
МРТ – 1/1, CIFAR-10	1,44	91,9
AlexNet – 32/32, ImageNet	240	57,2
SqueezeNet – 32/32, ImageNet	4,8	57,5
ResNet-34 – 32/32, ImageNet	21,8	73,27
МРТ – 1/1, ImageNet	19,3	52,07
МРТ – 1/32, ImageNet	13,7	74,03

Исходя из сегодняшнего состояния теории и практики нейросетевых вычислений разделение вывода и обучения представляется по-прежнему актуальным. Более того, появляются новые подходы к формированию и обучению нейросетей, например, на базе хеш-функций [5], выбора подсетей [4] и другие. Поэтому для формирования и обучения нейросетей требуются автоматизированные рабочие места с множеством алгоритмов формирования и обучения, а также наборов данных для достижения оптимального результата обучения.

Для вывода наиболее интересны нейросети с бинарными весами 1 или минус 1 и ассиметричной BPReLU (Biased parametric ReLU) функцией активации [6] с 1-, 2-, 32-разрядным выходом.

Представляется, что высокопроизводительная энергоэффективная реализация аппаратной платформы для выводов бинарных нейросетей может быть сделана на базе специализированных ПЛИС или СБИС, архитектура которых учитывает большую разреженность матриц весов и только два возможных значения их элементов: 1, минус 1. Поиск такой архитектуры – актуальная проблема.

Литература

1. Iandola F. et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size // arXiv:1602.07360v4 [cs. CV] 4 Nov 2016.

2. Alex K., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems // Lake Tahoe, NV, USA, 3–6 December 2012.

3. Корнеев В. В. Направления повышения производительности нейросетевых вычислений // Программная инженерия. 2020. Т. 11. № 1. С. 21–25.

4. Diffenderfer J., Kailkhura B. Multi-Prize Lottery Ticket Hypothesis: finding accurate binary neural networks by pruning a randomly weighted network // arXiv:2103.09377v1 [cs.LG] 17 Mar 2021 00:31:24 UTC (2,988 KB)

5. Chen B., Medini T., Farwell J., Gobriel S., Tai C., Shrivastava A. SLIDE : in defense of smart algorithms over hardware acceleration for large-scale deep learning systems // arXiv:1903.03129v2 [cs.DC] 1 Mar 2020

6. Zhang Y., Pan J., Liu X., Chen H., Chen D., Zhang Z. FracBNN: Accurate and FPGA-Efficient Binary Neural Networks with Fractional Activations // arXiv:2012.12206v1 [cs.LG] 22 Dec 2020 17:49:30 UTC (2,281 KB).

APPLICATION SPECIFIC INTEGRATED CIRCUITS FOR IMPLEMENTING NEURAL NETWORK INFERENCE

G. S. Elizarov, V. N. Konotoptsev, V. V. Korneev

Research Center “Kvant”, Moscow

The results of the experimental design of the VLSI circuit for the implementation of the inference of the convolutional neural network SqueezeNet are presented. The comparison of the predicted parameters of this VLSI circuit with various hardware platforms in terms of performance and performance per watt on the ImageNet test dataset is given.

Key words: convolutional neural networks, VLSI circuit implementation of neural network inference.

УДК: 004.2 + 004.4
DOI 10.53403/9785951505071_2022_185

АРХИТЕКТУРА СПЕЦИАЛИЗИРОВАННЫХ БОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

С. Г. Елизаров[1], *В. В. Корнеев*[2]

[1]Дизайн-центр цифровой микроэлектроники, МГУ им. М. В. Ломоносова, Москва
[2] Научно-исследовательский институт «Квант», Москва

В докладе рассматривается архитектура С-СБИС, ориентированная на технологические нормы 28 нм и менее для решения распараллеливаемых вычислительно сложных задач символьной обработки и дискретной математики. Для «нативного» распараллеливания программ применен подход «легких потоков», когда фрагменты кода единой программы передаются с очень низкими накладными расходами на множество исполнительных ядер, а результаты вычислений сохраняются в общей памяти. Описаны основные аппаратные блоки разработанной С-СБИС, в том числе вычислительные ядра, накристальная коммуникационная сеть, специализированный на проблемную область процессор с SIMD архитектурой.

Ключевые слова: СБИС, архитектура

Особенности архитектуры СБИС, обусловленные технологией изготовления

Архитектура специализированных больших интегральных схем С-СБИС должна разрабатываться с ориентацией на эффективную реализацию алгоритмов области специализации. При этом критериями эффективности реализации выступают производительность, потребляемая мощность, производительность на единицу мощности и т. д. в зависимости от требований проблемной области.

В связи с тем, что в настоящее время коммерчески доступны технологические процессы 28/16/7 нм… в предлагаемой архитектуре С-СБИС будем учитывать технологический сдвиг [1]. Современные технологии микроэлектроники вносят свои ограничения в создаваемые архитектуры С-СБИС, проявившиеся при достижении технологической нормы 90 нм, а также дальнейшие ограничения, начиная с норм 28 нм. Эти ограничения обусловлены: 1) усиливающимся по мере роста степени интеграции энергопотреблением интегральных схем, существенно определяемым суммарной длиной проводников, доставляющих команды и данные к функциональным устройствам; 2) ограничениями площади локальных кластеров на СБИС, состоящих из функциональных устройств и источников команд и операндов для них, использующих синхронное тактовое дерево, а также переход к архитектурам класса GALS (Globally Asynchronous, Locally Synchronous), в которых реализована глобальная асинхронность при передаче между синхронными вычислительными устройствами разных кластеров; 3) проблемой «темного кремния», заключающейся в схемотехнической возможности формирования схемы, которая при ее реализации на кристалле СБИС приведет к превышению предельно допустимой рассеиваемой мощности, что делает необходимым целый комплекс мер по обеспечению снижения потребляемой мощности, как на технологическом, так и на архитектурном уровне.

Поэтому специализация должна реализовываться в рамках ограничений, задаваемых технологией изготовления кристалла. Пусть требуется создать специализированный микропроцессорный кристалл, предельно достижимой производительности при вышеприведенных ограничениях. Из-за ограничений на время распространения тактового сигнала, синхронная схема специализированного процессорного элемента (ПЭ) может занимать площадь и содержать количество транзисторов, не

более наперед определенных для используемых технологических норм и тактовой частоты. Поэтому архитектура будет высоко параллельной класса GALS.

Более того, ограничения по потребляемому току в статике и динамике, а также выделяемому и отводимому теплу, приводят к тому, что в каждом такте может изменяться состояние только части из общего количества транзисторов. Поэтому рациональным решением служит введение в ПЭ блоков памяти, создающих эффект большого процента не переключающихся в каждом такте транзисторов. Оставшиеся после реализации блоков памяти транзисторы используются в специализированных функциональных устройствах. На большом кристалле может разместиться много (тысячи) указанных ПЭ, функционирование которых требует загрузки и выгрузки их блоков памяти для смены программ и данных ПЭ, а также инициализации выполнения в них программ. Такие архитектурные решения использованы в кристаллах разной специализации, например: PEZY-SC2 [2], ориентированном на численные методы решения дифференциальных уравнений, и Graphcore Colossus IPU [3], созданном для решения задач искусственного интеллекта.

Специализация архитектуры кристалла определяет, служит ли он только полем однородных ПЭ, загружаемых и управляемых из внешнего по отношению к кристаллу хост-компьютера, как Graphcore Colossus, либо реализуется иерархическая гетерогенная кластерная архитектура, в которой на каждом уровне иерархии вычислительные устройства имеют локальные блоки памяти, как в PEZY-SC2. В этом архитектурном варианте связь между хост-компьютером и специализированным кристаллом частично перекладывается на управляющие процессоры кластеров, что снижает требования к производительности хост-компьютера и пропускной способности канала связи между ним и специализированным кристаллом или множества объединенных специализированных кристаллов.

Программирование вычислительных систем, состоящих из хост-компьютеров и поля однородных вычислительных модулей (ВМ) или, более того, иерархии гетерогенных кластеров представляет сложную проблему, требующую от программиста знания технических деталей аппаратной реализации.

В отличие от упомянутых выше суть представляемой в докладе архитектуры С-СБИС при использовании иерархии гетерогенных кластеров состоит в реализации всех функций по организации и управлению хост-компьютера в собственно С-СБИС. То есть хост-компьютер может передать программу и исходные данные в процессор управления С-СБИС и получить из него результат выполнения. Программа для процессора управления создается с использованием привычных программистам C, C++, OpenCL, подходов и библиотек. Используется «нативный» подход к распараллеливаю вычислений, при котором от программиста требуется только указать, это может выполняться параллельно.

Архитектура С-СБИС

Логическая структура С-СБИС

Основные концепции представляемой архитектуры и программирования изложены в [4]. Для получения адекватных оценок эффективности предлагаемой архитектуры проведено проектирование в САПР С-СБИС для размера кристалла 6×6 мм^2. Выбор такого размера обусловлен сравнительной простотой доступа на микроэлектронные производства с технологией 28 нм.

Архитектура разработанного кристалла приведена на рис. 1. Логически С-СБИС состоит из универсальной компоненты и компоненты специализации. Универсальная компонента служит для организации вычислений, доставки данных и выгрузки результатов из специализированной компоненты. Соответственно, специализированная компонента обеспечивает высоко эффективные вычисления. Эта архитектура MALT (Manycore Architecture with Lightweight Threads) [5] служит фундаментом для построения семейств специализированных СБИС. Представленная ниже С-СБИС принадлежит семейству MALT-C, специализация которого решение идеально распараллеливаемых вычислительно сложных задач символьной обработки и дискретной математики.

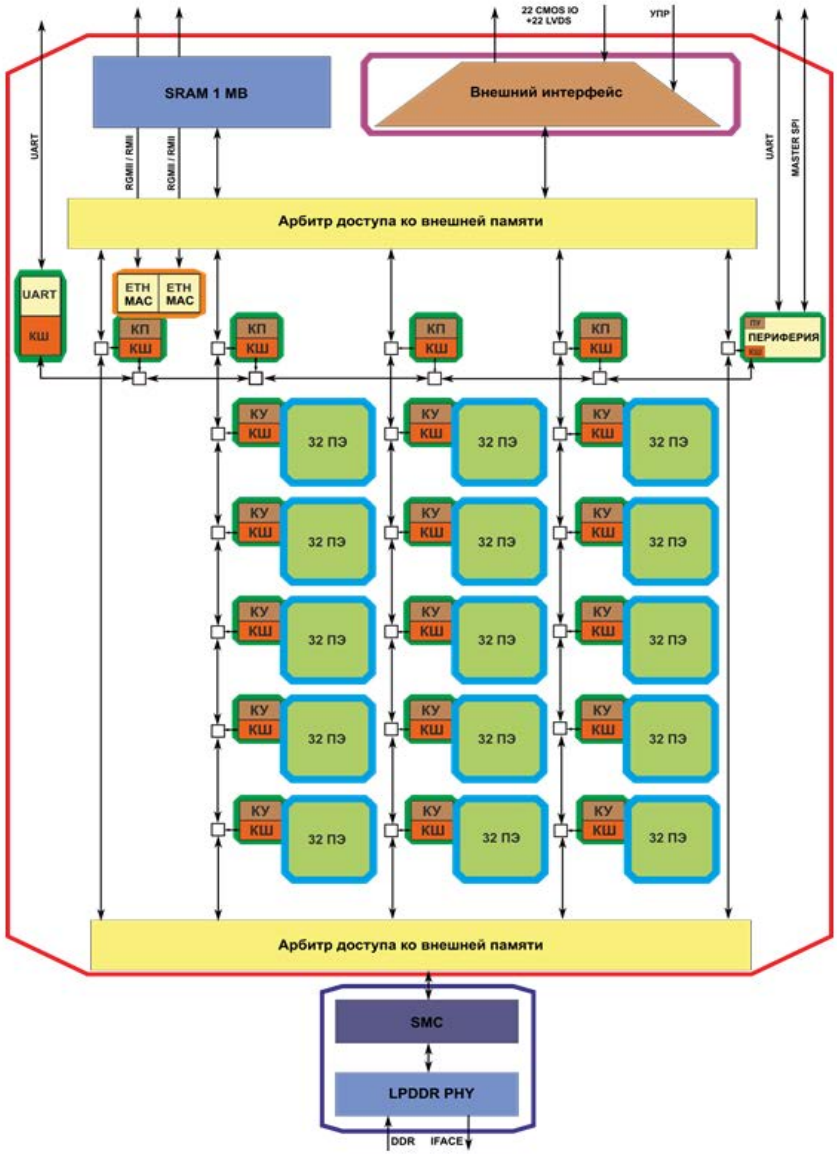


Рис. 1. Архитектура П-СБИС

Универсальная компонента С-СБИС

Универсальная компонента кристалла С-СБИС включает 3 типа процессоров: процессор управления ПУ, коммуникационные процессоры КП и вычислительные процессоры КУ.

Процессор управления включает:

– 32-битное процессорное RISC-ядро общего назначения MB-Lite, по набору инструкций совместимое с архитектурой MicroBlaze;

– ПЗУ инструкций (на шине инструкций MB-Lite);

– декодер адреса (шины данных MB-Lite);

– кеш инструкций;

– блок локальной памяти;

– контроллер КШ пакетной шины;

– контроллер периферийных интерфейсов.

Исполняемый код программы может частично храниться в ПЗУ инструкций (ROM-памяти), либо подгружаться через используемую пакетную шину с промежуточным хранением инструкций

в кеш-памяти (8 Кбайт). Данные для исполняемой программы могут быть загружены по пакетной шине в блок локальной памяти. Благодаря низко латентному доступу к локальной памяти данных достигается эффективная работа алгоритмов, требующих небольших объемов данных. Кеш данных отсутствует, по причине значительных накладных расходов по обеспечению кеш-когерентности.

На шине данных процессорного ядра присутствует декодер адреса, обеспечивающий подключение к шине данных как блоков памяти, так и других устройств с регистровым интерфейсом, отображаемым на эту шину.

Процессор управления функционирует под операционной системой Minix3. Вычислительный процессор КУ имеет тот же состав оборудования как у управляющего процессора, за исключением того, что в нем отсутствует контроллер периферийных интерфейсов, но добавлен интерфейс с кластером ПЭ. К каждому вычислительному процессору подключен один кластер (32 ПЭ).

Коммуникационные процессоры КП не имеют вышеупомянутых контроллера периферийных интерфейсов, а также интерфейса с кластером ПЭ и по набору инструкций совместимы с архитектурой MicroBlaze.

Пакетная шина с малыми задержками, используемая для объединения абонентов внутри С-СБИС построена на базе коммутаторов КОМ канального уровня 2×2. Один вход и выход коммутатора служат для подключения абонента посредством контроллера КШ пакетной шины, а оставшиеся вход и выход используются для соединения с предшествующим и последующим коммутаторами с образованием конвейера. На краю конвейера находится оконечный блок шины или терминатор шины, в случае отсутствия абонента.

Коммутаторы с использованием каналов (линков) объединяются в однонаправленный конвейер. Абоненты шины посредством контроллеров шины КШ подключаются к соответствующим портам коммутаторов. Абонент может быть подключен к двум разнонаправленным конвейерам, как это показано на рис.1. Такая топология выбрана исходя из требований к коммуникационной подсистеме – возможность одновременных передач данных между несколькими абонентами шины, направленность потоков данных, низкая задержка и малое количество затрачиваемых ресурсов.

Ширина портов коммутатора и линка – 65 бит: 64 бита – данные и бит, который служит для указания, текущее слово шины представляет собой заголовок.

Пакетная шина реализует принцип червячной передачи (wormhole), то есть, в каждый момент времени коммутатор хранит одно слово передаваемого пакета.

Пакетная шина использует статическую маршрутизацию. Заголовок пакета содержит номер абонента, которому предназначен запрос, в каждом узле коммутационной сети номер текущего абонента сравнивается со значением в заголовке и, исходя из этого, осуществляется маршрутизация сообщений.

Максимальная длина пакета на канальном уровне – 256 слов или 2 КБ. Одновременно шина может осуществлять передачу нескольких сообщений.

Количество пакетов, одновременно передаваемой по шине, определяется количеством коммутаторов. В системе из 16 абонентов, соединенных однонаправленным конвейером, может находиться максимально 8 пакетов по 2 слова в каждом или 1 пакет длиной 16 и больше слов.

При передаче данных используется коммутация без буферизации. То есть, длинный пакет разделен на части и, в зависимости от заголовка, часть пакета, на следующем такте, попадает в соответствующий выходной порт. При наличии у обоих входных портов пришедших данных, предпочтение отдается первому порту коммутатора. Для того, чтобы не прерывать работу абонента шины, когда приходят данные, реализована входная буферизация у второго порта.

Передача сообщений по шине на протокольном уровне является тегированной, что позволяет более эффективно использовать приходящие данные со стороны абонента.

На протокольном уровне реализовано несколько типов сообщений, таких как сообщения различной длины, broadcast-сообщения, запись и чтение в удаленную память массивов различной длины (DMA).

На кристалле С-СБИС реализовано 6 пар двунаправленных пакетных шин:

– одна для связи процессора управления с коммуникационными процессорами и периферийными устройствами;

– три для связи коммуникационных процессоров с соответствующими вычислительными процессорами;

– одна для доступа процессора управления к внутри кристальному блоку памяти (1 МБ) и блоку внешней памяти DDR;

– одна для обмена между сетевыми контроллерами и блоками внутри кристальной и внешней памятей.

Разделяемая ПУ, КУ, КП память расположена непосредственно на кристалле, а также может подключена через интерфейс DDR или LVDS. Возможны комбинации этих вариантов. Пропускная способность накристальной памяти значительно выше пропускной способности DDR или LVDS. Но размер накристальной памяти ограничен и обычно не превышает 1 Мегабайта. В случае использования накристальной памяти совместно с внешней памятью адресное пространство начинается в накристальной памяти и продолжается во внешней. Доступ к блокам памяти выполняется через SMC (smart memory controller) «умный» контроллер памяти, выполняющий атомарные операции и синхронизацию доступов к памяти на базе FE-битов [4].

Специализированная компонента С-СБИС

Для высоко эффективной параллельной обработки цифровой и символьной информации вычислительные процессоры КУ используют специализированную систему с SIMD архитектурой, показанной на рис. 2.

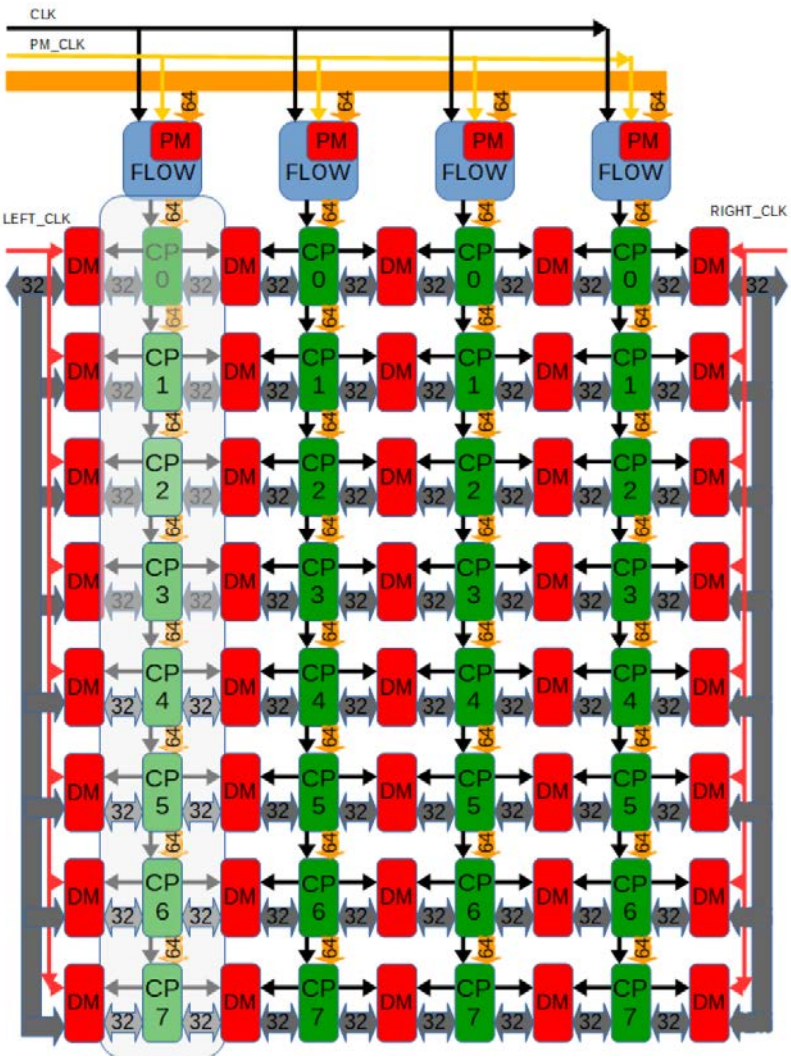


Рис. 2. Архитектура специализированной SIMD системы

Специализированная система состоит из четырех независимых устройств выдачи потока команд, к каждому из которых подключен конвейер из 8-ми процессорных элементов ПЭ, образующих в совокупности 8-ми процессорную систему с SIMD архитектурой. ПЭ выполняет операции над 32 разрядными словами.

Каждый такт команда поступает на выполнение в очередной ПЭ конвейера. Таким образом каждый ПЭ выполняет со сдвигом на такт одну и ту же команду. Ветвление возможно пропуском выполнения поступившей команды в зависимости от результата выполнения предыдущей команды. Каждый ПЭ имеет доступ к двум блокам локальной памяти SDRAM слева и справа от него, объемом каждый 1280 32-разрядных слов.

Вычислительный процессор КУ загружает программу в устройства выдачи потока команд, разные или одну и ту же, а также записывает или читает данные из крайних столбцов блоков памяти. Это обусловливает программное обращение к остальным блокам памяти через регистры соответствующих ПЭ. При этом у ПЭ нет прямого доступа ни к локальной памяти КУ, ни к общей разделяемой памяти.

Все устройства выдачи потока команд и ПЭ функционируют синхронно по-тактно. Проблемы возможных конфликтов по доступу двух ПЭ к разделяемому блоку памяти решаются на уровне программирования.

Структура ПЭ приведена на рис. 3.

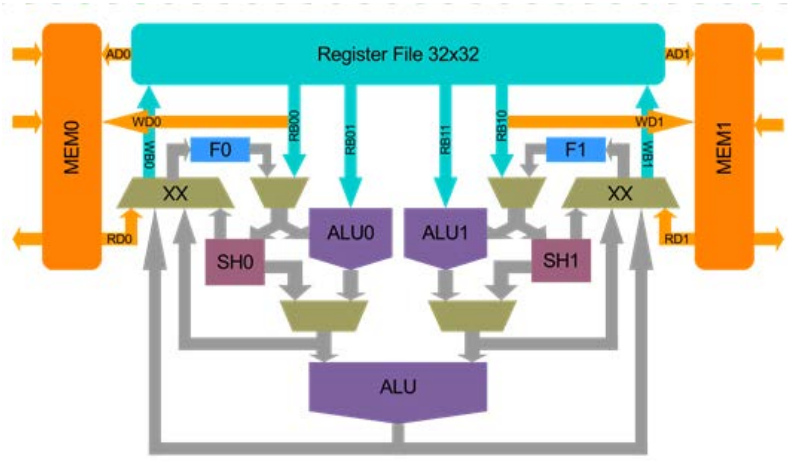


Рис. 3. Структура ПЭ

Процессорные элементы построены по архитектуре VLIW и могут одновременно выполнять 5 действий (3 АЛУ и 2 сдвигателя) с 4 операндами. Ветвление возможно только посредством временной остановки определенных процессорных элементов в зависимости от результата выполнения предыдущей команды. Таким образом, максимальная производительность ПЭ при тактовой частоте 1Ггц достигает $5\times32\times10^9 = 1{,}6\times10^{11}$ операций в секунду.

Максимальная производительность С-СБИС определяется суммой производительностей ПЭ и, например, для приведенного на рис.1 варианта кристалла площадью 6×6 мм^2 с 15 ПЭ, составляет $2{,}4\times10^{12}$ операций в секунду. Энергопотребление при тактовой частоте 1 Ггц по результатам моделирования в САПР составляет не более 10 Вт.

Возможности применения

С использованием СБИС семейства MALT-C возможно построение вычислительных систем для различных проблемных областей. Ориентация на область использования закладывается, например, на уровне конструктивного элемента платы, как показано на рис. 4.

Пример построения стойки вычислительной системы приведен на рис. 5.

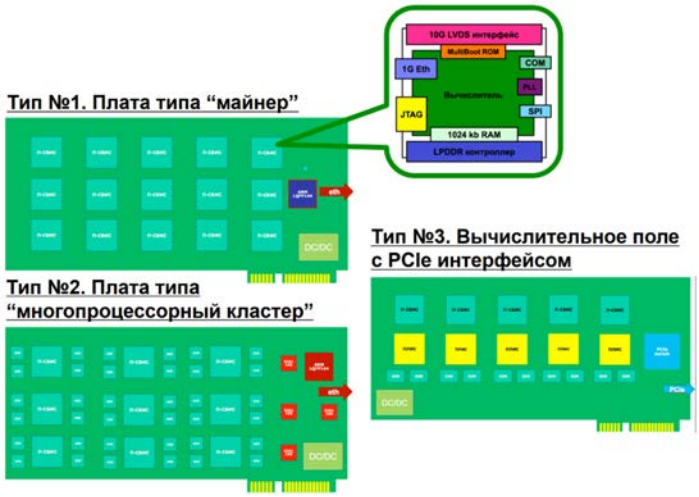


Рис. 4. Платы для возможных проблемных областей применения



Рис. 5. Пример конструктивного исполнения стойки проблемно-ориентированной вычислительной системы

Средства программирования

Программист, в зависимости от своих предпочтений и требований к оптимизации прикладного кода, может выбрать один из следующих подходов:

– C++ для MALT. Для программирования скалярных ядер поддерживается стандарт C++17. Работа с легкими потоками thread осуществляется через конструкции C++ std::thread, std::mutex и другие. MALT выступает в роли классического многоядерного процессора, поэтому портировать существующее ПО и библиотеки легко.

– OpenCL для MALT. Реализация стандарта OpenCL для работы со скалярными и векторными ядрами MALT. Реализация OpenCL поставляется вместе с библиотекой проблемно-ориентированных алгоритмов, оптимизированных под MALT. Теперь перейти на MALT для пользователей ускорителей AMD, NVIDIA, ARM стало проще.

– C для MALT + низкоуровневая библиотека. Реализует возможности архитектуры, позволяет в ручном режиме оптимизировать и распараллеливать программы целевых классов.

Исполнение программы инициируется процессором управления ПУ, которым служит «обычный» многопотоковый процессор, функционирующий под управлением POSIX-совместимой ОС Minix3. Для «нативного» распараллеливания программы на множество легких потоков, фрагмент кода единой программы ПУ отправляется им по пакетной шине с очень низкими накладными рас-

ходами на выбранный коммуникационный процессор КП. В свою очередь, КП определяет один или подмножество исполнительных вычислительных процессоров КУ, из числа подключенных к одной с ним пакетной шине. Вычислительные процессоры при этом получают указатель на фрагмент кода, передаваемого им для исполнения. Результаты сохраняются в общей памяти. Такой подход реализован, например, в библиотеке «легких тредов» Qthreads [6], созданной в Sandia National Laboraroties. Основная идея – обеспечить механизм порождения параллельно исполняемых потоков на уровне программы процессора, а не на уровне операционной системы. В Qthread порождение потока описывается вызовом вида:

void start_thread(int(*)(int), int, ...);

Вызов может быть произведен из любого вычислительного процессора. Такой вызов совместно с атомарными, не блокируемыми операциями инкремента и некоторыми другими стандартными операциями создает возможность порождать (spawn) и объединять (join) группы параллельно выполняемых потоков, создавая последовательно-параллельную программу.

Литература

1. Корнеев В. В., Тарасов И. Е. Особенности архитектуры массово-параллельных проблемно-ориентированных СБИС // Программная инженерия. 2019. Т. 10, № 4. С. 160–166.

2. Torii S., Ishikawa H. ZettaScaler: Liquid immersion cooling Manycore based Supercomputer // ISC 2017. Frankfurt am Main, Germany, june 18–22, 2017.

3. Jia Zh., Tillman B., Maggioni M., Scarpazza D. Dissecting the Graphcore IPU Architecture via Microbenchmarking. Technical Report. December 7, 2019. arXiv:1912.03413v1 [cs.DC] 7 Dec 2019.

4. Елизаров С. Г., Лукьянченко Г. А., Корнеев В. В. Технология параллельного программирования экзафлопсных компьютеров// Программная инженерия. 2015. № 7. С. 3–10.

5. Архитектура MALT [Электронный ресурс]: сайт Malt System. – Режим доступа: https://maltsystem.ru.

6. Wheeler K., Murphy R., Thain D. Qthreads: An API for programming with millions of lightweight threads // In Proceedings of the 22nd IEEE International Parallel & Distributed Processing Symposium, April 2008, pp. 1–8.

THE ARCHITECTURE OF THE APPLICATION SPECIFIC INTEGRATED CIRCUITS FOR HIGH-PERFORMANCE COMPUTING

S. G. Elizarov[1], *V. V. Korneev*[2]

[1] Design Center for Digital Microelectronics, MSU, Moscow
[2] Research Center “Kvant”, Moscow

The report discusses the S-VLSI circuit architecture focused on technological standards of 28 nm or less for solving computationally hard parallelizable problems of symbolic processing and discrete mathematics. For native parallel programming we use the ‘lightweight threads’ approach dividing the program into multiple threads performing relatively short code fragments from a single program with very low overheads. The results are saved in the shared memory. Main units of the S-VLSI circuit including universal cores computing kernels, on-chip communication network, specialized processor with SIMD-architecture are described.

Key words: VLSI circuit, architecture.

УДК 004.272.44
DOI 10.53403/9785951505071_2022_193

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ТЕСТА HPCG ДЛЯ ППВС «БУРАН»

Д. Н. Змеев, А. В. Климов, А. С. Окунев, Н. Н. Левченко

Институт проблем проектирования в микроэлектронике РАН, Зеленоград

HPCG (High Performance Conjugate Gradients) –- новый тест для сравнения суперкомпьютерных систем после LINPACK HPC. Считается, что он лучше отражает характеристики рабочей нагрузки реальных научных приложений. Рассматривается реализация его центральной части – сглаживателя Гаусса-Зейделя – на проектируемой параллельной потоковой вычислительной системе (ППВС) «Буран». В отличие от эталонного кода HPCG для MPI, который использует смешанный метод Якоби – Гаусса–Зейделя, мы реализуем чистый алгоритм Гаусса-Зейделя на предлагаемом языке потоков данных UPL, обеспечивающем автоматическое извлечение потокового параллелизма. Таким образом, мы нарушаем условия теста, используя другой вычислительный алгоритм, который частично компенсирует большие задержки сети и памяти.

Ключевые слова: потоковая модель вычислений, архитектура потока данных, параллельные вычисления, высокопроизводительные вычисления, тест HPCG, метод сопряженных градиентов, метод Гаусса–Зейделя.

1. Введение

Тест HPCG (High Performance Conjugate Gradients) [1] является попыткой создать новую метрику для сравнения суперкомпьютерных систем. Он призван дополнить используемый в настоящее время для этой цели тест HPL (High Performance LINPACK), поскольку задачи линейной алгебры, из которых тот состоит, не предъявляют в должной мере требований к свойствам аппаратуры, которые критичны для многих современных задач. Это, прежде всего, требования к подсистеме памяти и коммуникационной сети, их пропускной способности и латентности. А алгоритмам HPL обычно удается обходить эти проблемы.

Авторы теста HPCG и его эталонного кода требуют, чтобы при тестировании новой суперкомпьютерной системы использовался именно их численный метод и принципы его распараллеливания, и чтобы не вносились изменения в код, которые позволяли бы обходить недостатки аппаратуры. В этой статье мы демонстрируем как раз такой альтернативный способ реализации.

Мы рассматриваем возможность реализации данного теста на проектируемой в нашей организации параллельной потоковой вычислительной системе (ППВС) «Буран» [2], реализующей потоковую модель вычислений с динамически формируемым контекстом [3]. От обычных фоннеймановских вычислителей ее отличает использование принципа потока данных, состоящего в активации вычислительных узлов по готовности аргументов, а от классических потоковых архитектур – использование контекста как объекта данных «первого класса», не ограниченного формами, диктуемые стандартными шаблонами программирования (циклы, доступ к элементам массива, рекурсия, и т. п.) в распространенных языках.

Специфика модели вычислений позволяет использовать другой вариант алгоритма предобуславливателя, используемого в тесте, который, с одной стороны, обеспечивает более быструю сходимость, а с другой стороны, при распределенном выполнении является менее требовательным

к латентности памяти и коммуникационной сети. Это особенно актуально, если учесть, что именно латентность сетей и памятей большого размера (в отличие от пропускной способности) с годами улучшается крайне медленно [4]. Также важно отметить, что этот вариант алгоритма в нашей модели вычислений реализуется достаточно просто, тогда как его программирование, например, в MPI настолько сложно, что этот путь даже и не рассматривается.

Статья имеет следующую структуру. В разделе 2 рассмотрен алгоритм теста и проблемы, возникающие при его распараллеливании. В разделе 3 математически изложен фрагмент задачи сглаживателя и его вычислительный граф. В разделе 4 рассмотрены варианты декомпозиции для распараллеливания. В разделе 5 вводится язык UPL, на котором записывается стандартный алгоритм одного прохода по Гауссу–Зейделю, и разъясняется его работа. В разделе 6 изложены принципы работы потокового вычислителя. В разделе 7 определяется функция распределения вычислений по пространству (процессорным ядрам), обеспечивающая хорошее масштабирование. Возникающие при этом блоки могут использоваться для укрупнения узлов (раздел 8). А раздел 9 говорит о том, как распределение по времени помогает решить проблему «стены памяти». Результаты экспериментального моделирования приведены в разделе 10. В Заключении подводятся итоги и перечисляются нерешенные проблемы.

2. Описание задачи теста

В тесте HPCG решается задача итерационного решения системы линейных алгебраических уравнений (СЛАУ) вида $Ax = b$. Основная операция в каждой итерации – это умножение большой разреженной матрицы A на плотный вектор (текущее приближение) x. Матрица A является симметричной и положительно определенной. Портрет (места ненулевых элементов) матрицы A не случаен, он получен из отношения соседства на регулярной прямоугольной трехмерной сетке (рис. 1), где у каждой ячейки не более 26 соседей, а каждое уравнение связывает значение в ячейке со значениями в соседних ячейках. которой содержат элементы искомого вектора x. Ячейке сетки соответствует линейное уравнение, связывающее значение в ячейке со значениями в ее соседях. Задачей теста является решение этой системы уравнений. Если сетка имеет размеры n_x, n_y, n_z, то матрица A имеет порядок $N = n_x n_y n_z$.

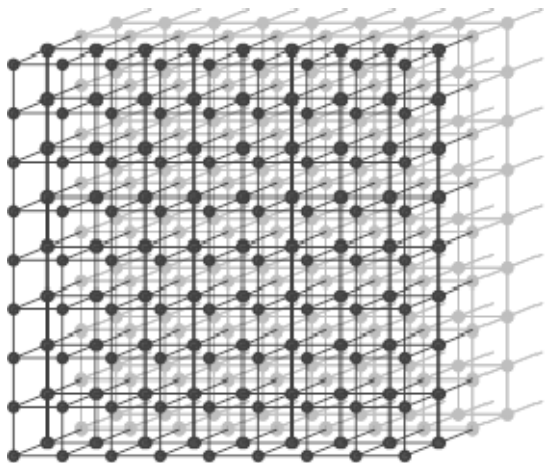


Рис. 1. Структура трехмерной сетки. Показана только часть связей (без «диагональных»)

Общее число итераций метода CG можно уменьшить, если сделать предобусловливание матрицы A. Иначе говоря, умножить ее на матрицу-предобуславливатель M, сближающее ее с единичной, и решать уравнение $MAx = Mb$ вместо исходного. Вместо матричного умножения MA вы-

годнее делать умножение матрицы M на вектор z всякий раз, когда в методе CG выполняется умножение матрицы A на вектор x (и один раз для вектора правых частей b). А такое умножение равносильно приближенному решению уравнения $Az = r$ итерационным методом с применением небольшого фиксированного числа линейных итераций – сглаживателей.

В качестве базового сглаживателя применяется один или несколько шагов итераций Симметричного метода Гаусса–Зейделя (СГЗ) решения уравнения $Az = r$, который сводится к попеременному решению двух треугольных систем уравнений: с нижне-треугольной матрицей $L + D$ и верхне-треугольной матрицей $U + D$, полученными разложением матрицы $A = L + D + U$ (здесь $D = diag(A)$).

Для ускорения сходимости метода CG применяется многосеточный сглаживатель, так называемый V-цикл [5]: строятся 4 уровня все более грубых сеток посредством операций рестрикции и пролонгации, а между ними как раз и применяется СГЗ. Считается, что это сочетание (многосеточность и сглаживатель Гаусса–Зейделя при переходе между уровнями) дает хорошие результаты в плане улучшения сходимости CG [5, 6].

Однако, в общем случае, метод Гаусса–Зейделя трудно распараллеливается: в работе [7], например, автору удается достичь лишь ускорения в 2 раза для 4 процессоров, и дальше оно не растет. Причина этого в том, что для плотных матриц каждая переменная зависит от предыдущей. А в разреженном случае некоторые группы переменных могут определяться независимо, и это дает дополнительное пространство для параллелизма. Тем не менее, считается [6], что в матрице, порожденной трехмерной сеткой, параллелизма немного, и извлечь его – трудная задача.

Поэтому, как правило (в том числе в HPCG), применяют смешанный метод Гаусса–Зейделя–Якоби. Он состоит в разбиении пространства ячеек сетки на прямоугольные блоки, каждый из которых обрабатывается своим отдельным процессором. Каждый процессор выполняет одну итерацию СГЗ локально (при этом для всех «внешних» переменных берутся значения из предыдущей итерации), после чего все процессоры обмениваются значениями вычисленных переменных с соседями (рис. 2). В вычислительном плане это другой метод: он легко программируется как параллельный, но медленнее сходится [6, 8].

Ниже мы реализуем чистый метод СГЗ на потоковом языке UPL [9], извлекая максимум параллелизма из разреженной структуры матрицы A.

3. Симметричный итерационный метод Гаусса–Зейделя

Пусть $A = L + D + U$ – симметричная положительная определенная матрица, z – вектор правой части. Итерационно решается уравнение $Az = r$. Имея приближение $x^{(v)}$, вычисляем приближение $x^{(v+1)}$ из уравнений (верхний индекс в скобках – номер итерации):

$$Lx^{(v+1)} + Dx^{(v+1)} + Ux^{(v)} = r, \text{ если } v \text{ четно, и}$$
$$Lx^{(v)} + Dx^{(v+1)} + Ux^{(v+1)} = r, \text{ если } v \text{ нечетно.}$$

Вычислительный процесс проходит по следующим формулам (матрично-векторным):

$$u^{(0)} = x^{(0)} = 0$$

$$\begin{cases} l^{(2k+1)} = Lx^{(2k+1)} \\ x^{(2k+1)} = D^{-1}\left(r - u^{(2k)} - l^{(2k+1)}\right) \end{cases}, k = 0,1, \ldots$$

$$\begin{cases} u^{(2k)} = Ux^{(2k)} \\ x^{(2k)} = D^{-1}\left(r - l^{(2k-1)} - u^{(2k)}\right) \end{cases}, k = 1,2, \ldots$$

Для удобства введены две (векторные) переменные: l и u. Наличие неявности по x не мешает, поскольку она разрешается благодаря треугольности матриц L и U: для вычисления -го элемента

вектора l достаточно иметь значения компонентов вектора x с меньшими индексами, а для u – с большими. Поэтому итерации с нечетными номерами вычисляются прямым проходом (вдоль векторов), а с четными – обратным.

Выполнение одной итерации, например нечетной (с номером $2k+1$), можно рассматривать как решение системы линейных алгебраический уравнений (СЛАУ) с треугольной матрицей $L+D$ относительно переменных $x^{(2k+1)}$:

Граф зависимостей между узловыми переменными, показан на рис. 2. (Здесь и далее для упрощения и наглядности используется 2-мерная сетка, при этом все сказанное без проблем обобщается на 3-мерный случай). Узлы – это ячейки сетки, а дугам соответствуют ненулевые элементы матрицы A (здесь из 144 строк). Для каждой связи оставлено одно из двух направлений согласно выбору части L или U матрицы. Порядок соответствует лексикографическому по координатам ячеек $\langle \dots, i_2, i_1 \rangle$ (младший индекс i_1, следующий по старшинству i_2, и т. д.)

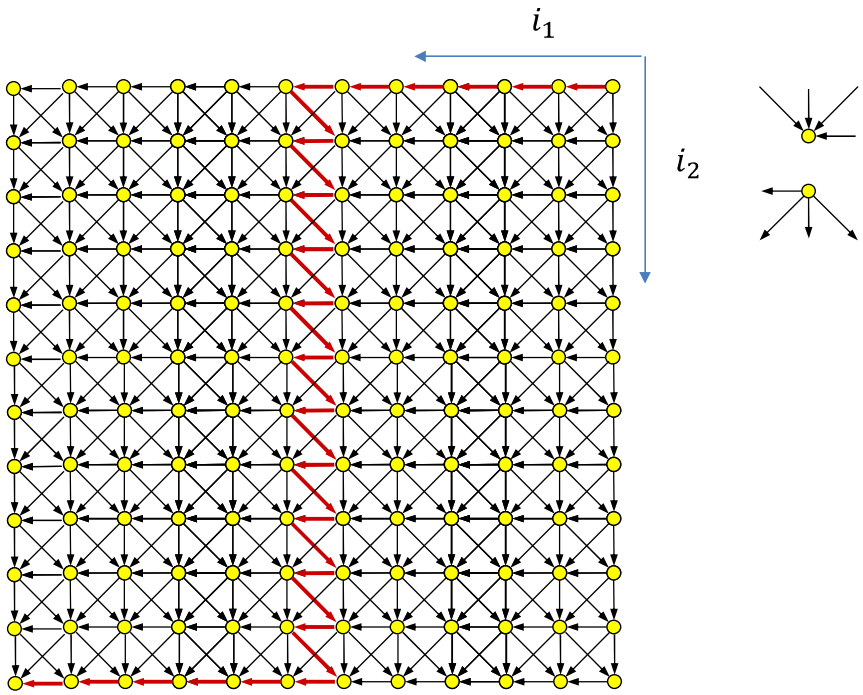


Рис. 2. Граф зависимостей одного прохода сглаживателя по Гауссу–Зейделю.
Красным цветом выделен один из критических путей

4. Варианты декомпозиции для распареллеливания

В эталонной реализации HPCG на C++ для MPI+OpenMP [10] для распараллеливания используется декомпозиция сетки на прямоугольные области, каждая из которых помещается в свой процесс. При этом умножение матрицы A на текущее приближение может быть «честно» выполнено автономно в каждой области при условии, что значения внешних переменных доставлены из других процессов. Однако, сглаживатель Гаусса-Зейделя при такой декомпозиции выполняется неточно: одна итерация метода выполняется локально в каждом прямоугольнике (см. рис. 3), используя всегда старые значения переменных из соседних процессов (прямоугольников), после чего новые значения переменных вдоль границ областей пересылаются соседям. Фактически это смешанная итерация Гаусса–Зейделя–Якоби: локально по Гауссу–Зейделю, глобально по Якоби. Такой вариант легко программируется, но требует большего числа итераций для достижения того же качества в смысле сходимости [6, 8].

Мы добиваемся распараллеливания вычислительного графа «чистого Гаусса–Зейделя», показанного на рис. 2. В этом мы отходим от условий теста HPCG. Наша цель – показать, что в модели вычислений ППВС можно легко осуществлять способы распараллеливания, которые трудны для MPI и OpenMP, и потому обычно отвергаются.

Множество входящих и исходящих дуг каждой ячейки показано на рис. 2 справа (в трехмерном случае их число равно 13). Положительные линейные комбинации этих векторов в системе координат сетки образуют конус зависимостей.

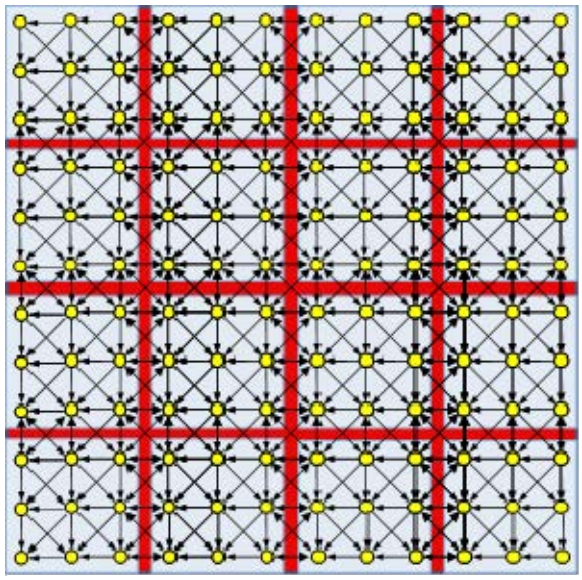


Рис. 3. Обычная декомпозиция на прямоугольные области, применяемая в HPCG. Внутри областей итерации по Гауссу–Зейделю, между областями – по Якоби

Согласно общим принципам, описанным в [11], такие зависимости допускают распараллеливание путем разбиения на блоки-параллелепипеды с гранями, параллельными граням конуса зависимостей (в котором следует оставить k граней, где k – размерность сетки). В данном случае для $k = 2$ это стороны параллелограмма на векторах (0,1) и (1,–1). Для $k = 3$ – параллелепипед на векторах (0,0,1), (0,1,–1), (1,–1,–1). По каждому направлению (вектору) можно выбрать свой размер блока. На рис. 4 показаны блоки 3×3.

Индуцируемый граф зависимостей между блоками ациклический, поскольку в блочной системе координат, основанной на указанных векторах, направления всех связей неотрицательные. Каждый блок готов к выполнению, когда завершены блоки сверху и справа от него. Поэтому здесь уместен потоковый параллелизм. Он создает фронт волны вычислений, параллельный большой диагонали блока, движущийся в направлении вектора (2,1).

Для другой части матрицы будет тот же граф, но с обратными стрелками.

5. Программа в потоковой модели вычислений

Потоковая программа решателя треугольной СЛАУ вида $\boldsymbol{Lx = b}$, представленная в графической версии языка UPL [9], показана на рис. 5. Эта программа применима к произвольной разреженной треугольной матрице L (здесь она содержит диагональ, которая должна быть целиком ненулевой), с любым «портретом» (включая плотный), а не только построенным из регулярной сетки.

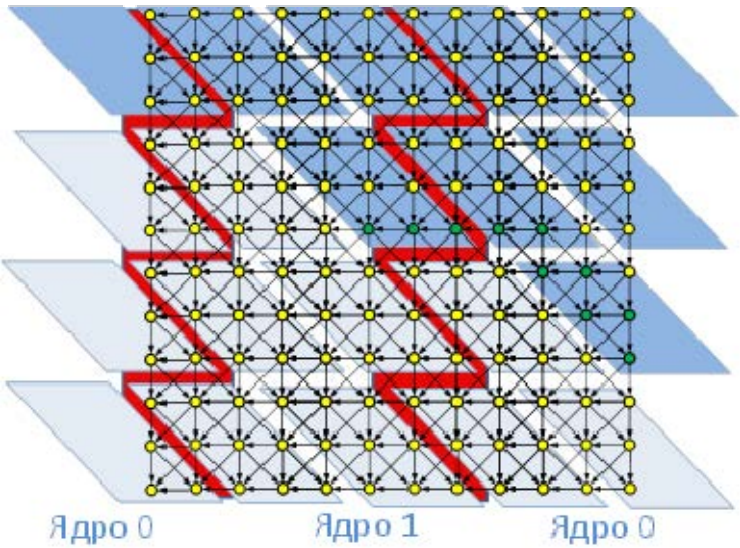


Рис. 4. Разбиение на блоки-параллелограммы. Каждый блок получает значения переменных от блоков справа и сверху и выполняется по приходу этих данных. Вычисленные блоки – темно-синие. Фронт движется «по диагонали» сверху вниз справа-налево. Красный зигзаг – распределение блоков по вычислительным ядрам. По пересекающим его стрелкам данные передаются между ядрами

Каждый узел, изображенный прямоугольным блоком, выполняет одну скалярную операцию. Все узлы имеют один или два индекса, *i,j*, стоящие на фоне шестиугольников: *i* – номер уравнения, *j* – номер переменной. (Этот номер в нашем случае будет иметь структуру $\langle i_3, i_2, i_1 \rangle$, но в программе это никак не проявляется). По стрелкам передаются токены со значениями переменных на входы узлов, изображенные в виде желтых овалов с именами. Токен со значением *x*, идущий на вход a узла *N* с индексом *I* обозначается как $x \to N.a\langle I \rangle$.

Значение *x* определяется либо формулой, записанной внутри узла, либо формулой, стоящей на стрелке (на фоне маленького прямоугольника), либо значением первого входа. Индекс узла-получателя *I* стоит на стрелке на фоне розового шестиугольника (а по умолчанию формируется из одноименных индексов источника). У начала стрелки может стоять условие выдачи токена (например, из узла *L* токен выдается на узел Mul при $i > j$, а на узел Div при $i = j$). Узел (с конкретным индексом) активируется, когда на всех его входах присутствуют токены. В момент активации токены удаляются, а их значения (с общим индексом) становятся параметрами процедуры узла, которая будет исполнена и породит новые токены.

Входные узлы слева в форме больших желтых овалов – это источники входных данных. Аналогичный узел справа принимает выходной результат. Каждый скалярный элемент (коэффициент или значение переменной) передается или принимается отдельным токеном вместе с индексом. Над входными узлами записаны области допустимых значений индексов, но это лишь неформальные комментарии.

Матрица *L* разреженная в том смысле, что она представляется как множество токенов с ненулевыми элементами матрицы, каждый со своей парой индексов $\langle i, j \rangle$. Кроме матрицы *L* и правых частей *b* также на вход подаются число ненулевых элементов в -й строке, $k\langle i \rangle$, и число ненулевых элементов в *j*-м столбце, $m\langle j \rangle$. В отличие от матрицы *L*, вектора b, k, m плотные, т. е. содержат в виде токенов все свои элементы, включая нулевые.

Узел Sum – это особый, суммирующий узел. Он имеет суммирующий вход *s*, заданный с префиксом (+), указывающим операцию редукции, и суффиксом [*k*], указывающим, что число слагаемых придет на другой одноименный вход *k*. Узел исполняется, когда на вход s придет *k* токенов, а значение *k* придет на другой вход, *k*. Результатом становится сумма значений этих *k* токенов. Значение *k* может быть нулевым, тогда по его приходу узел Sum сразу сработает с выдачей нулевого

результата, не ожидая токенов на входе *s* (и не используя его, даже если он и придет). Здесь в качестве одного из слагаемых всегда приходит правая часть $b\langle i\rangle$. Поэтому k должно быть числом ненулевых элементов, включая диагональный.

Узел Div⟨j⟩ получает сумму *s* от узла Sum⟨j⟩ и значение диагонального элемента *d* от входного узла L⟨j,j⟩, а посылает частное s/d на узел X⟨j⟩ и на выход Xout⟨j⟩.

Еще одна особенность – токен, посылаемый узлом X⟨j⟩. Он содержит кратность #m и множественный индекс вида ⟨*,j⟩. Такой токен будет взаимодействовать.

Дана: нижне-треугольная матрица L (с диагональю) и вектор b.
Матрица L разрежена, все диагональные элементы отличны от 0.
k_i – число ненулевых элементов в строке i (включая диагональный).
m_j – число ненулевых элементов в столбце j (исключая диагональный).
Найти: вектор x, такой что $Lx=b$.
Формула решения:

$$x_i = (b_i - \sum_{j=1}^{i-1} x_j L_{ij}) / L_{ii}, \text{ для } i \text{ от } 1 \text{ до } N$$

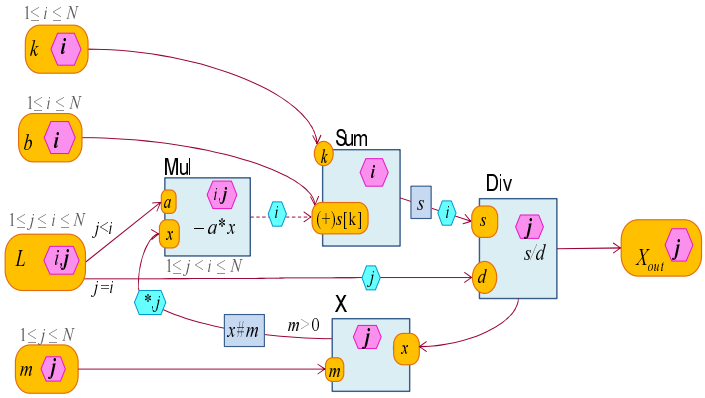


Рис. 5. Программа решения треугольной СЛАУ на графическом потоковом языке UPL

Данная программа будет хорошо и правильно работать, только если исходные данные правильные, т. е. *k* и *m* это действительно числа ненулевых элементов. В противном случае может возникнуть зависание или по окончании работы на входах некоторых узлов останется «грязь».

Применительно к нашей «сеточной» матрице эта программа «ничего не знает» ни про блоки, ни про фактические зависимости. Если матрица треугольная (при каком-либо порядке строк), то индуцируемый ею граф зависимостей ациклический, что необходимо и достаточно для завершения вычисления.

Процесс выполнения программы происходит следующим образом. Первым сработает узел Sum⟨i⟩, на который придет токен $1 \to \sum.k\langle i\rangle$ (для нижнетреугольной матрицы $i = 1$) и значение правой части $b_i \longrightarrow \sum.s\langle i\rangle$.

В результате будет выдан токен $b_i \to \div.s\langle i\rangle$, который «спарится» с токеном $L_{ii} \to \div.d\langle i\rangle$. Активированный узел Div⟨j⟩ (для $j = i$) вычислит значение переменной $x_j = s/d$, выдав токен $x_j \to X.x\langle j\rangle$. Узел X⟨j⟩ (получив также $m_j \to X.m\langle j\rangle$) выдаст токен $x_j \to Mul.x\langle ,j\rangle$. Этот токен «спарится» со всеми имеющимися токенами вида $L_{ij} \to Mul.a\langle i,j\rangle$ (с различными *i*, которых должно быть ровно m_j), всякий раз порождая активацию узла Mul, выдающую произведение $(-l_{ij} * x_j) \to \sum.s\langle i\rangle$. Оно сложится с такими же слагаемыми в узле $\sum\langle i\rangle$. И если число слагаемых достигло k_i, процесс продолжится для всех таких *i*. Процесс завершится вычислением значений всех переменных, если и только если граф зависимостей согласно матрице *L* является ациклическим.

6. Принципы работы потокового вычислителя

Схема устройства, способного вычислять потоковую программу, показана на рис. 6

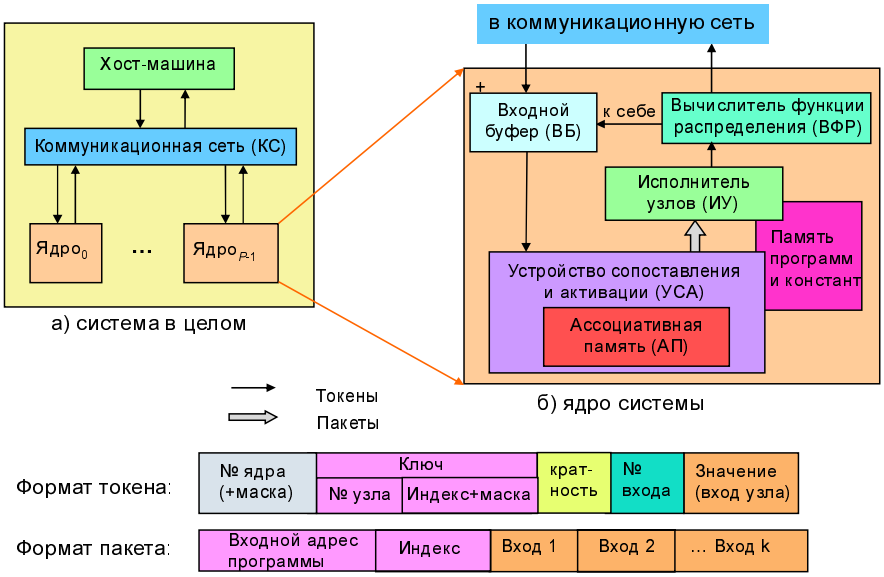


Рис. 6. Структура исполняющей системы

В каждом ядре имеется полнофункциональный вычислитель, отвечающий за свою часть адресного пространства задачи, где роль виртуальных адресов играют ключи вида ⟨имя узла, индекс⟩. (Маска ключа отмечает позиции со звездочками, что позволяет такому ключу «спариваться» с другими ключами с различными значениями в этой позиции). На основе ключа в ВФР определяется значение номера ядра, в который будет направлен токен. (При непустой маске ключа, если ФР существенно зависит от замаскированных полей, этот номер также будет иметь ненулевую побитовую маску, и тогда токен будет направлен в некоторое множество ядер [12]).

Со входа ядра (через ВБ) токены поступают в УСА, в АП которого хранятся ранее поступившие токены. Там обнаруживаются наборы из токенов, поступивших на все входы виртуального узла с конкретным индексом. Тогда этот узел активируется: токены набора изымаются из АП (при наличии кратности в токене, она уменьшается на 1 и, если получается не 0, токен остается в АП) и одновременно формируется пакет с данными для всех входов, который передается (через буфер пакетов) в ИУ, где он будет исполнен позднее. Результатом выполнения пакета будет образование одного или нескольких (возможно, ни одного) токенов, поступающих через ВФР в КС (или в свой ВБ).

7. Распределение узлов по ядрам ППВС

Каждой ячейке на рис. 2 отвечает одна (j-я) переменная и уравнение с тем же номером, из которого она определяется. В программе ей соответствует группа узлов с индексом j (или i, когда j отсутствует). На рисунке эти индексы выделены жирным шрифтом. Будет удобно (и экономно), если все узлы одной группы выполняются в одном ядре, иначе говоря, если функция распределения зависит только от выделенного индекса. Тогда только одна дуга на схеме переходит (возможно) из одного ядра в другое: это дуга из узла Mul в узел Sum, она выделена пунктиром. По ней передаются произведения j-й переменной на элементы j-го столбца матрицы.

Таким образом, одно ядро отвечает за некоторое множество таких групп, или ячеек. Будет так же экономно, если одно множество содержит тесно связанные ячейки, а связей между ними мало. Учитывая, что время передачи токена между ядрами (латентность) может быть заметно больше, чем время передачи «к себе», важно также, чтобы критические пути (на рис. 2 один такой путь вы-

делен красным) пересекали границу между ядрами малое число раз. Это критическое число обозначим как CN.

При разрезании по вертикалям $CN = n_x/d + 2n_y$. (Здесь и ниже d – ширина «полосы»). Это много. Если резать по горизонталям, $CN = n_y/d$, а если по диагоналям $i_1 + i_2 = const$, то $CN = (n_x + n_y)/d$. Уже лучше, но тогда очень медленными будут «разгон» и «торможение», и загрузка ядер будет сильно неполной в начале и конце прогона. Оптимальным будет комбинированное разрезание по красным зигзагам, показанным на рис. 4, для которого $CN = K + 2n_y/d$, где d – размер «зуба», K – число зигзагов (или число ядер).

При традиционном параллельном программировании (на OpenMP или MPI) построить такое разрезание было бы очень непросто (особенно в 3-мерном случае), и притом его пришлось бы кодировать прямо в программе – в виде сложных границ циклов и кода формирования и передачи сообщений. В нашей вычислительной модели мы должны только задать функцию распределения, вычисляющую номер ядра по индексу j. Нужная нам функция F на функциональном языке может быть записана так:

```
fun F2D(j) = let
     val j1 = j mod n
     val j2 = j / n
     val B1 = (i1+i2) / d
     val B2 = i2 / d
in
    (K*d*(B1–B2)/nx) mod K
```

Сначала переходим к двумерному индексу ⟨j1,j2⟩, затем к «блочным» координатам ⟨B1,B2⟩. Далее проекция блока на горизонтальную ось (B1–B2) нормируется к интервалу [0..K]. Здесь K – количество ядер, nx –размер сетки по горизонтали. Заключительное mod K нужно, чтобы подклеить неполные блоки слева к неполным блокам справа.

В трехмерном случае аналогичная функция будет немного сложнее:

```
fun F3D(j) = let
     val j1 = j mod nx
     val j2 = (j / nx) mod ny
     val j3 = j / (nx * ny)
     val B1 = (i1+i2+2*i3) / d
     val B2 = (i2+i3) / d
     val B3 = i3 / d
  in
    ( (K1*d*(B1–B2-B3)/nx) mod K1,
      (K2*d*(B2-B3)/ny) mod K2 )
```

Функция F3D выдает номер ядра в виде пары $\langle p_1, p_2 \rangle$, предполагая, что ядра образуют двумерную решетку. В результате пространство ячеек (с координатами $\langle j_3, j_2, j_1 \rangle$) разбивается на трехмерные блоки, имеющие форму параллелепипедов и блочные координаты $\langle B_3, B_2, B_1 \rangle$. На рис. 7 изображен блок $\langle 0,0,0 \rangle$. Блоки с $B_3 = 0$ своей нижней горизонтальной гранью лежат в горизонтальной плоскости (i2,i1), их верхняя горизонтальная грань смещена в горизонтальном направлении на вектор $\langle 0, -d, -d \rangle$.

Выходные формулы отображают блоки на ядра $\langle k_1, k_2 \rangle$ двумерной решетки. При этом нижняя (горизонтальная) плоскость 0,j2,j1 разрезается на K_1 частей по оси j_1 и K_2 частей по оси j_2, и в одну часть попадают целиком блоки, начальный угол которых проектируется (вдоль оси j_3) в эту часть. Таким образом, ячейки, выполняющиеся в одном ядре, образуют вертикальные колонны с зубчатыми гранями (в полной аналогии с двумерным случаем, изображенным на рис. 4).

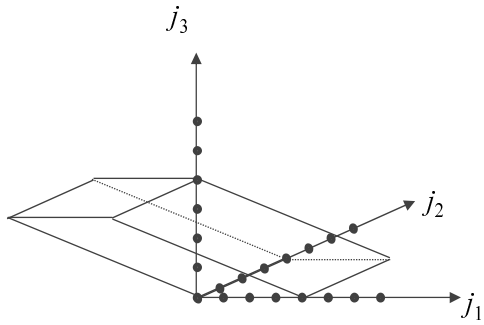


Рис. 7. Один блок $\{0,0,0\}$ разбиения трехмерной сетки на блоки ($d = 4$)

Важно отметить, что функция распределения задается вне программы, и при соблюдении определенных принципов [12] ее выбор не может нарушить правильность программы, а может только повлиять на скорость ее работы.

8. Блоки и укрупнение узлов

Смысл блоков пока состоял только в том, чтобы все узлы одного блока попали целиком в одно ядро, и чтобы близкие блоки тоже по возможности попали в то же ядро. Но есть и другой смысл: расчет всех ячеек одного блока можно задать как один узел потоковой программы, который активируется, когда для него готовы все входные аргументы. Это, во-первых, группы значений переменных, вычисленных предыдущими смежными блоками (на рис. 4 зеленым цветом выделены входные переменные для очередного блока в ядре 0) и, во-вторых, строки матрицы системы, связанные с ячейками блока. Значения, передаваемые из одного источника целесообразно объединить в один «векторный» токен. Здесь их размеры будут 3,2,5 и 36 чисел с плавающей запятой.

Такое укрупнение узлов может существенно улучшить быстродействие за счет значительного снижения накладных расходов на передачи мелких токенов и активации мелких узлов. Но за это придется заплатить резким усложнением программы узла. Мы предполагаем, что такого рода преобразования программы должны производиться автоматически [11]. Пользователю надо будет только составить несложное задание на преобразование, указав младшую часть функции распределения, характеризующую узлы, подлежащие группировке. Чтобы минимизировать передачи между ядрами, несколько смежных блочных узлов могут направляться в одно ядро – посредством старшей части той же функции распределения.

9. Распределение по времени

Строки матрицы подаются многократно для каждой итерации. Их подкачка является основным фактором, ограничивающим производительность. В ППВС можно будет задать функцию распределения по времени (этапам), согласно которой эти блоки будут по скоростному каналу заблаговременно подкачиваться в ассоциативную память ограниченного объема (аналог кэша) [13]. Быстродействие будет ограничиваться только пропускной способностью канала процессор-память.

В традиционных архитектурах аналогичный эффект можно пытаться достичь за счет сложного механизма умного предсказания будущих запросов на чтение. А в архитектуре ППВС «Буран» мы опираемся на то, что отправитель токена «знает», кому (какому узлу, с какими индексами) будут нужны передаваемые данные. И при наличии функции распределения по времени становится известным приблизительное время их потребления. Можно построить относительно несложный механизм, который будет обеспечивать своевременную подкачку токенов в рабочую ассоциативную память [13].

10. Эксперименты

Измерения проводились на параллельной блочно-регистровой модели ППВС «Буран» – ПБРМ, которая моделирует работу системы по событийному принципу с точностью до такта. «Параллельной» является моделируемая ППВС, а сама ПБРМ однопоточная, т. е. довольно медленная, и поэтому позволяет в разумное время прогонять лишь небольшие задачи.

Мы провели моделирование на двумерной задаче с сеткой 80×80 ячеек (1600 переменных). Моделирование одного прохода занимало около 2 минут. Целью моделирования было показать масштабируемость задачи с увеличением числа ядер с 2 до 8. Дальше ее нет из-за слишком малого объема работы, приходящегося на одно ядро, и растущих затрат на коммуникации. Если здесь масштабируемость есть, то можно говорить, по принципу подобия, о такой же возможной масштабируемости задачи размером 800×800 при росте количества ядер от 20 до 80.

Основным фактором и источником хорошей масштабируемости является возможность выполнять работу на фоне передач токенов между ядрами, особенно при высокой латентности (задержках) в коммуникационной сети. Поэтому мы проводим моделирование для разных значений параметра латентности сети, определяющего задержку на передачи токенов между ядрами: от 1000 до 8000 тактов. При этом пропускную способность сети, определяемую темпом передач, считаем достаточно высокой и постоянной: один токен каждые 100 тактов (на канал). Если производительность с ростом задержек не убывает, или убывает малозаметно, то это обычно называют толерантностью (нечувствительностью) к задержкам, и будет свидетельством хорошей потенциальной масштабируемости.

На рис. 8 приведены графики зависимости времени работы (в тактах) от величины задержки при передаче токенов между ядрами (тоже в тактах). Мы провели две серии экспериментов с двумя разными функциями распределения:

· Light: $P(i) = i\,1/(nx/K)$,

· Hard: $P(i) = (K * d * (B1 - B2)/nx)\%K$, где $B1 = (i1 + i2)/d$, $B2 - i\,2/d$.

Функция Light дает разрезание области на прямые вертикальные полосы, а функция Hard соответствует вертикальным зигзагам на рис. 4. Здесь K – число ядер, $d = 5$. Для каждой функции имеется несколько кривых, соответственно для 2, 4, 6 и 8 ядер.

С ростом задержки общее время растет, сначала медленно, потом быстрее. Участок медленного роста – это и есть область толерантности к задержке. Для функции Light она почти сразу исчезает (для 2х ядер от 2000, а для 4 и более – сразу от 1000). Для функции Hard толерантность заметно лучше: для 2-х ядер вплоть до задержки в 7000 тактов, для 4, 6 и 8 ядер, соответственно, до 5000, 3000 и 1000). Стартовые позиции (при нулевых задержках) у обеих функций примерно одинаковы, но затем сложная Hard заметно опережает простую Light.

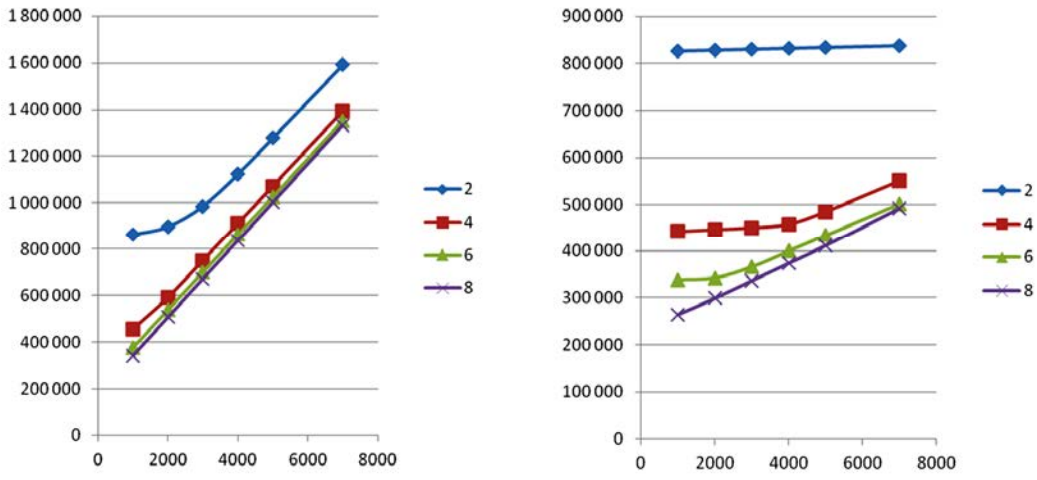


Рис. 8. Время работы в тактах в зависимости от задержки L при передаче между ядрами для различных конфигураций по числу ядер K (меньше – лучше). Варианты Light и Hard

На рисунке также можно видеть для функции Hard заметный прирост быстродействия с ростом числа ядер до 6, а при малых задержках до 8 (там, где велико расстояние между графиками). В то же время при функции Light прирост заметен только до 4 ядер. Таким образом, мы здесь видим не только пределы масштабируемости (роста быстродействия с увеличением числа ядер), но и основную причину ее раннего прекращения – латентность коммуникационной сети (зависящую от выбранной функции распределения).

11. Заключение

Рассмотрена задача распараллеливания симметричного сглаживателя Гаусса-Зейделя, применяемого в тесте HPCG. В отличие от обычного явного распараллеливания с разбиением на прямоугольные блоки и вычислениями по методу Якоби на уровне блоков (а по методу Гаусса–Зейделя только внутри блоков) мы добиваемся распараллеливания вычислений целиком по методу Гаусса-Зейделя (что ведет к лучшей сходимости метода) просто за счет выполнения на принципах потока данных. Для улучшения масштабируемости мы применяем разбиение области на косые параллелограммы (параллелепипеды). Такая декомпозиция слишком сложна для традиционного программирования средствами MPI. Но в потоковой модели вычислений, свойственной архитектуре разрабатываемой системе ППВС «Буран», такая нетривиальная декомпозиция, достигается просто путем написания функции распределения в несколько строк. И важно, что это дополнение не является частью основной программы и не может испортить ее корректность. Ошибки в нем могут ухудшить время работы, но не могут привести к неправильному результату.

Пока мы испытали этот подход только на потактовой модели на небольшой задаче для двумерного случая и только для одной итерации решения треугольной системы. Результаты обнадеживают. И теперь надо провести эксперименты с трехмерной сеткой.

Однако, для практического применения остается еще много нерешенных проблем. Вот главные:

Для устранения избыточных накладных расходов, связанных с мелкозернистым параллелизмом, разработать компилятор, который сможет автоматически породить укрупненные узлы, соответствующие блокам-параллелепипедам.

Доработать механизм распределения по времени, обеспечивающий высокоскоростную своевременную подкачку отложенных токенов из большой обычной памяти (на кристалле или вне его) в рабочую ассоциативную память небольшого размера.

Перейти от потактовой программной модели к реальной системе ППВС «Буран» в кремнии.

Благодарности

Авторы признательны Ю. Г. Бартеневу за обсуждение и полезные советы.

Литература

1. Dongarra J., Heroux M.A., Luszczek P. High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems // The International Journal of High Performance Computing Applications. 2015. Volume: 30 issue: 1. P. 3–10.

2. Стемпковский А. Л., Левченко Н. Н., Окунев А. С, Цветков В. В. Параллельная потоковая вычислительная система – дальнейшее развитие архитектуры и структурной организации вычислительной системы с автоматическим распределением ресурсов // Информационные технологии 2008. № 10. С. 2–7.

3. Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л. Вопросы применения и реализации потоковой модели вычислений // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. Вып. 2. С. 100–106. URL: http://www.mes-conference.ru/data /year2016/pdf/D114.pdf.

4. Chang K. Understanding and Improving the Latency of DRAM-Based Memory Systems. – PhD thesis. – Carnegie Mellon University, Pittsburgh, PA – May, 2017. URL: https://arxiv.org/pdf/1712.08304.pdf .

5. Brezina M., Hu J., Tuminaro R. Algebraic multigrid // Encyclopedia of parallel computing. David Padua, ed. – Springer, 2011. P. 23–33.

6. Ahmadi A., Khademi A., Smith M.C. A parallel Jacobi-embedded Gauss-Seidel method // IEEE Transactions on Parallel and Distributed Systems. – Vol. 32, No 2 – June, 2021. P. 1452–1464.

7. Shang Y. A distributed memory parallel Gauss-Seidel algorithm for linear algebraic systems // Computers & Mathematics with Applications. Vol. 57, Issue 8. – April, 2009. P. 1369–1376.

8. Adams M., Brezina M., Hu J., Tuminaro R. Parallel multigrid smoothing: polynomial versus Gauss-Seidel // J. Comput. Phys. – 188 (2003). P. 593–610.

9. Климов А. В., Окунев А. С. Графический потоковый метаязык для асинхронного распределенного программирования // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2016. № 2. С. 151–158. URL: http://www.mes-conference.ru/data/year2016/ pdf/D149.pdf .

10. HPCG Benchmark [Electronic resource]: URL: https://www.hpcg-benchmark.org/software.

11. Klimov Ark.V. On the Possibility of Parallel Programs Synthesis from Algorithm Graph Specification // Proceedings of the 21st Conference on Scientific Services & Internet (SSI-2019), Novorossiysk-Abrau, Russia, September 23-28, 2019. – CEUR Conf. Proc. – Vol. 2543, 2020. P. 219–233. URL: http://ceur-ws.org/Vol-2543/rpaper20.pdf.

12. Климов Арк.В. Средства верификации распределения вычислений в потоковой архитектуре ППВС «Буран» // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС) – 2020. Вып. 4. С.236–243. URL: http://www.mes-conference.ru/data/year2020/pdf/D107.pdf.

13. Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л. Суперкомпьютеры, иерархия памяти и потоковая модель вычислений // Программные системы: теория и приложения: электрон. научн. журн. – 2014. Т. 5. № 1(19). С. 15–36. URL: http://psta.psiras.ru/read/ psta2014_1_15-36.pdf.

FEATURES OF HPCG BENCHMARK IMPLEMENTATION FOR THE “BURAN” PDCS

D. N. Zmejev, A. V. Klimov, A. S. Okunev, N. N. Levchenko

Institute for Design Problems in Microelectronics RAS, Zelenograd

HPCG (High Performance Conjugate Gradients) is a new benchmark for comparing supercomputer systems after LINPACK HPC. It is believed to better represent the workload properties of real scientific applications. We consider the implementation of its central part, the Gauss-Seidel smoother, on our projected parallel dataflow computing system (PDCS) "Buran". Unlike the HPCG reference code for MPI that uses a mixed Jacobi – Gauss-Seidel method, we implement the pure Gauss-Seidel method in our proposed dataflow language UPL providing automatic extraction of dataflow parallelism. Therefore, we violate the benchmark conditions by using a different computation algorithm that partially compensates high network and memory latencies.

Key words and phrases: dataflow computation model, dataflow architecture, parallel computing, high-performance computing, HPCG benchmark, conjugate gradients method, Gauss-Seidel method.

УДК 004.4
DOI 10.53403/9785951505071_2022_206

ПРОБЛЕМЫ СОЗДАНИЯ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПРОГРАММИРОВАНИЯ ЛОГИКИ ПОВЕДЕНИЯ АГЕНТОВ В МУЛЬТИАГЕНТНЫХ СИСТЕМАХ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДВУХСТОРОННИХ БОЕВЫХ ДЕЙСТВИЙ

К. В. Иванов, М. В. Галкин, А. И. Сайфуллин, Р. Н. Сайфуллина, Д. В. Девятых

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе рассмотрены ключевые аспекты создания инструментальных средств программирования логики агентов в мультиагентных системах и озвучены проблемы, возникающие при разработке. Выполнен краткий обзор существующих инструментальных средств программирования логики агентов, в котором отмечены как зарубежные, так и отечественные программные продукты.

Основной акцент доклада – это имитационное моделирование двухсторонних боевых действий, ориентированное на специалистов предметной области. Для этого выделены три типа агентов: «тактическая единица», «тактическая группа» и «объединение». Показана структура их связи и взаимодействия, сформулированы особенности и специфика для каждого типа агентов. Для каждого типа агентов предложено решение, определяющее облик наиболее подходящего инструментального средства для программирования логики их поведения, который был бы удобен для применения специалисту предметной области.

Ключевые слова: имитационное моделирование, агент, мультиагентная система, программирование, логика поведения, система управления.

Введение

В современной литературе встречается множество определений понятия «мультиагентная система». Однако, в контексте затрагиваемых данным докладом проблем, целесообразно обратиться к определению Дж. Люгера, который под мультиагентной системой (МАС) понимал вычислительную программу, решатели которой расположены в некоторой среде и каждый из них способен к гибким, автономным и социально-организованным действиям в направлении предопределенной цели [1]. В данном докладе рассматривается подкласс мультиагентных систем – мультиагентные системы имитационного моделирования (МСИМ) двухсторонних боевых действий (ДБД).

Термин «агент» происходит от латинского глагола «agere», что означает «действовать», «двигать», «править», «управлять». В энциклопедическом словаре Ф. А. Брокгауза и И. А. Ефрона дано следующее определение: «агент – деятель, лицо, действующее по поручению или полномочию другого» [2]. Это определение правильно выражает суть агента МСИМ, который может функционировать автономно от имени своего владельца (человека-оператора) и решать разнообразные задачи по обработке информации. Для успешной работы агент должен обладать достаточными интеллектуальными способностями, должен иметь возможности взаимодействия с владельцем для получения заданий и передачи результатов, должен ориентироваться в среде своего существования и принимать необходимые решения.

Две базовые характеристики – автономность и целенаправленность – позволяют отличать агента от других программных и аппаратных объектов (модулей, подпрограмм, процедур и т. п.). Авто-

номность агента означает его способность взаимодействовать со средой без прямого участия других агентов, для чего он должен уметь контролировать свое внутреннее состояние и выполняемые действия. Целенаправленность означает, что соответствующий агенту программный код выполняется не автоматически, а по решению самого агента в зависимости от ситуации и согласно поставленным перед агентом целям.

Наличие целесообразности поведения требует, чтобы агент обладал свойством реактивности. Такой уровень интеллекта соответствует рефлекторному поведению животного. Если же агент обладает знаниями о среде, собственных целях и способах их достижения, то такой агент может быть назван разумным (когнитивным). Таким образом, может быть проведена граница между интеллектуальными и неинтеллектуальными агентами.

Еще одна характеристика агента – коммуникабельность, предполагает необходимость решения своих задач совместно с другими агентами и наличие развитых протоколов коммуникации.

Обзор существующих инструментальных средств программирования логики агентов

Перечень систем, ориентированных на решение задач, связанных с программированием логики агентов, довольно обширен. Приведем наиболее известные проекты: AgentBuilder [3], Ascape [4], Bee-gent [5], Cable [6], Decaf [7], FIPA-OS [8], Grasshopper [9], Gypsy [10], JADE [11], JASON [12], JATLite [13], JAFMAS [14], MAML [15], ProcessLink [16], Swarm [17], Zeus [18].

Среди отечественных разработок большую известность получили инструментальные средства группы интеллектуальных систем Санкт-Петербургского института информатики и автоматизации РАН, возглавляемой В. И. Городецким [19]. Концептуальной основой развиваемой коллективом В. И. Городецкого технологии разработки МАС является выделение «общности» реализуемых функций и представление их в виде классов и структур данных, являющихся составной частью программного инструментария. Реализуемая в этом подходе идея состоит в выделении наибольшего количества классов и структур данных, повторно используемых в агентных приложениях. В системе MAS DK создана конструкция «Типовой агент» («Generic Agent»), являющаяся базой для последующей специализации классов и структур данных, клонирования экземпляров агентов и индивидуальной доработки специфических компонентов. Этот процесс поддерживается соответствующим набором редакторов.

Отметим, что подавляющее большинство из приведенных выше систем ориентированы на разработку МАС “с нуля” и, по сути, представляют собой платформы для создания мультиагентной системы разработчиком-программистом. Нашей же целью является определение принципов и подходов к созданию инструментария, позволяющего не разработчику, а конечному пользователю МСИМ БДБ производить наполнение системы агентами, обладающими требуемой логикой поведения. При этом пользователь не должен отвлекаться на решение сопутствующих, общих проблем, связанных с созданием МАС (разработка пользовательских интерфейсов, средств коммуникаций агентов и т. п.), и может не являться программистом высокой квалификации. Исходя из этого, указанные системы целесообразно рассматривать с точки зрения языков и семантик, применяемых для программирования логики агентов.

Так, например, в системе JASON реализована операционная семантика, формально определенная для языка AgentSpeak(L). Данный язык определяет агента через спецификацию множества базовых убеждений и множества планов. Атомарное убеждение является обычным предикатом первого порядка. Атомарные убеждения или их отрицания обозначаются соответствующими литералами. Начальное множество убеждений является набором базовых атомарных убеждений. В языке AgentSpeak(L) определяются два типа целей: достижимые цели и тестовые цели. Достижимая цель устанавливается тогда, когда агент стремится достичь определенного состояния мира, в котором связанный предикат является истинным. Практически для этого инициируется выполнение вложенных планов. Тестовая цель возвращает унификацию для связанного предиката с одним из убеждений агента, в случае неудачи такая цель отвергается. Переключающее событие может инициировать выполнение некоторого плана. Событие может быть внутренним, когда достигнута определенная

подцель, или внешним, порождаемым в результате модификации убеждений после восприятия состояния внешней среды. Планы используют базовые действия, которые агент способен выполнить в среде. Действия также определяются как предикаты первого порядка, но обозначаются специальными символами. План образуется переключающим событием, за которым следует конъюнкция литералов, образующих контекст. Контекст должен быть последовательностью таких текущих убеждений агента, для которых данный план применим. Остаточная часть плана есть последовательность базовых действий и целей, которые агент достигает (или проверяет), когда план выбран для исполнения.

Тело плана – это последовательность операций, реализующих предназначение плана:

– математических операций;

– вызовов внутренних функций JASON;

– действий, направленных на объекты среды функционирования МАС;

– коммуникационных актов.

Обобщенная синтаксическая конструкция плана выглядит следующим образом:

«Метка

Имя события : Контекстные ограничения <- Тело плана.»

Программы агентов интерпретируются циклически. В каждом цикле обновляется список событий, порождаемых либо перцепцией из внешней среды, либо выполнением подцелей, определенных в теле плана. Далее интерпретатор унифицирует выбранное событие с переключающими событиями в заголовках планов, определяя, таким образом, множество релевантных планов. Проверяя содержимое контекста этих планов, интерпретатор определяет множество применимых планов и выбирает один применимый план из этого множества. Когда все формулы в теле плана оказываются выполненными, то цель считается достигнутой и план также удаляется из очереди планов. Цикл работы интерпретатора повторяется [20].

Ниже приведен пример программы на языке AgentSpeak(L) в которой вычисляется факториал числа 5:

```
fact(0,1).
+fact(X,Y): X < 5
<- +fact(X+1, (X+1)*Y).
+fact(X,Y): X == 5
<- .print("fact 5 == ", Y).
```

Программа содержит два разных плана, которые активируются при возникновении одного и того же убеждения fact(X,Y), но в разных контекстных условиях - «X < 5» и «X == 5». Благодаря наличию начального убеждения «fact(0,1).» и первому плану, у агента будут последовательно формироваться следующие убеждения:

```
fact(1,1).
fact(2,2).
fact(3,6).
fact(4,24).
fact(5,120).
```

При возникновении последнего убеждения первый план уже не активируется, зато активируется второй план, который выведет результат вычисления факториала.

Типы агентов в мультиагентных системах имитационного моделирования двухсторонних боевых действий

Рассмотренный выше язык AgentSpeak(L), безусловно, является мощным средством программирования логики агентов. Однако, в отличие от языков программирования «общего назначения» (C++, Java, Python и т. д.), язык AgentSpeak(L) имеет высокую степень специализации применительно к рассматриваемой предметной области, достаточно универсален в рамках этой предметной области, и, как следствие, довольно сложен в практическом применении.

Для того чтобы определить требования к инструментарию программирования логики агентов МСИМ БДБ, который был бы удобен для применения специалисту данной предметной области, попытаемся систематизировать типы применяемых агентов и определить характерные решаемые ими задачи. Рассмотрим наиболее общий случай, когда необходимо моделировать действия каждой отдельно взятой тактической единицы в связке с многоуровневой системой управления (СУ) силами. В качестве тактической единицы будем рассматривать элементарную, с точки зрения моделируемой системы управления, сущность. Такими сущностями могут являться корабль, самолет, рота (взвод, отделение – в зависимости от того, какая требуется степень детализации) солдат. Тактической группой будем называть несколько однотипных тактических единиц, объединенных под единым командованием для решения общей тактической задачи. Тактические группы и тактические единицы могут входить в состав объединений, которые представляют собой формирования, создаваемые для решения задач оперативно-тактического, оперативного и оперативно-стратегического уровней. Одни объединения могут входить в состав других.

Таким образом, можно выделить три типа агентов, между которыми выстроена иерархия с точки зрения системы управления (см. рис. 1): тактические единицы, тактические группы, объединения.



Рис. 1. Иерархия типов агентов МСИМ БДБ

Охарактеризуем задачи, которые решают агенты указанных типов в процессе моделирования.

Тактическая единица, действуя самостоятельно или в составе тактической группы, решает тактические задачи, поставленные перед ней вышестоящим командованием. В роли командира могут выступать либо агенты типа «Тактическая группа» и «Объединение», либо человек, который формирует команды, задавая исходные данные моделирования (сценарии). Человек также может выступать в роли командира тактической единицы и отдавать приказы на выполнение различных тактических задач, если применяется автоматизированный режим моделирования, который допускает интерактивное вмешательство человека в процесс моделирования. Таким образом, тактическая единица должна уметь решать устанавливаемые ей тактические задачи, действуя автономно (принимая самостоятельно решения в установленных рамках, планируя и осуществляя действия, меняющие окружающую среду), целенаправленно и коммуникабельно (см. «Введение»). Поскольку автономность подразумевает способность агента взаимодействовать со средой, тактическая единица должна обладать сенсорами, с помощью которых она сможет получать информацию о состоянии среды. Тактическая единица должна обладать способностью восприятия и анализа информации, поставляемой сенсорами, в процессе выполнения тактической задачи. Целенаправленность подразумевает, что информация, полученная агентом от сенсоров или других тактических единиц, используется для анализа текущей ситуации и выбора способа действия для достижения цели, заданной в рамках выполняемой тактической задачи. Коммуникабельность подразумевает способность агента решать свои задачи совместно с другими агентами, например, получая приказы от агента типа «тактическая группа» или «объединение».

Тактическая группа представляет собой объединение нескольких тактических единиц. Исходя из того, что в нашем случае каждая тактическая единица представлена отдельным агентом, т. е. сущностью, имеющей собственную логику поведения, тактическую группу целесообразно рассматривать в виде агента, формирующего и уточняющего индивидуальные задачи для тактических единиц. При этом агент действует в соответствии с приказами, получаемыми от вышестоящих в иерархии системы управления агентов, а также использует информацию о среде, получаемую от собственных сенсоров, сенсоров подчиненных тактических единиц, либо поставляемую другими агентами.

Объединения представляют собой формирования, состоящие из тактических единиц, тактических групп и других объединений. Поэтому, как и в случае с тактическими группами, их целесообразно представлять в виде агентов, формирующих и уточняющих задачи для подчиненных агентов. Логика поведения агента «объединение» в некотором смысле дублирует логику поведения агента «тактическая группа» – и в том и в другом случае производится анализ информации, поступающей от подчиненных тактических единиц, средств и системы управления, и на основе этого анализа формируются задачи для подчиненных объектов. Различаются лишь масштабы собираемых данных и подчиненных сил.

Таким образом, мы видим, что агенты всех выделенных нами типов призваны решать задачи, поставляемые вышестоящими в иерархии системы управления агентами. В процессе решения этих задач агенты типа «объединение» и «тактическая группа» осуществляют управление подчиненными агентами, а агенты типа «тактическая единица» управляют собственным состоянием, задавая параметры своего передвижения в пространстве, а также параметры приданных им средств обнаружения, поражения и связи.

Программирование логики агентов типа «тактическая единица»

Поскольку назначение агента «тактическая единица» состоит в решении тактических задач, поставленных перед ним вышестоящим командованием, будем считать, что логика поведения агента определена в том случае, если заданы алгоритмы решения всех возможных (в рамках конкретного модельного эксперимента) тактических задач. Как было отмечено выше, алгоритм решения тактической задачи для агента данного типа должен определять порядок перемещения тактической единицы в пространстве и управления ее системами в интересах достижения поставленных в задаче целей. При этом необходимо обеспечить анализ информации о текущей обстановке и реакцию на ее изменение в процессе решения задачи.

Применительно к рассматриваемой нами предметной области, упомянутые выше алгоритмы, как правило, строго регламентированы боевыми уставами. Например, алгоритмы движения подводных лодок (ПЛ) и использования ими вооружения в процессе выполнения поставленных задач четко описаны и легко представимы в виде блок-схемы. То же самое касается алгоритмов поведения ПЛ в ситуациях, которые могут возникнуть в процессе выполнения поставленных задач в результате изменения боевой обстановки. Введем термин «поведенческие модели», объединяющий алгоритмы поведения агента в рамках решения конкретных тактических задач и алгоритмы реакций на изменение обстановки.

Учитывая вышесказанное, для программирования поведенческих моделей удобным представляется инструмент, который бы позволил оператору формировать алгоритмы в виде графической схемы, основными элементами которой являются:

– процессоры – реализуют микропрограммы управления ТЕ («движение с заданными параметрами» и т. п.);

– функции – обеспечивают расчет входных параметров для процессоров («расчет параметров движения» и т. п.);

– связи – определяют условия и порядок передачи управления между процессорами и функциями.

Собранная по такому принципу схема может выглядеть следующим образом:

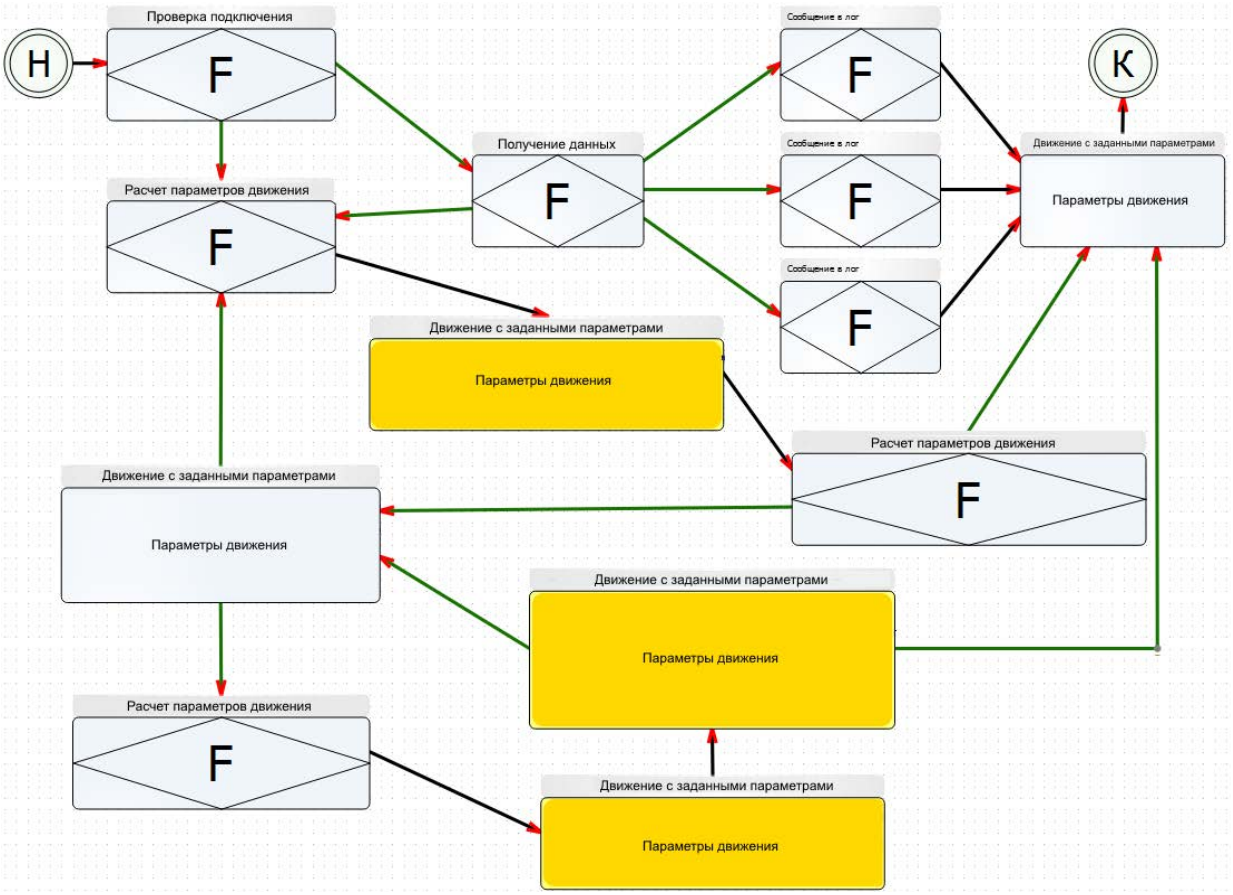


Рис. 2. Пример схемы алгоритма поведения агента типа «тактическая единица»

Для полного описания логики работы агента «тактическая единица», кроме алгоритмов его поведения в рамках решения конкретных тактических задач и реакций, необходимо задать правила обработки информации об изменении окружающей обстановки, поступающей от сенсоров, и активации того или иного алгоритма решения задачи или реакции. Назовем эти правила термином «логика системы управления». В процессе работы системы управления можно выделить следующие этапы:

1. Обработка информации, поступающей от сенсоров, и распознавание событий, имеющих семантическое значение в рамках описания логики работы агента (получение приказа, обнаружение цели и т. п.)

2. Анализ информации о произошедших событиях, текущем состоянии агента и окружающей среды, выбор актуальной поведенческой модели

3. Активация актуальной поведенческой модели

4. Алгоритмы этапов 1 и 2 могут быть довольно сложными, требующими для удобного описания таких языковых конструкций как циклы, ветвления, множественный выбор и т. п. Поэтому в качестве инструмента программирования логики системы управления целесообразно рассматривать язык общего назначения (например, Python), расширенный библиотеками проблемно-ориентированной направленности. В нашем случае эти библиотеки могут содержать такие функции как:

– опрос локальных сенсоров;
– отправка сообщения по каналу радиосвязи;
– получение сообщений по каналу радиосвязи;
– получение текущей активной поведенческой модели;
– активация поведенческой модели;
– расчет расстояния до точки на карте;
– получение модельного времени;

и т. п.

Очевидно, что рассмотренные ранее поведенческие модели дополняют логику системы управления и вместе с ней определяют полную логику поведения агента «тактическая единица». Тем не менее, принимая во внимание указанные особенности алгоритмизации, авторам представляется целесообразным выделение для программирования поведенческих моделей и логики системы управления отдельных интерфейсных модулей, обладающих вышеприведенными свойствами.

Программирование логики агентов типа «тактическая группа»

Введенные нами понятия «поведенческие модели» и «логика системы управления» можно применить и к агентам типа «тактическая группа». Задачей агентов данного типа является формирование и уточнение индивидуальных задач для подчиненных тактических единиц, исходя из получаемых приказов и информации о среде. Учитывая это, в качестве поведенческих моделей агентов «тактическая группа» будем рассматривать алгоритмы назначения частных задач подчиненным тактическим единицам и расчета входных параметров для этих задач, а в качестве логики системы управления - алгоритмы обработки приказов и информации, получаемой от подчиненных тактических единиц, активации поведенческих моделей.

При таком подходе, относительно желаемого образа инструмента программирования логики системы управления агентов типа «тактическая группа» можно сделать выводы, аналогичные выводам, сделанным для агентов «тактическая единица». Так же, как и в случае с агентами «тактическая единица», здесь целесообразно применение языка общего назначения, расширенного библиотеками проблемно-ориентированной направленности. Состав функций этих библиотек должен быть аналогичен составу, приведенному для агентов типа «тактическая единица», но дополнен функциями, позволяющими обрабатывать информацию о расположении, свойствах и текущем состоянии подчиненных тактических единиц:

– получение списка подчиненных тактических единиц;
– получение информации о расположении подчиненной тактической единицы;
– получение текущей задачи подчиненной тактической единицы;
– получение текущего состояния подчиненной тактической единицы;

и т. п.

Поскольку в процессе выполнения поведенческих моделей агентов данного типа в основном производятся различного рода вычисления, подразумевающие оценку текущей обстановки и расчеты значений входных параметров подчиненных тактических единиц, для программирования их логики также целесообразно применение языка общего назначения с расширениями, обеспечивающими доступ к геоинформационным данным, информации о тактико-технических характеристиках и текущем состоянии подчиненных тактических единиц.

Программирование логики агентов типа «объединение»

Как было отмечено выше, задачи агентов типа «объединение» схожи с задачами агентов типа «тактическая группа». Но если агент типа «тактическая группа» решает задачи тактического уровня и при принятии решения оценивает обстановку в рамках нескольких подчиненных тактических единиц, то агент типа «объединение» решает задачи оперативно-тактического, оперативного и оперативно-стратегического уровней и при принятии решений оценивает обстановку в рамках нескольких тактических групп.

С повышением уровня управления возрастает сложность формализации действий агента, осуществляющего управление. Если для тактического уровня алгоритмы управления и действий в большинстве случаев четко определены боевыми уставами, то в оперативном искусстве определения таких алгоритмов отсутствуют. Основной целью оперативных маневров является создание превосходства над противником на главном направлении в операции за счет изменения состава группировок, оперативного построения сил, использования резервов, переноса усилий на другое

операционное направление. Таким образом, можно сказать, что критерием достижения основной цели оперативных маневров является обеспечение превосходства над противником на главном направлении, причем этот критерий не меняется от одной задачи к другой, в отличие от случая с тактическими маневрами, в которых критерии достижения основной цели определяются индивидуально по каждой задаче. В процессе моделирования агенту типа «объединение» необходимо непрерывно производить анализ всей имеющейся информации о своих силах и силах противника с целью проверки выполнения условия достижения основной цели – превосходства над противником на главном направлении, и если обнаружено, что это условие не выполнено, обеспечить его выполнение вышеперечисленными способами.

Тем не менее, основной проблемой при создании инструмента программировании логики поведения агента типа «объединение» остается отсутствие возможности четкой формализации действий при принятии решения и определения детерминированных алгоритмов. В качестве решения данной проблемы может быть рассмотрено применение математического аппарата нечеткой логики и построенных на его основе нейронных сетей. Задачу управления силами можно декомпозировать на две основные подзадачи: анализ текущего соотношения сил на главном направлении и принятие компенсирующих мер для обеспечения выгодного для себя соотношения сил на главном направлении, исходя из имеющихся сил и средств. Первая подзадача может быть решена посредством простого сравнения сил по выделенным критериям, либо с применением более сложной методики оценки боевой эффективности. Для решения второй задачи целесообразно применение экспертной системы, дополненной нейронной сетью. Экспертная система (ЭС) позволит выработать решение на основе базы данных решающих правил. Если ЭС столкнется с ситуацией, которую эксперты не рассмотрели заранее, нейронная сеть может отнести непредвиденную ситуацию к одной из рассмотренных ранее. В том случае, если ситуацию не удалось отнести к одной из ранее рассмотренных, нейронная сеть, исходя из имеющихся сил и средств, а также доступной информации о противнике, определит наиболее выгодную, с точки зрения решаемой оперативной задачи конфигурацию сил, при которой будет обеспечено превосходство над противником на главном направлении. В этом случае будет определено новое правило.

В то время как экспертные системы используют правила импликации и логический вывод, нейронные сети имеют способность к обучению. Эта совокупность качеств делает нейронные сети и экспертные системы достойными претендентами на формирование гибридной интеллектуальной системы [21].

Таким образом, технология программирования логики поведения агента будет сводиться к процессам заполнения базы данных решающих правил и обучения нейронной сети.

Для экспертной системы должны быть определены сущности предметной области (цели, способы решения, возможные варианты событий и др.) и измеряемые характеристики (вероятности наступления событий, коэффициенты значимости целей, приоритетность решений и др.). При описании правил целесообразно использовать следующие базовые понятия: объект, показатель (признак) и процедура сравнения. В нашем случае объектами являются агенты типа «объединение», «тактическая единица», «тактическая группа». В качестве показателей могут быть использованы пространственно-временные характеристики, физические свойства объектов (степень поражения объекта, количество боеприпасов и т. д.). Процедура сравнения должна учитывать: причинно-следственные связи между объектами, степень влияния одних объектов на другие.

Основными типами алгоритмов обучений нейронных сетей [22] являются: обучение с учителем, обучение с подкреплением, обучение без учителя. В нашем случае целесообразно применение именно алгоритмов обучения с подкреплением. Данный подход позволит решить проблемы отсутствия данных о крупных боевых столкновениях, ограниченности доступа к результатам и эффективности учений потенциального противника, что делает обучение с учителем принципиально невозможным без размеченных наборов данных, отвечающих требованиям репрезентативности. Анализ литературных источников подтверждает высокий уровень интеллектуальных способностей агентов, обучаемых с подкреплением (системы AlphaGo [23], DeepMind [24], OpenGym AI [25]). И хотя агенты, обученные с помощью перечисленных программных продуктов, не использовались для детального моделирования двусторонних боевых действий, те принципы, на которых строились

модели принятия решений агентов, могут быть реализованы и для решения актуальных задач имитационного моделирования с использованием библиотек для машинного обучения с открытым исходным кодом и не накладывающих лицензионных ограничений.

В ходе обучения с подкреплением программный агент совершает наблюдения и предпринимает действия внутри среды моделирования, возвращающей награды и штрафы. Цель агента – максимизация долгосрочных наград и сокращение штрафов, это довольно абстрактная установка, конкретная реализация которой может достигаться с использованием таких приемов машинного обучения как градиенты политики и глубокие Q-сети. Ниже приведен распространенный вариант алгоритма REINFORCE (REward Increment = non-Negative Factor × Offset Reinforcement × Characteristic Eligibility):

- Запуск моделирования, при котором для политики в форме нейросетевой модели вычисляются (но не применяются) градиенты, определяющие изменение вероятности совершения выбранного действия;
- После прогона нескольких эпизодов вычисляются оценки каждого действия, например, с использованием алгоритма дисконтной ставки;
- Перемножение вектора градиента и оценки действия для изменения вероятностей совершения действия;
- Применение результирующих векторов-градиентов для изменения весовых коэффициентов нейросетевой модели с использованием алгоритмов оптимизации (в простейшем случае градиентного спуска).

Одной из проблем данного подхода является оценка действий, ввиду задержки в получении наград от среды. Совершив ряд действий и, достигнув определенного результата, агент должен оценить, какие из совокупности действий приводили к достижению результата, а какие мешали. Очевидно, что конечный результат определяется не последним действием, а всей совокупностью. Одним из распространенных подходов решения проблемы назначения доверительных коэффициентов является использование дисконтной ставки, при которой оценка совокупности действий определяется по следующей формуле:

$$N = \sum_{t} n_t \times r^t,\ i = 0 \ldots (M-1), 0 \leq r \leq 1.$$

где N – оценка совокупности действий; M – количество совершенных агентом действий, n_i – награда за отдельное действие, r – значение дисконтной ставки. Чем выше значение дисконтной ставки, тем более значимыми будут предыдущие действия.

Заключение

В результате проведенного анализа подходов к построению инструментальных средств программирования логики поведения агентов в мультиагентных системах имитационного моделирования двухсторонних боевых действий была выполнена систематизация типов агентов и выстроена иерархия с точки зрения системы управления. Установлено, что с повышением уровня иерархии возрастает степень абстрактности применяемых алгоритмов. Для каждого типа агентов предложено решение, определяющее облик наиболее подходящего инструментального средства для программирования логики их поведения:

Для агентов типа “тактическая единица” – конструктор на основе графических схем;

Для агентов типа “тактическая группа” – язык общего назначения, расширенный библиотеками проблемно-ориентированной направленности:

Для агентов типа “объединение” – заполнение базы данных решающих правил и обучение нейронной сети.

Литература

1. Люгер Д. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.

2. Энциклопедический словарь Ф. А. Брокгауза и И. А. Ефрона. – СПб.: Полрадис, 1993.

3. AgentBuilder: An Integrated Toolkit for Constructing Intelligent Software Agents [Electronic Resourse] / Reticular Systems, 1999. – Mode of access : http://www.agentbuilder.com.

4. Brookings R. S. Ascape: An Agent Based Modeling Framework in Java [Elec-tronic Resourse] / R. S. Brookings. – [Electronic Data]. - Brookings Center on Social and Economic Dynamics, 2000. – Mode of access : http://www.brook.edu/es/dynamics/models/ascape/.

5. Bee-gent Multi-Agent Framework [Electronic Resourse] / Toshiba Corporation Systems and Software Research Laboratories, 2000. [Electronic Data]. – Mode of access: http://www2.toshiba.co.jp/beegent/index.htm.

6. Dee, C. CABLE: A multi-agent architecture to support military command and control / C. Dee, P. Millington, B. Walls, T. Ward // Proceedings of PAAM 2000. -Manchester, UK, April 2000. P. 322–330.

7. Graham, J. Tools for Developing and Monitoring in Distributed Multi-Agent Systems / J. Graham, V. Windley, D. McHugh, F. McGeary, D. Cleaver, K. Decker // Pro-ceedings of the Workshop on Agents in Industry. Barcelona, Spain, June 2000. – http://www.eecis.udel.edu/~decaf/.

8. Poslad, S. J. The FIPA-OS agent platform: Open Source for Open Standards [Electronic Resourse]/ S. J. Poslad, S. J. Buckle, R. Hadingham // Proceedings of PAAM 2000. – Electronic Data. – Manchester, UK, April 2000. – P. – Mode of access : http://fipa-os.sourceforge.net.

9. Grasshopper, Release 2.2. Basics and Concepts (Revision 1.0), March 2001. – IKV++GmbH. – 70 p.

10. Jazayeri M. Gypsy: A Component-based Mobile Agent System [Electronic Re-sourse] / M. Jazayeri, W. Lugmayr // Proceedings of the 8th Euromicro Workshop on Parallel and Distributed Processing (PDP2000). – Electronic Data. – Rhodos, Greece, 2000. – http://www.infosys.tuwien.ac.at/Staff/lux/Gypsy.

11. Bellifemine, F. JADE – A FIPA-compliant agent framework [Electronic Re-sourse] / F. Bellifemine, A. Poggi, G. Rimassa // Proceedings of PAAM 2000. – Electronic Data. – Manchester UK, April 2000. – P. – Mode of access : http://sharon.cselt.it/projects/jade.

12. Bordini R. H., Hubner J. F., Wooldridge M. Programming Multi-Agent Systems in AgentSpeak with Jason. – Jonh Wiley&Sons: Chichester, 2007. – 294 p.

13. JATLite – http : // java.stanford.edu/java-agent/html.

14. JAFMAS – http://www.ececs.uc.edu/~abaker/JAFMAS.

15. Raphael M. J. A knowledge base for knowledge-based multiagent system con-struction / M. J. Raphael, S. A. Deloach. // National Aerospase and Electronics Conf. (NAECON), Dayton, OH, October 10-20, 2000. – http://www.cis.ksu.edu/~sdeloach/publications/Conference/arams-naecon.pdf .

16. Kouadri, G. A generic framework for context-based distributed authorizations/ G. Kouadri, P. Brezillon // Proc. Of the fourth Int. Conf. on Modeling and Using Context, Stanford, California (USA), June 23–25, 2003. P. 326–333.

17. Reference Guide for Swarm 2.1.1. [Electronic Resourse] / Swarm Develop-ment Group. – Electronic Data. – Mode of access : http:// www.santafe.edu/projects/swarm/swarmdocs/set/book930.html.

18. Collis, J. The Zeus Agent Bilding Toolkit : ZEUS Technical Manual [Electron-ic Resourse] / J. Collis, D. Ndumu // Intelligent Systems Research Group, BT Labs, Release 1.0. 1999. – Electronic Data. – Mode of access : http://193.113.209.147/projects/agents/index.htm.

19. Городецкий В. И. MAS DK: инструментарий для разработки многоагентных систем и примеры приложений / В. И. Городецкий, О. В. Карсаев, И. В. Хотенко, А. В. Хабалов // Труды Международного конгресса «Искусственный интеллект в XXI веке» (ICAI 2001), 3–8 сентября 2001 г. М.: Физматлит, 2001. С. 249–262.

20. Швецов А. Н. Агентно-ориентированные системы: от формальных моделей к промышленным приложениям. // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008. С. 75–76.

21. Алдошина А. Н. Экспертная система на основе нейросетевых технологий для мониторинга и диагностики корпоративной локальной сети // Молодой ученый – 2016. № 18(122). С. 37.

22. Hastie T., Tibshirani R. and Friedman J. “Elements of Statistical Learning”, Springer, 2009.

23. Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016) doi:10.1038/nature16961.

24. Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerch-ner. 2017. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. In Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research), Vol. 70. PMLR, 1480–1490.

25. Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym.(2016). arXiv:cs.LG/1606.01540.

PROBLEMS OF DEVELOPING TOOLS FOR THE AGENT BEHAVIOR LOGIC PROGRAMMING IN MULTIAGENT SYSTEMS FOR SIMULATION OF TWO-SIDED MILITARY OPERATIONS

K. V. Ivanov, M. V. Galkin, A. I. Sayfullin, R. N. Sayfullina, D. V. Devyatykh

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper considers the key aspects of developing tools for the agent behavior logic programming in multiagent systems and problems encountered in the development process are articulated. A brief review of available agent logic programming tools, including both foreign and Russian software products, is given.

The paper is focused on the simulation of two-sided military operations for experts in this subject. Three types of agents are identified: a tactical unit, a tactical group, and a combination. The agent communication and interaction structure, as well as the specific features of each agent type are presented. The solution is suggested for each type of agents, which determines the most relevant and easy to use tool for experts in this area to program the agent behavior logic.

Key words: simulation, agent, multiagent system, programming, behavior logic, control system

УДК 519.8
DOI 10.53403/9785951505071_2022_217

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ НЕПРЕРЫВНЫХ ФУНКЦИЙ БОЛЬШОЙ РАЗМЕРНОСТИ НА ОСНОВЕ ГИБРИДИЗАЦИИ АЛГОРИТМА ОПТИМИЗАЦИИ «СЕРЫХ ВОЛКОВ»

О. В. Коваленко, Д. В. Ежов, И. А. Крючков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Биоинсперированный алгоритм «Серых волков» [1] хорошо себя показал на задачах оптимизации большой размерности [2]. В статье рассматривается улучшение точности версии алгоритма «серых волков» из [2] на овражных функциях Розенброка за счет добавления в рой серых волков дополнительной роли – движения части волков по сходящейся спирали, которая была позаимствована из другого биоинсперированной алгоритма «Желтой седловой барабульки» [4].

Ключевые слова: оптимизация, биоинспирированный алгоритм, алгоритм серых волков, функция Розенброка, экстремум, алгоритм желтой барабульки.

1. Описание проблемы

Алгоритм оптимизации «Серых волков» (GWO – Grey Wolf Optimization) [1] – это один из недавних биоинспированных алгоритмов оптимизации, основанный на имитации загонной охоты стаи серых волков. В среднем алгоритм очень хорошо зарекомендовал себя на различных типах функций, как по точности поиска экстремума, так и по скорости сходимости. Особенно поражает скорость сходимости алгоритма. Практически уже при 100 итерациях алгоритм находит приемлемое решение на всех тестовых функциях и на любой размерности задачи. Основные показатели эффективности алгоритма оптимизации «Серых волков» лучше, чем у алгоритма оптимизации роем частиц, который в классе биоинсперированых алгоритмом оптимизации является «классическим». При этом вычислительная сложность алгоритм оптимизации «Серых волков» [2] сравнима с алгоритмом оптимизации роем частиц. Ввиду многочисленных достоинств алгоритма оптимизации «Серых волков» за короткое время с момента первой публикации алгоритма [1] появилось много работ по его модификации. Особо привлекает модификация алгоритма, предложенная в работе [2] для поиска экстремума высокоразмерных функций с числом независимых переменных равным 10000. Единственным недостатком алгоритма, предложенного в работе [2], является сравнительно невысокая точность найденных координат минимума функции Розенброка. Причем невысокая точность найденного экстремума проявилась на всех размерностях функции Розенброка. Хотя функция Розенброка унимодальная (она имеет один минимум), но она является «овражной» функцией [3], на которой подавляющее большинство алгоритмов поиска экстремума (включая классические градиентные методы) многомерных функций дают низкую точность решения, которая практически не улучшается с увеличением числа итераций.

В варианте для N переменных функция Розенброка определяется следующим образом:

$$f(X) = \sum_{i=1}^{N-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2], \quad \forall x \in \mathbb{R}^N \qquad (1)$$

Эта функция имеет один глобальный минимум, равный 0, когда $x_i = 1, i \in [1, N]$, где N – размерность.

Предпринимались попытки улучшить качество решения данного алгоритма и избежать «застревания» в локальном экстремуме на мультимодальных функциях, например, были введены весовые коэффициенты при расчете координат каждого «волка» на следующем шаге, как описано в [2]. Тем не менее, эти попытки давали лишь процентное улучшение качества решения на различных классах тестовых функциях, и наиболее неточными выглядят решения для «овражной» функции.

2. Канонический вид алгоритма Серых волков

В каноническом алгоритме каждый волк в стае является текущей точкой пространства поиска решений, с вычисленным значением фитнес функции. По качеству решения вся стая делятся на 4 основных видов. По-одному волку (носители текущих наилучших трех решений) составляют категорию лидеров. Лидеры в оригинальном алгоритме называются $\alpha,\ \beta,\ \delta$. все остальные волки составляют категорию ω.

Математически идея загона жертвы статей волков выглядит следующим образом. Позиция каждого волка в стае на $t+1$ шаге алгоритма определяющего окружение жертвы можно записать как

$$X_i(t+1) = X_{p,i} - A_i \cdot \left| C_i \cdot X_{p,i} - X_i(t) \right|, i = 1, \dots, N \quad (2)$$

t – номер итерации, i – номер компоненты N мерного пространства, $\vec{X}(t)$ – вектор позиции рассматриваемого волка, $\overrightarrow{X_p}$ – вектор позиции рассматриваемой жертвы, которая символизирует оптимум фитнес функции и которая неизвестна.

$$A_i = 2a \cdot r_{1,i} - a \quad (3)$$
$$C_i = 2r_{2,i},$$

$r_{1,i}$ и $r_{2,i}$ – случайные величины, которые равномерно распределены на отрезке $[0,1]$, и разыгрываются независимо для каждой компоненты i N – мерного пространства, a некоторый коэффициент, который линейно уменьшается от 2 к 0 с ростом номера итерации по закону

$$a(t) = 2 - \frac{2t}{MaxIter}, \quad (4)$$

где $MaxIter$ - заданное максимальное число итераций в алгоритме.

Так как глобальный оптимум не известен, то организуются следующая сходящаяся процедура.

Пусть волки волков $\alpha,\ \beta,\ \delta$ характеризуют три лучшие текущие на шаге t позиции. Тогда новые позиции на шаге $t + 1$ для всех волков рассчитываются по следующим уравнениям.

Для каждого волка с ролью ω считаются свои вектора $\vec{X}_1, \vec{X}_2, \vec{X}_3$. Они вычисляются с учетом известных позиций волков $\alpha,\ \beta,\ \delta$ по следующим формулам

$$X_{1,i}^{(k)}(t+1) = X_i^{(\alpha)}(t) - A_i^{(\alpha)} \cdot \left| C_i^{(\alpha)} \cdot X_i^{(\alpha)}(t) - X_i^{(k)}(t) \right| \quad (5)$$

$$X_{2,i}^{(k)}(t+1) = X_i^{(\beta)}(t) - A_i^{(\beta)} \cdot \left| C_i^{(\beta)} \cdot X_i^{(\beta)}(t) - X_i^{(k)}(t) \right| \quad (6)$$

$$X_{3,i}^{(k)}(t+1) = X_i^{(\delta)}(t) - A_i^{(\delta)} \cdot \left| C_i^{(\delta)} \cdot X_i^{(\delta)}(t) - X_i^{(k)}(t) \right| \quad (7)$$

где i – номер компоненты N мерного пространства, k – номер волка с ролью ω, тогда итоговая позиция любого рассматриваемого волка определяется как

$$\vec{X}(t+1) = \frac{\vec{X}_1(t+1) + \vec{X}_2(t+1) + \vec{X}_3(t+1)}{3} \quad (8)$$

Алгоритм 1

Вход: Популяция размером *M*, общее количество итераций $MaxIter$, *N* – размерность пространства поиска.

1. Случайно инициализируется позиции *M* волков $\vec{X}_i(i = 1,2,\dots,M)$ в области поиска, инициализируется $t = 0$ – текущая итерация

2. while $(t < MaxIter)$

2.1. Рассчитывается фитнес функция для каждого волка $y_i = f(\vec{X}_i)\ (i = 1,2,\dots,M)$

2.2. Выбирается три наилучших решения из $\{y_i, i = 1,2,\dots N\}$ и определяются им соответствующие $\vec{X}^{(\alpha)}$, $\vec{X}^{(\beta)}$, $\vec{X}^{(\delta)}$

2.3. *for i* = 1:*M* % по каждому индивиду

for j = 1:*N* % по каждой компоненте размерности пространства

рассчитываются X_1, X_2, X_3 по (5)–(7),при новых r_1 и r_2,

рассчитываются компонента вектора новой позиции волка по (8)

end

end

2.4. $t = t + 1$

end while

3. return $\vec{X}^{(\alpha)}$

Заметим, что для того чтобы выполнялся на последней итерации шаг с $a(MaxIter) = 0$, в псевдокоде итераций ложно быть не $MaxIter$, а $MaxIter$ +1, т. е. в строчке 2 псевдокода необходимо написать *while* $(t < MaxIter + 1)$. Это как правило повышает точность окончательного решения на большинстве тестовых функциях.

Модификация ERGWO

В работе [2] оригинальный алгоритм из [1] был модифицирован и хорошо проявил себя на большеразмерных задачах. Модификации имели следующий вид. Вместо формулы (4) на итерациях параметр *a* менялся по закону

$$a(t) = a_{initial} - (a_{initial} - a_{final})\mu^{-t}, \tag{9}$$

где $a_{initial}$ – начальное значение параметра, которое принимается 2 как в оригинальном алгоритме, a_{final} – финальное значение параметра, которое принимается равным 0, $\mu \in [1.0001,1.005]$. $t \in \{0,1,2,\dots,MaxIter\}$.

В отличие от формулы (4), формула (9) выглядит неправильной, она дает значение a_{final} при $t = 0$ и далее стремится к значению $a_{initial}$, т. е. $\lim_{t\to\infty} a(t) = a_{initial}$. Это наводит на мысль, что в статье [2] была допущена опечатка. Чтобы сохранить убывающий тренд зависимости оригинальной статьи [1], скорее всего формула (9) должна была бы выглядеть как

$$a(t) = a_{final} + (a_{initial} - a_{final})\mu^{-t}, \tag{10}$$

Тогда, чтобы получить $a(MaxIter) = 0$, a_{final} должна быть отрицательной и вычисляться по формуле

$$a_{final} = -\frac{a_{initial}}{(\mu^{MaxIter} - 1)} \tag{11}$$

Проведенные тесты показали, что можно использовать более простую формулу для адаптации параметра *a* с ростом номера итерации

$$a(t) = 2\mu^{-t}, \tag{12}$$

При этом использование формул (4). (10) и, (11) дают примерно одинаковый результат, который зависит от общего числа итераций. В среднем предпочтительно использовать оригинальную формулу (4) или упрощенную (12).

Однако и следующие изменения оригинального алгоритма, сделанные в [2], выглядят как ошибка.

Вместо формулы (8), которая дает координату волка, как среднеарифметическое значение между координатами, посчитанными по направлением к волкам лидерам α, β, δ, в статье [2] предложено рассчитывать усреднение по формуле:

$$\vec{X}(t+1) = \frac{w_1\vec{X}_1(t+1) + w_2\vec{X}_2(t+1) + w_3\vec{X}_3(t+1)}{3}, \quad (13)$$

где

$$w_1 = \frac{|\vec{X}_1(t+1)|}{|\vec{X}_1(t+1)| + |\vec{X}_2(t+1)| + |\vec{X}_3(t+1)|}, \quad (14)$$

$$w_2 = \frac{|\vec{X}_2(t+1)|}{|\vec{X}_1(t+1)| + |\vec{X}_2(t+1)| + |\vec{X}_3(t+1)|}, \quad (15)$$

$$w_3 = \frac{|\vec{X}_3(t+1)|}{|\vec{X}_1(t+1)| + |\vec{X}_2(t+1)| + |\vec{X}_3(t+1)|}. \quad (16)$$

Формулы (13)–(16) выглядят как ошибка потому, что они зависят от выбора системы отсчета, а формула (13) является взвешенным усреднением, дополнительно деленным на 3.

Удивительным является, что так введенное «усреднение» улучшило точность оригинального алгоритма.

Нетрудно увидеть, что формула (13), по сути, корректирует делитель в формуле (8). Вместо делителя 3, выражения (13)–(16) соответствуют делителю 9 в формуле (8) на последних итерациях, где $w_1 \approx w_2 \approx w_3$, так как из-за сходимости в конце итерационного процесса $\vec{X}_1 \approx \vec{X}_2 \approx \vec{X}_3$.

Увеличение делителя в формуле (8) резко ускоряет сходимость на начальных итерациях, особенно при большеразмерных задачах. На средних размерностях опытным путем было получено, что в среднем замена делителя 3 на делитель 4 в формуле (8) улучшает результат.

Однако окончательная точность алгоритма на многих тестовых функциях остается хуже, чем у алгоритма PSO при увеличении числа итераций до 500. Очевидно, что агентам в алгоритме GWO и его модификациях явно не хватает исследовательской составляющей, особенно во второй половине итерационного процесса.

Идея алгоритма желтой седловой барабульки

В работе [4] новый алгоритм поиска глобального оптимума, инспирированный поведением желтой седловой барабульки (Yellow Saddle Goatfish). В алгоритме все особи разбиваются на заранее заданное количество групп. В каждой группе есть основные роли: 1) одна рыба преследователь (chaser fish) и 2) рыбы блокировщики (blocker fish).

В n-мерном пространстве поиска новая позиция рыбы блокировщика определяется по формуле

$$\Phi_l^{(i)}(t+1) = \Phi_l^{(i)}(t) + \gamma \cdot Levy_{t,l}^{(i)}(\lambda), \quad (17)$$

где l – номер группы, $Levy(\lambda)$ является случайным полетом Леви, который определяется по формуле

$$Levy_{t,l}^{(i)}(\lambda) = \frac{u_l}{|v_l|^{1/\lambda}}\left(\vec{\Phi}_l(t) - \vec{\Phi}_{best}(t)\right), \quad (18)$$

$\vec{\Phi}_{best}(t)$ – наилучшее на шаге t решение, u_l и v_l – нормально распределенные случайные величины, разыгрываемые для каждой пространственной компоненты i независимо

$$u = N(0, \sigma_u^2),$$
$$v = N(0, \sigma_v^2), \quad (19)$$

$$\sigma_u = \left(\frac{\Gamma(1+\lambda) sin\frac{\pi\lambda}{2}}{\Gamma\left(\frac{1+\lambda}{2}\right) \lambda 2^{(\lambda-1)/2}} \right)^{1/\lambda}, \quad \sigma_v = 1, \tag{20}$$

где $\Gamma(x)$ гамма функция.

Заметим, что соотношение (18) дает нулевое смещение для наилучшего решения. В статье [4] предложено наилучшее текущее решение подвергать случайному смещению, где каждая компонента вектора позиции независимо смещается как

$$S = \gamma \frac{u}{|v|^{1/\lambda}}. \tag{21}$$

Предложение спорное и нуждается в численной проверке. Скорее всего, оно играет дестабилизирующую роль, уводя наилучшего преследователя от глобального оптимума.

Нас интересует уравнение изменения координат для индивидов второй роли. Они двигаются по случайной логарифмической спирали вокруг рыбы преследователя в группе согласно уравнению

$$\varphi_i^{(g)}(t+1) = D_i^{(g)}(t) \cdot e^{b\rho_i} \cdot cos2\pi\rho_i + \Phi_i^{(i)}(t), \tag{22}$$

где i – номер пространственной компоненты, g – индекс рыбы блокировщика в рассматриваемой группе, l – номер группы, $\vec{\Phi}_i(t)$ – позиция рыбы преследователя в группе l, ρ случайная величина, равномерно распределенная на отрезке $[a, 1]$, который определяет, как близко блокировщик к рыбе-преследователю, рассматривая путь по спирали. Для усиления эксплуатационных свойств а линейно уменьшается от 1 до –2 по мере того, как номер итерации увеличивается. Позиция рыбы преследователя достигается, когда $\rho = -2$. Следовательно, самая близкая позиция к рыбе преследователю получается, когда $\rho \approx -2$, а самая дальняя, когда $\rho = 1$. Уравнение (22) применяется для каждой компоненты вектора позиции блокировщика. То есть для каждой компоненты разыгрывается случайная величина ρ и случайная величина r, входящая в формулу (23).

Параметр b в уравнении (22) константа, которая определяет форму и направление спирали, в этом подходе b = 1. Член Dg является расстоянием между текущим положением рыбы блокировщиком φ_g^l и рыбой преследователем $\boldsymbol{\Phi}_l$ в кластере $\boldsymbol{c}_l$, описанным как:

$$D_i^{(g)}(t) = \left| r_i \cdot \Phi_i^{(i)}(t) - \varphi_i^{(g)}(t) \right|, \tag{23}$$

где r_i случайное число, равномерно распределенное на отрезке $[-1,1]$.

На каждом шаге внутри каждой группы выбирается наилучшее решение. Рыбе с наилучшим решением в кластере назначается роль преследователь. Среди преследователей выбирается наилучшее глобальное решение.

Предложенная модификация алгоритма серых волков

В данной работе предложен параллельный алгоритм, основанный на гибридизации варианта алгоритма оптимизации «Серых волков», предложенного в работе [2], и алгоритма поведения Желтой Седловой Барабульки (Yellow Saddle Goatfish behavior) [4], за счет добавления новой роли в алгоритм серых волков, поведение которой соответствует поведению рыб блокировщиков из алгоритма [4].

Алгоритм «Серых волков» получил следующие нововведения:

1. В рой добавлена новая роль s – волки загоняющие жертву по сходящейся логарифмической спирали. Волки с это ролью начинают двигаться в многомерном пространстве (плоскость движения при этом получается как результат разигрывания случайных величин в формуле (21) и (22) вокруг предполагаемого (текущего) наилучшего решения, число (процент) волков, двигающихся по спирали, является динамической величиной;
2. Измерен алгоритм перераспределения ролей;
3. Экспериментально подобрано оптимальная доля волков, двигающихся по спирали
4. Число лидеров может быть переменной величиной.

Переменное число волков-лидеров

Вместо формул (5)–(8) для каждого индивида с ролью ω будем использовать следующие формулы

$$X_{\zeta,i}^{(k)}(t+1) = X_i^{(\zeta)}(t) - A_i^{(\zeta)} \cdot \left| C_i^{(\zeta)} \cdot X_i^{(\zeta)}(t) - X_i^{(k)}(t) \right|, \quad (24)$$

где ζ – номер лидерской роли, $\zeta \in \{\alpha, \beta, \delta, \dots\}$,

$$X_i^{(k)}(t+1) = \sum_{\zeta \in \{\alpha,\beta,\delta,\dots\}}^{L} w_\zeta X_{\zeta,i}^{(k)}(t+1), \quad (25)$$

L – количество лидеров в стае, w_ζ – вес лидерской роли с нормировкой в виде неравенства

$$\sum_{\zeta \in \{\alpha,\beta,\delta,\dots\}}^{L} w_\zeta \leq 1. \quad (26)$$

Веса подбираются опытным путем. Мы предлагаем использовать для трех ролей веса $w_\alpha = w_\beta = w_\delta = \frac{1}{4}$

Уравнение движения для волков с ролью s рассчитывается согласно уравнению

$$X_i^{(g)}(t+1) = D_i^{(g)}(t) \cdot e^{b\rho_i} \cdot cos 2\pi\rho_i + X_i^{(\alpha,\beta,\dots,L)}(t+1), \quad (27)$$

g номер волка с ролью s, $X_i^{(\alpha,\beta,\dots,L)}(t+1)$ – позиция жертвы, $D_i^{(g)}(t)$ – рассчитывается по формуле аналогичной формуле (23)

$$D_i^{(g)}(t) - \left| r_i \cdot \Phi_i^{(i)}(t) - X_i^{(\alpha,\beta,\dots,L)}(t+1) \right|. \quad (28)$$

Позиция жертвы рассчитывается по формуле

$$X_i^{(\alpha,\beta,\dots,L)}(t+1) = \sum_{\zeta \in \{\alpha,\beta,\delta,\dots\}}^{L} w_\zeta X_i^{(\zeta)}(t+1). \quad (29)$$

Сортировка волков

Алгоритм сортировки волков включает в себя следующее:

1. Определение координат предполагаемого наилучшего решения;
2. Определения координат всех волков, включая волков-лидеров;
3. Определение расстояния от всех волков до предполагаемого наилучшего решения;
4. Сортировка волков таким образом, что, волки, расположенные дальше всего от предполагаемого оптимального решения, принимают роль ω. Волки не вошедшие в число лидеров, и находящиеся ближе к оптимальному решению принимают роль s.

Данное улучшение позволяет динамически «перетасовывать» волков так, что самые дальние начинают подтягиваться к остальным, не оставаясь слишком далеко и, тем самым, не выпадая из числа волков, имеющих шанс найти лучшее решение, чем предполагаемое на данной итерации.

Определение доли волков двигающихся по спирали

Одним из нововведений стал внедренный алгоритм расчета оптимального числа волков, двигающихся по спирали.

В качестве исходной точки эмпирически был найден условно-оптимальный процент волков, двигающихся по спирали согласно уравнению (27), от общего числа – 0.9. Далее, в качестве изначального значения устанавливается оно. В процессе итераций, если значение фитнесс-функции не улучшается (меняется) в течение определенного числа итераций, процент волков с ролью s начинает

изменяться в диапазоне [0.85, 0.95]. Эксперименты показали, что выход за пределы этого диапазона с вероятностью p > 0.9 не приводит к улучшению качества решения.

При изменении значения фитнесс-функции (улучшении решения) процент волков, двигающихся по спирали, фиксируется на том значении, на котором произошло улучшение.

Алгоритм 2

Вход: Популяция размером *M*, общее количество итераций $MaxIter$, *N* –размерность пространства поиска.

1. Случайно инициализируются позиции *M* волков $\vec{X}_1(i = 1,2,\dots,M)$ в области поиска, инициализируется $t = 0$ – текущая итерация

2. while $(t < MaxIter)$

2.1. Распределение популяции по процессам

2.2. Вычисляется фитнес функция для каждого волка $y_t = f(\vec{X}_t)\ (i = 1,2,\dots,M)$.

2.3. Выбирается *L* наилучших решений из $\{y_t, i = 1,2,\dots N\}$ и определяются им соответствующие $\vec{X}^{(\zeta)}(t), \zeta \in \{\alpha, \beta, \delta, \dots, L\}$.

2.4. *If* (решение не улучшилось в течение определенного числа итераций):

Динамически меняем процент волков с ролью s (растет доля из не лидеров от 0.85 до 0.95 с шагом 0.01)

end if

2.5. *for i* = 1:*M* % по каждому индивиду

for j = 1:*N* % по каждой компоненте размерности пространства

If(*i* < *L* + 1)

по формуле (24) и (25) рассчитываются новые позиции лучших решений

новые

else If(*L* < *i* < *S*) % *S* – номер упорядоченных волков, отсекающий роль *s* от роли ω

рассчитываются $X_j^{(i)}(t+1)$ по формулам (28) и (27),

else

рассчитываются $X_{\alpha,j}^{(i)}, X_{\beta,j}^{(i)}, X_{\delta,j}^{(i)}, \dots$ по формуле (24), при новых r_1 и r_2

рассчитываются компонента вектора новой позиции волка по (26)

end If

end for

If(i==L+1)

По формуле (30) рассчитывается позиция жертвы

end If

end for

2.6. $t = t \mid 1$

end while

3. *return* $\vec{X}^{(\alpha)}$

3. Полученные результаты

В результате внесенных изменений в алгоритм, точность решения значительно повысилась на всех классах функций и на всех размерностях тестовых задач. Размерность задач варьировалась от 2 до 20000.

Рассмотрим улучшение качества решения на примере функции Розенброка. Результаты представлены в табл. 1 и на рис. 1 (для наглядности применена логарифмическая шкала).

Таблица 1

Результаты расчетов

Размерность	Без новой роли s	С ролью s	Коэффициент улучшения решения (разы)
2	0,467702	0,000003	155900,6667
4	2,996389	0,000018	166466,0556
40	38,974482	0,0077	5061,621039
400	398,99956	0,011349	35157,24381
1000	998,999865	0,007255	137698,1206
2000	1999	0,000888	2251126,126
3000	2998,999983	0,013721	218570,0738
4000	3998,975809	0,026884	148749,2862
5000	4998,989612	0,019844	251914,413
6000	5999	0,000036	166638888,9
7000	6998,991559	0,000184	38037997,6
8000	7998,979025	0,008828	906091,8696
9000	8998,982006	0,002281	3945191,585
10000	9998,98238	0,006632	1507687,331
20000	19998,962761	0,000616	32465848,64

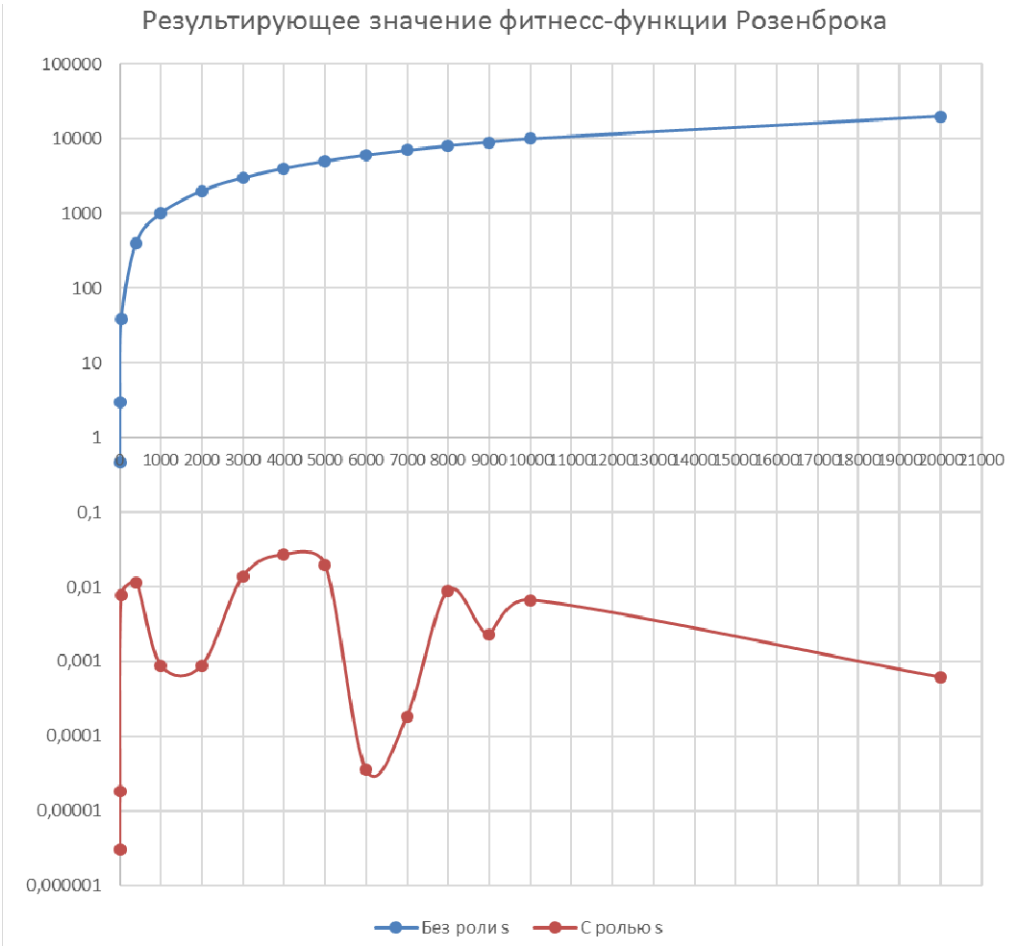


Рис. 1. График зависимости качества решения

На рис. 2 приведены графики зависимости счетного времени от размерности задачи для функции Розенброка.

На рис. 2 видно, что время расчета алгоритма без учета расчетов фитнес функции имеет очень пологий, почти линейный вид.

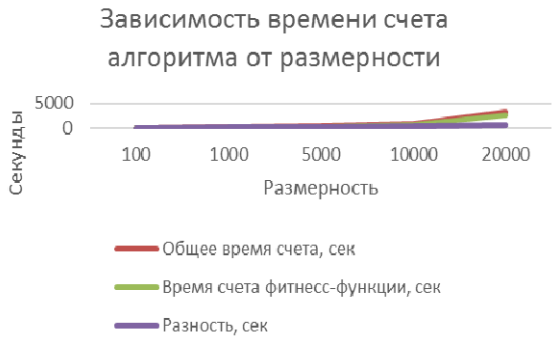


Рис. 2. Зависимость времени счета от размерности задачи для алгоритма в целом, для расчета фитнес функции и их разности

Заключение

В результате описанных выше модификаций удалось значительно улучшить качество конечного решения.

В дальнейшем планируется численно определить значимость s и ω ролей в точности алгоритма. Не до конца выяснены наилучшие динамические зависимости для констант алгоритма и числа лидеров.

Литература

1. Mirjalili S., Mirjalili S. M., Lewis A. «Grey wolf optimizer» // Advances in Engineering Software. 2014. Vol. 69. P. 46–61.

2. Long W., Cai S., Jiao J., Tang M. «An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization» // Soft Computing. – 2019. Vol. 3.

3. Rosenbrock H. H. «An automatic method for finding the greatest or least value of a function» // The Computer Journal. 1960. T. 3. P. 175–184.

4. Zaldivar D., Morales B., Rodriguez A., Valdivia-G A., Cuevas E., Pérez-Cisneros M. «A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior» // BioSystems. 2018. Vol. 174. P. 1–21.

PARALLEL ALGORITHM TO OPTIMIZE CONTINUOUS HIGH-DIMENSIONAL FUNCTIONS ON THE BASIS OF HYBRIDIZATION OF THE GREY WOLF OPTIMIZER

O. V. Kovalenko, D. V. Ezhov, I. A. Kryuchkov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

A bio-inspired Grey Wolf Optimization (GWO) algorithm [1] is well-reputed in problems of high dimensionality optimization [2]. The paper describes accuracy improvement of the Grey Wolf algorithm option from [2] on Rosenbrock «ravine» functions due to giving an additional role to the swarm of grey wolves – movement of a part of wolves along the converging spiral borrowed from another bio-inspired Yellow Saddle Goatfish algorithm [4].

Key words: optimization, bio-inspired algorithm, Grey Wolf algorithm, Rosenbrock function, extremum, Yellow Saddle Goatfish algorithm.

УДК 532.5
DOI 10.53403/9785951505071_2022_226

МОДЕЛИРОВАНИЕ ЗАДАЧ ГИДРОДИНАМИКИ НА СЕТКАХ, СОДЕРЖАЩИХ НЕСОГЛАСОВАННЫЕ СЕТОЧНЫЕ ИНТЕРФЕЙСЫ

А. В. Коротков, С. В. Лашкин, А. С. Козелков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Данный доклад посвящен описанию универсального сеточного интерфейса (УСИ), реализованного в составе пакета программ ЛОГОС на базе алгоритма работы GGI (General Grid Interface) [1], и его адаптации для расчета задач гидродинамики на несогласованных интерфейсах методом SIMPLE [2,3].

В докладе приводятся примеры использования УСИ при расчете задач на сеточных моделях, состоящих из фрагментов неструктурированной несогласованной сетки. Показано, что реализованный алгоритм позволяет производить расчет на сеточных моделях, состоящих из несогласованных фрагментов сетки, при этом не уступает по качеству и скорости счета аналогичным расчетам на геометрически сшитой сетке.

Ключевые слова: General Grid Interface, универсальный сеточный интерфейс, SIMPLE, ЛОГОС, несогласованный сеточный интерфейс.

Введение

В настоящее время значительно возросла потребность в численном моделировании гидродинамики в сложных промышленных объектах с использованием моделей большой размерности. Это связано, в первую очередь, с тем что уровень развития современных вычислительных систем позволяет за приемлемое время получать результаты с учетом все большего количества физических факторов на все более подробных сетках. Расчетные сетки таких моделей могут содержать более миллиарда расчетных точек (ячеек).

Опыт показывает, что построение такого рода сеточных моделей достаточно трудоемко, ресурсоемко и, как правило, занимает много времени. При решении инженерных задач процесс оптимизации конструкции часто требует внесения изменений в исходную геометрию, что автоматически требует полного перестроения сеточной модели и существенно увеличивает время разработки.

Одним из основных способов решения данной проблемы, является разбиение сложной исходной геометрии на более простые фрагменты и построение сетки отдельно для каждого геометрического фрагмента, с учетом особенностей протекающих физических процессов. Такой подход упрощает задачу и позволяет получить сетки лучшего качества. Например, выборочное сгущение сетки в отдельных фрагментах, где это необходимо, дает возможность получить более точное решение без существенного увеличения количества ячеек в сеточной модели. В качестве примера можно привести конструкции, в которых, присутствуют детали существенно меньшего, по сравнению со всей областью, размера, такие как маленькие препятствия и тонкие каналы. Необходимо отметить, что сеточная модель, полученная путем разбиения исходной геометрии на отдельные фрагменты, будет иметь несогласованные сеточные интерфейсы на смежных границах сеточных фрагментов.

Для обеспечения возможности расчета задач на моделях, содержащих несогласованные внутренние сеточные интерфейсы, в составе пакета программ ЛОГОС был разработан и реализован алгоритм универсального сеточного интерфейса (УСИ). В основу УСИ был положен принцип GGI [1],

позволяющий установить связь между смежными областями сетки, содержащими несогласованные сеточные интерфейсы. Основными преимуществами разработанного УСИ являются отсутствие требований к выравниванию топологии на поверхностях стыкуемых областей сеток и возможность счета на сеточных моделях с частично пересекающимися границами на поверхностях стыкуемых областей сеток.

Данный доклад посвящен описанию УСИ и его адаптации для расчета задач гидродинамики на несогласованных интерфейсах методом SIMPLE. В докладе приводятся примеры использования УСИ. Показано, что реализованный алгоритм позволяет производить расчет на сеточных моделях, состоящих из несогласованных фрагментов сетки и при этом не уступает по качеству и скорости счета аналогичным расчетам на геометрически сшитой сетке.

1. Исходные уравнения и метод дискретизации

Нестационарные трехмерные турбулентные течения вязкого теплопроводного газа описываются системой уравнений, включающей уравнения неразрывности, Навье–Стокса и энергии. В случае осреднения данной системы по Рейнольдсу [4] ее можно записать в неконсервативной (полудивергентной) форме:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla(\rho u) = 0; \\ \rho\left(\dfrac{\partial u}{\partial t} + (u\nabla)u\right) - \nabla\tau = -\nabla p; \\ \dfrac{\partial(\rho H)}{\partial t} - \dfrac{\partial p}{\partial t} + \nabla(\rho u H) + \nabla q = \nabla(u\tau), \end{cases} \qquad (1)$$

где ρ – плотность моделируемой среды, t – время, $\boldsymbol{u} = \{u, w, v\}$ - вектор скорости осредненного течения в трехмерной декартовой системе координат, $\boldsymbol{\tau} = \boldsymbol{\tau}_{\mu} + \boldsymbol{\tau}_{t}$ – сумма молекулярной и турбулентной составляющих вязкой части тензора напряжений, p – давление, H – полная энтальпия, $\boldsymbol{q} = \boldsymbol{q}_{\mu} + \boldsymbol{q}_{t}$ – сумма молекулярной и турбулентной составляющих вектора плотности теплового потока.

Для решения системы уравнений (1) в основном используют три подхода: методы конечных разностей, элементов и объемов [2–4].

Оптимальным методом дискретизации уравнений (1) является метод конечных объемов. Метод конечных (контрольных) объемов – это численный метод интегрирования систем дифференциальных уравнений, который обладает хорошими консервативными свойствами, основным преимуществом которого является возможность счета на произвольных неструктурированных сетках. Данный метод дискретизации используется в пакете программ ЛОГОС.

Процесс дискретизации методом конечных объемов включает в себя два основных этапа – дискретизация расчетной области и решение дискретного аналога дифференциальных уравнений (1).

На этапе дискретизации расчетная область разбивается на конечное число контрольных объемов, полностью ее заполняющих. В итоге дискретная модель должна содержать информацию о координатах узлов, способах формирования контрольных объемов, состоящих из граней и ячеек, а также о количестве и типе используемых граничных условий.

Для решения дискретного аналога дифференциальных уравнений (1), в пакете программ ЛОГОС используется гранево-ячеечный формат представления дискретной модели. Данный формат является наиболее оптимальным с точки зрения реализации вычислительных алгоритмов, так как основные соотношения в методе контрольных объемов записаны относительно потоков через грани соседних ячеек.

Дискретизацию дифференциальных уравнений методом конечных объемов рассмотрим на примере дискретизации уравнения переноса скалярной величины φ:

$$\frac{\partial \rho\varphi}{\partial t} + \nabla(\rho \boldsymbol{u}\varphi) - \nabla(\Gamma \nabla\varphi) = Q, \tag{2}$$

где Γ – коэффициент диффузии, Q – источниковое слагаемое.

Процесс получения дискретного аналога подразумевает интегрирование исходного уравнения (2) по выделенному контрольному объему ячейки P и времени. Получаем полный вид дискретного аналога дифференциального уравнения переноса скалярной величины (2) для контрольного объема P:

$$\frac{\rho_P^n \varphi_P^n - \rho_P^{n-1}\varphi_P^{n-1}}{\Delta t} V_P + \sum_{f=face(P)} \rho_f^n \varphi_f^n (\boldsymbol{u}_f \boldsymbol{S}_f) + \sum_{f=face(P)} \Gamma_f (\nabla\varphi_f^n \boldsymbol{S}_f) = Q V_P, \tag{3}$$

где $\boldsymbol{S}_f = S_f \boldsymbol{n}_f$ – площадь, ориентированная по нормали к грани f, S_f – площадь грани f, ρ_f^n – плотность на грани f на текущем временном шаге, φ_f^n – скалярная величина на грани f на текущем временном шаге, Γ_f – коэффициент диффузии на грани f, ρ_P^n – плотность в центре контрольного объема P на текущем временном шаге, φ_P^n – скалярная величина в центре контрольного объема P на текущем временном шаге, ρ_P^{n-1} – плотность в центре контрольного объема P на предыдущем временном шаге, φ_P^{n-1} – скалярная величина в центре контрольного объема P на предыдущем временном шаге, $f = face(P)$ – суммирование по всем граням, ограничивающим контрольный объем P.

В случае наличия внешних граней и граней, лежащих на внутренних интерфейсах, в контрольном объеме P конвекционное и диффузионное слагаемые на текущем временном шаге примут вид:

$$\sum_{f=face(P)} \rho_f \varphi_f (u_f S_f) = \sum_{f=face(P_{inner})} \rho_f \varphi_f (u_f S_f) + \sum_{f=face(P_{bound})} \rho_f \varphi_f (u_f S_f) + \sum_{f=face(P_{interface})} \rho_f \varphi_f (u_f S_f), \tag{4}$$

$$\sum_{f=face(P)} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f) = \sum_{f=face(P_{inner})} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f) + \sum_{f=face(P_{bound})} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f) + \sum_{f=face(P_{interface})} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f), \tag{5}$$

где $f = face(P_{inner})$ – суммирование по всем внутренним граням, ограничивающим контрольный объем P, $f = face(P_{bound})$ – суммирование по всем внешним граням, ограничивающим контрольный объем P, $f = face(P_{interface})$ – суммирование по всем интерфейсным граням, ограничивающим контрольный объем P.

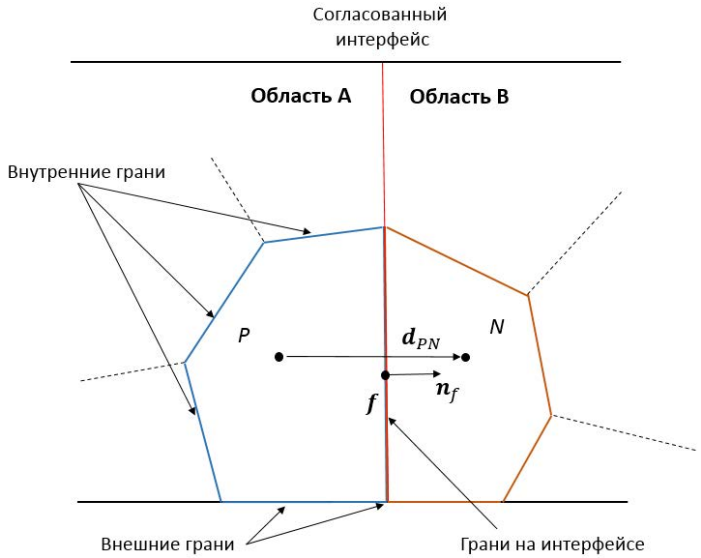


Рис. 1. Схематичный пример сетки с согласованным внутренним интерфейсом

Для случая, когда контрольный объем P лежит на согласованном интерфейсе (рис. 1), все значения неизвестных величин ρ_f, φ_f, $\boldsymbol{u}_f$ и Γ_f относятся к центру грани и могут быть вычислены по любой из известных схем аппроксимации [4].

Вычисление градиента $\nabla\varphi_f$ на произвольных неструктурированных сетках выполняется с учетом поправки на неортогональность векторов $\boldsymbol{d}_{PN}$ и нормали к грани $\boldsymbol{n}_f$ (рис. 1):

$$\nabla\varphi_f = (\varphi_N - \varphi_P)\frac{\boldsymbol{S}_f}{\boldsymbol{S}_f\boldsymbol{d}_{PN}} + \overline{\nabla\varphi_f} - (\overline{\nabla\varphi_f}\boldsymbol{d}_{PN})\frac{\boldsymbol{S}_f}{\boldsymbol{S}_f\boldsymbol{d}_{PN}}, \tag{6}$$

где $\overline{\nabla\varphi_f} = \lambda_f\nabla\varphi_P + (1-\lambda_f)\nabla\varphi_N$ – осредненная величина градиента на грани, λ_f – интерполяционный фактор для грани f.

Градиент в центре контрольного объема $\nabla\varphi_P$ может быть вычислен по алгоритму Грина-Гаусса [4]:

$$\nabla\varphi_P = \frac{1}{V_P}\sum_{f=face(P)}\varphi_f\boldsymbol{S}_f, \tag{7}$$

где $\varphi_f = \lambda_f\varphi_P + (1-\lambda_f)\varphi_N$ – значение величины φ на грани f, λ_f – интерполяционный фактор для грани f.

Окончательный вид дискретного аналога дифференциального уравнения переноса скалярной величины (3) удобнее представить в виде:

$$a_P\varphi_P + \sum_{N=nb(P)} a_N\varphi_N = b_N^{\varphi}, \tag{8}$$

где a_p – диагональный коэффициент, a_N – недиагональный коэффициент и b_P^{φ} – правая часть, $N=nb(P)$ – суммирование по всем соседним ячейкам N контрольного объема P.

Алгебраическое уравнение (8) с учетом граничных условий составляется для каждого контрольного объема, находящегося в расчетной области. В результате формируется система линейных алгебраических уравнений (СЛАУ), которая может быть решена любыми итерационными методами (Гаусса–Зейделя, сопряженных градиентов, Якоби и так далее).

Описанный алгоритм дискретизации исходных уравнений методом конечных объемов применим для всех дифференциальных уравнений, в том числе и для системы уравнений Навье–Стокса. Однако следует отметить, что для расчета величин ρ_f, φ_f, $\boldsymbol{u}_f$, Γ_f в [4] и [5], величины $\nabla\varphi_f$ в [6] и величины φ_f в [7] на гранях, лежащих на несогласованных интерфейсах стандартные схемы аппроксимации неприменимы, так как необходимо учитывать соседство грани f с несколькими ячейками на противоположной стороне интерфейса.

2. Описание работы УСИ в составе пакета программ ЛОГОС

В пакете программ ЛОГОС каждый из внутренних интерфейсов содержит два смежных набора граней, сетки которых в общем случае могут быть не согласованы в области пересечения. Для согласования решения в области пересечения несогласованных областей А и В создается УСИ, в котором выполняется переход от исходных граней к виртуальным (рис. 2).

Для определения параметров виртуальных граней УСИ в области пересечения исходные грани области А последовательно проецируются на все исходные грани области В.

В результате пересечения образуется набор виртуальных граней, в состав которого могут входить внутренние виртуальные грани, определяющие связь двух ячеек, и внешние виртуальные грани, на которых могут быть заданы граничные условия (рис. 3).

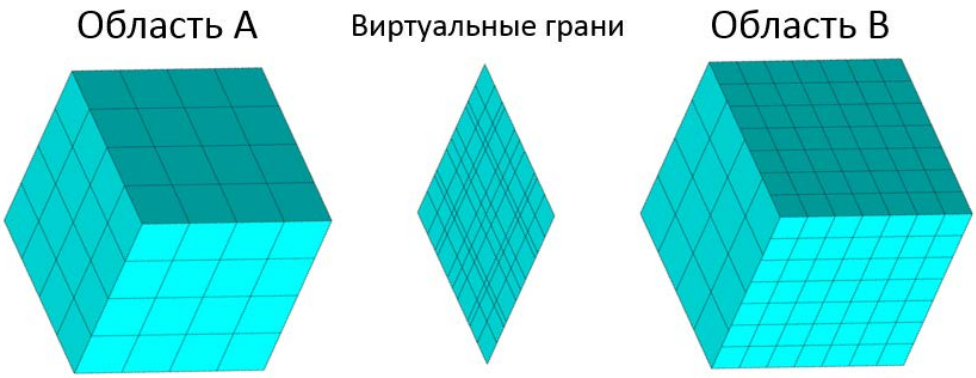


Рис. 2. Формирование УСИ

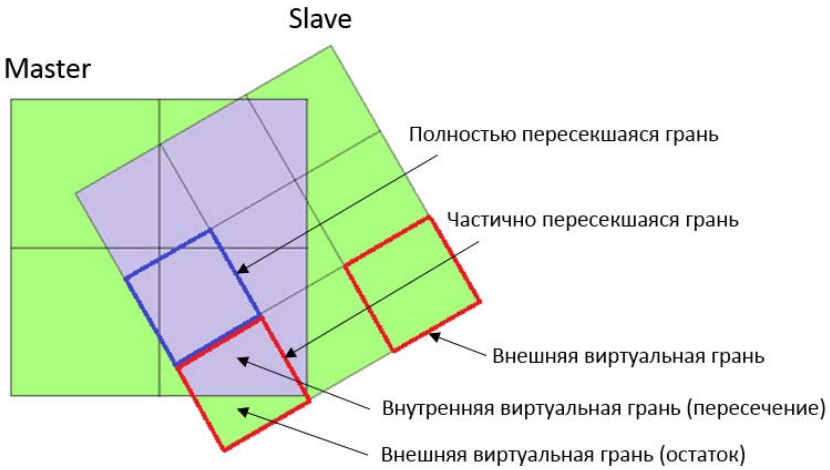


Рис. 3. Пример образования виртуальных граней на несостыкованном интерфейсе

Полученный набор виртуальных граней образует виртуальный интерфейс, позволяющий установить связь между ячейками несогласованных смежных областей сетки и выполнить обработку непересеченных частей сетки в соответствии с выбранными граничными условиями.

Рассмотрим переход от исходных к виртуальным граням на примере контрольного объема P, лежащего на несогласованном УСИ (рис. 4).

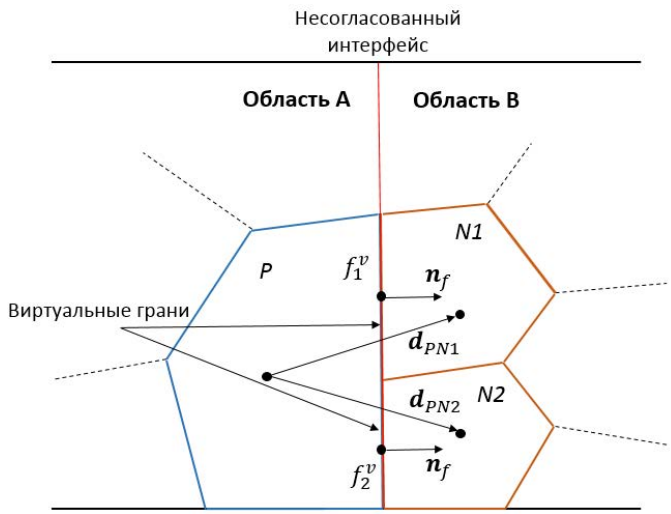


Рис. 4. Схематичный пример сетки с несогласованным внутренним интерфейсом

Дискретный аналог конвекционной составляющей (4) для контрольного объема P, лежащего на несогласованном УСИ примет вид:

$$\sum_{f=face(P)} \rho_f \varphi_f (\boldsymbol{u}_f \cdot \boldsymbol{S}_f) = \sum_{f=face(P_{inner})} \rho_f \varphi_f (\boldsymbol{u}_f \boldsymbol{S}_f) + \sum_{f=face(P_{bound})} \rho_f \varphi_f (u_f S_f) + \sum_{f=face(P_{interface})} \sum_v \rho_f^v \varphi_f^v (u_f^v S_f^v), \quad (9)$$

где $f = face(P_{interface})$ – суммирование по всем интерфейсным граням, ограничивающим контрольный объем P, ρ_f^v, φ_f^v, $\boldsymbol{u}_f^v$ – плотность, скалярная величина и скорость на виртуальной грани v, образовавшейся на исходной грани f, $\boldsymbol{S}_f^v$ – площадь виртуальной грани v ориентированная по нормали к исходной грани f.

Дискретный аналог для диффузионной составляющей (5) для контрольного объема P, лежащего на несогласованном УСИ примет вид:

$$\sum_{f=face(P)} \Gamma_f (\nabla\varphi_f \cdot \boldsymbol{S}_f) = \sum_{f=face(P_{inner})} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f) + \sum_{f=face(P_{bound})} \Gamma_f (\nabla\varphi_f \boldsymbol{S}_f) + \sum_{f=face(P_{interface})} \sum_v \Gamma_f^v (\nabla\varphi_f^v \boldsymbol{S}_f^v), \quad (10)$$

где Γ_f^v, $\nabla\varphi_f^v$ – коэффициент диффузии и величина градиента скалярной величины на виртуальной грани v, образовавшейся на исходной грани f, $\boldsymbol{S}_f^v$ – площадь виртуальной грани v, ориентированной по нормали к исходной грани f.

Все значения неизвестных величин ρ_f^v, φ_f^v, $\boldsymbol{u}_f^v$ и Γ_f^v относятся к центру виртуальной грани v, образовавшейся на исходной грани f, и могут быть вычислены по любой из известных схем [4].

Значение градиента $\nabla\varphi_f^v$ в центре виртуальной грани v, образовавшейся на исходной грани f, с учетом поправки на неортогональность:

$$\nabla\varphi_f^v = (\varphi_N - \varphi_P)\frac{\boldsymbol{S}_f^v}{\boldsymbol{S}_f^v \boldsymbol{d}_{PN}} + \overline{\nabla\varphi_f^v} - (\overline{\nabla\varphi_f^v}\,\boldsymbol{d}_{PN})\frac{\boldsymbol{S}_f^v}{\boldsymbol{S}_f^v \boldsymbol{d}_{PN}}, \quad (11)$$

где $\overline{\nabla\varphi_f^v} = \lambda_f^v \nabla\varphi_P + (1-\lambda_f^v)\nabla\varphi_N$ – осредненная величина градиента на грани, λ_f^v – интерполяционный фактор для виртуальной грани v, образовавшейся на интерфейсной грани f.

В свою очередь значение градиента $\nabla\varphi_P$ в центре контрольного объема P, лежащего на УСИ, вычисляется по алгоритму Грина–Гаусса [4]:

$$\nabla\varphi_P = \frac{1}{V_P}\Big(\sum_{f=face(P_{inner})} \varphi_f \boldsymbol{S}_f + \sum_{f=face(P_{bound})} \varphi_f \boldsymbol{S}_f + \sum_{f=face(P_{interface})} \varphi_f^{av} \boldsymbol{S}_f\Big), \quad (12)$$

где $\varphi_f^{av} = \dfrac{\sum_v \varphi_f^v \boldsymbol{S}_f}{\sum_v \boldsymbol{S}_f}$ – усредненная скалярная величина на интерфейсной грани f.

В результате описанных выше преобразований, в дискретный аналог уравнения переноса (8) для каждого контрольного объема P, лежащего на несогласованном интерфейсе добавляются слагаемые, учитывающие связь ячеек через виртуальные грани. Таким образом замена исходных граней виртуальными на УСИ позволяет объединять не состыкованные фрагменты сеточной модели в одну расчетную область, согласовав решение СЛАУ в области несогласованных интерфейсов.

3. Тестовые расчеты с использованием УСИ

3.1. Охлаждение твердотельных блоков в плоском канале

Рассмотрим работу УСИ на примере расчета задачи охлаждения шести твердотельных блоков с объемным тепловыделением, установленных в плоском твердотельном канале, ламинарным потоком несжимаемой жидкости при вынужденной конвекции [6].

Геометрия расчетной области схематично представлена на рис. 5.

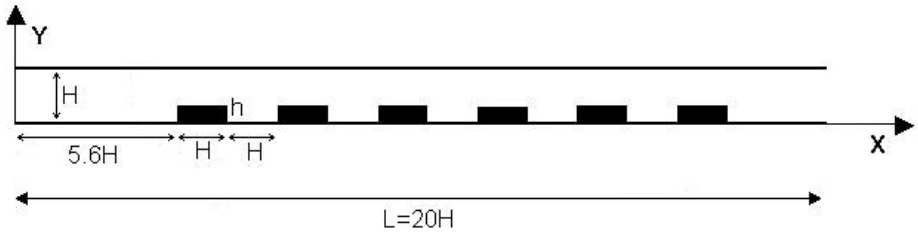


Рис. 5. Схематическое изображение расчетной области

Плоский прямоугольный канал имеет высоту $H = 1$ и длину $L = 20H$. Внутри канала, вплотную к нижней стенке, расположены шесть одинаковых твердотельных блоков длинной H и высотой $h = 0.25H$. Левая грань первого блока находится на расстоянии $5.6H$ от входа. Промежутки между блоками одинаковы и равны H. Каждый блок равномерно нагревается объемным источником тепла мощностью $Q = 71.42857$. Стенки канала неподвижны, непроницаемы и теплоизолированы (граничные условия Wall_No_Slip, Adiabatic_Wall). На левой стороне канала расположен вход (Inlet), где задается нулевая температура и единичный вектор скорости, направленный вдоль оси X. На правой границе расположен выход (Outlet).

Равномерная прямоугольная расчетная сетка в области, заполненной несжимаемой жидкостью, построена так, чтобы границы твердотельных блоков точно совпадали с гранями ячеек. На рис. 6 показана результирующая расчетная сетка, состоящая из несогласованных регионов.

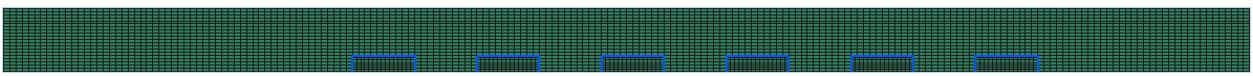


Рис. 6. Результирующая расчетная сетка

Сетка твердотельных блоков состоит из более мелких ячеек. На рис. 7 показан фрагмент сетки на границе двух регионов в области одного из твердотельных блоков.

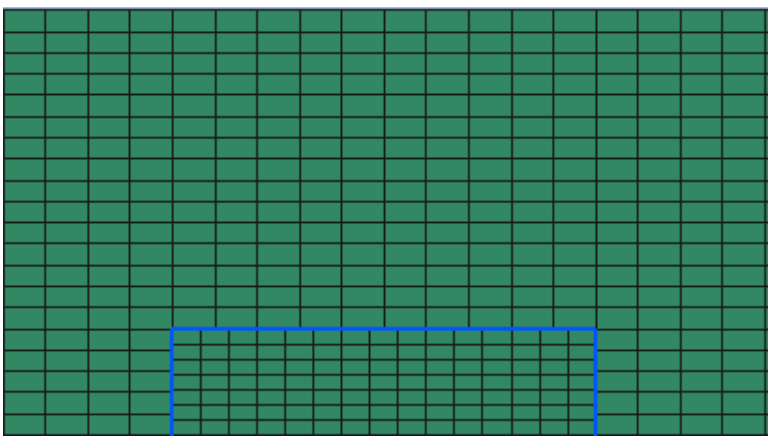


Рис. 7. Фрагмент расчетной сетки

Для согласования регионов на границе жидкости и твердого тела используется УСИ. Для сравнения результатов дополнительно проведен расчет аналогичной задачи на предварительно сшитой в пакете программ ЛОГОС сетке.

При расчете были заданы физические параметры среды таким образом, чтобы обеспечивались значения безразмерных величин, указанных в [6]: $Re = 100$, $Pr = 0.7$. Расчет задачи был выполнен в параллельном режиме на 12 процессорах. На рис. 8 приведены поля температуры из [6] (вверху) и рассчитанные в пакете программ ЛОГОС (внизу) отображенные в одном масштабе. График на рис. 9 показывает профили температуры вдоль прямой $y = 0.1$.

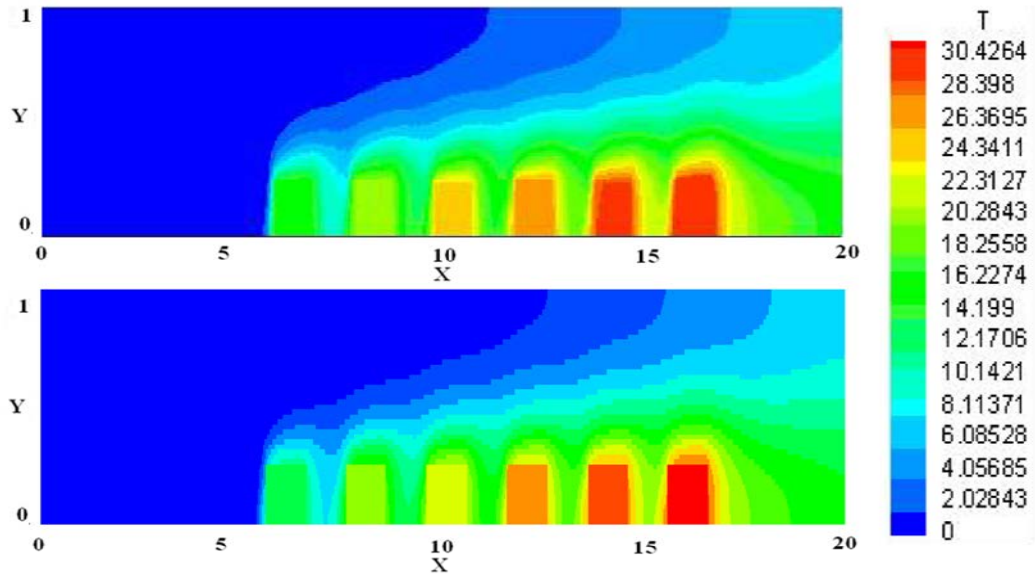


Рис. 8. Сравнение полей температуры

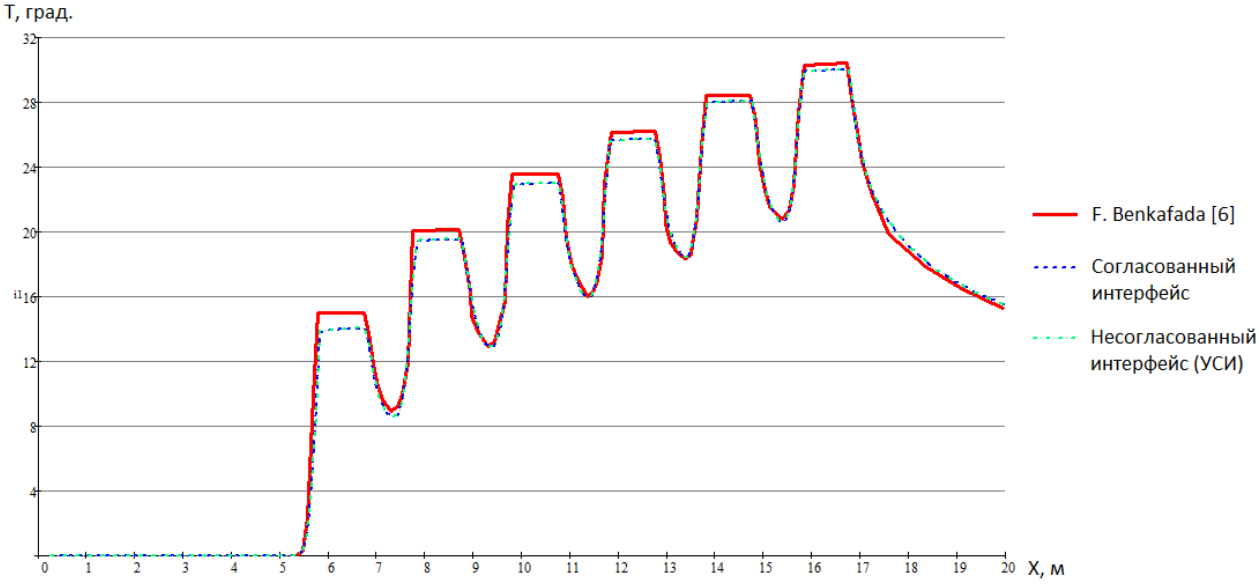


Рис. 9. Сравнение профилей температуры вдоль канала

На графике рис. 9 показано, что рассчитанная температура в центрах твердотельных блоков в случае расчета на сшитой сетке с большой точностью совпадает с аналогичными значениями, полученными в случае расчете на сетке с использованием УСИ. Результат расчета в центрах твердотельных блоков отличается от [6] не более чем на 1 градус.

3.2. Задача турбулентного течения несжимаемой жидкости в круглой трубе с внезапным сужением

Производится расчет турбулентного течения несжимаемой жидкости в круглой трубе с внезапным сужением. Геометрическая модель, показанная на рис. 10, представляет собой прямой участок трубы со ступенчатым уменьшением внутреннего диаметра (с 50 до 30 мм). Требуется определить потери полного давления при течении несжимаемой жидкости в трубопроводе со ступенчатым уменьшением внутреннего диаметра между сечением 1 и сечением 2.

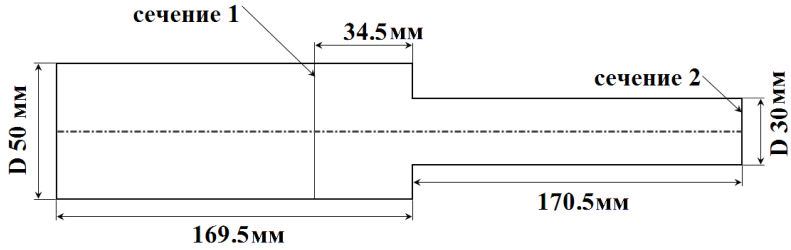


Рис. 10. Геометрия канала с сужением

Расчет проводится на сетке, состоящей из двух несогласованных областей (фрагмент трубы до сужения и фрагмент трубы после сужения) сетки которых были построены независимо друг от друга. Фрагменты расчетной сетки в области сужения представлены на рис. 11 и 12. Расчетная сетка содержит 1.7 млн. ячеек.

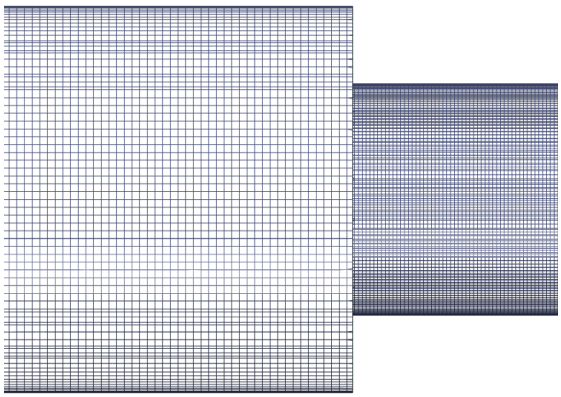


Рис. 11. Фрагмент сетки в продольном сечении в районе сужения

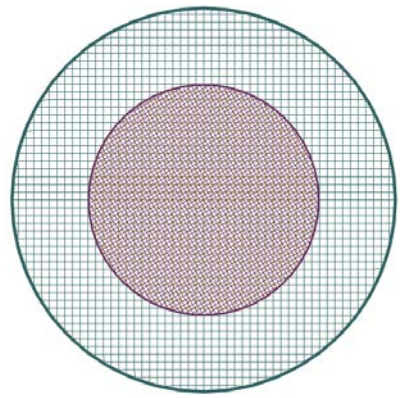


Рис. 12. Расчетная сетка в поперечном сечении сужения

Физические параметры среды, представлены в табл. 1.

Таблица 1

Параметры среды и граничные величины

Параметр	Значение	Размерность
Плотность жидкости, ρ	998,2	кг/м^3
Динамическая вязкость, μ	0.00101	кг/(м·с)
Интенсивность турбулентности	0.05	%
Длина перемешивания	0.001	м

Для моделирования турбулентного перемешивания в расчетах используется k-Omega, SST, All Reynolds модель турбулентности. Для аппроксимации конвективных слагаемых в расчетах используется схема UD первого порядка точности. На рис. 13 показано расположение, а в табл. 2 приведены типы и параметры граничных условий расчетной модели.

Таблица 2

Граничные условия

Граница	Тип граничного условия в ЛОГОС	Граничные условия
Вход	Вход дозвуковой (Inlet)	U, V = 0 м/с W = 0.4227 м/с
Выход	Давление (Pressure)	P = 0 Па
Стенка	Стенка без проскальзывания (WallNoSlip)	U, V, W = 0 м/с
УСИ	Интерфейс (Interface)	–
Внешние (не имеющие пересечений) грани на УСИ	Стенка без проскальзывания (WallNoSlip)	U, V, W = 0 м/с

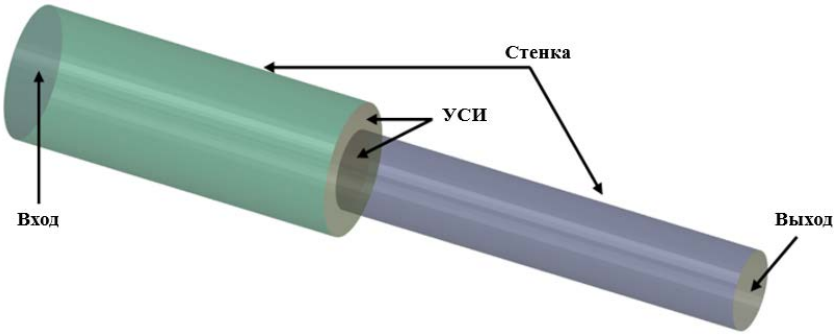


Рис. 13. Граничные условия

Расчет задачи был выполнен в параллельном режиме на 80 процессорах. Для согласования решения в области пересечения несогласованных границ между фрагментами расчетной сетки (в области внезапного сужения) используется УСИ. Для сравнения результатов дополнительно проведен расчет аналогичной задачи на предварительно сшитой в пакете программ ЛОГОС сетке.

На рис. 14 представлено поле модуля скорости в трубопроводе при расчете на сшитой сетке и на сетке, содержащей УСИ.

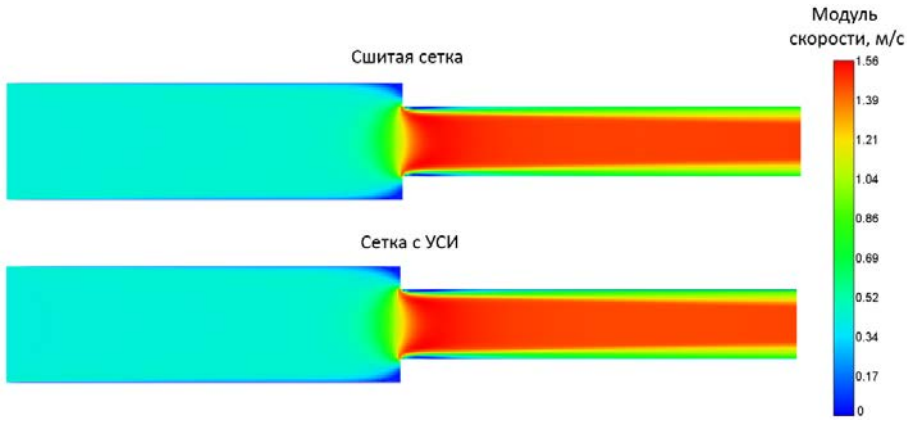


Рис. 14. Поле модуля скорости в трубопроводе

Результаты расчета приведены в табл. 3.

Таблица 3

Результаты расчета

Источник данных	Перепад полного давления между сечениями трубы 1 и 2, Па	Время расчета на 80 процессорах, с
Эмпирический расчет [7]	339	–
Расчет на сшитой сетке	351,8	502
Расчет на сетке, содержащей несогласованный интерфейс (с использованием УСИ)	351,6	510

3.3. Расчет гидравлического сопротивления диффузора

Выполняется расчет потерь полного давления при течении несжимаемой жидкости в трубопроводе с диффузором. Геометрическая модель представляет собой трубопровод с коническим диффузором. Схематичная геометрия модели и расчетная область приведены на рис. 15.

Расчет проводится на сетке, состоящей из трех несогласованных областей (фрагмент трубы до диффузора, диффузор и фрагмент трубы после диффузора). Фрагменты расчетной сетки в области диффузора представлен на рис. 16. Расчетная сетка содержит 1.7 млн. ячеек.

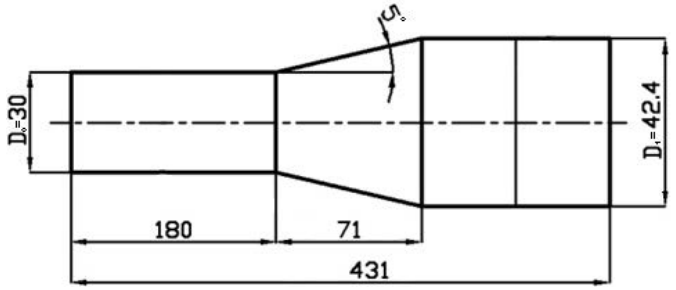


Рис. 15. Геометрия трубопровода с диффузором (мм)

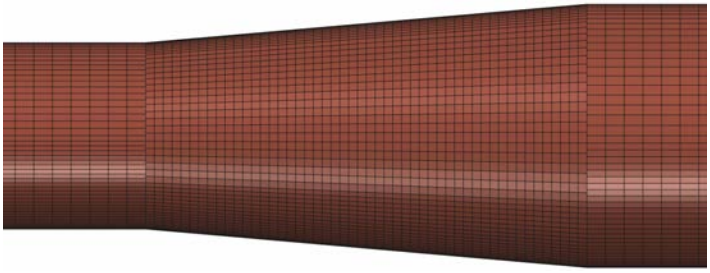


Рис. 16. Фрагмент сетки в районе диффузора

Физические параметры среды, представлены в табл. 4.

Таблица 4

Параметры среды и граничные величины

Параметр	Значение	Размерность
Плотность жидкости, ρ	772,9	кг/м^3
Динамическая вязкость, μ	0.002682	кг/(м·с)
Интенсивность турбулентности	0.05	%
Длина перемешивания	0.001	м

Для моделирования турбулентного перемешивания в расчетах используется k-Omega, SST Low модель турбулентности. Для аппроксимации конвективных слагаемых в расчетах используется схема UD первого порядка точности. На рис. 17 показано расположение, а в табл. 5 приведены типы и параметры граничных условий расчетной модели.

Таблица 5

Граничные условия

Граница	Тип граничного условия в ЛОГОС	Граничные условия
Вход	Массовый расход (Inlet)	Q = 3.159662 кг/с
Выход	Давление (Pressure)	P = 0 Па
Стенка	Стенка без проскальзывания (WallNoSlip)	U, V, W = 0 м/с
УСИ	Интерфейс (Interface)	–

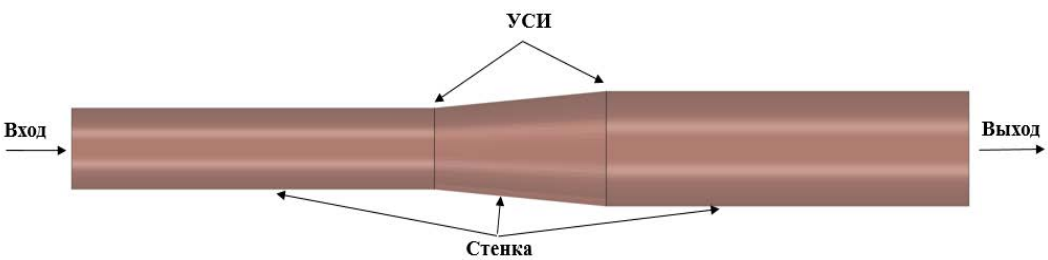


Рис. 17. Граничные условия

Расчет задачи был выполнен в параллельном режиме на 80 процессорах. Для согласования решения в области пересечения несогласованных границ между фрагментами расчетной сетки (в области диффузора) используется УСИ. Для сравнения результатов дополнительно проведен расчет аналогичной задачи на предварительно сшитой в пакете программ ЛОГОС сетке.

На рис. 18 представлено поле модуля скорости в диффузоре. Результаты расчета приведены в табл. 6.

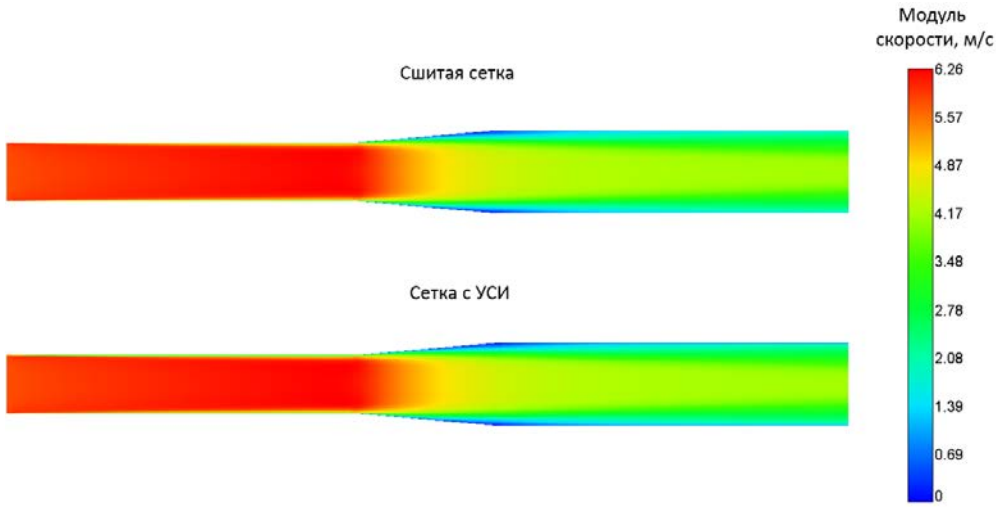


Рис. 18. Поле модуля скорости в трубопроводе

Таблица 6

Результаты расчета

Источник данных	Перепад полного давления между сечениями трубы 1 и 2, Па	Время расчета на 80 процессорах, с
Эмпирический расчет [7]	2573	–
Расчет на сшитой сетке	2816	674
Расчет на сетке, содержащей несогласованный интерфейс (с использованием УСИ)	2819	917

Заключение

В докладе представлено описание алгоритма работы универсального сеточного интерфейса (УСИ), реализованного в пакете программ ЛОГОС. УСИ позволяет проводить численные расчеты на сеточных моделях, содержащих несогласованные сеточные интерфейсы. В докладе приведены примеры использования УСИ при расчете задач на сеточных моделях, состоящих из фрагментов неструктурированной несогласованной сетки. Полученные результаты численного моделирования хорошо согласуются с численными результатами, опубликованными в открытых источниках. Показано, что реализованный алгоритм позволяет производить расчет на сеточных моделях, состоящих из несогласованных фрагментов сетки и при этом не уступает по качеству и скорости счета аналогичным расчетам на геометрически сшитой сетке.

Литература

1. Martin Beaudoin, Hrvoje Jasak. Development of Generalized Grid Interface for Turbomachinery simulation with OpenFOAM // Open Source CFD International Conference: Berlin, 2008.

2. Patankar. S. Numerical Heat Transfer And Fluid Flow. □ New York, Hemisphere Publishing Corporation, 1980.

3. Jasak H. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flow. Thesis submitted for the degree of doctor // Department of Mechanical Engineering, Imperial College of Science. – London, 1996.

4. Ferziger J.H., Peric M. Computational Method for Fluid Dynamics. – New York: Springer-Verlag, 2002.

5. Ченцов О. В., Скворцов А. В. Обзор алгоритмов построения оверлеев многоугольников // Вестник Томского государственного ун-та. 2003. № 280.

6. F. Benkafada, K. Talbi, M. Afrid, The enhanced cooling of heated blocks mounted on the wall of a plane channel filled with a porous medium // Revue des Energies Renouvelables. 2008. Vol. 11, № 2. P. 197–206.

7. Идельчик И. Е. Справочник по гидравлическим сопротивлениям. М.: Машиностроение, 1992.

SIMULATION OF HYDRODYNAMIC PROBLEMS ON MESHES WITH UNMATCHED MESH INTERFACES

A. V. Korotkov, S. V. Lashkin, A. S. Kozelkov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

This report describes a universal mesh interface (UMI) realized in LOGOS software package on the basis of GGI (General Grid Interface) operation algorithm [1], and its adaptation for the computations on hydrodynamic problems on unmatched interfaces using SIMPLE-method [2, 3].

The report shows examples of UMI implementation in computations for problems on mesh models that comprise parts of an unstructured nonconforming mesh. It is shown that the implemented algorithm allows calculating on mesh models that consist of nonconforming parts of the mesh, and it is not worse in quality and in its count rate than the analogous computations on the geometrically-linked mesh.

Key words: General Grid Interface, universal mesh interface, SIMPLE, LOGOS, nonconforming mesh interface.

УДК 517.9
DOI 10.53403/9785951505071_2022_239

О КРИТИЧЕСКОМ ПОВЕДЕНИИ ДИНАМИЧЕСКИХ СИСТЕМ ПРИ ВЫСОКОИНТЕНСИВНОМ ВОЗДЕЙСТВИИ

Е. В. Кошелева, Н. И. Сельченкова, С. С. Соколов, И. Р. Трунин, А. Я. Учаев

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Работа посвящена установлению общих закономерностей динамических деструктивных процессов (динамического разрушения и диспергирования), протекающих в металлах при высокоинтенсивном внешнем воздействии в динамическом диапазоне неравновесных состояний $t \sim 10^{-6} \div 3 \cdot 10^{-10}$ с. Установлены количественные характеристики кинетических процессов, которые являются откликом системы на внешнее воздействие. Определены фрактальная размерность, показатель Херста диссипативных структур – продуктов диспергирования, гидродинамических мод, кластеров центров разрушения.

Масштабная инвариантность возникающих диссипативных структур указывает на то, что неравновесная система достигла критического состояния. Также признаком того, что система находится в критическом состоянии является обращение в бесконечность сжимаемости (восприимчивости). Используя парный потенциал кристаллической решетки определено выражение для сжимаемости (восприимчивости) нагружаемого образца, которая зависит от плотности центров разрушения.

Проведено численное моделирование процессов диспергирования, возникающих при воздействии импульсов релятивистских электронов на металлические образцы с нанесенными возмущениями в виде пирамидок, с использованием неструктурированных многогранных лагранжевых сеток ТИМ-3D и двумерной лагранжево-эйлеровой расчетной методики; проведено численное моделирование процесса динамического разрушения тонких металлических фольг при воздействии импульсов релятивистских электронов с использованием расчетной методики ТИМ-2D.

Предлагаемая методология исследования высокоинтенсивных динамических деструктивных процессов основана на подходе, который позволяет изучать количественные характеристики кинетических процессов, являющихся откликом неравновесных систем на внешнее воздействие, что и позволяет прогнозировать поведение неисследованных металлических образцов различной геометрии в экстремальных условиях.

Ключевые слова: динамические деструктивные процессы, динамическое разрушение, диспергирование, высокоинтенсивное внешнее воздействие, экстремальные условия, масштабная инвариантность, критическое состояние, сжимаемость, парный потенциал кристаллической решетки.

Покажем, что металлы в явлении динамического разрушения (на пороге макроразрушения, когда возникает связность в каскаде центров разрушения) находятся в критическом состоянии, а сам процесс динамического разрушения является аналогом критических явлений.

В связи с расширением области применения в научных исследованиях высокоэнергетических импульсных установок (импульсные ускорители – наносекундный диапазон воздействия, термоядерные реакторы – микросекундный диапазон воздействия, импульсные ядерные реакторы – миллисекундный диапазон воздействия, фемтосекундные лазеры) требуется знание поведения конден-

сированных сред в экстремальных условиях, поскольку мощностные возможности таких установок напрямую связаны со стойкостью элементов и узлов, входящих в их состав.

Следовательно, является актуальным исследование кинетических релаксационных процессов, протекающих в неравновесных конденсированных средах различной природы и геометрии, и установление временной границы сохранения функциональных свойств конструкционных материалов при различных видах высокоинтенсивного внешнего воздействия.

В работе приведен анализ результатов исследования количественных характеристик диссипативных структур на мезоуровне, возникающих в динамически нагруженных образцах различной геометрии, ряда металлов с целью установления универсальных признаков динамических релаксационных процессов.

Результаты исследований процесса динамического разрушения и диспергирования ряда металлов показали, что каскады различного рода диссипативных структур на различных масштабно-временных уровнях (начиная от наноуровня и до макроуровня) являются фрактальными и перколяционными кластерами [1].

Самоорганизация возникает в диссипативных неравновесных системах с высоким уровнем нелинейности, плотность поглощенной энергии ~ 0,05–0,2 $E_{св}$ кристаллической решетки ($E_{св}$ – энергия связи). Необходимо отметить, что для возникновения диссипативных структур необходимо, чтобы размер неравновесной системы превысил некоторое критическое состояние, чтобы могли возникнуть положительные обратные связи между элементами диссипативных структур. Причем, критический размер неравновесной системы зависит от степени ее неравновесности и интенсивности протекающих процессов. Так, в явлении динамического разрушения металлов фрактальный каскад центров разрушения образуется на временах $t \sim 10^{-9}$ с при плотности поглощенной энергии проникающих излучений $E \sim 0{,}8$ ($H_э + L_{пл}$) (линейные размеры неравновесной системы $l \sim (5 \div 10) \cdot 10^{-3}$ мм, $H_э$ – энтальпия, $L_{пл}$ – теплота плавления); а при плотности поглощенной энергии, приводящей к макроразрушению $E \sim 0{,}2$ ($H_э + L_{пл}$), времена возникновения фрактального перколяционного кластера $t \sim 5 \cdot 10^{-7}$ с (линейные размеры неравновесной системы $l \sim 3 \cdot 10^{-1}$ мм). Масштабы элементов диссипативных структур существенно меньше масштабов неравновесных систем [1]. Примером изучения неравновесных систем в области науки о материалах является исследование разрушения конструкционных материалов [2, 3].

Изучение эффектов самоорганизации неравновесных систем и установление количественных характеристик релаксационных процессов имеет важное значение.

Особый интерес и большую прикладную значимость вызывает изучение явлений самоорганизации неравновесных систем в наносекундной, микросекундной областях неравновесных состояний, например, поведение конденсированных сред при высокоинтенсивном внешнем воздействии. Именно на таких временах $t \sim 10^{-6} – 3 \cdot 10^{-10}$ с начинают развиваться многоволновые процессы и неустойчивости в нагруженных образцах с нанесенными возмущениями [1].

Одно общее свойство неравновесных нелинейных систем – реализация будущего состояния через последовательность бифуркаций делает эволюцию системы необратимой.

И. Р. Пригожин в работе [3] в параграфе 8 «Энтропия как принцип отбора» пишет: «Как мы уже неоднократно подчеркивали, в природе существуют системы с обратимым поведением, допускающие полное описание в рамках законов классической или квантовой механики. Но большинство интересующих нас систем … ориентировано во времени на макроскопическом уровне. Их … однонаправленность во времени отражает нарушение временной симметрии на микромасштабном уровне. Необратимость существует либо на всех уровнях, либо не существует ни на одном уровне. Она не может возникнуть, словно чудо, при переходе с одного уровня на другой».

Необратимость процесса диспергирования, процесса динамического разрушения конструкционных материалов проявляется в том, что кинетика динамических диссипативных процессов на микро- и мезоуровнях формирует поведение неравновесной системы на макроуровне.

В теории фазовых переходов К. Вильсона [4] показана универсальность поведения систем, претерпевающих фазовые переходы. «Основные свойства системы вблизи точек фазового перехода не зависят от свойств системы на микроуровне, неважно рассматриваем ли мы переход жидкость-пар, структурную перестройку в деформируемом кристалле или переход в магнитную фазу». Авторы

работы [4] указывают «когда степени свободы ведут себя взаимосвязано, поведение системы определяется … фактом кооперативного поведения и природой самих степеней свободы. Гамильтонианы взаимодействия играют при этом лишь второстепенную роль. В теории критических явлений возникло понятие универсальности, что все гамильтонианы взаимодействия приводят к одинаковому критическому поведению».

Это утверждение подтверждают данные табл. 1, в которой приведены табличные значения $\frac{H_э + L_{пл}}{H_{субл}}$, $\frac{E_{кр}}{H_{субл}}$, $\frac{E_{кр}}{H_э + L_{пл}}$, критическая плотность поглощенной энергии $E_{кр}$, приводящая к разрушению ряда металлов, для значения долговечности $t \sim 10^{-8}$ с ($H_{субл}$ – энтальпия сублимирования) [5].

Отношение $\frac{E_{кр}}{H_{субл}}$ при $t \sim 10^{-8}$ с и отношение $\frac{H_э + L_{пл}}{H_{субл}}$ для приведенных в табл. 1 металлов имеют близкие значения, что свидетельствует о подобии поведения металлов в явлениях динамического разрушения и диспергирования

Таблица 1

Энергетические параметры металлов и параметры нагружения, типы симметрий решеток

Элемент	Ат. №	$\frac{H_э + L_{пл}}{H_{субл}}$, [5]	$E_{кр}$, Дж/г при $t \sim 10^{-8}$ с, [5]	$\frac{E_{кр}}{H_э + L_{пл}}$ при $t \sim 10^{-8}$ с, [5]	$\frac{E_{кр}}{H_{субл}}$ при $t \sim 10^{-8}$ с	Тип решетки, [6]
Al	13	0,151	–	–	–	ГЦК
Ti	22	0,16	950	0,537	0,086	ГЦК
Fe	26	0,18	754	0,585	0,105	ОЦК
Co	27	0,16	–	–	–	ГЦК
Ni	28	0,15	618	0,533	0,080	ГЦК
Cu	29	0,144	394	0,575	0,083	ГЦК
Zn	30	0,157	–	–	–	ГЦК
Mo	42	0,18	819	0,537	0,097	ОЦК
Cd	48	0,18	80	0,494	0,089	гексагональная
Sn	50	0,12	100	0,525	0,063	алмазная
Ta	73	0,186	379	0,495	0,092	ОЦК
W	74	0,165	522	0,570	0,094	ОЦК
Pb	82	0,116	51	0,510	0,059	ГЦК

Для неравновесных систем применим принцип Ле Шателье–Брауна, который говорит о том, что внешнее воздействие на термодинамическую систему вызывает в ней такие процессы, которые ослабляют внешнее воздействие [7]. Количественной характеристикой принципа Ле Шателье–Брауна является обобщенная восприимчивость. Для механических систем обобщенной восприимчивостью является сжимаемость. Например, для разрушаемого динамически твердого тела сжимаемость связана с плотностью центров разрушения [1]. На пороге макроразрушения сжимаемость неограниченно возрастает точно также, как скорость накопления центров разрушения и скорость центрообразования.

Предлагаемый подход позволил установить кинетические характеристики процесса динамического разрушения – скорость накопления центров разрушения $N(t)$ и скорость центрообразования $J(t)$, которые имеют вид

$$N(t) = A_1 \cdot \exp\left(\frac{t/t_p}{A_2}\right)^{0,2} \left(1 - t/t_p\right)^{-1,2}, \quad (1)$$

$$J(t) = A_3 \cdot \exp\left(\frac{t/t_p}{A_2}\right)^{0,4} \left(1 - t/t_p\right)^{-2,2}, \quad (2)$$

где A_1, A_2, A_3 – коэффициенты, t_p – время разрушения (долговечность), t – текущее время [1].

На рис. 1 приведен вид скорости центрообразования в приведенных координатах ряда металлов ($J_{max}(t)$ – максимальная скорость центрообразования).

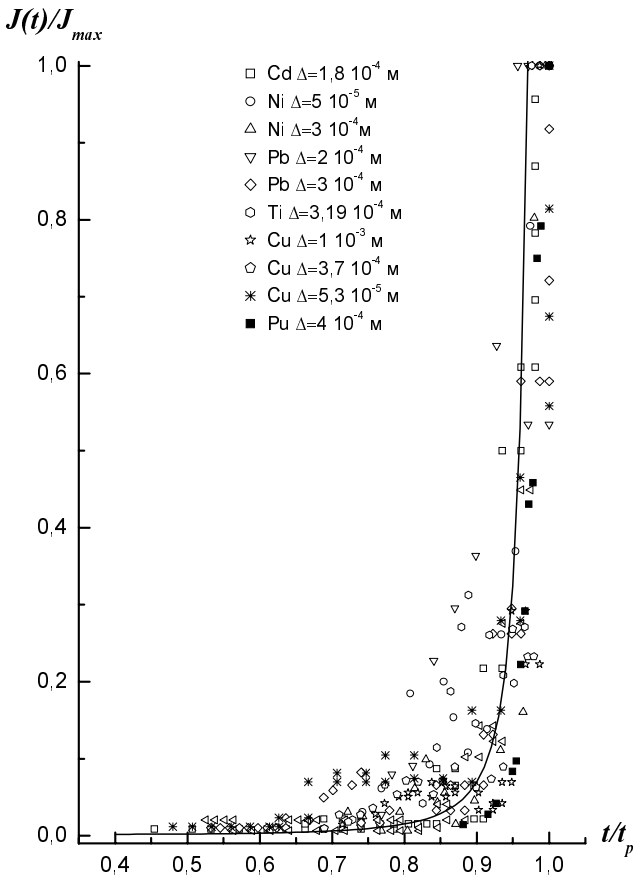


Рис. 1. Скорость центрообразования в приведенных координатах t/t_p, $J(t)/J_{\max}(t)$ [1]

Методом Монте-Карло определялись плотности поглощенных энергий в исследуемых образцах и толщины дозирующих фильтров для спектра электронов ускорителя РИУС-5. Из уравнения состояния Ми–Грюнайзена находились расчетные значения критических давлений, приводящих к разрушению для образцов различных материалов.

Проведены исследования процесса динамического разрушения плоских образцов металлов (Cu $\Delta \sim 0{,}2$ мм) при воздействии релятивистских электронных пучков ускорителя РИУС-5 и образцов Cu, Al с нанесенными возмущениями в виде пирамидок высотой $h \sim 0{,}7$ мм, общей толщиной $\Delta \sim 1{,}5$ мм с углом раствора $\alpha \sim 60^0, 45^0, 120^0$.

В зависимости от угла раствора нанесенных возмущений в виде пирамидок происходит смена механизма релаксационных процессов в нагруженных образцах, когда происходит кинетический фазовый переход от кинетики диспергирования конструкционных материалов к кинетике откольного разрушения (переход от фрагментации к поврежденности).

Многоволновое движение среды приводит к движению элементов среды в пирамидке по схеме «трансляция (сдвиг) + поворот». При указанной постановке исследований процесса диспергирования металлических образцов, после возникновения откольного разрушения происходит кумуляция энергии ударной волны при ее отражении от боковых граней. Для возмущений с углом раствора $\alpha \sim 45^0$, $\alpha \sim 60^0$ угол отражения от граней имеет значение соответственно $\varphi \sim 23^0$, $\varphi \sim 30^0$. При таких значениях φ происходит практически полное отражение ударной волны от свободной поверхности грани конструкционных материалов и кумулирование ее энергии в вершинах нанесенных возмущений.

Расчетное давление на тыльной стороне в облучаемой области для медного образца $P \sim 7$–9 ГПа в образцах с нанесенными возмущениями с углом раствора $\alpha \sim 45^0$, $\alpha \sim 60^0$, $\alpha \sim 120^0$. Давление в облученных образцах определялось с использованием уравнения состояния Ми–Грюнайзена. В образцах с углом раствора $\alpha \sim 45^0$, $\alpha \sim 60^0$ при таких значениях давления происходят кумулятивные выбросы из вершин пирамидок (см. рис. 2). В верхней части пирамидок отсутствует часть объема, вследствие диспергирования среды.

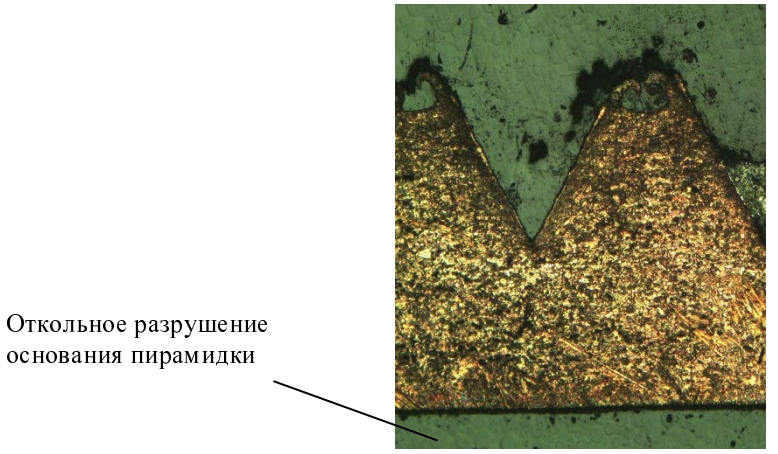


Рис. 2. Вид образца из меди с нанесенными возмущениями в виде пирамидок с углом раствора $\alpha \sim 45^0$ после ударно-волнового нагружения

В образце с углом раствора $\alpha \sim 120^0$ близ граней пирамидок начинает формироваться откольный слой (см. рис. 3). Расчетное давление в области откольного разрушения тыльной стороны $P \sim 7$ ГПа.

При увеличении давления в облучаемой области медного образца с углом раствора $\alpha \sim 120^0$ происходит откольное разрушение областей пирамидок близ граней (см. рис. 3). Расчетное давление в области откольного разрушения тыльной стороны $P \sim 9$ ГПа.

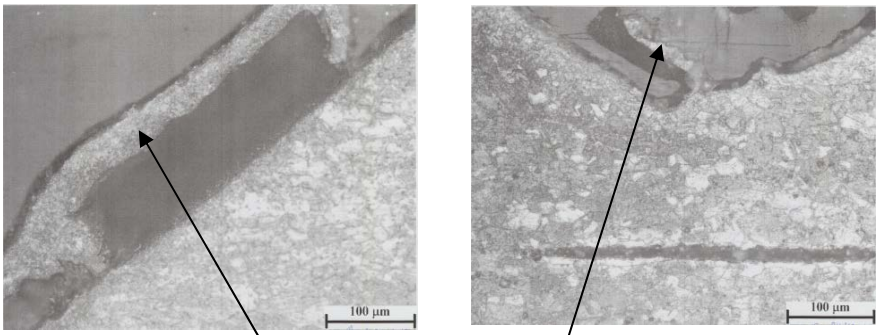
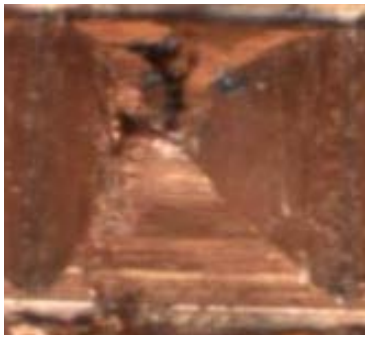
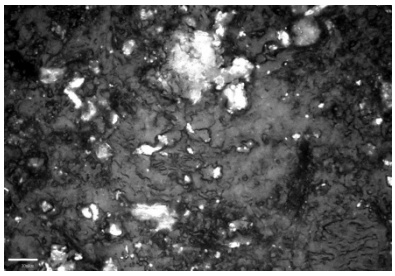


Рис. 3. Вид нагруженных образцов из Cu с углом раствора $\alpha \sim 120^0$ с откольными слоями близ граней пирамидки, $P \sim 9$ Гпа: а – откольный слой левой грани пирамидки; б – откольный слой правой грани пирамидки

На рис. 4,а приведен вид разрушенной медной пирамидки ($\alpha \sim 60^0$) и шлиф медного образца после воздействия релятивистских электронных пучков, на рис. 4,б – продукты диспергирования на преграде и вид частиц диспергированного металла на преграде после воздействия релятивистских электронных пучков [5].



x45



10 мкм

а б

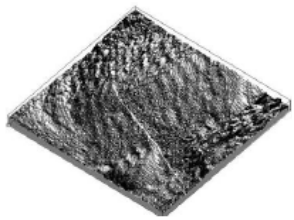
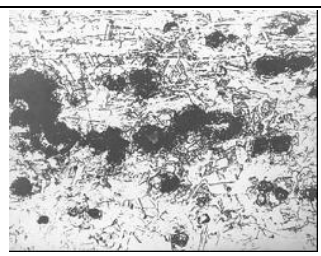
Рис. 4. Вид разрушенной медной пирамидки ($\alpha \sim 60^0$) (а); вид частиц диспергированного металла на преграде (б) [5]

С использованием разработанного математического пакета программ интерактивной системы анализа изображений (ИСАИ) определены количественные характеристики (спектр распределения по размерам) ряда диссипативных структур (плотность гидродинамических мод, спектр продуктов диспергирования, спектр шероховатости центров разрушения). Математическая обработка спектров по размерам позволила определить фрактальную размерность и показатель Херста [1].

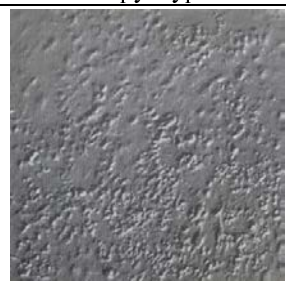
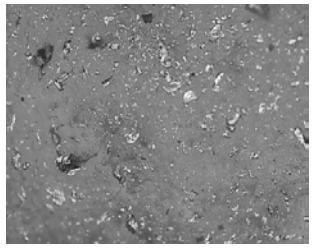
В табл. 2 приведены результаты математической обработки диссипативных структур Cu, Al на различных масштабных уровнях (нано, мезо I, мезо II, макро) и продуктов диспергирования.

Таблица 2

Внешний вид и количественные характеристики диссипативных структур на различных масштабных уровнях

Масштабный уровень	Образец; толщина	Вид диссипативных структур	Фрактальная размерность, d_f	Показатель Херста, H
нано	Cu; 8 ~ 1 мм	 внутренняя поверхность центров разрушения	~ 1,67	–
мезо I	Cu; высота пирамидки h ~ 0,5 мм	 полосы скольжения	~ 1,69 – bottom ~ 1,62 – top	~ 0,86 – bottom ~ 0,85 – top
мезо II	Cu; 8 ~ 0,2 мм	 центры разрушения	~ 1,82	–

Продолжение таблицы 2

Масштабный уровень	Образец; толщина	Вид диссипативных структур	Фрактальная размерность, d_f	Показатель Херста, H
макро	Cu; 8 ~ 0,2 мм	 шероховатость поверхностей разрушения	~ 1,59	~ 0,58
макро	Al	 продукты диспергирования	~ 1,51 – Al ~ 1,52 – Cu	~ 0,76 – Al ~ 0,64 – Cu

Количественные характеристики диссипативных структур – фрактальная размерность, показатель Херста – на различных масштабных уровнях и количественные характеристики продуктов диспергирования имеют близкие значения для Cu Al после воздействия релятивистских электронных пучков.

Фрактальные размерности диссипативных структур, возникающих в процессе динамического разрушения конструкционных материалов на четырех масштабных уровнях при воздействии релятивистских электронных пучков приведены в табл. 3.

Таблица 3

Фрактальные размерности диссипативных структур на четырех различных масштабных уровнях при воздействии релятивистских электронных пучков

Вид воздействия	Уровень исследований			
	Наноуровень (шероховатость внутренних поверхностей центров разрушений)	Микроуровень		Макроуровень (шероховатость поверхности разрушения)
		мезоуровень I (каскад полос скольжения кристаллической решетки)	мезоуровень II (каскад центров разрушения)	
Релятивистские электронные пучки	1,565 ± 0,097	1,684 ± 0,126	1,496 ± 0,113	1,486 ± 0,074

Ранее было показано [1] что в явлении динамического разрушения металлов соблюдается концентрационный критерий, когда линейный масштаб элементов диссипативных структур и среднее расстояние между элементами диссипативных структур в широких масштабно-временных диапазонах подчиняется соотношению

$$\tilde{N} = \frac{r}{D},$$

где r – среднее расстояние между элементами диссипативных структур, D – размер элементов диссипативных структур, C – концентрация.

Переход неравновесной системы разрушаемого, диспергируемого тела с одного масшабно-временного уровня на следующий подчиняется концентрационному критерию, когда концентрация

элементов диссипативных структур на любом масштабно-временном уровне достигает критического значения. Концентрационный критерий выражает масштабно – инвариантную природу диссипативных структур. Необходимо отметить, что каскады диссипативных структур на различных масштабно-временных уровнях являются фрактальными кластерами.

Из вышеизложенного следует, что масштабно – инвариантный концентрационный критерий укрупнения элементов диссипативных структур действует в масштабно – инвариантном ансамбле (кластере) элементов диссипативных структур. В рамках масштабно – инвариантного распределения элементов диссипативных структур в неравновесной системе и масштабно – инвариантного концентрационного критерия элементов диссипативных структур происходит катастрофическое развитие необратимых деструктивных процессов. Именно масштабная инвариантность каскада диссипативных структур на различных масштабно-временных уровнях является признаком критического состояния неравновесной системы. Подобно переходу жидкость – газ, ферромагнитному переходу, переходу порядок – беспорядок в сплавах в критических точках, поведение, например, плотностной переменной описывается степенной зависимостью.

Самоорганизация системы в критическое состояние, когда существуют взаимодействия между элементами системы, распространяется на всю систему. При эволюции системы каждый масштабно-временной уровень достигает критического состояния. Результаты исследований, приведенные в работах [1, 5] показали, что эффекты самоорганизации сильно неравновесных систем могут происходить на масштабах от десятков нанометров до долей миллиметров и временных интервалах доли-десятки наносекунд.

Установление универсальных релаксационных признаков, общих для неравновесных систем различной природы позволяет прогнозировать поведение неисследованных конденсированных сред в экстремальных условиях.

На рис. 5 приведена зависимость среднего расстояния $r = N^{-1/3}$ между элементами диссипативных структур от их среднего размера <*D*> для: каскад центров разрушения (воздействие релятивистских электронных пучков), Cu (Δ = 0,37 мм), бронза (Δ = 0,3 мм), Fe (Δ = 0,5 мм) – мезоуровень II; микротрещины: полимеры – мезоуровень I; каскад полос скольжения (ударно-волновое нагружение) Ti; шероховатость на поверхности (мезоуровень I) Cu; шероховатость на поверхности (нано-уровень): Au, Cu; полосы скольжения в грани Cu пирамидки (в верхней и нижней частях) [1, 5].

Данные, представленные на рис. 5, показывают, что процесс укрупнения (переход с одного масштабно-временного уровня на другой) полос скольжения кристаллической решетки и шероховатость поверхности разрушения Cu образца при амплитудно-временных характеристиках внешнего воздействия подчиняются концентрационному критерию.

Процессы динамического разрушения и диспергирования конструкционных материалов, согласно данным рис. 5, являются аналогами критических явлений. Амплитудно-временные диапазоны неравновесных состояний для исследованных металлов $P \sim 1$–20 ГПа, $t \sim 10^{-6} - 3 \cdot 10^{-10}$ с.

Существует несколько видов распределения вероятностей величин, характеризующих различные физические процессы. Наиболее распространенные – это экспоненциальное распределение $P(x) \sim e^{-x}$, нормальное (Гауссово) распределение $P(x) \sim \exp\left\{-\dfrac{(x-m)^2}{2\sigma^2}\right\}$, степенное распределение $P(x) \sim x^{-d_f}$ (см. рис. 6).

Экспоненциальное распределение описывает, например, распределение молекул газа по энергиям. Нормальному распределению подчиняются величины, получающиеся при сложении большого числа независимых случайных слагаемых.

Примером степенного распределения на рис. 6 является распределение по размерам каскадов центров разрушения в различных металлах, полос скольжения кристаллической решетки, распределение продуктов диспергирования.

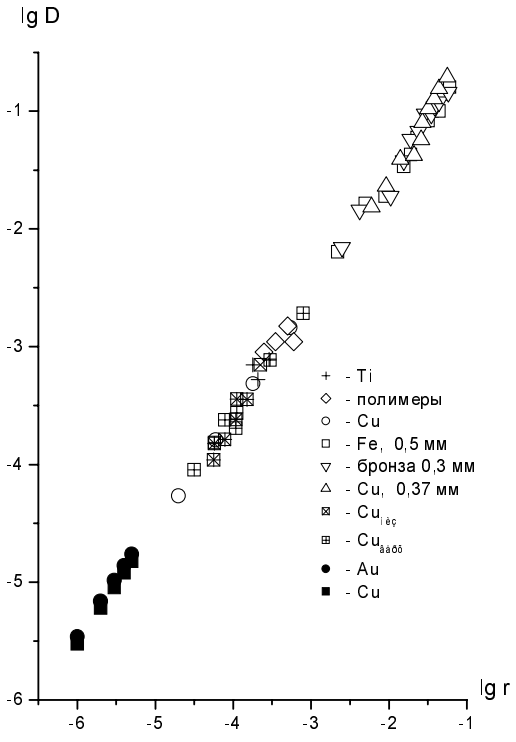


Рис. 5. Зависимость среднего расстояния $r = N^{-1/3}$ между элементами диссипативных структур от их среднего размера <*D*>. Размеры *r* и <*D*> приведены в см

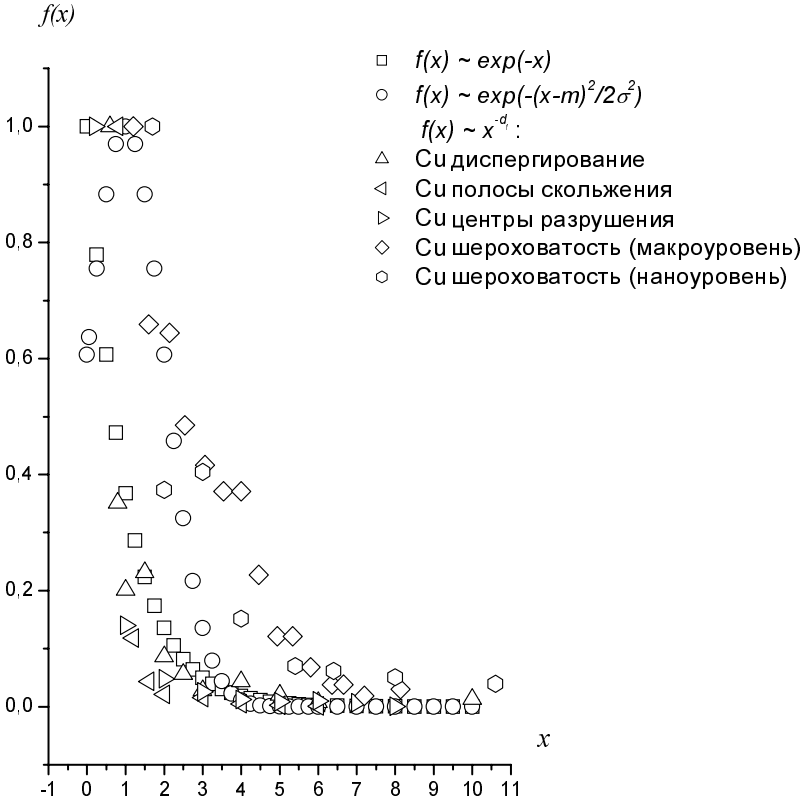


Рис. 6. Распределение вероятностей величин, характеризующее различные физические процессы

На рис. 6, 7 приведен типичный вид плотности вероятности величин, распределенных в соответствии с экспоненциальным, нормальным и степенным законами с различными представлениями данных по осям. В полулогарифмическом масштабе по оси ординат, нормальное и экспоненциальное распределение соответственно представляются в виде параболы и прямой. На рис. 6 (двойной логарифмический масштаб) степенной закон распределения имеет вид прямой, что говорит о скейлинговом поведении, т. е., об отсутствии выделенных масштабов.

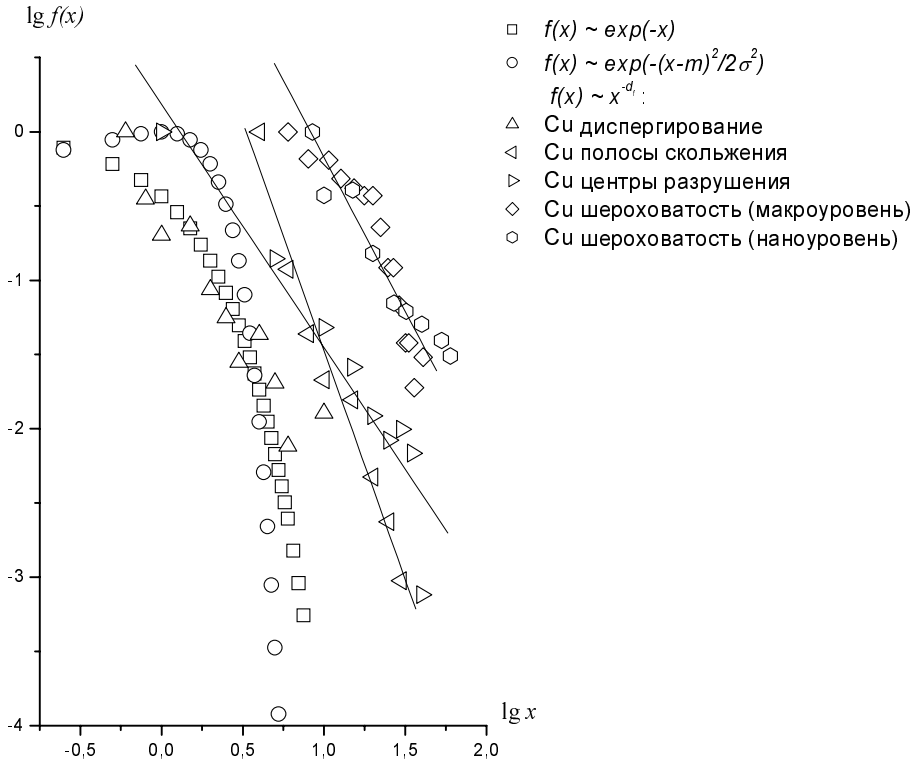


Рис. 7. Распределение вероятностей величин, характеризующее физические процессы в двойном логарифмическом масштабе

Для построения графика нормального распределения среднее значение m и дисперсия σ выбирались равными единице, $x \in [0 \div 10]$; для экспоненциального распределения $x \in [0 \div 10]$.

Разница между нормальным и степенным распределением носит принципиальный характер. Если статистика системы описывается нормальным распределением, то 99,7 % событий отклонятся от среднего значения не более чем на 3σ (так называемое правило трех сигм), а вне диапазона 5σ находятся всего лишь 10^{-6} событий. Поэтому можно пренебречь редкими событиями вне интервала 3σ.

При степенном распределении плотность вероятности на хвостах распределения существенно больше, чем при нормальном распределении и редкими событиями пренебречь нельзя, например, частота f столкновений метеоритов с Землей, как функция их диаметра D подчиняется степенному распределению $f(D) \sim D^{d_f}$. При увеличении диаметра метеорита с 1 метра до 100 метров частота столкновения с Землей уменьшается с одного события в год до 10^{-4} событий в год. Распределение метеоритов по размерам, которые возникают при фрагментации небесных тел, подчиняется степенному закону, Земля в данном случае играет роль преграды [1].

Масштабная инвариантность возникающих диссипативных структур указывает на то, что неравновесная система достигла критического состояния.

Свяжем сжимаемость $\chi = \dfrac{\partial V}{\partial P}$ образца нагруженного металла с плотностью центров разрушения N, используя парный потенциал $\varphi(r)$ взаимодействия атомов в кристалле.

Объемный модуль упругости выражается через вторую производную потенциала парного взаимодействия и межатомного расстояния a, см., например, [8, 9]: $K \sim \frac{\varphi''(0)}{\alpha a}$ при $x \to 0$. Выражение для K в сечении тела можно выразить следующим образом $K = NK_{am}$. $N \sim 1/a^2$ – число атомов на см^2, K_{am} – «упругий модуль» на атом. Если в сечении тела (разрушаемого образца) существует каскад центров разрушения, то это означает, что в сечении имеется n_p разорванных межатомных связей. Следовательно, выражение для K можно представить:

$$K' = (N - n_p)K_{\text{ат}}, \tag{3}$$

K' – объемный модуль «пористого» металла.

Число разорванных межатомных связей как функцию плотности N_P центров разрушения выразим следующим образом

$$n_p = \frac{N_p^{2/3}\overline{D}^2}{a^2}, \tag{4}$$

где N_p – объемная плотность центров разрушения, $\overline{D}$ – средний размер центров разрушения

$$\overline{D} = \frac{\sum N_i(D_i)\cdot D_i}{\sum N_i(D_i)}, \tag{5}$$

$N_i(D_i)$ – число центров разрушения размером D_i.

Выражение для K' через N_p принимает вид

$$K' = K_{\text{ат}}\left(\frac{1}{a^2} - \frac{N_p^{2/3}\overline{D^2}}{a^2}\right) = \frac{K_{\text{ат}}}{a^2}\left(1 - N_p^{2/3}\overline{D^2}\right). \tag{6}$$

Выражение для восприимчивости (сжимаемости)

$$\chi \sim \frac{a^2}{K_{\text{ат}}(1 - N_p^{2/3}\overline{D^2})} \sim A\frac{1}{(1 - N_p\overline{D^2})}, \tag{7}$$

A – коэффициент.

Это выражение показывает, что чем больше число центров разрушения N_P, тем больше χ.

Используя выражение для временной зависимости плотности центров разрушения $N(t)$ [1]

$$N(t) \sim B\left(1 - t/t_p\right)^{-1,2}, \tag{8}$$

B – коэффициент,

и выражение (5); учитывая $\Sigma N_i = N_{tot}$ имеем

$$\overline{D} \sim N_{tot}^{-1};\ \ D^2 \sim N^{-2}. \tag{9}$$

Согласно выражению (7)

$$\overline{D}\cdot N^{2/3} = N^{-2}\cdot N^{2/3}. \tag{10}$$

Тогда восприимчивость χ описывается выражением

$$\chi \sim \frac{1}{\left(1 - \frac{t}{t_p}\right)^{2,53}}. \tag{11}$$

На рис. 8 приведен вид восприимчивости.

В работе [7] отмечено, что скорость звука в конденсированных средах вблизи критической точки стремится к нулю.

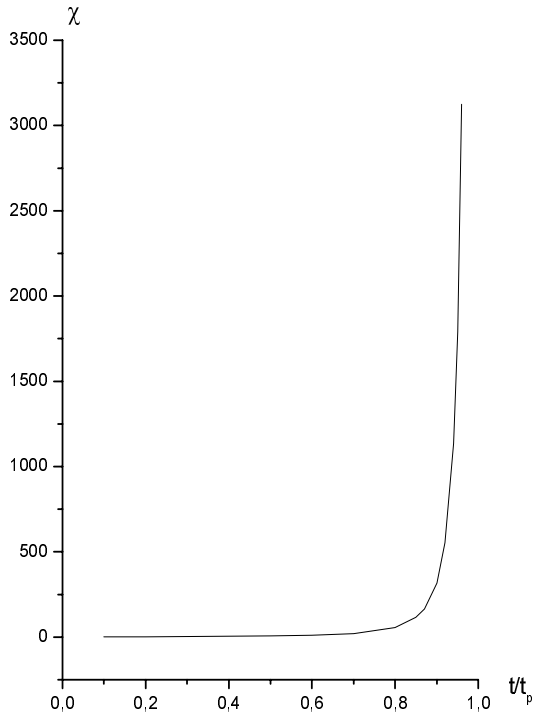


Рис. 8. Временная зависимость восприимчивости
на масштабе времени разрушения

В низкочастотном пределе скорость звука V_0 имеет вид

$$V_0 = \left(\frac{\partial P}{\partial \rho}\right)^{1/2} = V\left(-\frac{1}{\mu}\frac{\partial P}{\partial V}\right)^{1/2}; \qquad (12)$$

в критическом состоянии $\left(\frac{\partial P}{\partial V}\right) = 0$ [7].

Вышеизложенное показывает, что явление динамического разрушения является аналогом критических явлений, что определяет выбор математического аппарата для адекватного описания этого явления.

В неравновесных диссипативных системах в большинстве случаев для описания распределения элементов системы имеет место степенное распределение $f(x) \sim x^{-\alpha}$; x – размер элемента системы, α – параметр распределения.

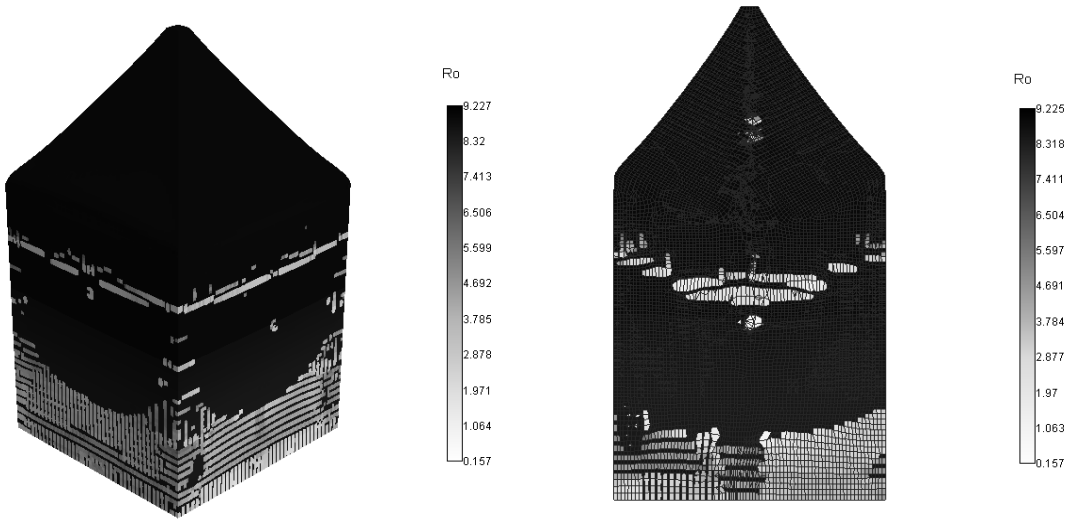
Особенность таких распределений – отсутствие выделенного характерного размера, именно такие распределения характерны для масштабно-инвариантных множеств, например, для мультифракталов.

Степенные распределения являются статистическим выражением масштабной инвариантности. Распределение числа $N(D)$ центров разрушения от размера имеет вид $N(D) \sim D^{-\alpha}$, распределение числа $N(l)$ элементов шероховатости поверхности от их размера l имеет вид $N(l) \sim l^{-\beta}$, распределение продуктов диспергирования по размерам $N(d) \sim d^{-\gamma}$, распределение числа элементов шероховатости внутренней поверхности центра разрушения $N(l) \sim l^{-\beta}$, α, β, γ – параметры распределения. Приведенные примеры распределения элементов диссипативных структур в кластерах показывают, что кластеры являются фрактальными множествами [1–4].

Процессы, происходящие в масштабно-инвариантных системах, устроены одинаково на всех уровнях организации физической системы. Здесь нет, как в простых системах выделенного масштаба, который бы отвечал за «основные процессы» к описанию которых можно было бы свести изучение свойств системы. Тем самым является невозможным сведение системы к набору независимых объектов, что и означает ее сложность.

Проведено численное моделирование нагруженных медных мишеней с нанесенными возмущениями в виде пирамидок ($\alpha \sim 60^0$) от момента нагружения до макроразрушения с использованием лагранжевой методики ТИМ-3D [11]. Моделируемые параметры нагружения близки к натурным.

На рис. 9 приведено распределение плотности в объеме пирамидки. Расчет 3D с использованием модели откольного разрушения на момент времени $t \sim 3 \cdot 10^{-6}$ с.



а б

Рис. 9. Распределение плотности: в объеме образца – а; в осевом сечении – б. Расчет 3D с использованием откольного разрушения. Время 3 мкс

С использованием численной методики для расчета двумерных течений [12] проведено численное моделирование диспергирования в нагруженных образцах из Al с нанесенными возмущениями в виде пирамидок с углом раствора $\alpha \sim 90^0$.

С использованием численной методики для расчета ТИМ-2D [12] проведено численное моделирование воздействия импульсов релятивистских электронных пучков на медную фольгу (рис. 11).

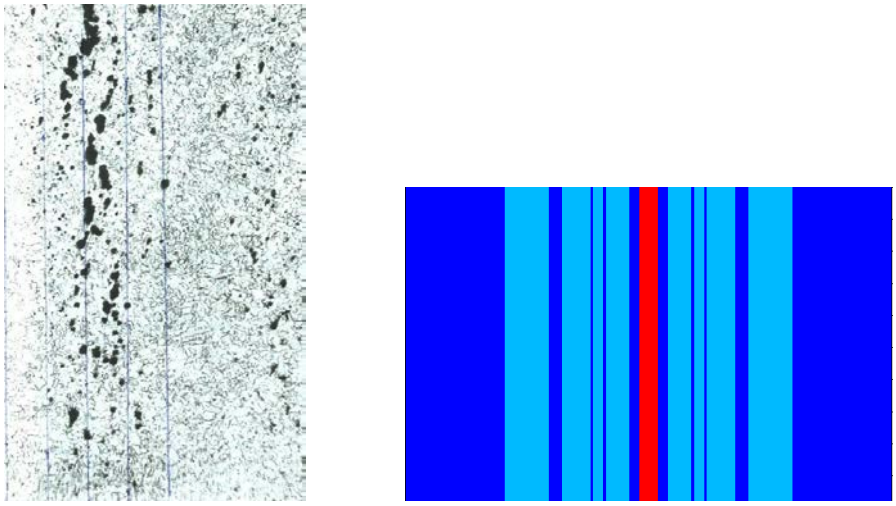
Плотность поглощенной энергии электронного пучка для спектра электронов ускорителя РИУС-5 определялась методом Монте-Карло. Давление в области откольного разрушения определялось из уравнения состояния Ми–Грюнайзена.

С использованием математического пакета программ ИСАИ получены количественные характеристики поверхности разрушения нагруженных образцов. Определена фрактальная размерность и показатель Херста шероховатости поверхности разрушения образцов при воздействии релятивистских электронных пучков ускорителя РИУС-5.



Рис. 10. Вид струеобразования из образца Al на момент времени 17 мкс

Фундаментальные свойства процесса динамического разрушения и диспергирования металлов, установленные в указанной редакции исследований, вытекают из неравновесного, диссипативного характера явления, которому присущи масштабная инвариантность возникающих пространственно-временных диссипативных структур. Масштабная инвариантность возникающих диссипативных структур указывает на то, что неравновесная система достигла критического состояния. Из вышеизложенного следует, что явление динамического разрушения и диспергирования металлов являются аналогами процессов, протекающих в критических точках, что и определяет применение определенного математического аппарата для описания указанных явлений. Реализация будущего состояния неравновесной диссипативной системы через последовательность бифуркаций делает эволюцию системы необратимой.



а б

Рис. 11. Шлиф Cu образца после воздействия релятивистских электронных пучков (а); поле поврежденности образца на момент разрушения при $\Delta = 0{,}037$см и $E_0 = 255$ Дж/г; ω – отношение разрушенного объема V_p к исходному V_0; $\omega = V_p/V_0$; голубой цвет $\omega = 0$, красный цвет $\omega = 1$ [1]

Литература

1. Кошелева Е. В., Пунин В. Т., Сельченкова Н. И., Учаев А. Я. Общие закономерности иерархических релаксационных процессов в металлах при воздействии импульсов проникающих излучений: монография – Саров: РФЯЦ-ВНИИЭФ, 2015. – 211 с.
2. Пригожин И., Николис Г. Познание сложного: Введение. Пер. с англ. / Предисл. Г. Г. Малинецкого. Изд. 4-е. – М.: ЛЕНАНД, 2014. – 360 с. (Синергетика: от прошлого к будущему. № 8).
3. Пригожин И., Стенгерс И. Порядок из хаоса: Новый диалог человека с природой. Пер. с англ. / Общ. ред. Аршинова В. И., Климонтовича Ю. Л. и Сачкова Ю. В. – М.: Прогресс, 1986. – 432 с. (Знание – Сила, 1997. № 9).
4. Вильсон К., Когут Дж. Ренормализационная группа и ε-разложение. Пер. с англ. Загребнова В. А. / Под ред. Федянина В. К. М.: Мир, 1975. (Новости фундаментальной физики, выпуск 5).
5. Kosheleva E. V., Sel'chenkova N. I., Sokolov S. S., Trunin I. R. and Uchaeva A. Ya. On the Similarity of Dynamic Destructive Processes in Metals of Different Geometry under the Effect of External Impacts with Various Amplitude-Time Characteristics // Physics of Atomic Nuclei. 2018. Vol. 81, No. 10. P. 1477–1483.
6. Введение в физику твердого тела. Ч. Китель. Перевод с четвертого американского издания А.А. Гусева и А. В. Пахнева под общей редакцией А. А. Гусева. Учебное руководство. Главная редакция физико-математической литературы издательства «Наука», М., 1978. – 792 с.
7. Базаров И. П. Термодинамика. Учеб. пособие для ун-тов. Изд. 2-е, перераб. и доп. – Москва: Высшая школа, 1976. – 447 с.
8. Ашкрофт Н., Мермин Н. Физика твердого тела. – Москва: Мир, 1979. – 422 с.
9. Штремель М. А. Прочность сплавов. Часть I. Дефекты решетки: Учебник для вузов. – Москва: МИСИС, 1999. – 384 с.
10. Соколов С. С., Панов А. И., Воропинов А. А., Новиков И. Г. и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 3. С. 37–52.
11. Янилкин Ю. В., Шанин А. А., Ковалев Н. П. и др. Комплекс программ ЭГАК для расчетов двумерных течений многокомпонентной среды // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1993. Вып. 4. С. 69–75.

12. Соколов С. С., Воропинов А. А., Новиков И. Г. и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29–43.

ON CRITICAL BEHAVIOR OF DYNAMIC SYSTEMS AT HIGH-INTENSE ACTION

E. V. Kosheleva, N. I. Sel'chenkova, S. S. Sokolov, I. R. Trunin, A. Ya. Uchaev

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper is dedicated to establishing of common regularities of dynamic destructive processes (dynamic failure and dispersion), passing in metal sat high-intense external action in the dynamic range of non-equilibrium states $t \sim 10^{-6} \div 3 \cdot 10^{-10}$ s. Specification of quantity of kinetics processes is determined. They are system's response to the external action. There were specified fractal dimensionality, Hearst index of dissipative structures – dispersion products, hydrodynamic oscillations, failure centers clusters.

Scale invariance of originating dissipative structures points out to the fact that non-equilibrium system has approached a critical state. Also the evidence that the system is in a critical state, is conversion the compressibility (receptivity). Using a pair potential of crystal lattice, there was specified an expression for compressibility (receptivity) of the loaded sample, depending on the density of failure centers.

There was carried out a numerical modeling of dispersion processes, appearing at relativistic electron pulse effects up on metal samples with applied perturbations in the form of pyramids, using un irregular Lagrange technique TIM-3D and two-dimensional Lagrangian–Eulerian calculation procedure; numerical modeling of the dynamic failure process of thin metallic foils under the action of relativistic electron pulses using TIM 2 calculation procedure was performed.

The proposed methodology to study high-intense dynamic destructive processes is based on the approach, which permits to investigate specification of quantity of kinetics processes, being a response of non-equilibrium systems to external action, what allows behavior prediction of unstudied different geometry metal samples under extreme conditions.

Key words: dynamic destructive processes, dynamic failure, dispersion, high-intense external action, extreme conditions, scale invariance, critical state, compressibility, pair potential of crystal lattice.

УДК 004.94:528.952
DOI 10.53403/9785951505071_2022_254

СРЕДСТВА МУЛЬТИАГЕНТНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ВИЗУАЛИЗАЦИОННО-ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ «ОПТИМУС»

И. А. Крючков, Д. В. Ежов, А. В. Огородников

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Визуализационно-интеграционная платформа «ОптИМУС» предоставляет возможности для разработки программных комплексов, использующих технологии мультиагентного имитационного моделирования. В составе платформы выполнение расчетов на вычислительных системах (в том числе и с распределенной памятью) обеспечивает специально разработанный мультиагентный решатель. В докладе показаны подходы, использованные при разработке данного решателя, отмечены особенности и преимущества.

Ключевые слова: мультиагентное имитационное моделирование, ВИП «ОПТИМУС», агент, решатель.

Во ФГУП «РФЯЦ-ВНИИЭФ» разработана визуализационно-интеграционная платформа «ОптИМУС»[1], предназначенная для обеспечения возможности эффективной и качественной разработки специализированных программных комплексов поддержки принятия решений, имитационного моделирования и оптимизации с высококачественным представлением результатов.

Базовой технологией имитационного моделирования в ВИП «ОптИМУС» является подход мультиагентного моделирования, с помощью которого анализируемая система представляется в виде совокупности программных агентов. Имитационная модель, построенная по агентной технологии, описывает поведение децентрализованных агентов, определяющих поведение всей системы в целом. При этом поведение агентов определяется на индивидуальном уровне, а глобальное поведение возникает как результат деятельности множества агентов [2].

ВИП «ОптИМУС» представляет собой расширяемый набор компонентов для разработки программных комплексов имитационного мультиагентного моделирования и включает в себя два программных пакета (ПП):

– программный пакет мультиагентного моделирования Оптимус.Решатель;

– программный пакет препостпроцессинга (препостпроцессор) мультиагентных задач Оптимус.Препост с графическим пользовательским интерфейсом (ГПИ).

Краткая схема работы ВИП «ОптИМУС» представлена на рис. 1.

ВИП «ОптИМУС» предоставляет средства мультиагентного имитационного моделирования, которые организуют взаимодействие по формализованным правилам произвольного количества агентов разных типов, запрограммированных в соответствии с определенными требованиями (рис. 2). Для всех агентов используется единый механизм продвижения по времени и обеспечивается информационное взаимодействие (обмен событиями, сообщениями и т. д.).

Разработка агентов является ключевым и одним из наиболее сложных элементов при разработке имитационных мультиагентных моделей. В общем случае, для реализации поведения агентов могут использоваться компилируемые средства программирования (языки программирования C/C++, Fortran, Java и т. д.) или интерпретационные средства (Python, UnrealScript и т. д.).

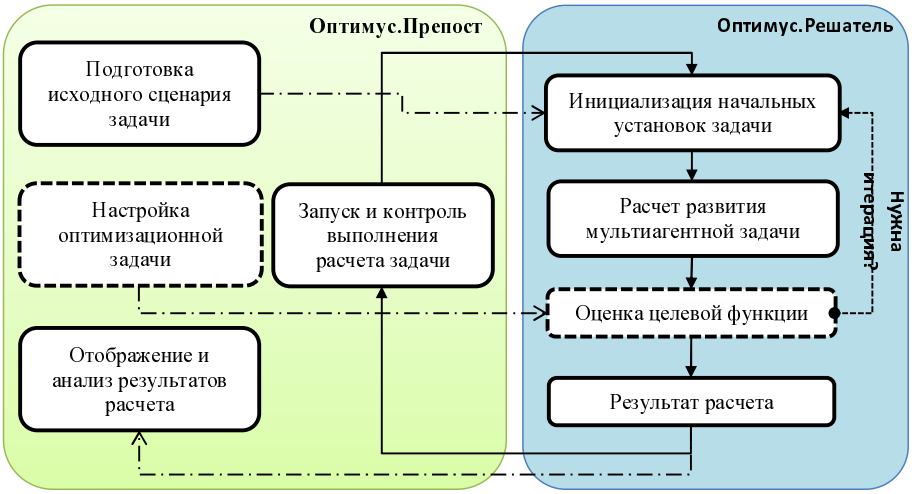


Рис. 1. Краткая схема работы ВИП «ОптИМУС»



Рис. 2. Разнообразие типов и вариантов программных агентов

Разработка агентов может вестись не только непосредственным написанием программного кода, но и с помощью средств визуального программирования. Этот способ создания программ представляет собой манипулирование графическими объектами.

Вычислительные системы, используемые для выполнения расчетов, также могут быть различны: от персональных компьютеров до супер-ЭВМ, работающих под управлением различных операционных систем (Windows, различные версии Linux).

В связи с этими и другими аспектами существуют различные реализации специальных мультиагентных решателей, построенных на различных принципах и технологиях. Наиболее популярными являются реализации на языке программирования Java (AnyLogic [3], Jade [4], Repast [5] и т. д.). Их главным недостатком является ограниченная возможность использования вычислительных систем с распределенной памятью. Это обстоятельство накладывает серьезные ограничения на размер исследуемых систем. При этом программные продукты на Java зачастую предоставляют богатые пользовательские интерфейсы и обеспечивают использование определенных стандартов разработки агентов.

Среди программных продуктов, предназначенных для использования распределенных вычислительных систем, выделяются реализации на технологиях HLA и MPI. Качественные решения, использующие HLA, являются сугубо коммерческими (например, vsTasker [6]) и не доступны для тестирования. Доступные решения не реализуют специфику мультиагнетных систем, и, как следствие, слабо пригодны без серьезных доработок для решения практических задач.

Одной из наиболее популярных программ для мультиагентного моделирования на распределенных вычислительных средствах является RepastHPC (реализация на языке программирования C++ с использованием технологии MPI). Но в RepastHPC нет стандартизованного и поддерживаемого библиотекой способа организации взаимодействия между агентами: для воздействия на агента, необходимо вызывать методы объекта взаимодействия или редактировать его параметры напрямую. Для развития этих возможностей требуется создание модулей или надстроек над Repast. Таким образом, разработка и расширение агентной системы могут быть весьма проблематичными.

В связи с этим были сформулированы требования к средствам мультиагентного моделирования:

- обеспечение функционирования агентов;
- обеспечение взаимодействия агентов;
- продвижение агентов в рамках модельного времени;
- параллельные вычисления на вычислительных системах с распределенной памятью;
- сохранение результатов;
- накопление протоколов и статистик;
- удобство разработки агентов;
- формализованность и расширяемость функционала агентов;
- формализованность и расширяемость способов взаимодействия агентов;
- взаимодействие с компонентами ВИП «ОптИМУС» (Препост, БД, оптимизатор, Opal++ и т. д.);
- поддержка стандарта FIPA.

Основой для организации мультиагентных взаимодействий стала поддержка стандарт FIPA (Foundation for Intelligent Physical Agents) [7]. Изначально FIPA предназначен для создания стандартов взаимодействия между собой агентных приложений и агентных систем. Эти стандарты включают в себя спецификации для инфраструктуры агентов и агентных приложений.

Общий вид агентной системы, реализованной на базовых принципах стандарта FIPA, изображен на рис. 3.

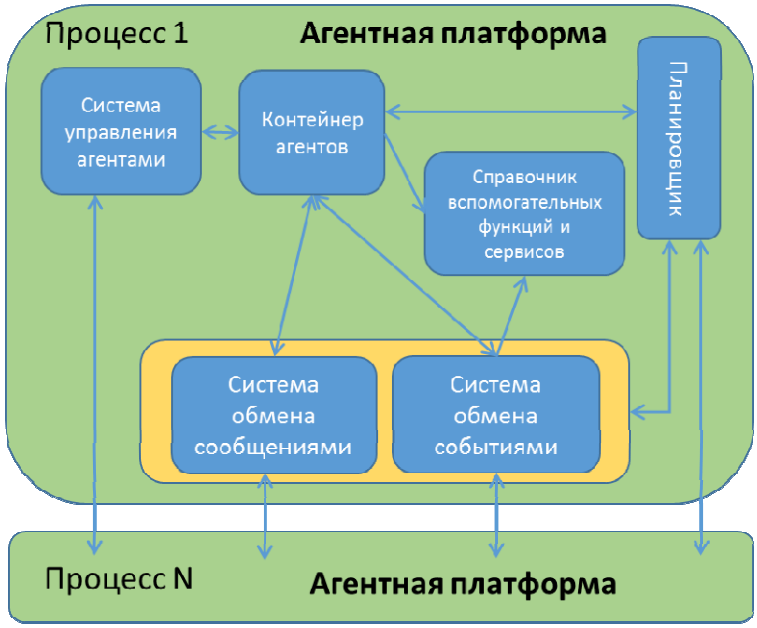


Рис. 3. Схема взаимодействия управляющих компонент мультиагентного решателя

Таким образом, агентная система состоит из следующих логических компонентов, каждый из которых представляет собой набор логических возможностей:

– агентная платформа (Agent Platform, AP) – обеспечивает инфраструктуру, в которой агенты могут создаваться и функционировать. Агентаная платформа функционирует на каждом вычислительном процессе;

– агент (Agent) – программный объект (существует в единственном экземпляре на каком-либо вычислительном процессе), реализующий автономную функциональность. Агенты обмениваются информацией с помощью систем обмена сообщениями и событиями;

– сервис (Service) – программный объект (существует на каждом вычислительном процессе), реализующий функциональность, необходимую для использования агентам. Сервисы получают информацию непосредственно от агентов или от системы обмена событиями;

– справочник вспомогательных функций и сервисов – компонент, предоставляющий для агентов доступ к сервисам;

– система управления агентами (Agent Management System, AMS) – осуществляет управление доступом к агентам. AMS поддерживает актуальное состояние контейнера агентов. Каждый агент обязан зарегистрироваться в AMS для того, чтобы получить корректный идентификатор;

– система обмена сообщениями (Message Transport Service, MTS) – средство обмена информацией (передачи сообщений) между агентами (в том числе и на разных вычислительных процессах);

– система обмена сообытиями (Event Transport Service, ETS) – средство публикации и доставки событий (порождаемых агентами) в мультиагентной системе для агентов и сервисов (в том числе и на разных вычислительных процессах);

– планировщик – специальное средство, которое обеспечивает общую синхронизацию вычислительных процессов, продвижение модельного времени, активацию агентов и т. д.

Цель архитектурной поддержки стандарта FIPA состоит в том, чтобы обеспечить быстрое и формализованное создание агентных систем, которые легко расширяются и дополняются новыми типами агентов.

Ключевой сущностью мультиагентного моделирования является агент. Агент может представлять собой объект произвольной сложности, как по составу, так и по логике поведения. В связи с этим мультиагентный решатель должен обеспечивать универсальные средства декомпозиции составных частей и логики, а также гарантировать возможности взаимодействия с другими частями ВИП «ОптИМУС». Например, для сохранения результатов работы (протоколы работы, апроксиматоры движения и т. д.), для формирования логики поведения агента на визуальном языке Opal++ или для взаимодействия с геоинформационными средствами. На рис. 4 в общем виде приведена структурная схема агента.

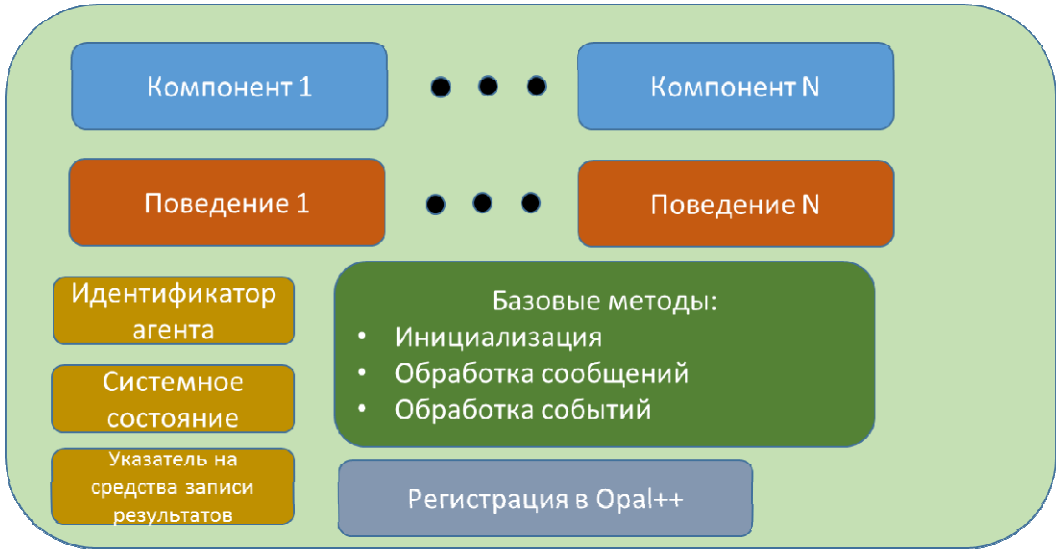


Рис. 4. Схема взаимодействия управляющих компонент мультиагентного решателя

Все процессы жизненного цикла агентов протекают в модельном времени. Модельное время является воспроизведением физического времени в модели. Очевидно, что физическое время величина непрерывная, в то время как модельное время – дискретная. При этом каждый программный агент может обладать собственной дискретностью времени. В связи с этим каждый агент функционирует с собственной дискретностью по времени в рамках общего модельного времени. Сущностью имитационного моделирования является продвижение модельного времени от 0 до конечной величины и выполнение событий, связанных с определенными значениями модельного времени. Событие в модели – это программный образ значимого с точки зрения разработчика модели изменения в моделируемой системе. Соответственно, одной из основных задач имитационного моделирования является правильное отображение порядка и временных отношений между изменениями в моделируемой системе на порядок выполнения событий в модели, т. е. события в модели должны выполняться в хронологическом порядке в модельном времени (см. рис. 5).

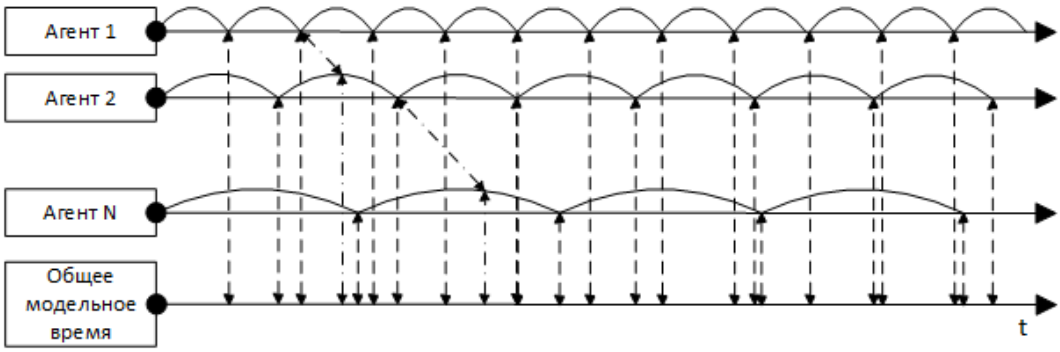


Рис. 5. Пример схемы продвижения по времени

В мультиагентном решателе ВИП «ОптИМУС» используется последовательная консервативная система продвижения по времени, при которой каждый агент моделируемой системы имеет возможность рассчитывать свое состояние и свой следующий модельный шаг. При этом внешние события, порожденные другими программными агентами, могут вызвать какой-либо функционал данного в необходимый для них момент времени (в рамках общего времени моделирования).

Таким образом, модуль имитационного моделирования берет на себя обязанности по созданию условий для своевременного вызова функций агентов, доставки вызовов, сообщений и событий между агентами.

Агент характеризуется набором атрибутов, описывающих его состояние в конкретный момент времени (модельный шаг). При расчете своего состояния на следующий шаг, агент имеет возможность использовать данные о состоянии других агентов на тот же самый шаг, на который выполняется расчет.

В процессе работы агент может передавать информацию к другим агентам через среду моделирования, которая гарантирует ее своевременную доставку до адресата.

Каждый агент в имитационном моделировании выполняет какую-либо свою предметную функцию. На рис. 6 приведен пример логики функционирования сложного интеллектуального агента.

При этом один и тот же тип агента, в зависимости от своих внутренних состояний (определяющихся в поведениях) может в конкретные моменты модельного времени выполнять разные поведения. Для этого, каждый агент может иметь несколько сущностей «Поведение» и переключать их в соответствии с необходимостью. Таким образом, каждое поведение является самостоятельным объектом и может быть реализовано различными способами. В ВИП «ОптИМУС» доступны три способа реализации поведений: программный код на C++, скриптовый и визуальный код на Opal++.

Любые процессы в мультиагентном моделировании не могут обойтись без взаимодействия агентов между собой. В решателе обеспечиваются три уровня взаимодействия:

– внутри агента – это взаимодействие компонент агента между собой;

– событийно – взаимодействие по принципу публикация-подписка. То есть одни агенты публикуют различные события в общее пространство, а другие агенты подписываются на эти события и получают их в требуемые моменты;

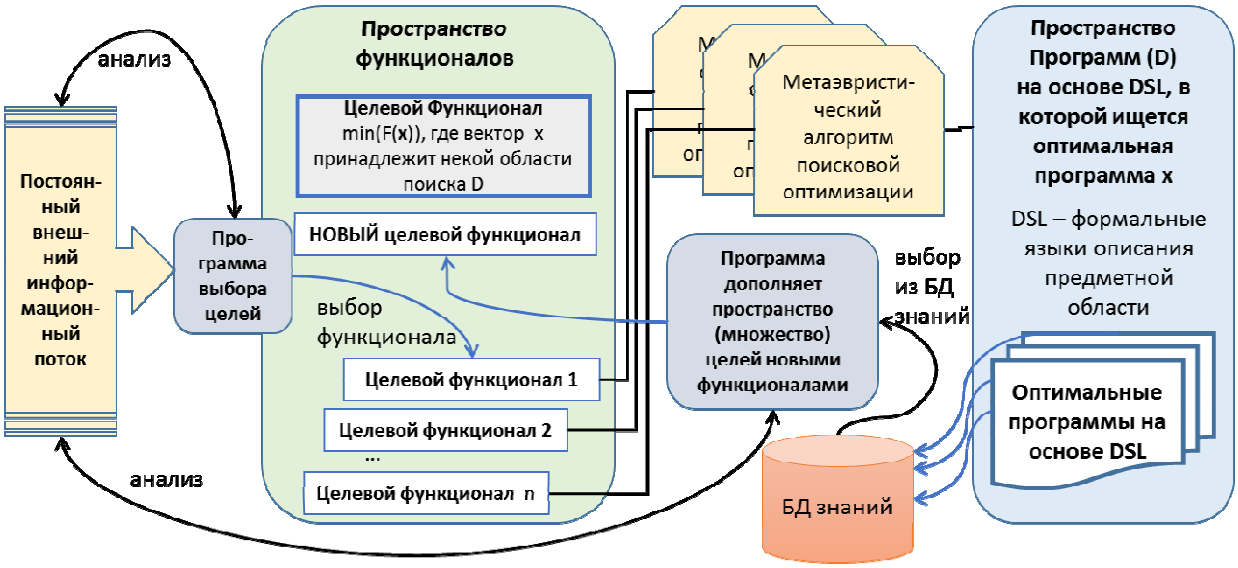


Рис. 6. Схема взаимодействия управляющих компонент мультиагентного решателя

– адресно – взаимодействие по принципу прямой передачи сообщения между известными абонентами.

Эти уровни взаимодействия покрывают практически весь спектр необходимой коммуникации между агентами.

Таким образом, в составе ВИП «ОптИМУС» реализован и развивается решатель мультиагентного имитационного моделирования. Данный решатель обеспечивает базовый уровень технологий при разработке мультиагентных систем и их функционирование на вычислительных системах с распределенной памятью [7]. Формализованный подход к структуре программных агентов и принципам их взаимодействия минимизирует затраты разработчика на технические моменты реализации, освобождая ресурсы для творческой работы с логикой агентов.

Литература

1. Свидетельство о государственной регистрации программы для ЭВМ № 2019617157 «Визуализационно-интеграционная платформа для оптимизационного имитационного моделирования и управления системами (ВИП «ОптИМУС»)» / Коваленко О. В., Крючков И. А., Ежов Д. В., Огородников А. В., Ерошкина И. В., Собанин Д. С., Хочкин Н. И., Варгина Е. Ф., Тихомиров Ю. В., Рыжих А. В., Васильева Е. А., Кондратьев А. Б.

2. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М., 2006.

3. Комплекс моделирования AnyLogic [Электронный ресурс]: Официальный ресурс – Режим доступа: https://www.anylogic.ru.

4. Комплекс моделирования JADE [Электронный ресурс]: Материалы по реализации – Режим доступа: http://jade.cselt.it/papers/2003/WhitePaperJADEEXP.pdf.

5. Набор инструментов мультиагентного моделирования Repast Agent Simulation Toolkit [Электронный ресурс]: Официальный ресурс исходных кодов – Режим доступа: http://repast.sourceforge.net.

6. Стандарт FIPA [Электронный ресурс] Foundation for Intelligent Physical Agents – Режим доступа: http://www.fipa.org.

7. Коваленко О. В., Крючков И. А., Огородников А. В., Ежов Д. В., Собанин Д. С. Возможности визуализационно-интеграционной платформы ОптИМУС для имитационного моделирования вооружений, военной и специальной техники // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2019. Вып. 4.

MULTI-AGENT SIMULATION MODELING TOOLS OF "OptIMUS" VISUALIZATION-INTEGRATION PLATFORM

I. A. Kryuchkov, D. V. Ezhov, A. V. Ogorodnikov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

An "OptIMUS" visualization-integration platform (VIP) has been worked out in FSUE RFNC-VNIIEF and is designed to provide tools for multi-agent simulation modeling that establishes the interaction by formalized rules of a random number of agents of different types programmed in compliance with certain requirements. The paper shows the approaches used to realize the described means and the ways to implement these means in practical problems.

Key words: "OptIMUS" visualization-integration platform, multi-agent simulation, agent, imitation model.

УДК 536
DOI 10.53403/9785951505071_2022_261

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ТЕПЛОПРОВОДНОСТИ И ПРОЧНОСТИ НА МАКРОУРОВНЕ ПРИ СЕЛЕКТИВНОМ ЛАЗЕРНОМ ПЛАВЛЕНИИ В ПРОГРАММНОМ КОМПЛЕКСЕ «ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР»

В. Ю. Кузнецов, П. А. Машенькин, Ю. Н. Дерюгин, Р. А. Барабанов, Е. А. Филимонкин, В. А. Глазунов, А. Ю. Вишняков, О. С. Ларькина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В данной работе представлены результаты разработки первой версии расчетных модулей теплопроводности и прочности пакета программ «Виртуальный 3D-принтер» на макроуровне. Макроуровень характеризуется тем, что множество частиц порошка рассматривается как сплошная среда, характеризующаяся своими макропараметрами (например, пористость). Данные модули базируются на счетных модулях пакета программ ЛОГОС (разработка РФЯЦ-ВНИИЭФ) и адаптируются согласно проекту «Виртуальный 3D-принтер» по разработке программного обеспечения для 3D-принтеров. Приведены модели и алгоритмы численного решения процесса селективного лазерного плавления (СЛП) и оценки напряженно-деформируемого состояния детали, как в ходе плавления, так и в ходе нагружения готовой детали, полученной по технологии СЛП.

Ключевые слова: селективное лазерное спекание, 3D-принтер, оценка напряженно-деформируемого состояния детали.

Введение

СЛП – аддитивная технология послойного получения деталей посредством плавления металлического порошка при взаимодействии с лазерным излучением [1]. Блок моделирование является инструментом для анализа процесса СЛП. Процесс создания технологии печати детали на 3D-принтере является нетривиальным и требует существенных трат времени и денег. Виртуальные испытания технологии печати конкретной детали существенно могут снизить затраты на производстве.

В общих чертах процесс создания технологии печати детали можно представить на блок-схеме (рис. 1):

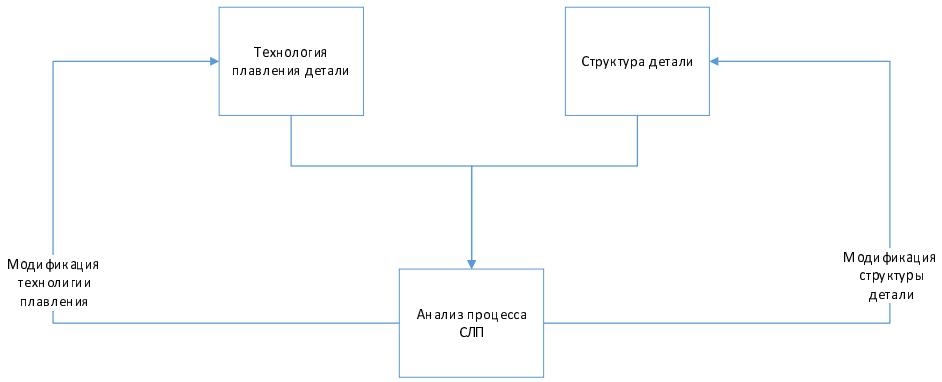


Рис. 1. Блок-схема создания технологии печати детали на 3D-принтере

Процесс создания технологии печати детали на 3D-принтере является итерационным. Из блок-схемы видно, что анализ процесса СЛП является центральным при проведении виртуальных испытаний. Под анализом процесса СЛП понимается анализ напряженно-деформируемого и теплового состояния детали, как в ходе ее выпекания, так и после. Функционал, необходимый для решения задачи теплопроводности, реализован на базе модуля ЛОГОС.HEAT [2]. Функционал, необходимый для решения прочностных задач, реализован на базе модуля ЛОГОС.SA [3].

В процессе печати изделия могут возникать области с повышенным напряжением, которые в свою очередь могут стать причиной возникновения различного рода дефектов (трещины, разрывы). Избежать данную проблему можно правильным подбором режима работы 3D-принтера при печати детали (модификация технологии плавления).

Технология СЛС позволяет осуществлять печать изделий с использованием ячеистых (решетчатых) структур. Замена «сплошных» областей деталей такими структурами привод к значительному снижению их массы (особенно важно в авиапромышленности), но при этом запас прочности должен удовлетворять требованиям безопасности. Изделие после построения в нем решетчатых структур должно подвергнуться серии «виртуальных» испытаний. Если в ходе данных испытаний деталь сможет выдержать заданные нагрузки, то возможно ее использование с заменой сплошных частей решетчатыми структурами (модификация структуры детали). В данном докладе это демонстрируется на примере детали «спинной кейдж».

Математическая модель и алгоритмы численного решения процессов теплопереноса

Процесс теплопереноса при моделировании лазерного сплавления включает в себя, главным образом, нагрев частиц порошка и металла за счет излучения лазера и процесса теплопереноса с учетом фазового перехода металла из твердого состояния в расплавленное. Теплопроводность описывается законом Фурье

$$q = -\chi \operatorname{grad} T, \tag{1}$$

где T – температура, К; χ – коэффициент теплопроводности вещества, Вт/(м·К).

Уравнение сохранения внутренней энергии представим в следующем виде:

$$\frac{\partial}{\partial t}(E) + \operatorname{div}(q) = Q, \tag{2}$$

где E – внутренняя энергия вещества, Дж/м^3; q – вектор плотности теплового потока, Вт/м^2; Q – источник тепла, описывающий нагрев от лазера, Дж/(м^3·с).

Внутренняя энергия вещества в элементарном объеме складывается от энергии металла, находящегося в твердом или жидком состоянии и воздуха, содержащегося в порах. Выразить эту энергию можно по следующей формуле:

$$E = m\rho_g C_g T + (1-m)\rho_m C_m T, \tag{3}$$

где m – пористость, м^3/м^3; ρ_g – плотность газа, кг/м^3; C_g – удельная теплоемкость газа, Дж/(кг·К); ρ_m – плотность металла, кг/м^3; C_m – удельная теплоемкость металла, Дж/(кг·К).

С учетом фазового перехода зачастую используется уравнение теплопроводности в обобщенной постановке:

$$\frac{\partial E}{\partial T}\frac{\partial T}{\partial t} + \operatorname{div}(q) = Q, \tag{4}$$

$$\frac{\partial E}{\partial T} = m\rho_g C_g + (1-m)\rho_m \left(C_m + L\delta\left(T - T^*\right)\right).$$

где δ – дельта функция Дирака; L – скрытая энергия фазового перехода, Дж/кг.

Учет энергии фазового перехода в нестационарном члене используется при решении классической задачи Стефана [4]. При этом значения теплофизических свойств металла определяются сглаженной функцией около температуры фазового перехода.

Система уравнений (1), (2), (4) решается методом контрольного объема. Аппроксимация нестационарного члена и теплового потока реализована подходом Эйлера и с использованием двухточечной аппроксимации потока с коррекцией на неортогональность сетки и возможную анизотропию тензора теплопроводности [5].

Блок-схема работы модуля «Тепло-макроуровень» представлена на рис. 2:

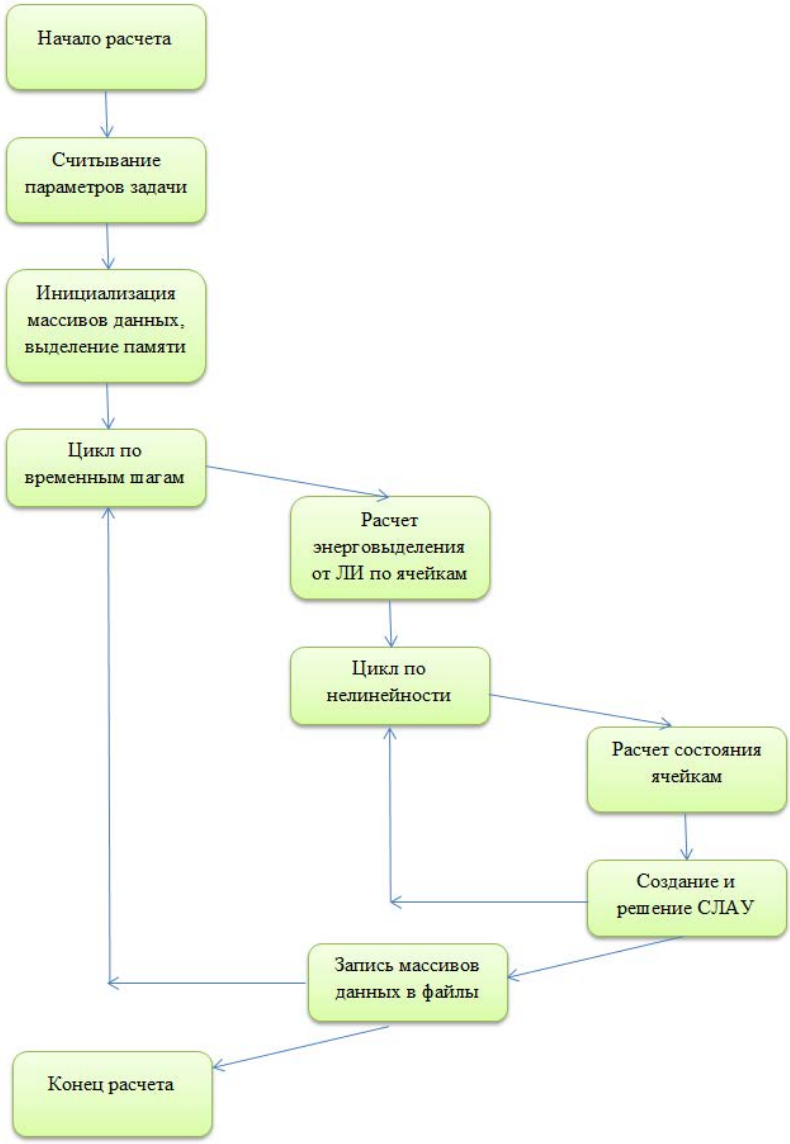


Рис. 2. Схема работы модуля «Тепло-макроуровень»

Верификация модуля «Тепло-макроуровень»

В настоящее время первая версия модуля является самостоятельным исполняемым файлом, данные для которого подготавливаются отчасти в создаваемой программе (геометрия и расчетная воксельная сетка).

Для демонстрации его работы приведем следующий тестовый пример. Рассмотрим процесс выпекания решетчатой структуры, изображенной на рис. 3. На рис. 4 представлен один слайсов (слой) со штриховкой данной детали. Толщина насыпного слоя порошка 0.15 мм. Пористость порошка – 0.4. Материал порошка – сталь. Учтена теплопроводность порошка в зависимости от пористости:

$$\chi = \chi_0 \left(1 - \sqrt[3]{4m^2}\right),\ 0 \le m \le 1/2,$$

где χ_0 – теплопроводность (для стали 13 Вт/(м·К)). Теплоемкость стали 485 Дж/(кг·К). Плотность стали 8000 кг/м^3. Температура плавления $T_{ф.п} = 1723.3\,K$. Удельная теплота плавления L = 84 кДж/кг.

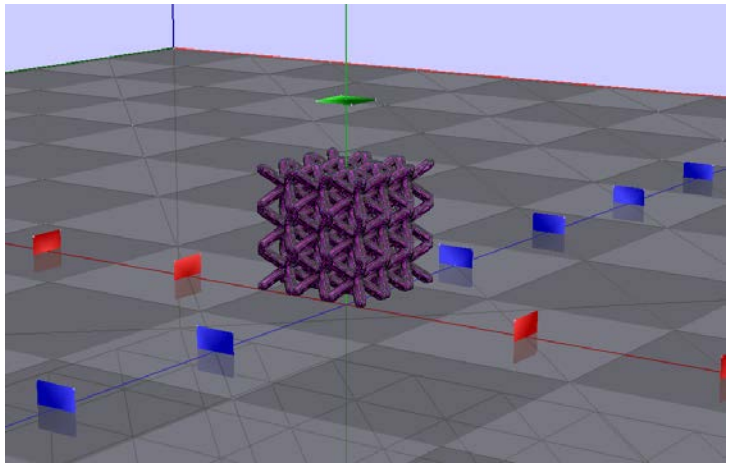


Рис. 3. Решетчатая структура на платформе

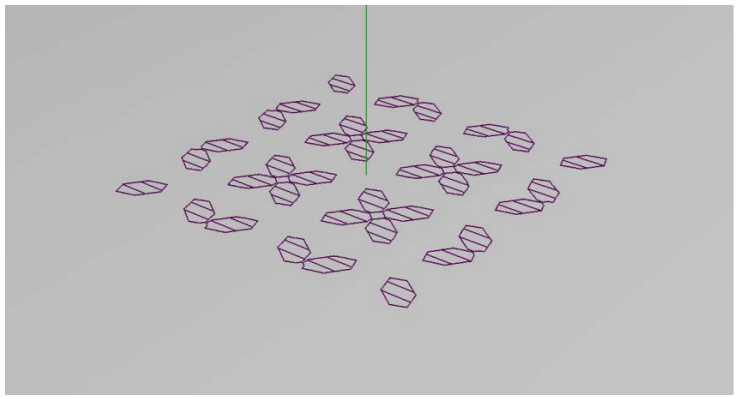


Рис. 4. Слайс детали со штриховкой сечения

Мощность источника энерговыделения 50 Вт. Время остывания (перерыв в работе лазера после прохождения слоя) 1 с. Результат моделирования приведен на рис. 5, 6. В дальнейшем в модуль будет интегрирована связь с другими модулями, уточнены граничные условия в задаче и использованы более точные зависимости для расчетов. Скорость сканирования и другие параметры лазера взяты из [6].

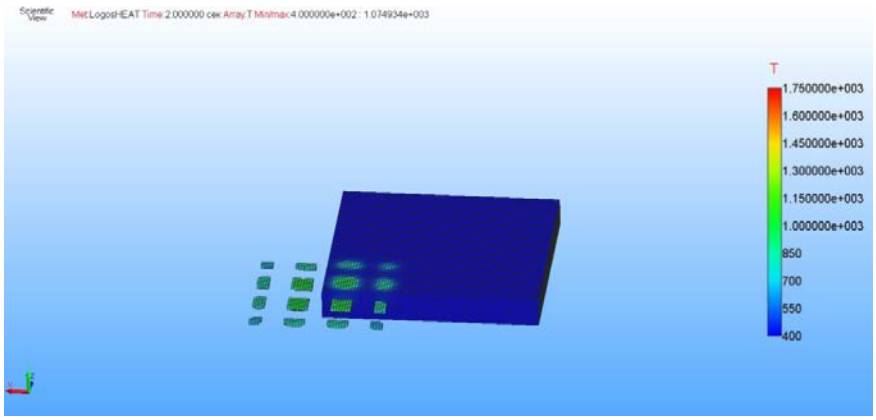


Рис. 5. Поле температуры на момент времени 2 с

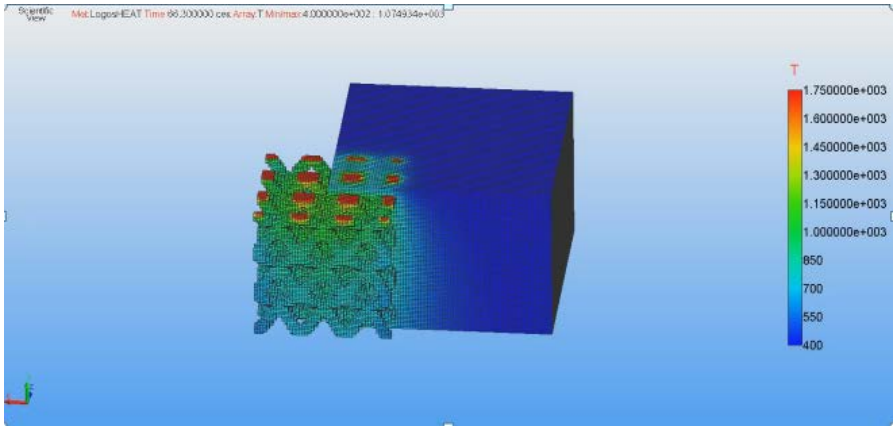


Рис. 6. Поле температуры на момент времени 66 с

Математическая модель и алгоритмы численного решения задачи напряженно-деформированного состояния (НДС)

Для анализа напряженно-деформированного состояния конструкций в пакете программ «Виртуальный 3D принтер» используется вычислительный модуль программного комплекса ЛОГОС – «ЛОГОС-Прочность». Данный модуль предназначен для решения широкого спектра задач механики твердого деформируемого тела. При моделировании процесса селективного лазерного спекания (СЛП) анализ напряженно-деформируемого состояния осуществляется с помощью статического анализа при квазистатических термосиловых воздействиях. Поле температуры является результатом решения «тепловой» задачи. Также по результатам решения «тепловой» задачи определяется массив элементов прочностного расчета (в прочностном расчете участвуют только те элементы, температура в которых ниже температуры плавления).

Для решения статических задач прочности используется распространенная схема МКЭ в форме метода перемещений. В этом случае неизвестными являются компоненты вектора узловых перемещений $\{U\}$ системы, находящейся в равновесном напряженно-деформированном состоянии под действием внешних прикладываемых нагрузок. В трехмерном случае вектор перемещений определяется как $\{U\}^T = \{u, \nu, \omega\}$, где u, v, w – проекции вектора перемещений на координатные оси x, y, z. В этом разделе при обозначении величин векторные величины выделяются фигурными скобками, матричные величины – квадратными.

Разные значения перемещений двух соседних точек вызывают деформации в материале и связанные с ними напряжения. В общем случае, деформации и напряжения в каждой точке конструкции определяются шестью компонентами деформаций $\{\varepsilon\}^T = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{xz}\}$ и шестью компонентами напряжений $\{\sigma\}^T = \{\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}\}$..

При малых деформациях связи между компонентами, определяющими НДС (перемещения, деформации, напряжения), выражаются линейными соотношениями. Линейная связь между перемещениями и деформациями определяется соотношением $\{\varepsilon\} = [B]\{U\}$, где [B] – матричный дифференциальный оператор:

$$[B]=\begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix}. \quad (5)$$

Если тело одновременно находится под воздействием внешних механических нагрузок и температурного поля, то деформации ε, возникающие в нем, выражаются в виде:

$$\{\varepsilon\}=\{\varepsilon^m\}+\{\varepsilon^t\}, \quad (6)$$

где $\{\varepsilon^t\}$ – температурная деформация; $\{\varepsilon^m\}$ – механическая деформация.

Температурные деформации для изотропного материала определяются соотношением:

$$\left\{\varepsilon^t\right\}=\alpha\Delta T\begin{Bmatrix}1\\1\\1\\0\\0\\0\end{Bmatrix}, \quad (7)$$

где α - коэффициент теплового расширения, 1/K; ΔT – изменение температуры, К.

Связь между компонентами деформаций и напряжений в линейно-упругом случае выражается соотношением:

$$\{\sigma\}=[D]\left\{\varepsilon^m\right\}. \quad (8)$$

В случае однородной изотропной линейно-упругой среды матрица материальных констант [D] имеет вид:

$$[D]=\begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \quad (9)$$

где λ и μ – константы Ламе, которые выражаются через модуль упругости первого рода E и коэффициент Пуассона ν: $\lambda=\frac{\nu E}{(1+\nu)(1-2\nu)}$, $\mu=\frac{E}{2(1+\nu)}$.

С учетом (5), зависимость напряжений от деформаций (6), запишется в следующем виде:

$$\{\sigma\}=[D]\left(\{\varepsilon\}-\left\{\varepsilon^t\right\}\right). \quad (10)$$

Для единицы объема упругого тела удельная энергия деформации записывается в виде:

$$W = \frac{1}{2}\{\varepsilon^m\}^T\{\sigma\} = \frac{1}{2}\left(\{\varepsilon\}^T - \{\varepsilon^t\}^T\right)\{\sigma\}. \quad (11)$$

Полная потенциальная энергия деформации, накопленная телом, определяется через интеграл по всему объему тела V:

$$\Pi = \iiint_V W dV = \frac{1}{2}\iiint_V \left(\{\varepsilon\}^T - \{\varepsilon^t\}^T\right)[D]\left(\{\varepsilon\} - \{\varepsilon^t\}\right)dV. \quad (12)$$

Работа внешних сил, действующих на тело, объема V, ограниченного поверхностью S, может быть записана в виде:

$$A = \iiint_V \{u\}^T \{F\}\, dV + \iint_S \{u\}^T \{P\}\, dS, \quad (13)$$

где $\{u\}^T$ – вектор перемещений, м; $\{F\}^T$ – вектор объемных сил для элемента dV, Н; $\{P\}^T$ – вектора поверхностных сил для элемента площади dS, Па.

Полная потенциальная энергия системы определяется как разница между потенциальной энергией деформации и работой, произведенной внешними нагрузками:

$$L = \Pi - A = \frac{1}{2}\iiint_V \left(\{\varepsilon\}^T - \{\varepsilon^t\}^T\right)[D]\left(\{\varepsilon\} - \{\varepsilon^t\}\right)dV - \iiint_V \{u\}^T \{F\}\, dV - \iint_S \{u\}^T \{P\} dS. \quad (14)$$

Согласно вариационному принципу Лагранжа, полная потенциальная энергия тела, находящегося в равновесии под действием внешних объемных и поверхностных сил, достигает минимума на истинном поле перемещений, т. е. на поле перемещений, удовлетворяющем полной системе уравнений теории упругости.

Этот вариационный принцип минимума потенциальной энергии системы, находящейся в равновесии, является определяющим при дальнейшем построении конечно-элементной дискретизации и формировании глобальной системы уравнений.

Процедура конечно-элементной дискретизации сплошной среды, реализованная в программном продукте «Виртуальный 3D принтер», схематически заключается в следующем:

- сплошная среда разделяется линиями/поверхностями на некоторое количество конечных элементов, без самопересечений заполняющих весь объем исходной системы;
- предполагается, что элементы связаны между собой в узловых точках, расположенных на границах элементов;
- значения перемещений в узлах конечных элементов являются искомым решением дискретной модели;
- для каждого типа конечного элемента определяется система базисных функций, однозначно определяющих перемещения любой точки элемента через перемещения узловых точек, функции перемещений однозначно определяют деформации в любой точке элемента через узловые перемещения;
- найденные деформации при заданных материальных свойствах позволяют определить напряжения внутри элемента и на его границах;
- определяется система сосредоточенных узловых сил, уравновешивающих граничные напряжения и распределенные нагрузки.

Основная концепция конечно-элементной аппроксимации состоит в замене непрерывной функции перемещений дискретным набором кусочно-непрерывных функций. Минимизация функционала исходной задачи осуществляется на совокупности этих кусочно-непрерывных функций, каждая из которых определена на своей подобласти – конечном элементе

Алгоритм решения задачи и взаимодействие модулей (обмен информацией)

На данный момент алгоритм решения задачи моделирования СЛС реализован следующим образом. На первом этапе производится решение задачи теплопроводности с моделированием лазера как нестационарного источника $Q(x, y, z, t)$. Вклад значения источника на временном шаге в ячейку рассчитывается способом, описанным в [6]. Окончанием первого этапа считается прохождением лазером всего слоя. Вторым этапом является этап остывания и насыпания следующего слоя порошка. После этого повторяется первый этап. Выдача информации для прочностного расчета выдается на определенные временные шаги. В качестве исходных данных прочностной модуль использует поле узловых температур и массивы индексов элементов, в которых материал находится уже в твердом (кристаллизованном) состоянии. По полученному полю температур производится расчет напряжено-деформированного состояния.

Взаимодействие через общие файлы имеет ряд существенных недостатков связанных с необходимостью постоянного чтения и записи файлов, поэтому в ближайшем будущем начнутся работы по установлению связи между комплексами с использованием технологий МИП (модульно-интеграционной платформы, разработка РФЯЦ-ВНИИЭФ).

Верификация модуля «Прочность макроуровень»

Проведем расчет НДС на поле температур, полученных из теплового расчета. Свойства материала представлены в табл. 1

Таблица 1

Свойства материала

Свойство материала	Значение (единицы)
Плотность	7.8e-6 (кг/мм^3)
Модуль упругости	210000 (МПа)
Коэффициент Пуассона	0.3
Коэффициент теплового расширения	11.1 e-6 1/K

На рис. 7 представлено поле напряжений по Мизесу в момент времени 66 с (момент остывания слоя). Из рисунка видно, что значение напряжений имеют не существенные значения.

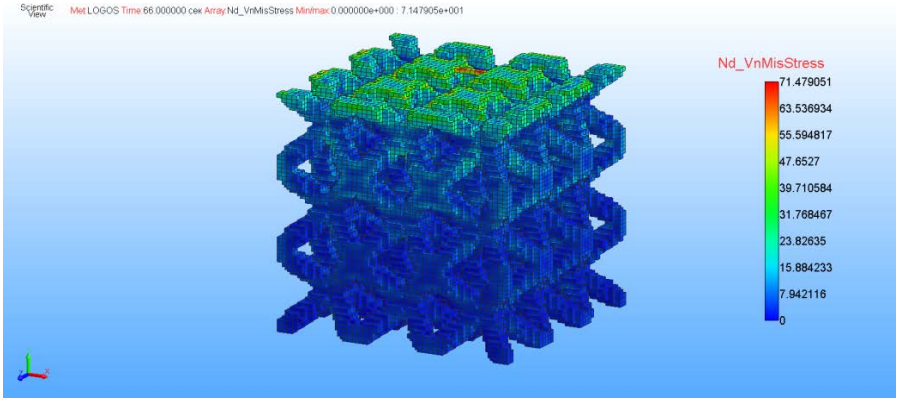


Рис. 7. Поле напряжений по Мизесу в момент времени 66 с

Виртуальные испытания

Далее приведем пример виртуальных испытаний различных вариантов конструкции детали «спинной кейдж». На рис. 8 представлен результат расчета на прочность детали без использования решетчатых структур. На рис. 9 и 10 изображены наборы граней и узлов для задания граничных

условий. Для верхнего набора граней (см. рис. 9) задается давление из расчета общего сжимающего усилия по оси Z равного 1000Н (примерно 7,63МПа). Для нижнего набора узлов нижней части детали задается условие жесткой заделки. В табл. 1 представлены свойства материала детали.

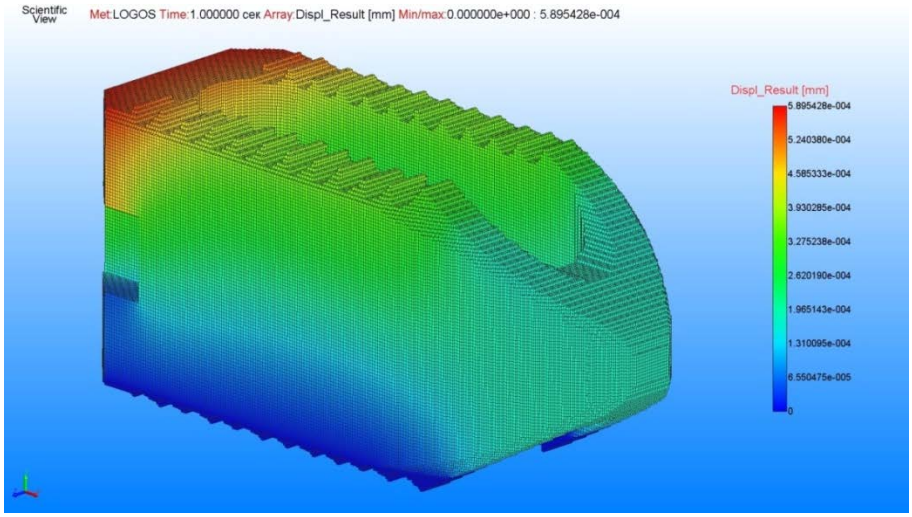


Рис. 8. Поле перемещений для сплошного спинного кейджа

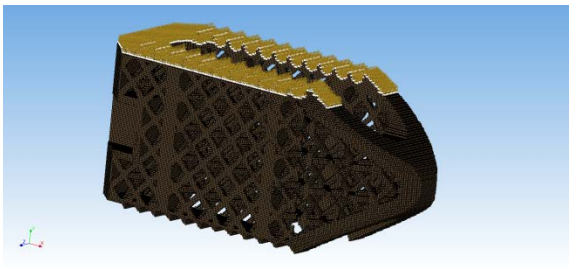


Рис. 9. Набор граней с приложенным давлением



Рис. 10. Набор узлов с жесткой заделкой

Отметим, что указанные нагрузки и свойства материала общие для всей серии расчетов. На рис. 11 и 12 представлены поля перемещений и напряжений по Мизесу для данной детали. Из данных рисков видно, что наличие отверстия в левой части детали способствует возникновению в данной части значительных значений перемещений и деформаций.

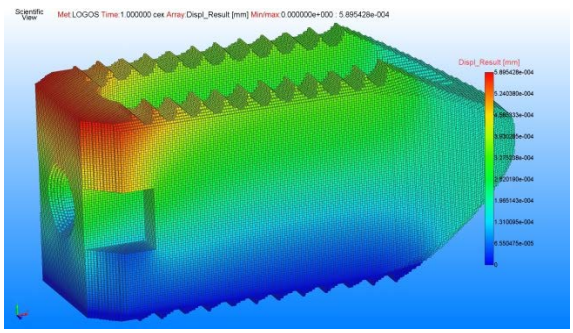


Рис. 11. Поле перемещений для сплошного спинного кейджа

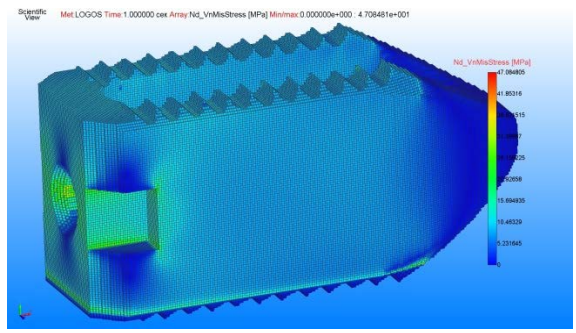


Рис. 12. Поле напряжений для сплошного спинного кейджа

Далее из детали были вырезаны обширные области (где в исходной детали находилась решетка) и произведен расчет для данной геометрии. На рис. 13–15 представлены результаты расчета данной детали. Видны области, которые испытывают максимальные напряжения.

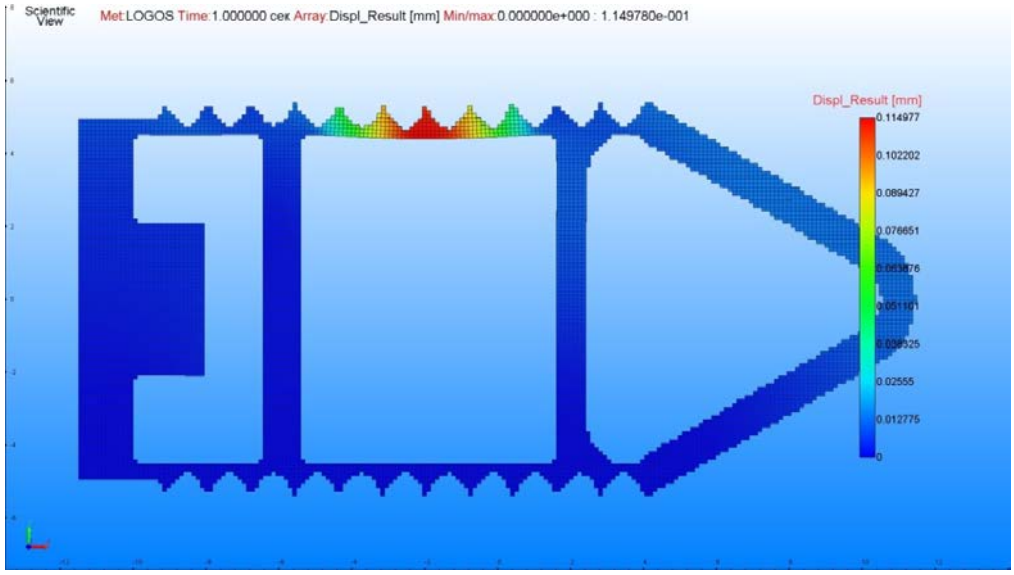


Рис. 13. Поле перемещений для полого спинного кейджа (вид сбоку)

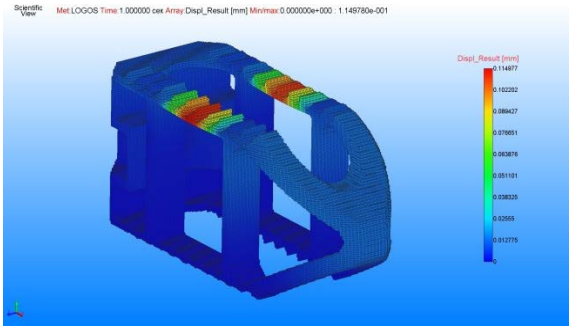


Рис. 14. Поле перемещений для полого спинного кейджа

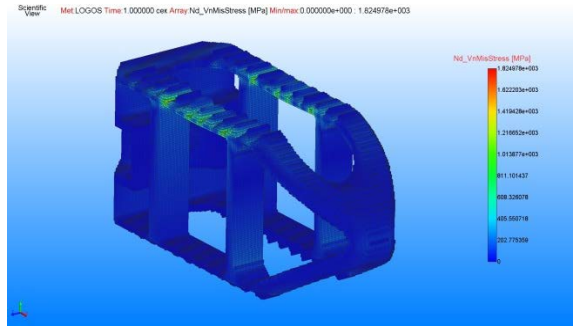


Рис. 15. Напряжения по Мизесу для полого спинного кейджа

Следующие варианты конструкции содержали различные тестируемые решетчатые структуры в полостях спинного кейджа.

Первый тип решеток – это BCC. Значения нагрузок аналогичны нагрузкам расчета сплошной детали. На рис. 16 и 17 представлены результаты расчетов.

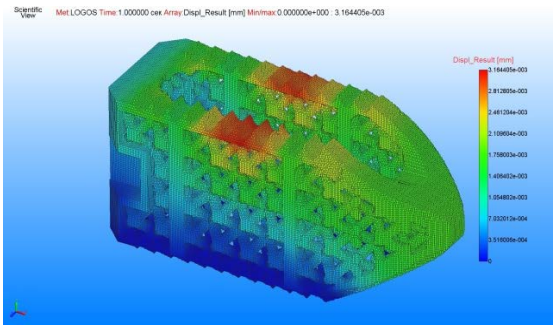


Рис. 16. Поле перемещений для BCC решетки

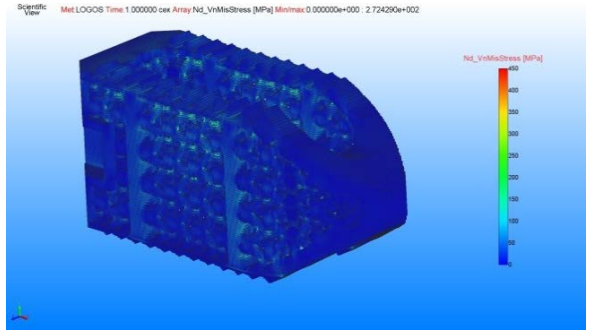


Рис. 17. Поле напряжений по Мизесу для BCC решетки

Следующий тип решеток – это BCC с вертикальными распорками. Данный тип решетки является более жестким по отношению к нагрузкам по оси Y. На рис. 18 и 19 представлены результаты расчетов этой конструкции.

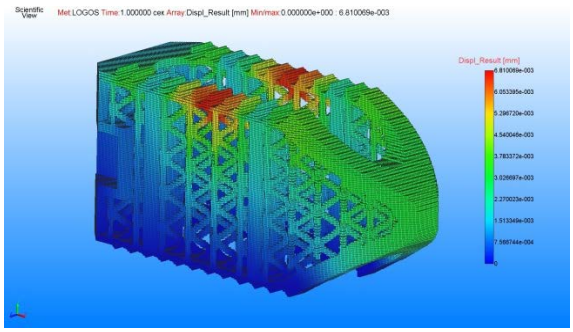


Рис. 18. Поле перемещений для решетки BCC с вертикальными распорками

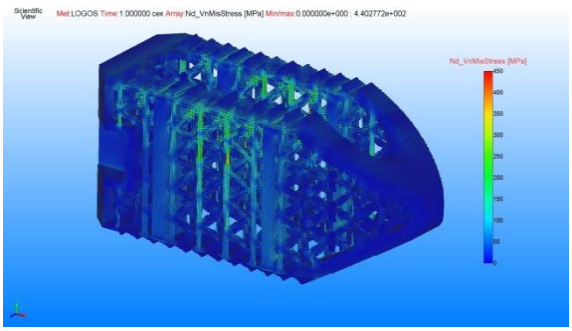


Рис. 19. Поле напряжений по Мизесу для решетки BCC с вертикальными распорками

На рис. 20 и 21 представлены результаты расчетов для решетчатой структуры сотового типа.

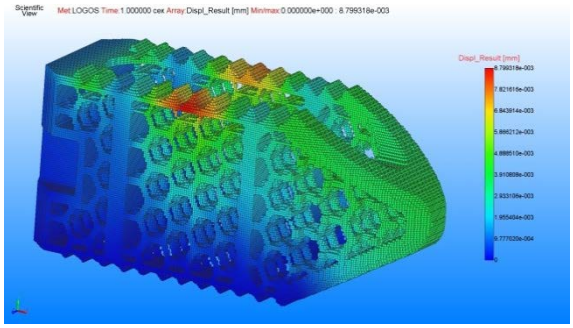


Рис. 20. Поле перемещений для решетчатой структуры сотового вида

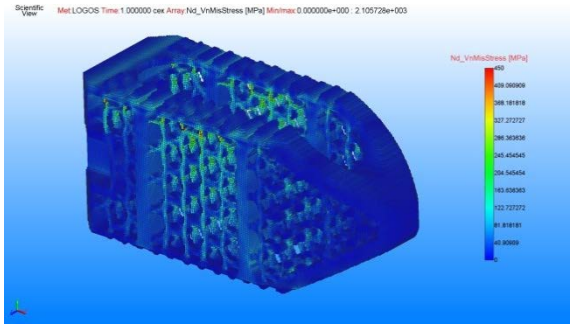


Рис. 21. Поле напряжений по Мизесу для решетчатой структуры сотового вида

На рис. 22–23 представлены результаты расчета для решетчатой структуры, состоящей из распорок, параллельных всем координатным осям.

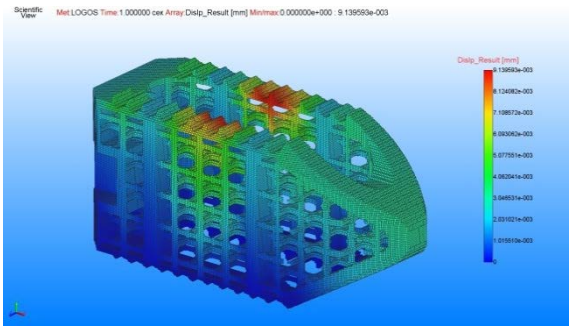


Рис. 22. Поле перемещений для BCC решетки

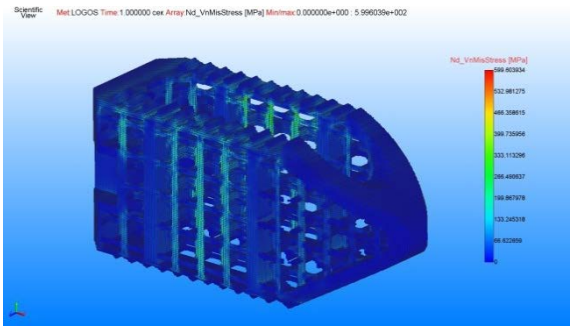


Рис. 23. Поле напряжений по Мизесу для BCC решетки

Из представленных выше результатов можно сделать вывод, что для указанных нагрузок почти все решетчатые структуры могут быть использованы для данной детали. Однако в силу большей жесткости лучше использовать решетчатые структуры, имеющие в своем составе вертикальные распорки.

Заключение

Описанная в данном докладе технология моделирования селективного лазерного спекания позволит проводить оптимизацию процесса производства деталей как с точки зрения затрат на производство детали так и снижения количество брака, а в будущем (с учетом реализации моделей

короблення) и затрат на постобработку изделий (шлифовка, полировка и т. д.). Использования блока «Виртуальных испытаний» и замена сплошных частей деталей на решетчатые структуры приведут к уменьшению массы детали, при этом сокращается количество затраченного на ее производства материала (порошка) при сохранении необходимого запаса прочности.

Литература

1. Шишковский И. В. Лазерный синтез функциональных мезоструктур и объемных изделий. – М.: ФИЗМАТЛИТ, 2009. – 419 с.

2. Вишняков А. Ю., Глазунов В. А., Дерюгин Ю. Н., Чистякова И. Н. Пакет программ ЛОГОС. Модуль расчета сопряженных и связанных задач теплопереноса // Труды МФТИ. 2014. Т. 6, № 4. С. 158–167.

3. Александрова О. Л., Барабанов Р. А., Дьянов Д. Ю., Косарим С. С., Наумов А. О., Спиридонов В. Ф., Филимонкин Е. А., Циберев К. В. Пакет программ ЛОГОС. Конечно-элементная методика расчета задач статической прочности конструкций с учетом эффектов физической и геометрической нелинейности // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. 2014. Т. 3. С. 3–17.

4. Самарский А. А., Вабищев П. Н. Вычислительная теплопередача. Москва 2003, 784 с.

5. Fersinger J. H., Peric M. Computational methods for fluid dynamics.

6. Zhang Y., Guillemont G., Bernacki M., Bellet M. Macroscopic thermal finite element modeling of additive metal manufacturing by selective laser melting process // Comput. Methods Appl. Mech. Engrg. 331. 2018. P. 514–535.

SIMULATION AT MACROLEVEL OF HEAT TRANSFER AND STRENGTH IN SELECTIVE LASER MELTING USING SOFTWARE PACKAGE “VIRTUAL 3D PRINTER”

V. Yu. Kuznetsov, P. A. Mashenkin, Yu. N. Deryugin, R. A. Barabanov, E. A. Filimonkin, V. A. Glazunov, A. Yu. Vishnyakov, O. S. Larkina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The report presents results of the development of the first version of compute modules for the simulation at macrolevel of heat transfer and strength using software package “Virtual 3D printer”. At macrolevel, a set of powder particles is considered to be a continuum, which is characterized by its macro parameters (for example, porosity). The developed modules are based on compute modules of the LOGOS software package (developed at RFNC-VNIIEF) and adapted under the “Virtual 3D printer “project aimed at the development of software for 3D printers. Models and algorithms are presented for the numerical simulation of the selective laser sintering (SLS) process and estimates of the stress-and-strain state of a component part both in the melting process and process of loading the component produced using the SLM procedure are given.

Key words: selective laser sintering, 3D printer, estimation of the stress-and-strain state of a component part.

УДК 519.6
DOI 10.53403/9785951505071_2022_273

ПОСТРОЕНИЕ СМЕЩЕННОЙ СЕТКИ ДЛЯ ВЫДЕЛЕНИЯ ПОГРАНИЧНОГО СЛОЯ В ПАКЕТЕ ПРОГРАММ «ЛОГОС»

В. В. Лазарев, О. Н. Борисенко

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе предложен алгоритм построения смещенной сетки по заданной поверхностной треугольной сетке. Подробно рассмотрены особенности геометрических моделей, поверхностной сетки и пользовательских данных, которые усложняют решение данной задачи. Представлены результаты построенных сеток.

Ключевые слова: пограничный слой, смещенная сетка, пакет программ ЛОГОС.

Введение

Компьютерное моделирование обтекания твердого тела жидкостью или газом имеет практическое значение при проектировании сложных изделий. Для инженера представляет интерес скорость потока, давление и температура около обтекаемого тела. При неудовлетворительном результате, геометрическая модель изменяется, и расчет проводится повторно. Для поиска оптимальной формы конструкции выполняется серия из ста и более расчетов. Для выполнения таких работ нужны инструменты для автоматической подготовки дискретной модели к расчету и эффективный параллельный расчетный код.

Большая часть времени компьютерного моделирования тратится на подготовку модели к расчету. Автоматическое построение на ней качественной расчетной сетки за приемлемое время остается нерешенной задачей [1]. Построение сетки может занимать до 50–90 % времени [2, 3] от всего цикла моделирования. Много времени тратится на удаление излишне мелких деталей и исправление ошибок в представлении геометрии.

В ИТМФ разрабатывается пакет программ инженерного анализа ЛОГОС [4], включающий в свой состав инструменты для решения задач аэрогидродинамики (CFD) – это построение расчетной сетки [5] и решение системы уравнений Навье – Стокса для вязкого турбулентного течения [6]. Одна из основных задач заключается в уменьшении общего времени моделирования за счет сокращения времени подготовки данных к расчету. Это создает дополнительные трудности при построении расчетной сетки.

Чаще всего в CFD расчетах применяется гибридная сетка, состоящая из трех типов ячеек. Вблизи тела выделяется пограничный слой – анизотропная сетка из призматических ячеек. Оставшаяся часть модели заполняется адаптивной шестигранной сеткой, которая измельчается в области, близкой к объекту моделирования. Пересекающиеся с пограничным слоем шестигранные ячейки усекаются, превращаясь при этом в многогранные ячейки. Подробное описание такой сетки и алгоритма ее построения опубликовано в работе [5].

Пограничный слой – это тонкий слой вокруг обтекаемого тела. Численное решение в нем характеризуется сильным градиентом скорости потока по нормали к поверхности тела и незначительным градиентом вдоль тела. Поэтому для расчета в этой подобласти наиболее эффективна анизотропная сетка.

Существуют различные способы выделения пограничного слоя. Один из них – это прямое построение анизотропной тетраэдрной сетки [7]. Сетка строится триангуляцией Делоне. Управление размерами ячеек и их вытягивание в нужном направлении достигается за счет искривления пространства полем метрических тензоров. Недостаток сетки из тетраэдров в том, что, по сравнению с призмами, при одинаковой плотности узлов для ее хранения используется больше памяти. Поэтому предпочтительнее использовать призмы, которые к тому же вытягиваются без вырождения угла. Существуют три основных подхода к построению пограничного слоя из призматических ячеек. Первый подход заключается в последовательном наращивании призм в квазинормальном к телу направлении [8, 9]. Второй подход выполняет смещение исходной сетки тела сразу на толщину слоя [10] с последующим разбиением на слои выделенного объема. Третий подход работает по уже существующей изотропной объемной сетке и вклинивает призматические ячейки между ней и телом [11].

Представленная в докладе работа является продолжением ранее выполненной работы [5]. В ней описывается алгоритм, основанный на втором подходе построения пограничного слоя. В отличие от третьего подхода, для него не нужна объемная сетка, а в отличие от первого – формируется гладкая поверхность границы слоя, что необходимо для дальнейшего корректного построения объемной сетки. Сложность построения смещенной сетки связана с особенностями геометрических моделей, поверхностной сетки и пользовательских данных.

Особенности, влияющие на построение смещенной сетки

Ниже приведены особенности входных данных, которые усложняют построение смещенной сетки – снижают устойчивость процесса построения, ухудшают качество результирующей сетки. Эти особенности можно условно разделить на несколько групп:

1. Малые детали в конструкции и их плотное размещение.
2. Узкие зазоры между деталями конструкции.
3. Острые стыки поверхностей.
4. Узкие поверхности, приводящие к вырожденным ячейкам.
5. Вогнутые области с высокой кривизной.
6. Неточный стык двух поверхностей.
7. Сгущение сетки на ровной поверхности.
8. «Дребезг» треугольников.
9. Большие величины смещения относительно размеров треугольников.

Рассмотрим подробнее пункты 1, 6, 7, 8, 9.

Малые детали в конструкции и их плотное размещение. Сложные геометрические модели могут содержать большое количество мелких деталей, таких, как элементы механизации крыла, подвес для вооружения, антенны, рассекатели. Размеры этих деталей значительно меньше габаритов области обтекания. Качественное описание этих деталей в сеточной модели приводит к необходимости уменьшения размера ячеек, что усложняет процесс и увеличивает время построения сетки.

Неточный стык двух поверхностей. Геометрическая модель в граничном сплайновом представлении не всегда корректно описывается. Несмотря на то, что оболочка является замкнутой в терминах топологических объектов, геометрически имеет существенную погрешность в местах пересечения поверхностей. Исходя из практических результатов, а также на основании [2], можно сказать, что погрешность стыковки двух поверхностей (CT, curve tolerance) на один порядок больше, чем допуск на объединение узлов (SPT, same point tolerance). При моделировании вязких течений, толщина первой ячейки пограничного слоя намного меньше величины CT [2]. Чтобы получить замкнутую сетку, нужно сшить сетки на поверхностях. Если $CT > SPT$, то в сетку добавляются дополнительные вырожденные треугольные ячейки, чтобы заполнить зазор. В результате могут возникать перехлесты поверхностей, либо скачки высот между поверхностями в местах их стыковки.

Сгущение сетки на ровной поверхности. Величину CT можно уменьшить с помощью локальных модификаций сплайновых поверхностей, но это приводит к осцилляциям кривизны на границе этих поверхностей. Такие осцилляции влияют на построение поверхностной сетки, образуя сгущения

сетки на «ровной» поверхности. На рис. 1 показан пример поверхностной сетки со сгущением в месте контакта крыла с закрылками.

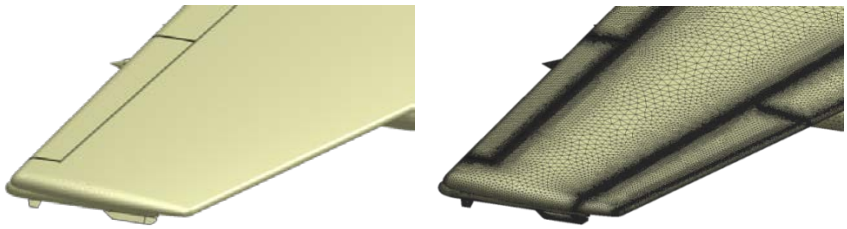


Рис. 1. Исходная геометрическая модель и поверхностная сетка со сгущениями

«Дребезг» треугольников. Дребезг треугольников – это такое расположение треугольников, при котором появляются сеточные ребра с большой величиной двугранного угла и не упорядоченные в единую кривую. Он возникает на стыке двух поверхностей с малым, относительно размера ячеек, радиусом скругления. На рис. 2 показан фрагмент геометрии, на котором выделены сеточные ребра с двугранными углами больше 30°.

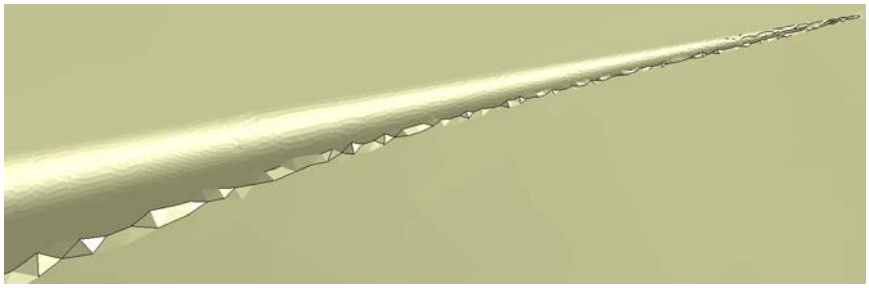


Рис. 2. Дребезг ячеек сетки в местах гладкой стыковки поверхностей

Дребезг обычно появляется на ребрах, которые расположены поперек асимптотической кривой и недостаточно точно аппроксимируют кривизну.

Это связано с тем, что поверхностная сетка строится фронтом от границы, который может не совпадать с асимптотическими линиями искривленной поверхности.

Большие величины смещения относительно размеров граней. Отношение толщины слоя к наименьшей длине ребра правильной грани (вырожденные грани не рассматриваются) может достигать 100 раз. Незначительная несогласованность направлений смещения смежных узлов приводит к перехлестам сетки, как показано на рис. 3.

Такая проблема возникает из-за существенного различия размеров сеточных элементов, описывающих мелкие детали конструкции и область моделирования в целом. Пользователь задает значение смещения, как правило, не учитывая возможность присутствия в сетке более детализированных конструктивных элементов, что и приводит к описанной выше проблеме. Задача алгоритмов построения сетки – взять на себя корректное разрешение таких ситуаций.

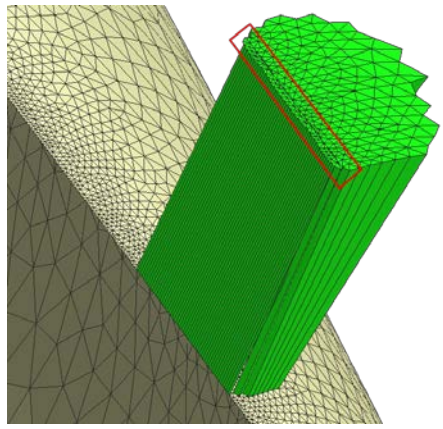


Рис. 3. Пример перехлеста граней смещенной сетки

Алгоритм построения смещенной сетки

Исходная геометрическая модель аппроксимируется замкнутой поверхностной треугольной сеткой без пересечений. Назовем ее исходной сеткой. Каждая грань исходной сетки принадлежит определенному набору граней, образующих замкнутую область. Такой набор граней называется патчем. Для каждого патча определен признак и величина смещения.

Для корректного описания топологических особенностей модели на исходной сетке должны быть выделены характерные кривые, которые используются для описания острых кромок и/или деталей поверхности и представляют собой набор ребер и узлов, которые обязательно должны присутствовать в сетке. Дополнительно в качестве характерных кривых выделяются границы между патчами и специально выделенные пользователем кривые. В алгоритме построения смещенной сетки вводится понятие характерной вершины. Характерная вершина – это узел характерной кривой, отвечающий одному из условий:

1) наличие излома кривой;
2) стыковка двух и более характерных кривых;
3) начальная или конечная точкой кривой.

Множество узлов исходной сетки, которые не относятся к характерным кривым и вершинам, называются гладкой областью.

Необходимо построить смещенную сетку, удовлетворяющую условиям:

1) близкое к заданным значениям расстояние между гранями исходной и смещенной сеток;
2) отображение гладких областей в гладкие области;
3) отображение характерных кривых в характерные кривые;
4) отсутствие пересечений между исходной и смещенной сетками.

Алгоритм построения смещенной сетки состоит из следующих этапов:

1) вычисление нормалей в узлах исходной сетки;
2) вычисление начальной смещенной сетки;
3) оптимизация направлений смещений;
4) удаление пересечений в сетках;
5) сглаживание изломов в смещенной сетке.

Для упрощения изложения предполагается, что смещение пограничного слоя проводится в направлении нормалей к граням исходной сетки.

Вычисление нормалей в узлах характерных кривых и вершинах подробно описана в работе [5]. Нормаль в узле гладкой области и характерной кривой вычисляется как взвешенные по углу и величине смещения нормали инцидентных граней:

$$\overline{n}_i = \frac{\sum_j^{N_i} \overline{n}_{ij}\, \alpha_{ij} d_{ij}}{\left|\sum_j^{N_i} \alpha_{ij}\, d_{ij}\right|}$$

где N_i – количество инцидентных узлу i граней, α_{ij} – угол в узле i грани j, $\overline{n}_{ij}$ – нормаль в инцидентной грани j, d_{ij} – смещение в этой грани. Заметим, что нормаль имеет единичную длину. В характерной вершине нормаль определяется через шар, вписанный в многогранный угол.

Смещение узла задается двумя параметрами – направлением и величиной смещения. Предполагается, что направление смещения всегда нормализовано. Координаты смещенного узла i могут быть в любой момент вычислены как:

$$\overline{p}_i' = \overline{p}_i + \overline{d}_i * v_i$$

где $\overline{p}_i$ – координаты узла исходной сетки, $\overline{d}_i$ – направление, а v_i – величина смещения. Величина $\overline{p}_i' - \overline{p}_i = \overline{d}_i * v_i$ называется вектором смещения.

Вектор смещения определяется только в тех узлах исходной сетки, которые необходимо сместить, т. е. в узлах смещаемых патчей. Начальные направления смещения совпадают с нормалями

исходной сетки. Направления смещений в узлах на границах смещаемого и несмещаемого патчей проецируются на поверхность несмещаемого патча, как приведено в работе [5].

Величина смещения в узле вычисляется как:

$$v_i = \min\left\{\frac{v_{ij}}{\max\left(\overline{d}_i\overline{n}_{ij}, \cos\frac{\pi}{4}\right)}\right\}.$$

Здесь v_{ij} – заданное пользователем смещение на инцидентной грани, $\overline{n}_{ij}$ – нормаль грани, $\overline{d}_i$ – направление смещения в узле.

Величина в знаменателе выражения представляет собой скалярное произведение нормали к грани и вектора смещения в узле. Она ограничена значением косинуса 45°, чтобы не допустить излишнего смещения узлов в местах острого стыка граней. На данный момент в алгоритме существует два типа узлов, для которых невозможно построить корректное смещение:

- узлы на границе смещаемого и несмещаемого патчей, если угол между нормалями граней этих патчей меньше 20°;
- узлы, в которых углы между направлением смещения и нормалью инцидентной грани больше 85°.

Для этих типов узлов величина смещения обнуляется, и смещенная сетка для них не строится. Второй тип узлов характерен острому стыку двух поверхностей, местам наложения одной поверхности на другую из-за неточного стыка, а также характерным вершинам, в которых не существует оптимального направления смещения.

В результате выполнения первых двух этапов построения смещенной сетки, приведенных выше, формируется сетка, в общем случае не удовлетворяющая условиям 2, 3 и 4. Третий этап – оптимизация направлений смещений – предназначен для решения следующих задач:

1) уменьшить отклонение от плоскости боковых граней призм, основаниями которых являются грани исходной и смещаемой сеток;

2) исключить самопересечения смещенной сетки;

3) уменьшить площадь перехлеста в смещенной сетке.

Оптимизация направлений смещений выполняется локально в местах перехлестов смещенной сетки и их окрестностей. Узел считается перехлестнувшимся, если существует исходящее из него ребро такое, что скалярное произведение векторов, сформированных на ребрах исходной и смещенной сетки, отрицательное. Вокруг каждого перехлестнутого узла формируется область точек, расстояние до которых не превышает величины смещения и сглаживается способом, изложенным в работе [5] для этого узлы сортируются, начиная от характерных вершин по направлению характерных кривых, а затем вглубь гладкой области.

На четвертом этапе в смещенной сетке удаляются пересечения локальным уменьшением величины смещения. Можно выделить три типа пересечения:

1) самопересечения граней смещенной сетки;

2) пересечения граней исходной и смещенной сеток;

3) перехлест в смещенной сетке.

Как правило, перехлест смещенной сетки обычно возникает на характерных кривых – в местах излома исходной сетки, как показано на рис. 4.

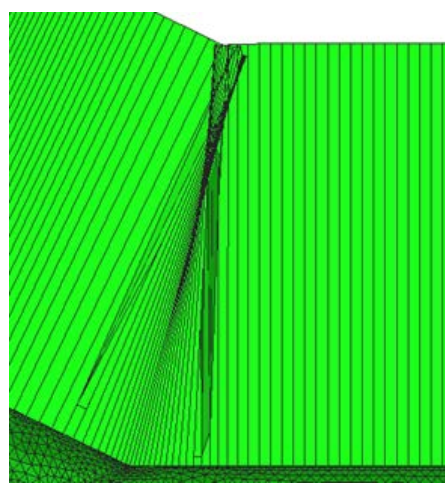


Рис. 4. Перехлест смещенной сетки в месте излома исходной сетки

Для устранения пересечений площадь перехлеста минимизируется при оптимизации направлений смещений. Если этого недостаточно, то уменьшается расстояние между смещенной и исходной сеткой. Одновременно с устранением перехлестов уменьшается искривление боковых граней призм. Величина смещения в узле поджимается так, чтобы все инцидентные ребра смежных узлов удовлетворяли условиям:

1) угол отклонения от исходных ребер не должно превышать 40°;

2) отношение их длин не должно быть больше, чем 100 раз.

Оставшиеся пересечения определяются парой граней f_i и f_j. Смещения в узлах граней уменьшаются на 30 %, кроме узлов, в которых смещение в два раза меньше наибольшего смещения в узлах второй грани или отношение разницы величин смещений в гранях к расстоянию до соседнего узла в исходной сетке превышает единицу.

Первое условие позволяет равномерно сжиматься двум противоположно растущим друг к другу областям смещенной сетки. Это условие позволяет исключить избыточное поджатие тонкого пограничного слоя, если одна из областей должна быть поджата по другим причинам, а вторая имела свою первоначальную толщину. Второе условие – это приближенный способ вычисления тангенса угла между исходным и смещенным ребром для общего случая, когда 4 точки (два смежных узла и их смещения) могут не лежать на одной плоскости. Оно исключает из поджатия узлы грани, которые не являются причиной пересечения.

Если смещение в узле меньше некоторого порогового значения, то оно обнуляется, что позволяет исключить вырождение пограничного слоя в узких областях.

Описанные выше действия выполняются до тех пор, пока не будут исключены все случаи пересечений.

В местах удаления пересечений и обнуления смещений образуются изломы, которые нарушают условия 2 и 3 построения смещенной сетки. Перед сглаживанием изломов необходимо ограничить отклонение нормалей граней смещенной сетки от граней исходной сетки значением 45°. Эти области отмечены на рис. 5 красными окружностями.

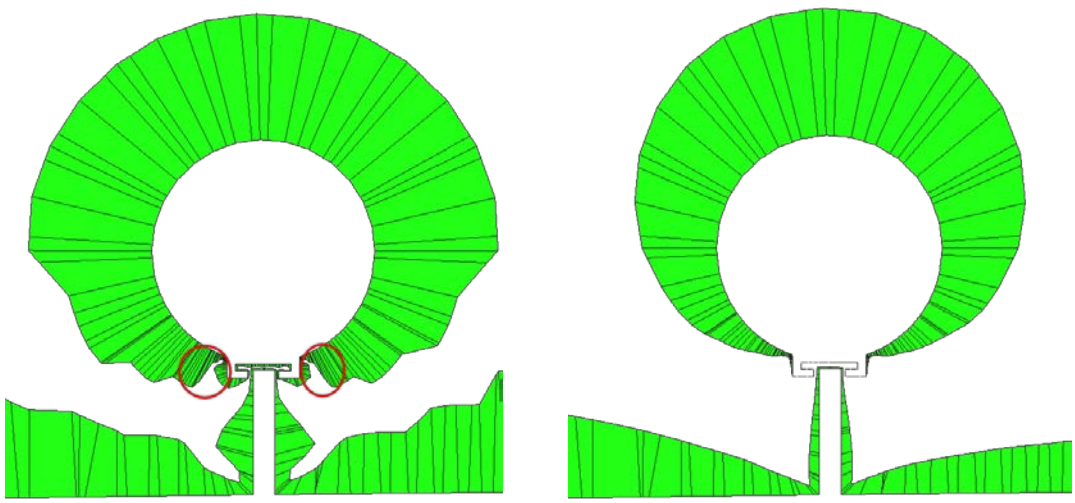


Рис. 5. Смещенная сетка до и после сглаживания изломов

Сглаживание изломов стартует с начального фронта. Начальный фронт составляют смещенные узлы, в которых нарушен критерий излома. Сглаживание происходит итерационно широким фронтом от начальной области вглубь до тех пор, пока невязка больше заданной величины. Невязка вычисляется как отношение величины сдвига на среднюю длину ребер в узле смещенной сетки. В каждом сглаживаемом узле вычисляется величина сдвига. Направление сдвига определяется направлением смещения. На каждой итерации смещение уменьшается на 30 % от части сдвига в узле.

Сдвиг в узле вычисляется по-разному в зависимости от области, которой принадлежит узел – гладкой, характерной кривой или вершине.

Для узла характерной кривой сдвиг вычисляется следующим образом. Пусть $\overline{p}_0$ – координаты смещенного узла, в котором есть излом. Соседями узла по кривой являются узлы $\overline{p}_1$ и $\overline{p}_2$. Пусть $\overline{p}$ – это проекция узла $\overline{p_0}$ на отрезок $(\overline{p_1}, \overline{p_2})$. Тогда величину сдвига можно вычислить по формуле $s = (\overline{p} - \overline{p}_0) \cdot \overline{d}$, как схематично показано на рис. 6.

Если узел принадлежит гладкой области, то для удаления излома используется решение задачи потока средней кривизны [12]:

$$\frac{\partial \bar{X}}{\partial t} = -\kappa_{\text{H}} \bar{n} \quad (1)$$

где κ_{H} – средняя кривизна поверхности, $\bar{n}$ – нормаль поверхности. Поток деформирует поверхность в нормальном направлении со скоростью равной ее средней кривизне. Для быстрого сглаживания в работе [12] предложена аппроксимация (1) для явной схемы и единичного шага, которая имеет следующий вид:

$$(\kappa_H \bar{n})_{normalized} = \frac{\sum_j^{N_i} \left(ctg(\alpha_j)+ctg(\beta_j)\right)\left(\bar{p}_i - \bar{p}_j\right)}{\sum_j^{N_i} \left(ctg(\alpha_j)+ctg(\beta_j)\right)} \quad (2)$$

где $\bar{p}_j$ – координаты смежного узла, α_j и β_j – углы противоположные инцидентному ребру j. Для смещенной сетки деформация (2) проецируется на направление смещения, и сдвиг в узле вычисляется по формуле $s = \bar{d} \cdot (\kappa_H \bar{n})_{normalized}$.

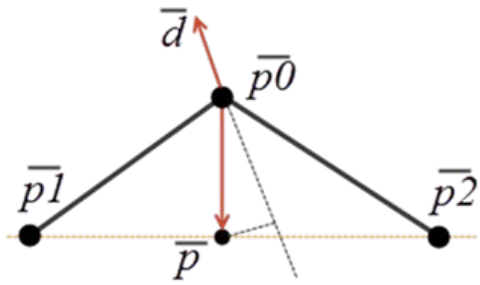


Рис. 6. Сдвиг узла характерной кривой

Если узел является характерной вершиной, то величина нового смещения в нем, вычисляется как средняя проекция векторов смещения смежных узлов на направление смещения данной вершины. Сдвиг будет вычисляться, как разность нового и старого смещений.

Критерий излома определяется отдельно для характерных вершин, характерных кривых и гладких областей.

Узел характерной вершины по определению является изломом. Если узел смещенной сетки не повторяет выпуклость (вогнутость) узла исходной сетки, то излом фиксируется для последующего сглаживания.

Поскольку границы характерных кривых принадлежат группе характерных вершин, то любой узел характерной кривой содержит два инцидентных с ним ребра, принадлежащих этой кривой. Угол между этими ребрами на исходной сетке должен быть согласован с углом ребер смещенной сетки с некоторой погрешностью $t_\angle$. В противном случае в узле характерной кривой фиксируется излом.

Не всегда гладкая область является действительно гладкой. Область может иметь нулевую кривизну в одном направлении и ненулевую в ортогональном – как, например, для цилиндрических поверхностей. В гладкую область может попасть «дребезг» треугольников, для которых двугранный угол на ребрах может достигать 30°– 40°. Это связано с тем, что при выделении характерных кривых, как правило, используется предельный угол, равный 44°. Попытки сгладить такие изломы на смещенной сетке приводят к уменьшению толщины пограничного слоя до нуля в выпуклых областях и увеличению в вогнутых. Поэтому излом на смещенной сетке определяется с учетом излома на исходной. Для каждого инцидентного узла ребра вычисляется двугранный угол со знаком – отрицательный для вогнутых областей и положительный для выпуклых. Двугранный угол на смещенной сетке должен повторять этот же угол на исходной сетке с некоторой погрешностью $t_\angle$. В результате анализа тестирования на базе моделей с различными значениями параметра $t_\angle$ была выбрана величина $t_\angle = 7°$.

Примеры построения смещенных сеток

В этом разделе представлены результаты построения смещенной сетки. Для наглядности смещенная сетка показана сечением треугольных призм, помещенных на исходную сетку. Геометрическая модель самолета Су – 47 была взята с ресурса [13].

На рис. 7 показано сечение сетки на крыле самолета. Исходная сетка показана бежевым цветом, смещенная – набором призматических ячеек зеленого цвета. Углубления показывают границу раздела крыла и предкрылка. Пограничный слой сжимается в этих углублениях, но затем относительно быстро набирает высоту.

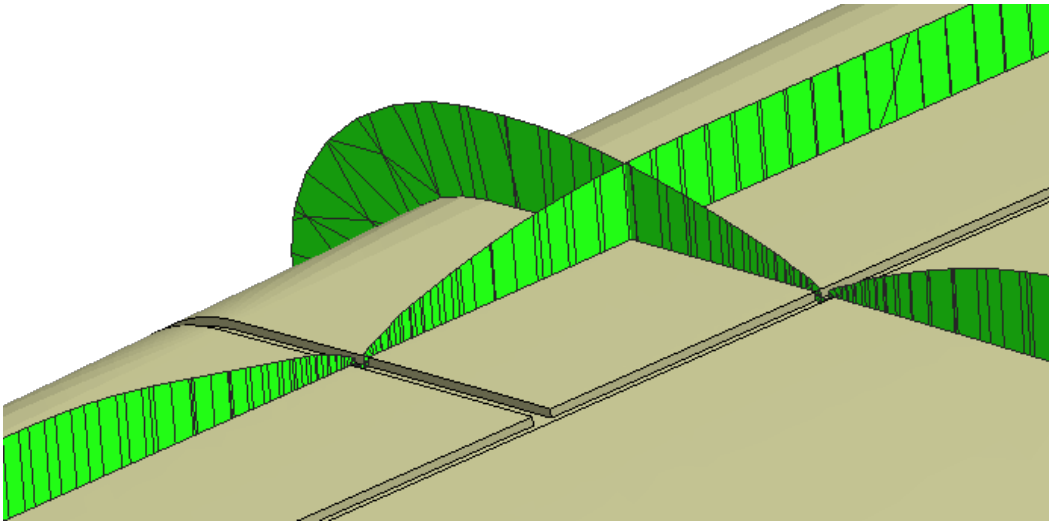


Рис. 7. Исходная и смещенная сетки на крыле самолета

На рис. 8 показаны смещенная сетка (справа) и ее сечение (слева) на головной части блока ракеты. Нос ракеты и края цилиндрических отверстий на конусе создают острый угол, в котором поджимается смещенная сетка. На рис. 8, где показана смещенная сетка, видно, что характерные кривые отверстий сохранились четко.

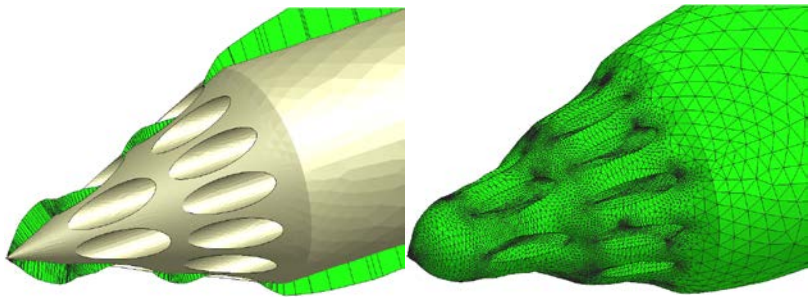


Рис. 8. Сечение и смещенная сетка на блоке ракет

На рис. 9 показаны смещенная сетка и ее сечение на приемнике воздушного давления. Данный случай показывает пример подробной детализации модели. Толщина штанги и размеры дополнительных элементов на ней меньше, чем заданная пользователем толщина пограничного слоя.

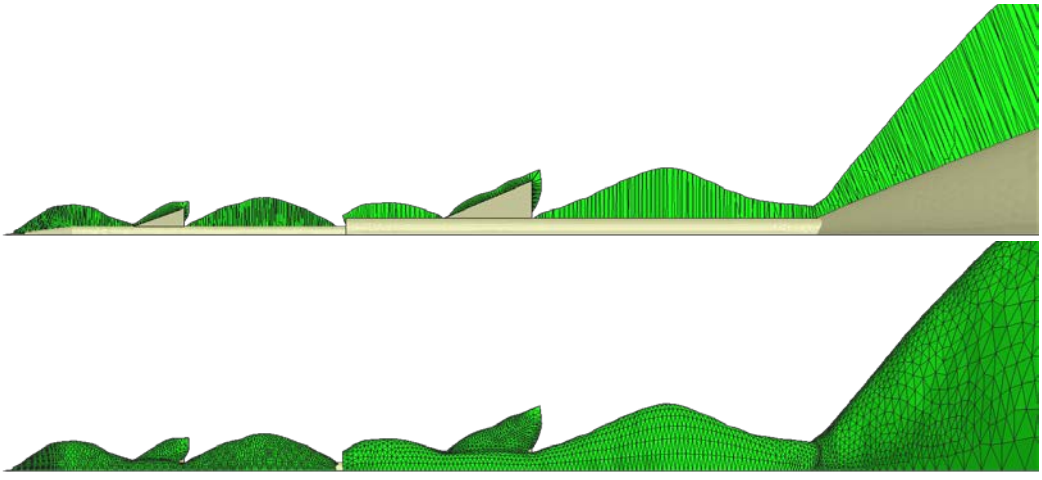


Рис. 9. Сечение и смещенная сетка на приемнике воздушного давления

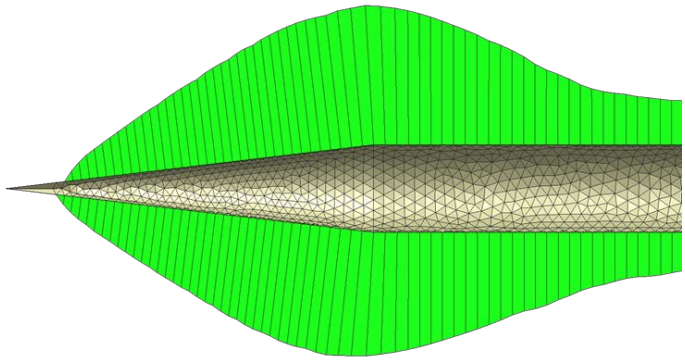


Рис. 10. Смещенная сетка со стороны плоскости симметрии
на приемнике воздушного давления

На рис. 10 показана смещенная сетка со стороны плоскости симметрии. Виден острый кончик, на котором произошло «схлопывание» пограничного слоя.

На рис. 11 показан срез смещенной сетки на стыке двух поверхностей. Погрешность стыковки была выше величины допуска при объединении узлов, поэтому отверстие «зашилось» дополнительными треугольными ячейками. Характерной особенностью этого случая является ступенька, высота которой уменьшается до нуля и разворачивается в обратную сторону. Из сечения видно, что смещение поджимается на ней быстро и гладко, после чего быстро выходит на нужную высоту.

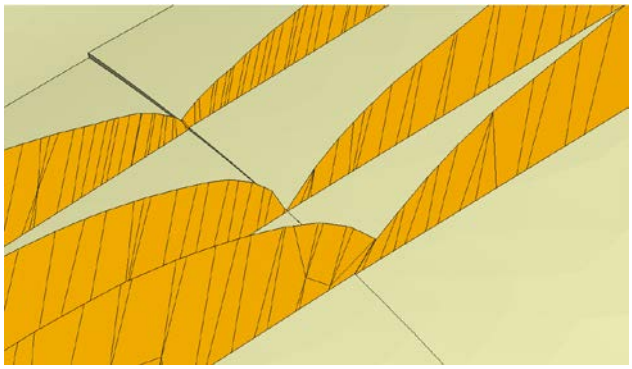


Рис. 11. Серия сечений смещенной сетки на месте
неточного контакта двух поверхностей

На рис. 12 показано сечение смещенной сетки в области «дребезга» ячеек. Видно, что смещенная сетка построена корректно.

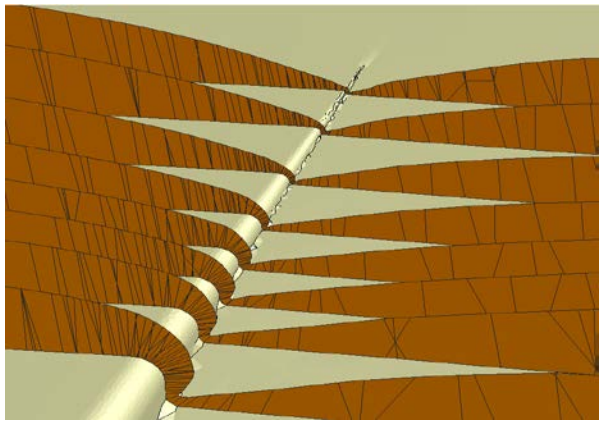


Рис. 12. Серия сечений смещенной сетки в области
дребезга треугольников

Заключение

В докладе представлен алгоритм вычисления смещенной треугольной сетки, являющийся развитием алгоритма, представленного в работе [5].

Наибольшее внимание было уделено сохранению необходимой толщины пограничного слоя при устранении пересечений различных видов, обработке острого стыка граней, «дребезга» граней и поддержке гладкости смещенной поверхности. Выделены особенности геометрических моделей, поверхностных сеток и пользовательских входных данных, которые усложняют решение данной задачи, указаны причины их появления. Представлены примеры построенных смещенных сеток.

Литература

1. Slotnick J., Khodadoust A., Alonso J., Darmofal D., Gropp W., Lurie E., Mavriplis D. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. – Hampton, Virginia: NASA Langley Research Center, 2014.

2. Steinbrenner J. P. The Link Between Mesh and Geometry in Pointwise // ITI CAE Geometry Workshop: Cambridge, UK, 14–15 September, 2017.

3. Sevilla R., Hassan O. Generation of meshes without de-featuring // CAE Geometry Workshop: Cambridge, UK, 14–15 September, 2017.

4. ЛОГОС: Пакет программ инженерного анализа и суперкомпьютерного моделирования [Электронный ресурс] – Режим доступа: www.logos.vniief.ru

5. Смолкина Д. Н., Борисенко О. Н., Черенкова М. В., Гиниятуллина А. Г., Кузьменко М. В., Чухманов Н. В., Потехина Е. В., Попова Н. В., Турусов М. Р. Автоматический генератор неструктурированных многогранных сеток в препроцессоре пакета программ "ЛОГОС" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. – 2018. – Вып. 2. – С. 25–39.

6. Козелков А. С., Жучков Р. Н., Уткина А. А., Володченкова К. Б. Моделирование турбулентных течений на сетках гибридной структуры с использованием схем высокого порядка точности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. – 2014. – Вып.3. – С. 18–31.

7. Borouchaki H., George P., Hecht F., Laug P., Saltel E. Delaunay mesh generation governed by metric specifications. Part I. Algorithms // Finite Elements in Analysis and Design. – 1997. – Vol. 25. – P. 61–83.

8. Kallinderis Y., Ward S. Prismatic grid generation for three-dimensional complex geometries // AIAA Journal. – 1993. – Vol. 31, No. 10. – P. 1850 – 1856.

9. Шмелева А. К., Воропинов А. А. Алгоритмы построения трехмерных неструктурированных сеток на основе поверхностной сетки для методики ТИМ // XVI Международная конференция «Супервычисления и математическое моделирование»: Труды конференции. Саров, 3–7 октября, 2016. С. 336–344.

10. Garimella R. V., Shephard M. S. Boundary layer mesh generation fo viscous flow simulations // International Journal for Numerical Methods in Engineering. – 2000. – Vol. 49. – P. 193–218.

11. Bottasso C. L., Detomi D. A Procedure for Tetrahedral Boundary Layer Mesh Generation // Engineering with Computers. – 2002. – Vol 18, № 1. – P. 66–79.

12. Desbrun M., Meyer M., Schröder P., Barr A. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow // SIGGRAPH 99: Conference Proceedings. August 1999. P. 317–324

13. GrabCAD: Design Community, CAD Library, 3D Printing Software [Electronic resource] – Mode of Access: https://grabcad.com/

OFFSET BOUNDARY-LAYER MESH GENERATION IN THE "LOGOS" SOFTWARE PACKAGE

V. V. Lazarev, O. N. Borisenko

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents an algorithm for constructing an offset mesh from a given triangular surface mesh and considers in detail the features of geometric models, surface meshes and user-defined data, which complicate this task. Results of mesh construction are reported.

Key words: boundary layer, offset mesh, LOGOS software package.

УДК 004.4.242
DOI 10.53403/9785951505071_2022_284

ВЫСОКОПРОИЗВОДИТЕЛЬНАЯ РЕКОНФИГУРИРУЕМАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА НА ОСНОВЕ ПЛИС XILINX ULTRASCALE+

И. И. Левин, Ю. И. Дорончeнко, А. И. Дордопуло, М. Г. Левина

ООО «Научно-исследовательский центр супер-ЭВМ и нейрокомпьютеров», Таганрог

В статье рассматривается перспективная реконфигурируемая компьютерная система на базе современных ПЛИС с погружной подсистемой охлаждения электронных компонентов. Система состоит из автономных вычислительных блоков. Мы представляем архитектуру базового компонента системы, такого как вычислительный модуль. Благодаря коммутационной структуре между вычислительными модулями и блоками, можно решать широкий класс ресурсоемких задач с многократными информационными обменами. Мы представляем новые подходы к питанию ПЛИС, которые потребляют до 180 Вт каждая. Благодаря уникальной конструкции вычислительного блока, можно достичь вычислительной плотности до 128 ПЛИС в конструкции 3U. Мы демонстрируем преимущества подсистемы погружного охлаждения. Кроме того, мы представляем описания технических решений для улучшения подсистемы охлаждения. Благодаря такому усовершенствованию, можно отводить до 20 кВт тепла в целом от одного вычислительного блока. Кроме того, мы рассматриваем оригинальный комплекс средств программирования на языке высокого уровня для многокристальных РВС.

Ключевые слова: реконфигурируемые вычислительные системы, производительность вычислений, ПЛИС, системы погружного охлаждения, вычислительная плотность, средства программирования, язык высокого уровня.

1. Введение

Научно-технический прогресс требует неуклонного повышения производительности суперкомпьютеров для решения вычислительно трудоемких задач. В то же время рост производительности традиционных многопроцессорных систем замедляется, повышение тактовых частот процессоров остановилось. Для реконфигурируемых вычислительных систем (РВС), построенных на основе программируемых логических интегральных схем (ПЛИС), удается поддерживать рост производительности вычислений в 2,5–3 раза с выходом каждого нового поколения ПЛИС (приблизительно раз в два года).

В отличие от вычислительных систем с традиционной архитектурой концепция РВС позволяет создавать вычислительные комплексы, обеспечивающие высокую реальную производительность вычислений для решения широкого круга задач различных проблемных областей за счет нивелирования противоречий между реализацией проблемно ориентированного устройства (обладающего максимальной производительностью) и необходимостью его универсального применения [1]. РВС достигают существенно более высокой реальной производительности и энергоэффективности за счет адаптации своей архитектуры к структуре решаемой задачи: пользователь может в РВС для каждой решаемой задачи создать специализированное вычислительное устройство, реализующее вычислительные операции информационного графа задачи с наименьшими задержками [2].

Современный уровень интеграции ПЛИС и достигаемые значения рабочей тактовой частоты реализаций алгоритмов на ПЛИС очень высоки – сотни миллионов эквивалентных вентилей способны работать на частотах до 700 МГц. Использование этих возможностей и близкое к максимальному заполнение ресурсов микросхемы делают невозможным отвод тепла от корпуса ПЛИС с использованием воздушных систем охлаждения при ограниченных габаритах изделия. Одной из труднейших проблем построения эффективных вычислителей является обеспечение приемлемых тепловых режимов работы как отдельных компонентов с высоким тепловыделением, так и функционально законченного вычислительного устройства в целом. Важным является выбор системы охлаждения. Проведенные в Научно-исследовательском центре супер-ЭВМ и нейрокомпьютеров (НИЦ СЭ и НК) исследования показали, что из различных вариантов жидкостного охлаждения наиболее эффективной является иммерсионная технология охлаждения [3].

2. РВБ «Арктур» на основе ПЛИС Xilinx UltraScale+

Каждый выпуск нового семейства ПЛИС сопровождается увеличением аппаратного вычислительного ресурса, ростом тактовой частоты вследствие уменьшения проектных норм производства, изменениями в технологии передачи данных, что подталкивает разработчиков реконфигурируемых суперкомпьютеров к поиску новых технических решений, позволяющих использовать эти новые преимущества.

Для построения РВБ нового поколения на основе ПЛИС Xilinx семейства UltraScale+ потребовался ряд изменений как в системе охлаждения, так и в конструкции изделия и его составных частей.

Корпус ПЛИС семейства UltraScale+ на 5 мм больше корпуса ПЛИС семейства UltraScale, используемых в предыдущих РВС, например РВБ «Неккар» [4]. В этой связи ширина печатной платы вычислительного модуля не позволяет разместить восемь вычислительных ПЛИС и контроллер. Для сохранения вычислительной плотности изделия из состава вычислительного модуля исключена ПЛИС контроллера. Функции контроллера (обеспечение процесса загрузки конфигураций ПЛИС, реализация интерфейса доступа к вычислительным ресурсам и мониторинг параметров) возложены на вычислительную ПЛИС, что потребует не более 5 % ее ресурса. Эскиз компоновки вычислительного модуля показан на рис. 1.

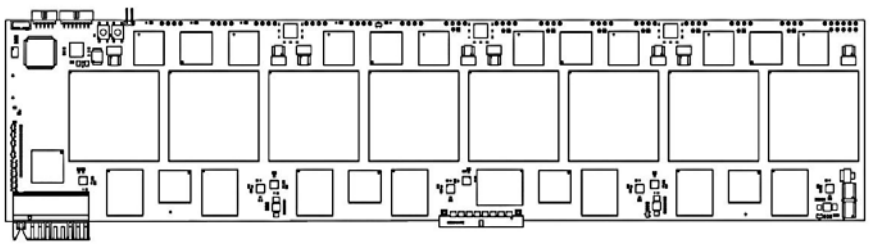


Рис. 1. Эскиз компоновки вычислительного модуля «Арктур»

Вычислительный ресурс новых микросхем UltraScale+ примерно в два раза превышает ресурс используемых ПЛИС предыдущего поколения. Следовательно, с учетом роста энергоэффективности потребление одной ПЛИС может вырасти до 170–200 Вт, что потребовало кардинальной переработки подсистемы вторичного электропитания вычислительного модуля.

В РВБ «Арктур» система питания спроектирована, исходя из расчетного потребления ПЛИС не менее 200 Вт. При таком потреблении вычислительный модуль с восемью ПЛИС может потреблять до 2 кВт, что делает обеспечение его питания от 12 В практически невозможным, так как ток в этом случае будет до 170 А. В этой связи в качестве основной шины питания выбрано напряжение 48 В, что позволяет снизить ток на плате вычислительного модуля в 4 раза до приемлемых значений 32 – 42 А. Нестабилизированные гальванические преобразователи 400/48 В могут быть реализованы распределенным образом, в том числе непосредственно на кросс-плате. Соответственно на вычислительном модуле выполняются преобразования из 48 В во все требуемые напряжения со стабилизацией.

Рост потребляемой мощности по ядру ПЛИС приводит к необходимости модернизации вторичного питания. При одноточечном размещении источника питания вследствие омического сопротивления возникает градиент напряжения, при этом падение при большом токе потребления становится существенным – до нескольких десятков милливольт. Корректировка топологии не приводит к существенному улучшению, поэтому необходимо использовать несколько источников напряжения ядра, что позволит сократить длины проводников и снизить падение напряжения в несколько раз.

Применение ПЛИС фирмы Xilinx семейства UltraScale+, выполненных по 16-нм технологии, позволит до трех раз повысить производительность вычислений за счет роста тактовой частоты и степени интеграции ПЛИС, причем без увеличения объема вычислительной системы. По заявлениям производителя в связи с новыми технологическими нормами производства снижение потребляемой мощности по сравнению с UltraScale может составить до 50 %. Однако при этом, несмотря на некоторый имеющийся резерв мощности разработанной системы иммерсионного жидкостного охлаждения, можно ожидать очередное стремление рабочих температур ПЛИС к своим критическим значениям.

РВБ «Арктур» обладает поддержкой широких возможностей информационного обмена. На рис. 2 показана структурная схема блока. МЗУ обеспечивает функциональную законченность изделия и выполнен в виде отдельной платы. Основными узлами МЗУ являются процессор Intel Core i5–6300U и ПЛИС семейства Kintex Ultrascale. ПЛИС выполняет функцию контроллера и обеспечивает взаимодействие между процессором и ПЛИС в первом вычислительном модуле. Связь между МЗУ и вычислительным модулем осуществляется через кросс-плату. МЗУ также обеспечивает аппаратный контроль датчиков уровня жидкости, датчика протока и параметров насоса с помощью микроконтроллера. На плату индикации выведены кнопки управления, индикаторы, а также интерфейсы для подключения клавиатуры, мыши, дисплея, Ethernet и оптический интерфейс.

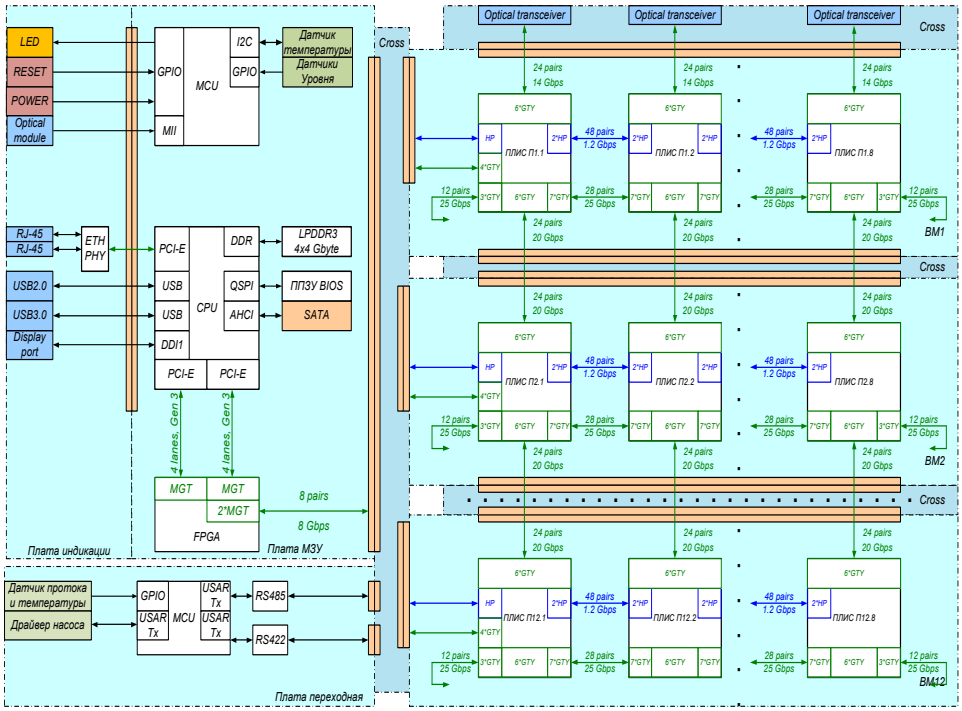


Рис. 2. Структурная схема РВБ «Арктур»

На рис. 3 показана структура информационных связей. В качестве основных связей между ПЛИС предполагается использование дифференциальных линий с подключенными к ним мульти-гигабитными трансиверами (MGT), вспомогательными связями являются дифференциальные линии, подключенные к HP – банкам.

На вычислительном модуле ПЛИС связаны горизонтальными и вертикальными линиями связей. Горизонтальные связи между ПЛИС представлены 28 и 12-ю дифференциальными парами

в каждом направлении со скоростью передачи данных до 25 Гбит/с по одной паре, что обеспечивает пропускную способность 700 и 300 Гбит/с независимо в каждом направлении. К вертикальным связям относятся 192 дифференциальные пары в каждом направлении с допустимой скоростью передачи данных до 20 Гбит/с по одной паре, что обеспечивает пропускную способность до 3840 Гбит/с независимо в каждом направлении.

Дополнительные линии связи представлены горизонтальной сетью от П1.1 до П1.8 на каждой плате (не показана на рис. 2) и содержит 48 дифференциальных линий с максимальной скоростью до 1,2 Гбит/с, каждая из которых может работать в любом направлении.

Кроме того, между платами существуют связи, предназначенные для служебных целей, таких как загрузка конфигурации ПЛИС, передача команд и данных от МЗУ к любому вычислительному модулю. Пропускная способность данного канала – 64 Гбит/с для обоих направлений независимо.

В настоящее время рассматривается два варианта производства РВБ «Арктур» – с использованием ПЛИС XCVU9P для вычислительно трудоемких задач и с использованием ПЛИС XCVU37P для задач с особыми требованиями к объему памяти. ПЛИС XCVU37P является одной из четырех в особой линейке ПЛИС UltraScale+, обладающей новым аппаратным ресурсом – встроенным модулем DDR памяти HBM2. Память HBM2 обладает объемом до 8 Гбайт и может обеспечивать многоканальный доступ 16×64 бит.

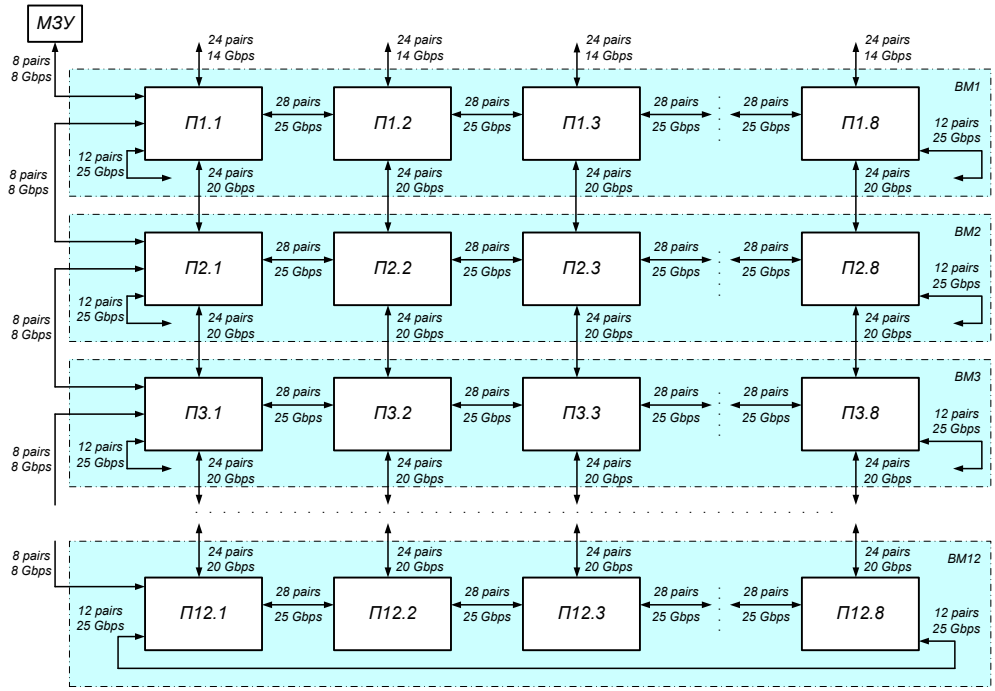


Рис. 3. Структура информационных связей РВБ «Арктур»

Все технические решения по созданию вычислительного модуля отмакетированы.

При построении вычислительных комплексов информационное взаимодействие между РВБ осуществляется через оптические каналы (рис. 4). Оптические приемо-передатчики установлены на отдельной кросс-плате и подключаются к первому вычислительному модулю через разъемное соединение. Скорость приема и передачи данных в каждом из оптических каналов достигает 14 Гбит/с. Полное количество оптических каналов составляет 384 и 192 канала на прием и 192 канала на передачу, это позволяет получить пропускную способность до 2688 Гбит/с независимо в каждом направлении.

ПЛИС вычислительного блока можно условно представить в виде строк и столбцов. Тогда строками будут являться ПЛИС, связанные между собой в пределах каждого вычислительного модуля РВБ, а столбцами – ПЛИС, связанные соответствующими межмодульными соединениями. Таким образом, обеспечивается информационная связь между соответствующими столбцами всех РВБ, объединенных оптическими каналами. Первый и последний РВБ в вычислительной стойке комплекса

могут быть как соединены между собой, образуя кольцевое соединение по столбцам, так и подключаться к РВБ соседних вычислительных стоек комплекса, наращивая объединенный в единое пространство ресурс. Для обеспечения изоморфности топологии информационных связей во всех РВБ первый канал столбца предыдущего РВБ всегда подключается ко второму каналу столбца последующего РВБ.

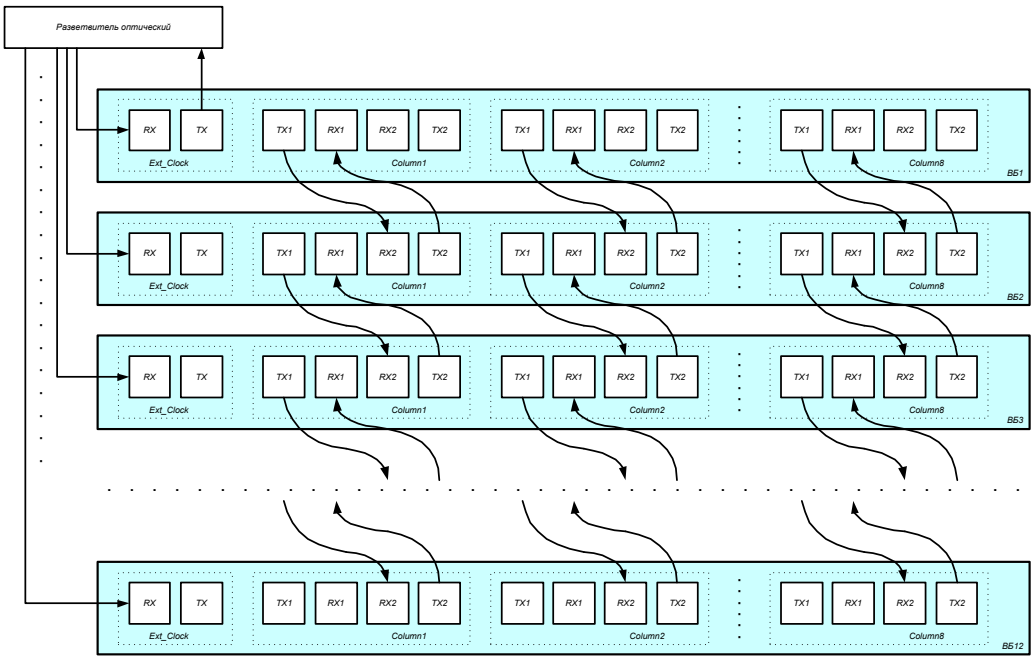


Рис. 4. Структура информационных связей между РВБ

Выполнена оценка производительности РВБ «Арктур» при использовании ПЛИС UltraScale+ XCVU9P на 12-ти вычислительных модулях.

Для оценки производительности использовалась конвейерная вычислительная структура, реализующая функцию фильтра с конечной импульсной характеристикой (КИХ) с симметричными коэффициентами. Вычислительный конвейер представляет собой последовательно соединенные каскады фильтрации. При реализации данной тестовой задачи цифровой обработки данных может быть достигнута производительность 240 Tflops (single precision). Таким образом, построение РВС из 16 РВБ «Арктур» обеспечит производительность при решении данной задачи 3,8 Pflops (single precision).

Для оценки производительности также использовалась задача решения СЛАУ с помощью конвейерной вычислительной структуры, реализующей функцию LU – разложения, поскольку данная функция является алгоритмической основой теста производительности компьютеров LINPACK. При реализации данной тестовой задачи может быть достигнута производительность 31,5 Tflops (double precision). При построении РВС на основе 16 РВБ «Арктур» при решении задачи LU – разложения производительность составит 500 Tflops (double precision).

3. Система охлаждения РВБ «Арктур»

Особенностью выпускаемых в НИЦ СЭ и НК реконфигурируемых вычислительных блоков (РВБ) для построения РВС является высокая плотность компоновки вычислительных компонентов. Каждый РВБ состоит из 12–16 базовых вычислительных элементов – вычислительных модулей. На одном вычислительном модуле устанавливается до восьми ПЛИС, каждая из которых требует обеспечения электропитанием с помощью микросхем модулей питания, которые также являются активными источниками тепла. Данное обстоятельство существенно увеличивает число тепловыделяющих элементов по сравнению с микропроцессорными модулями, состоящими из одного–двух процессоров.

В ступенчатом переносе тепла в иммерсионной (открытой) системе охлаждения принимают участие несколько компонентов. Эффективность системы жидкостного охлаждения обусловливается перспективными техническими решениями при реализации каждого компонента: применяемого хладагента, конструкции используемых радиаторов ПЛИС, насосного оборудования, теплообменников. Хладагент должен обладать наилучшей электрической прочностью, высокой теплопроводностью, максимально возможной теплоемкостью при низкой вязкости. Радиатор должен обеспечивать максимальную поверхность теплосъема, возможность организации циркуляции хладагента через радиатор, турбулентность потока хладагента в радиаторе, технологичность изготовления. Применяемый термоинтерфейс должен не деградировать и не вымываться хладагентом, иметь стабильно высокий коэффициент теплопроводности. Теплообменник должен обеспечивать высокий коэффициент теплопередачи между основным и вторичным контуром охлаждения. Для обеспечения циркуляции хладагента в объеме РВБ используется насос, который должен обладать необходимой производительностью.

Проведенные в НИЦ СЭ и НК работы позволили получить и апробировать технические решения по обеспечению эффективного теплопереноса каждым из компонентов системы охлаждения. В настоящее время в НИЦ СЭ и НК серийно производится РВБ «Неккар» (рис. 5). Рассмотрим основные составные части и особенности технологии производства данного изделия для общности понимания перспектив развития направления РВС с иммерсионной системой охлаждения.



Рис. 5. РВБ «Неккар»

Корпус РВБ «Неккар» [5] в конструктиве 3U 19” состоит из вычислительной секции и теплообменной секции. Вычислительная секция представляет собой герметичный контейнер с хладагентом, в которой размещаются вычислительные модули (12 штук), модули питания, плата управления, кросс-платы. Сверху вычислительная секция закрыта крышкой.

К вычислительной секции примыкает теплообменная секция, основными компонентами которой являются насос и теплообменник. Насос обеспечивает циркуляцию хладагента в РВБ по следующему замкнутому контуру: из вычислительной секции нагретый хладагент поступает в пластинчатый теплообменник, там охлаждается, затем вновь поступает в вычислительную секцию, охлаждает там нагретые электронные компоненты, отчего нагревается сам и вновь поступает в теплообменник. Теплообменник подключен к вторичному контуру охлаждения через штуцеры и предназначен для охлаждения хладагента, как правило, с помощью воды. Также в теплообменной секции располагается расширительный бачок для компенсации объемного расширения хладагента.

Каждый вычислительный модуль РВБ «Неккар», представленный на рис. 6, содержит восемь ПЛИС семейства Kintex UltraScale XCKU095T, а также контроллер вычислительного модуля, выполненный на ПЛИС Xilinx Kintex – 7. Для охлаждения ПЛИС используются радиаторы малой высоты оригинальной конструкции, установленные по особой технологии с помощью специально разработанного термоинтерфейса.



Рис. 6. Вычислительный модуль

Специалистами НИЦ СЭ и НК были проведены теплотехнические расчеты и предложена принципиально новая конструкция радиатора с оригинальными ламелями, которые позволяют осуществлять местное закручивание потока хладагента, переводя его из ламинарного в условно турбулентный поток.

Для осуществления загрузки ПЛИС и управления процессом вычислений в вычислительных модулях РВБ разработана оригинальная материнская плата на основе процессора Intel Skylake®Core I5–6300U – модуль загрузки и управления (МЗУ). Для МЗУ разработана оригинальная базовая система ввода-вывода (BIOS), позволяющая использовать все возможности системы на кристалле Intel Skylake и внешнего периферийного оборудования. Для теплоотвода от процессора используется радиатор оригинальной конструкции.

РВБ «Неккар» использует электропитание 400 В постоянного тока. Для электропитания вычислительных модулей разработан модуль питания, обеспечивающий четыре канала преобразования DC/DC 400/12 В мощностью до 4 кВт для электропитания четырех ПВМ. Максимальная потребляемая мощность РВБ – 12 кВт.

Для охлаждения электронных компонентов РВБ был разработан диэлектрический хладагент – масло маловязкое, диэлектрик МД–4,4 Gazpromneft для охлаждения электронных компонентов ЭВМ. Хладагент обладает высокой электрической прочностью и теплопроводностью, а также максимально возможной теплоемкостью при низкой вязкости.

Для эксплуатации РВБ и РВС на их основе требуется подключение с помощью подающих и обратных коллекторов через штуцеры (или балансировочные клапаны) и гибкие трубопроводы к вторичному контуру охлаждения, а также к источнику энергоснабжения и сетевому концентратору.

Уникальная вычислительная плотность РВБ «Неккар» позволяет достигать производительности при решении тестовой задачи цифровой обработки данных до 60 Tflops (single precision). Таким образом, построение РВС из 16 РВБ обеспечит производительность 1 Pflops. Производительность РВС на основе 16 РВБ «Неккар» при решении задачи LU – разложения (тест LinPack) составляет 160 Tflops (double precision).

Комплекс технических решений разработанной погружной системы жидкостного охлаждения позволил обеспечить в рабочем режиме температуру хладагента не более 33 ºС при потребляемой мощности до 90 Вт для каждой ПЛИС. При этом максимальная температура ПЛИС не превышает 57–60 ºС, что свидетельствует о резерве мощности, позволяющем обеспечить эффективное охлаждение для проектируемых РВС на основе перспективного семейства ПЛИС Xilinx UltraScale+.

При проектировании РВБ «Арктур» в рамках модернизации системы охлаждения решены следующие задачи:

– отработка оптимальной конструкции радиаторов за счет увеличения эффективной площади поверхности теплообмена между радиатором ПЛИС и хладагентом;

– увеличение производительности насоса подачи хладагента и повышение надежности системы жидкостного охлаждения посредством применения погружных насосов;

– повышение производительности теплообменника.

Для отвода большего количества тепла от ПЛИС необходимо увеличивать площадь теплосъема у радиаторов, монтируемых на ПЛИС, исключив при этом увеличение высоты радиатора. В рамках проведенных исследований рассматривалось множество вариантов конструкции радиаторов, часть из которых были эффективными, но очень дорогостоящими в технологии изготовления. Для радиатора удалось увеличить площадь теплосъема на 17 и 43 % для различных исполнений радиаторов, а если учесть, что с увеличением габаритов ПЛИС можно увеличить и габариты (длину и ширину) радиатора, площадь теплосъема возрастает более чем в два раза.

Циркуляцию хладагента в системе жидкостного охлаждения обеспечивает насос с приводом от электродвигателя постоянного тока. Электродвигатель является единственным элементом в РВБ с жидкостным охлаждением, требующим отдельного охлаждения, для чего в вычислительных стойках предусматривается принудительная циркуляция холодного воздуха. Предложено охлаждать электродвигатель хладагентом, циркулирующим в РВБ. Для этих целей разработаны и успешно тестируются два насоса, оснащенных погружными электродвигателями мощностью 0,5 и 1 кВт. Использование этого новшества позволяет снизить требования к вычислительным стойкам и поме-

щениям, где эксплуатируются РВС с иммерсионным охлаждением. Кроме того, значительно упрощается техническое обслуживание, а также повышается надежность за счет уменьшения количества открытых гидравлических соединений при эксплуатации.

Увеличение выделяемой тепловой мощности ПЛИС влечет увеличение мощности теплообменника не менее чем в 1,5 раза, что потребовало нарастить количество рабочих пластин с 24 до 36 штук, что неизменно повлияло на его высоту. При существующей схеме расположения гидравлических элементов сохранение габарита изделия 3U стало невозможным. Проблему могло решить расположение теплообменника на боковой поверхности, но в этом случае теплообменник переставал выполнять свою основную функцию – отвод тепла. Дело в том, что у современных теплообменников штуцеры для подключения теплоносителей одной среды, как правило, расположены в одной плоскости, параллельной боковой поверхности, а не по диагонали. Поэтому при установке теплообменника на боковой поверхности невозможно удалить воздух из теплообменника для среды, у которой штуцеры расположены внизу.

Для конструкции РВБ идеальным виделось диагональное расположение штуцеров сред на противоположных сторонах. Это позволило бы разделить подключение хладагента и воды на разных сторонах теплообменника, сократить количество гидравлической арматуры и минимизировать последствия нарушения герметичности подключения изделия. Кроме того, необходимо, чтобы теплообменник эффективно работал с более вязкой средой, чем вода, – хладагентом МД – 4,4 Gazpromneft. Это требовало применения пластин специальной конфигурации. В сотрудничестве с немецким производителем пластинчатых теплообменников фирмой «Funke» данная проблема была решена. За основу были взяты теплообменники серии TPL, которые разрабатывались специально для охлаждения гидравлических и моторных масел водой. Ширина канала теплообменников TPL до 80 % больше, чем у классических паяных водо-водяных теплообменников. Специальными встроенными турбулизаторами и эффективными диагональными направляющими создается максимальная эффективность. Теплообменники TPL обладают повышенными требованиями к технологии пайки, исключающими возможность перемешивания двух сред.

Претерпела изменения и система компенсации объемного расширения хладагента вследствие изменения его температуры. В РВБ «Неккар» для этих целей использовался расширительный бачок, сообщающийся с атмосферой, где изменялся уровень хладагента в зависимости от его объемного расширения. Такая система имела ряд недостатков: сложность конструкции расширительного бачка и наличие соединительных гидравлических узлов требовало тщательной герметизации изделия и регулярного контроля уровня хладагента. Кроме того, хладагент, находящийся в расширительном бачке, контактировал с атмосферным воздухом, что приводило к его дополнительному окислению и неизбежному сокращению срока службы.

В предложенном техническом решении роль компенсатора объемного расширения хладагента играют гофрированные резиновые рукава. При этом полностью исключен контакт хладагента с атмосферным воздухом и минимизированы места потенциальных протечек.

Эскиз конструкции РВБ «Арктур» с модернизированной погружной системой охлаждения представлен на рис. 7.



Рис. 7. Эскиз конструкции РВБ «Арктур» с модернизированной системой охлаждения

Все технические решения по модернизации системы охлаждения для РВБ «Арктур» проверены на различных стендах и макетах и показали возможность эффективного функционирования изделия.

4. Комплекс средств программирования реконфигурируемых вычислительных систем нового поколения

Оригинальный комплекс средств программирования РВС [2, 6] в течение 17 лет разрабатывается коллективом Научно-исследовательского центра супер-ЭВМ и нейрокомпьютеров (г. Таганрог), ведущего российского разработчика и производителя высокопроизводительных РВС. Отличительной характеристикой созданного комплекса является программирование многокристальных РВС на языке программирования высокого уровня с автоматической синхронизацией размещенных в каждой ПЛИС фрагментов задачи.

С 2017 года в НИЦ супер-ЭВМ и нейрокомпьютеров разрабатывается новый комплекс средств разработки прикладных программ для трансляции исходной программы на языке С в программу на языке COLAMO с последующей адаптацией под архитектуру РВС и синтезом конфигурационных файлов ПЛИС.

В отличие от автоматических распараллеливающих компиляторов [7] (Polaris, SUIF, VAST/Parallel, Intel/OpenMP и др. для вычислительных систем с общей памятью и ParaWise, Parallel Studio, САПФОР [7, 8], ОРС и ДВОР [9] и др. для систем с распределенной памятью), для создания параллельной программы в РВС используется альтернативный распараллеливанию последовательной программы подход. Для РВС задача исходно представляется в абсолютно параллельной форме в виде отражающего естественный параллелизм и конвейеризацию операций информационного графа [1, 2], который адаптируется к текущей конфигурации вычислительной системы с помощью методов масштабирования (индукции и редукции) производительности вычислительной структуры. При этом информационный граф может масштабироваться не только по данным, но и по итерациям, что сокращает число используемых каналов памяти (или ввода-вывода).

Структура комплекса и схема его функционирования представлены на рис. 8.

Комплекс средств программирования РВС содержит:

– конвертор «Ангел» для преобразования программы на языке программирования «С» в стандарте ISO/IEC 9899:1999 в абсолютно-параллельную программу на языке Colamo;

– конвертор «Русалка», преобразующий абсолютно-параллельную программу на языке Colamo в ресурсонезависимую (масштабируемую) программу на языке Colamo;

– программу «Прокруст», осуществляющую адаптацию параметров ресурсонезависимой программы на языке Colamo под архитектуру РВС;

– конвертор «Щелкунчик», осуществляющий редукцию производительности ресурсонезависимой программы при нехватке аппаратного ресурса;

– интегрированную среду разработки;

– транслятор языка программирования высокого уровня Colamo;

– синтезатор масштабируемых вычислительных структур Fire!Constructor;

– описания (паспорта) РВС и библиотеку элементов;

– средства диагностики и мониторинга;

– средства описания софт-архитектур и синтезатор Steam!Constructor;

– управляющую программу на языке С.

В отличие от большинства известных средств трансляции языка С в конфигурационные файлы ПЛИС (трансляторы C-to-RTL, MitrionC, CatapultC и др.) рассматриваемый комплекс средств программирования не синтезирует специализированный вычислитель на основе автоматной модели или процессорной парадигмы, а отображает последовательную программу в абсолютно параллельную форму, которую адаптирует под доступный аппаратный ресурс РВС или гибридной РВС с учетом информационных зависимостей данных.

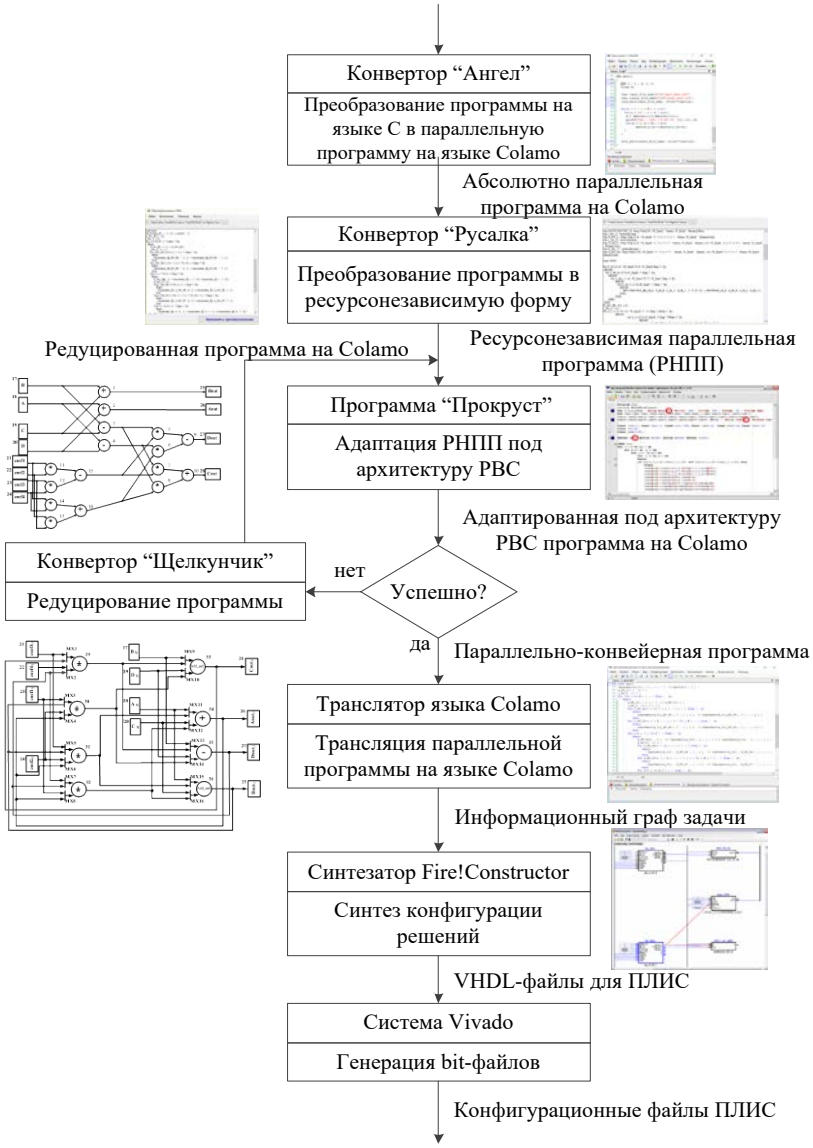


Рис. 8. Структура и схема функционирования комплекса средств программирования нового поколения

От наиболее близкого аналога Xilinx Vivado HLS (High-Level Synthesis) при схожей функциональности комплекс отличается поддержкой многокристальных решений с автоматической синхронизацией информационных и управляющих сигналов.

Входной программой является последовательная программа на языке программирования С в наиболее распространенном стандарте ISO/IEC 9899:1999 с GNU-расширением для компилятора gcc. Методика перевода последовательной программы на языке программирования С в программу на языке программирования COLAMO содержит 5 основных этапов.

Первый этап: перевод программы на языке программирования С в абсолютно-параллельную форму на языке программирования COLAMO. Результатом первого этапа является абсолютно параллельная программа в синтаксисе COLAMO, но требующая для своей реализации большое число каналов памяти.

Второй этап: анализ информационных зависимостей исходной программы, расщепление скалярных переменных и растягивание массивов по итерациям выполняется транслятором «Ангел». При сохранении информационных связей исходной последовательной программы обеспечивается выполнение правил единственной подстановки и однократного присваивания в параллельной программе

с помощью анализа информационных зависимостей переменных и массивов данных исходной программы. Если для адресации массива данных с явно выраженной информационной зависимостью используются не все переменные циклов, то те циклы, которые не используются для адресации элементов массива (потока) данных, являются для него итерационными (как правило, итерации действуют для самого нижнего вложенного цикла). На каждой итерации итерационного цикла будет осуществляться запись в одни и те же ячейки коммутационного массива, что приведет к нарушению правила однократного присваивания. Поэтому необходимо осуществить растягивание коммутационного массива по итерациям (исходный мемориальный массив остается без изменений).

Третий этап: переход в параллельно-конвейерную форму, разбиение массивов и циклов на векторную и потоковую составляющие выполняются процессором «Русалка».

Каждое измерение любого массива разбивается на 2 составляющие – векторную и потоковую с параллельным и последовательным типами доступа соответственно. Для этого используется следующая схема разбиения: длина потоковой составляющей задается равной R, а длина векторной определяется как $(T + R - 1)/R$, где T – исходная длина измерения массива). Векторная составляющая должна быть целочисленной, т.е. исходная длина измерения массива должна нацело делиться на R.

Четвертый этап: перевод в масштабируемую параллельно-конвейерную форму выполняется процессором «Прокруст». Если каналов памяти недостаточно для реализации базового подграфа, то необходимо выполнять редукцию программы [10]. Полученный коэффициент масштабирования K определяет параллелизм по каждому измерению массива. Полученный коэффициент K не всегда является итоговым, так как он характеризует максимальную степень масштабирования и на него нацело должен делиться исходный размер задачи N. Аналогичным образом определяются необходимый для реализации базового подграфа аппаратный ресурс (по числу LUT, DSP, Flip-Flop и других ресурсов ПЛИС) и коэффициент масштабирования по ресурсу Q.

Пятый этап: создание вычислительной структуры выполняется процессором «Прокруст».

Выполнение описанной методики позволяет синтезировать ресурсонезависимую параллельную прикладную программу, которая адаптируется под доступный вычислительный ресурс изменением значений нескольких констант в автоматическом режиме без существенной корректировки исходного текста программы. В результате выполнения всех 5-ти этапов создается масштабируемая прикладная программа на языке программирования COLAMO, которая с помощью существующих транслятора COLAMO, синтезатора Fire! Constructor и других средств разработки преобразуется в конфигурационные файлы ПЛИС многокристальных РВС.

В настоящее время создана экспериментальная версия комплекса средств разработки пятого поколения, включающая все программы и конверторы, с помощью которой осуществлена успешная трансляция ряда последовательных программ на языке С в программы на языке программирования высокого уровня COLAMO и далее в конфигурационные файлы ПЛИС для различных предметных областей. Успешно транслированы следующие прикладные программы математической физики: решение систем линейных алгебраических уравнений (СЛАУ) методом Гаусса, решение СЛАУ методом Якоби, решения СЛАУ с помощью верхней треугольной и нижней треугольной матриц (LU-разложение). В области символьной обработки решены 5 прикладных задач, при этом программы, транслируемые с помощью комплекса средств нового поколения, показывают высокую эффективность: не менее 85 % от созданных прикладными программистами на языке COLAMO и не менее 70 % от прикладных решений, разработанных специалистами-схемотехниками. Время трансляции последовательных программ на языке С в программы на языке программирования высокого уровня COLAMO и в конфигурационные файлы ПЛИС (без синтеза загрузочных конфигурационных bit-файлов) не превышает 30 минут, основное время трансляции составляет работа среды проектирования Xilinx Vivado.

5. Заключение

ПЛИС как элементная база реконфигурируемых супер-ЭВМ обеспечивают устойчивый, близкий к линейному, рост производительности РВС, открывая новые перспективы по созданию суперкомпьютеров петафлопсной производительности. В отличие от элементной базы многопроцессорных вычислительных систем, ПЛИС поддерживают развитие РВС за счет сохранения темпов роста аппаратного ресурса и рабочей тактовой частоты. Построение РВС из 16 РВБ «Арктур» (одна вычислительная стойка) обеспечит пиковую производительность вычислений до 4 Pflops (single precision), позволив при этом за счет мощной системы информационных связей решать широкий класс задач, в том числе сильносвязанных.

Эффективность применения иммерсионной системы жидкостного охлаждения в перспективных системах обуславливается техническими решениями при реализации каждого из компонентов системы – хладагента, радиаторов ПЛИС, насосного оборудования, теплообменников. При переходе к новой элементной базе требования к этим компонентам усиливаются, необходима модернизация готовых и поиск новых решений. Для построения нового РВБ «Арктур» на основе ПЛИС Xilinx UltraScale+ в ООО «НИЦ СЭ и НК» проведены масштабные работы по доработке компонентов системы охлаждения, что позволит осуществить эффективный отвод тепла, оставив при этом конструктив изделия 3U 19” без изменений.

Продолжение активного развития технологии построения РВС открывает возможности для создания вычислительных систем экзафлопсной производительности. Одним из шагов в этом направлении является создание перспективного РВБ «Арктур» на основе ПЛИС Xilinx UltraScale+, серийное производство которого намечено ООО «НИЦ СЭ и НК» в 2021 г.

Для эффективного программирования РВС и гибридных реконфигурируемых вычислительных систем разработана методика трансляции последовательной программы на языке С в программу на языке программирования высокого уровня COLAMO и далее, с помощью разработанных средств трансляции, – в конфигурационные файлы ПЛИС. Создаваемый комплекс средств разработки прикладных программ позволит существенно сократить время программирования многокристальных РВС с помощью трансляции исходной последовательной программы на языке программирования С в загрузочные конфигурационные файлы ПЛИС.

Совокупность разработанных аппаратно-программных средств на основе ПЛИС новых поколений открывает возможность создания высокопроизводительных вычислительных комплексов мирового уровня.

Литература

1. Каляев А. В., Левин И. И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Янус-К, 2003.

2. Гузик В. Ф., Каляев И. А., Левин И. И. Реконфигурируемые вычислительные системы / Учебное пособие под общей редакцией Каляева. И. А. 2016. – Ростов-на-Дону: Изд-во ЮФУ, 2016.

3. Levin I. I., Dordopulo A. I., Fedorov A. M., Gulenok A. A. Reconfigurable computer based on Virtex UltraScale+ FPGAs with immersion cooling system // XI международная научная конференция "Параллельные вычислительные технологии (ПаВТ) 2017 (Parallel computational technologies (PCT’2017) agora.guru.ru/pavt)", Казань, Республика Татарстан, Россия, 3–7 апреля 2017 г. – Челябинск: Издательский центр ЮУрГУ, 2017. С. 27–41. ISBN 978-5-696-04880-2 URL: https://link.springer.com/chapter/10.1007/978-3-319-67035-5_3 (дата обращения: 03.10.2019).

4. Каляев И. А., Дордопуло А. И., Левин И. И., Федоров А. М. Развитие отечественных многокристальных реконфигурируемых вычислительных систем: от воздушного к жидкостному охлаждению // Труды СПИИРАН. – Санкт-Петербург: Изд-во СПИИРАН ФГБУН Санкт-Петербургский институт информатики и автоматизации РАН, 2017. № 1(50). С. 5–31. DOI 10.15622/ sp.50.1 http://dx.doi.org/10.15622/sp.50. (дата обращения: 03.10.2019).

5. Levin Ilya, Dordopulo Alexey, Fedorov Alexander, Doronchenko Yuriy. High-Performance Reconfigurable Computer Systems with Immersion Cooling // Parallel Computational Technologies (12th International Conference, PCT 2018, Rostov-on-Don, Russia, April 2–6, 2018. P. 62–76. https://link.springer.com/content/pdf/10.1007%2F978-3-319-99673-8_5.pdf (дата обращения: 03.10.2019).

6. Gulenok A. A., Dordopulo A. I., Levin I. I., Gudkov V. A. Hybrid computer system programming technology with adaptation and scaling of calculations // Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. Vol. 6. № 1. P. 73–86. DOI: 10.14529/cmse170105.

7. Бахтин В. А., Жукова О. Ф., Катаев Н. А., Колганов А. С., Крюков В. А., Поддерюгина Н. В., Притула М. Н., Савицкая О. А., Смирнов А. А. Автоматизация распараллеливания программных комплексов // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19-24 сентября 2016 г., г. Новороссийск). – М.: ИПМ им. Келдыша М. В, 2016. С. 76–85. – doi: 10.20948/abrau-2016 – 31

8. Система САПФОР. URL:http://www.keldysh.ru/dvm/SAPFOR/ (дата обращения 03.10.2019)

9. Открытая распараллеливающая система. URL: http://www.ops.rsu.ru (дата обращения 03.10.2019).

10. Сорокин Д. А., Дордопуло А. И., Левин И. И., Мельников А. К. Решение задач с существенно-переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. 2012. № 2. С.49–56. DOI: 10.14489/issn.1810 – 7206

HIGH-PERFORMANCE RECONFIGURABLE COMPUTER SYSTEM BASED ON XILINX ULTRASCALE+ FPGAs

I. I. Levin, Yu. I. Doronchenko, A. I. Dordopulo, M. G. Levina

OOO “Scientific research centre of supercomputers and neurocomputers”, Taganrog

In the paper we review an advanced reconfigurable computer system on the base of modern FPGAs and with an immersion subsystem for electronic components cooling. The system consists of stand-alone computational blocks. We present the architecture of the system basic component such as a computational module. Owing to the structure of data connections among computational modules and among blocks, it is possible to solve a wide class of resource-intensive problems with multiple data exchange. We show new approaches to power supply of FPGAs which consume up to 180 Watt each. Owing to the unique design of the computational block it is possible to achieve the computational density of up to 128 FPGAs in a 3U construction. We show advantages of an immersion cooling subsystem. Besides, we describe technical solutions for improvement of the cooling subsystem. Owing to such improvement, it is possible to dissipate up to 20 KWatt of total heat from one computational block. Also, we review an original complex of high-level language programming tools for multichip RCSs.

Key words: reconfigurable computer systems, performance of calculations, FPGA, immersion cooling systems, computational density, programming tools, high-level language.

УДК 551.46
DOI 10.53403/9785951505071_2022_297

ДИНАМИЧЕСКИЕ ЭФФЕКТЫ ИНТЕНСИВНЫХ ВНУТРЕННИХ ВОЛН В ДВУХСЛОЙНОЙ ЖИДКОСТИ: ЛАБОРАТОРНЫЕ ЭКСПЕРИМЕНТЫ И ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ

Л. В. Лобовикова[1], *П. В. Лобовиков*[1], *О. Е. Куркина*[1], *А. А. Куркин*[1], *А. С. Козелков*[2]

[1] Нижегородский государственный технический университет им. Р. Е. Алексеева, Нижний Новгород
[2] Российский федеральный ядерный центр – Всероссийский НИИ экспериментальной физики, Саров

Представлены результаты лабораторных экспериментов по изучению динамики и структуры физических полей при распространении длинных волн на границе раздела в двухслойной жидкости. Большое внимание уделено характеристикам поля давления на дне и вертикальной стенках бассейна. Адаптированы и откалиброваны для расчетов численные модели (ЛОГОС, MITgcm) квази-идентичные лабораторной. Проведены численные эксперименты с различными конфигурациями начального состояния жидкости. Результаты численного моделирования качественно и количественно согласуются с результатами лабораторных экспериментов. Однако модель, построенная в рамках пакета ЛОГОС, обладает более высокой точностью при воспроизведении указанных физических процессов, особенно для динамических характеристик волновых полей, в частности, давления. Результаты работы позволяют полагать, что модель на основе ЛОГОС может быть в дальнейшем использована для детального моделирования распространения внутренних волн в стратифицированной жидкости и выявления различных режимов течений и оценок их динамической эффективности, как в условиях лаборатории, так и на морском шельфе.

Ключевые слова: лабораторные эксперименты, численное моделирование, внутренние волны, двухслойная жидкость, численная модель MITgcm, пакет программ ЛОГОС.

Введение

Исследование внутренних волн (ВВ) в стратифицированной жидкости – актуальная задача, интерес к которой не угасает уже более 50 лет [1–4]. Существует достаточно много численных моделей – глобальная модель циркуляции MITgcm [5], ЛОГОС Аэро-Гидро [6], численная модель ISWFoam, вычислительный комплекс IGWResearch [7], Princeton Ocean Model (POM) [8], Bergen Ocean Model (BOM) [9] и др., на основе которых изучается как генерация и взаимодействие ВВ, так и динамические эффекты интенсивных ВВ.

В настоящей работе представлен комплексный подход к решению задачи об исследовании динамики ВВ (в том числе в контексте геофизических приложений), включающий в себя как стадию лабораторного моделирования, так и адаптацию под цели исследования надежных вычислительных комплексов, прошедших валидацию на полученных экспериментальных данных. Были использованы глобальная модель циркуляции MITgcm разработанная в Массачусетском технологическом институте с бесплатным доступом и открытым кодом и пакет программ ЛОГОС Аэро-Гидро (далее ЛОГОС, http://logos.vniief.ru/), который является коммерческим продуктом РФЯЦ-ВНИИЭФ и на текущий момент имеет широкий спектр применения, возможности которого продолжают расширяться.

Полнонелинейная модель, на которой базируется MITgcm, основывается на трехмерных уравнениях гидродинамики трехмерной несжимаемой жидкости в приближении Буссинеска, что позволяет учитывать негидростатические процессы, в частности описывать генерацию и распространение ВВ, которые напрямую зависят от диффузионных, вязкостных и адвективных свойств жидкости, т. е. данная модель предполагает эмпирическое задание и варьирование коэффициентов, отвечающих за перечисленные эффекты.

В пакете программ ЛОГОС для моделирования динамики несжимаемой жидкости используется математическая модель многофазной среды, основанная на системе уравнений Навье – Стокса [10]. При этом, в отличие от MITgcm, ЛОГОС позволяет напрямую проводить моделирование динамики уединенных ВВ без привлечения эмпирических коэффициентов, используя метод конечных объемов с неструктурированными сетками и локальным уточнением сетки. Как и MITgcm, ЛОГОС поддерживает технологии кластерных и суперкомпьютерных вычислений, что позволяет значительно сократить время проведения расчетов для ресурсоемких инженерных задач. Необходимо отметить, что ранее пакет программ ЛОГОС не использовался для моделирования динамики стратифицированной жидкости и проведенное исследование позволило расширить диапазон его применимости к негидростатическим задачам.

Постановка задачи

Основные инструменты исследования в данной работе – лабораторное моделирование в лотке НИЛ Моделирования природных и техногенных катастроф НГТУ им. Р. Е. Алексеева и соответствующие численные эксперименты, воспроизводящие задачу с начальными условиями в жидкостях с двумя однородными слоями различной плотности. Лабораторные эксперименты проводились в гидродинамической установке, параметры которой приведены в табл. 1. Данная установка может быть использована для моделирования волновых процессов в замкнутых или почти замкнутых акваториях «вытянутой формы». Примером таких водоемов могут служить некоторые каналы, множество озер (например, Балатон, Веттерн и др.), бухты и гавани, фьорды.

Таблица 1

Параметры лабораторной установки

Длина резервуара	6.24 м
Ширина резервуара	0.5 м
Расстояние между левой стенкой лотка и воротами (L_g)	0.45 м
Толщина ворот (Δg)	0.03 м
Кварцевые преобразователи давления	p_{01}-p_{21}
Датчик атмосферного давления	p_{22}

Численные расчеты выполнены с помощью вычислительного комплекса MITgcm [11, 12] и пакета программ ЛОГОС Аэро-Гидро [13]. Их результаты сопоставлены с данными лабораторных экспериментов по генерации и распространению уединенной внутренней волны отрицательной полярности, схема, одного из которых приведена на рис. 1. Начальные параметры задачи: $\rho_1 = 995.75$ кг/м^3 – плотность пресной воды, $\rho_2 = 1019.49$ кг/м^3 – плотность соленой воды. Динамическая вязкость верхнего слоя $\mu_2 = 0.8937$мПа·с, и динамическая вязкость нижнего слоя $\mu_1 = 0.85336$ мПа·с, $h_1 = 0.05$ м, $h_2 = 0.2$ м, $H_1 = 0.05$ м, $H_2 = 0.2$ м. Уединенные внутренние волны генерируются методом так называемого «гравитационного коллапса» [14, 15] за счет извлечения вертикальной перегородки со скоростью 1 м/с. При этом происходит образование локализованного возмущения на границе раздела жидкостей и последующее его распространение. Поскольку уровень легкой жидкости в основной части бассейна меньше, уровня тяжелой, то уединенная внутренняя волна, в описанном случае, имеет отрицательную полярность.

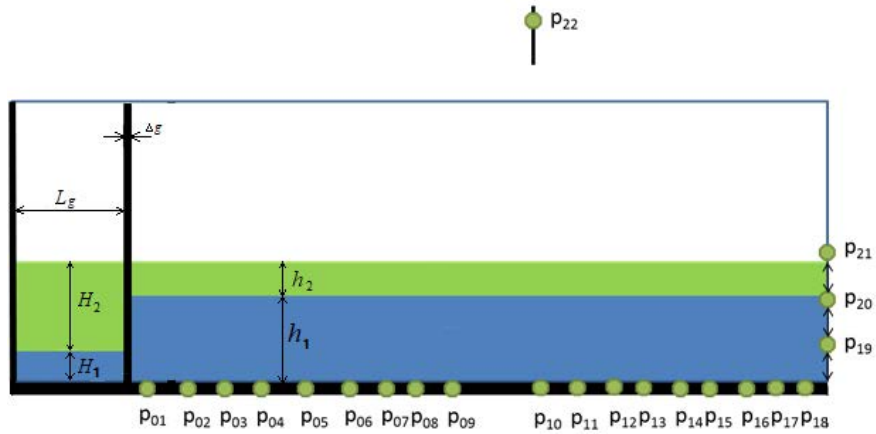


Рис. 1. Схема задачи

Начальные условия в численных моделях квази-идентичны лабораторной, т. е. полностью соответствовали схеме, представленной на рис. 1. Регистрация проведенных экспериментов осуществляется с помощью видеосъемки и кварцевых преобразователей давления, установленных на дне и вертикальной стенке гидродинамической установки. Для воспроизведения морфологических характеристик уединенной волны производится оцифровка видеокадров каждого эксперимента. Временные ряды давления в волновом поле как при численных, так и при лабораторных экспериментах записывались в точках p_{01}–p_{21} (рис. 1).

Основные результаты

Траектории движения уединенной ВВ в виде *X*-*T* диаграмм смещения границы раздела жидкостей по данным численного моделирования (MITgcm, ЛОГОС) и лабораторного эксперимента представлены на рис. 2. На рисунке четко прослеживается движение с постоянной скоростью уединенной ВВ отрицательной полярности вдоль трассы распространения, значительное увеличение ее максимальной амплитуды в момент столкновения с твердой стенкой, расположенной на правой границе расчетной области и генерация мелкомасштабных возмущений после ее взаимодействия со стенкой во всех экспериментах (численных и лабораторных). При этом наклон траекторий распространения солитоноподобной волны, характеризующий ее нелинейную скорость сохраняется.

Результаты сопоставления формы смещения границы раздела жидкостей в центре расчетной области приведены на рис. 3. Можно видеть, что наблюдается хорошее качественное соответствие по длине волны. При этом, отличие результатов численных экспериментов от лабораторных по амплитуде смещения составляет 7.2 % (MITgcm) и 9.8 % (ЛОГОС).

Для количественных оценок различий результатов лабораторного и численного моделирования можно использовать функцию взаимной корреляции временных рядов смещения границы раздела жидкости (рис. 4, слева). Как можно видеть на рис. 4 (справа) максимальная корреляция между экспериментальными данными и данными численных экспериментов достигает значения $K = 0.90$ с задержкой в 4 секунды (ЛОГОС) и $K = 0.86$ с задержкой в 2 секунды (MITgcm), что говорит о неплохой согласованности результатов численного и лабораторного моделирования.

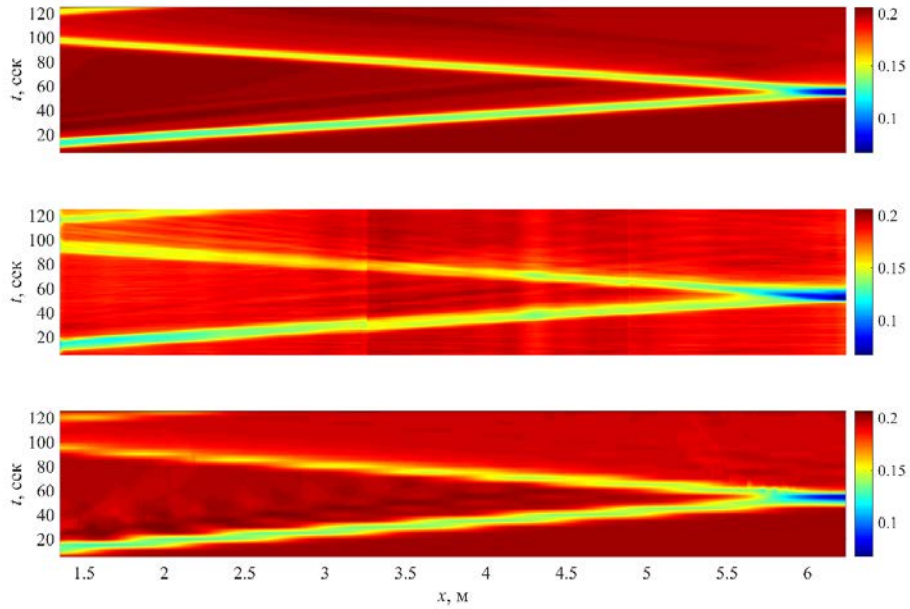


Рис. 2. Пространственно-временные диаграммы процесса смещения границы раздела жидкостей по данным численного моделирования в MITgcm (сверху), по данным видео лабораторного эксперимента (в центре) и по данным численного моделирования в программном пакете ЛОГОС (снизу); цветом показана амплитуда в м

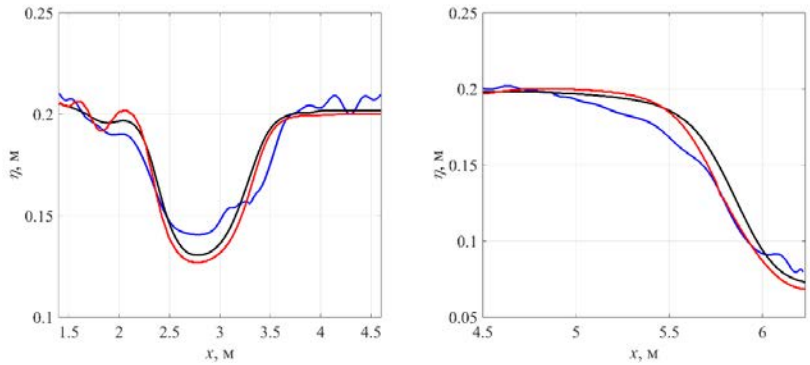


Рис. 3. Форма уединенной внутренней волны в центре установки (слева) и в момент взаимодействия со стенкой (справа); синяя линия – лабораторный эксперимент; черная – MITgcm; красная – ЛОГОС

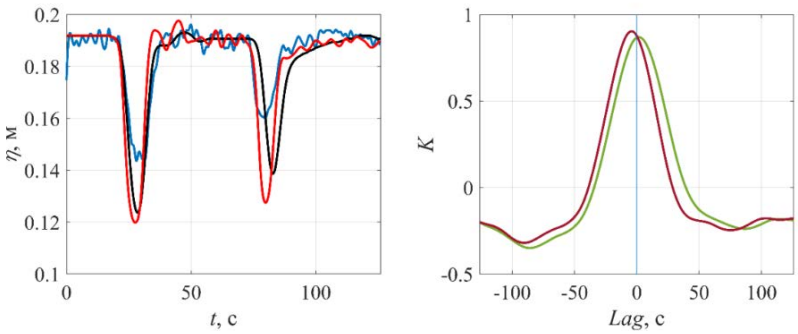


Рис. 4. Слева: смешение границы раздела от времени в точке x = 3.1 м. Синяя линия – лабораторный эксперимент; черная – MITgcm; красная – ЛОГОС; справа: корреляция смещения границы раздела, коричневым цветом – эксперимент-ЛОГОС, зеленым цветом – эксперимент-MITgcm

На рис. 5: по данным лабораторного и численного моделирования показано проявление процесса генерации и трансформации интенсивных ВВ отрицательной полярности в поле динамического давления на дне и вертикальной стенке лотка. Следует отметить, что как изменения давления, как и динамика этих изменений во времени, хорошо согласуются в лабораторных и численных экспериментах (разница в амплитуде давления за время эксперимента составила не более 26 %, как для MITgcm, так и для ЛОГОС).

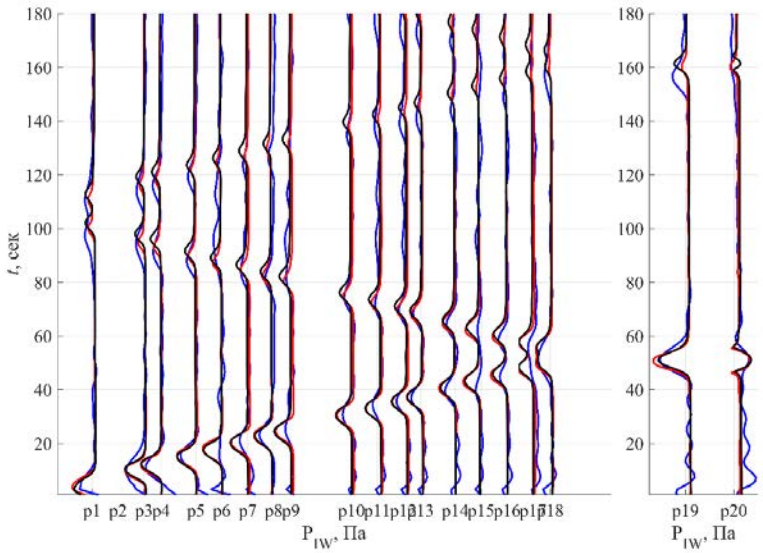


Рис. 5. Сравнение вклада ВВ в общее поле давления жидкости на дно и вертикальную стенку в рамках трех подходов: черная линия – численное моделирование средствами MITgcm; синяя линия – лабораторный эксперимент; красная линия – численное моделирование средствами пакета ЛОГОС

Максимальные вариации давления ВВ на датчике p_{10} (в центре расчетной области) (рис. 6) составляют ≈ 9.45 Па по данным пакета программ ЛОГОС и ≈ 7 Па по экспериментальным данным, а максимальная корреляция между экспериментальными данными и ЛОГОС достигает значения $K = 0.87$.

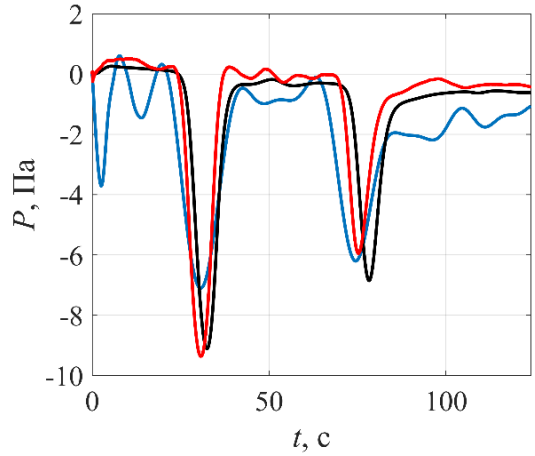


Рис. 6. Слева: временное распределение давления ВВ с датчика p_{10}. Синяя линия – лабораторный эксперимент; черная – MITgcm; красная – ЛОГОС

Заключение

В данной работе проанализированы особенности динамических полей, индуцированных сильно нелинейной уединенной внутренней волной в двухслойной жидкости. Результаты расчетов сопоставлены с данными лабораторных экспериментов, наблюдается их хорошее соответствие. Проведено качественное и количественное сравнение характеристик ВВ (амплитуда, ширина, корреляция), дана оценка величины придонного давления в интенсивной ВВ и давления на вертикальную стенку при ее отражении. Показано, что результаты лабораторного и численных экспериментов обладают корреляцией ($K > 0.8$), как для смещения границы раздела, так и для временных рядов давления. Комплексный анализ полученных данных позволяет говорить о достаточно высокой эффективности применения программ вычислительной гидродинамики к воспроизведению результатов лабораторных экспериментов с интенсивными ВВ в двухслойной жидкости. Однако пакет программ ЛОГОС, обладает более высокой точностью при воспроизведении указанных физических процессов, особенно динамических характеристик волновых полей, в частности, давления. Результаты работы позволяют полагать, что пакет программ ЛОГОС может быть в дальнейшем использован для детального моделирования распространения ВВ в стратифицированной жидкости и выявления различных режимов течений и оценок их динамической эффективности, как в условиях лаборатории, так и на морском шельфе.

Представленные результаты получены при поддержке стипендии Президента Российской Федерации молодым ученым и аспирантам (СП–3546.2021.5).

Литература

1. Boegman L., Stastna M. Sediment resuspension and transport by internal solitary waves // Annual review of fluid mechanics. – 2019. – V. 51. – P. 129–154.

2. Woodson C. B. The fate and impact of internal waves in nearshore ecosystems // Annual review of marine science. – 2018. – V. 10. – P. 421 – 441.

3. Lamb K. G. Internal wave breaking and dissipation mechanisms on the continental slope/shelf //Annual Review of Fluid Mechanics. – 2014. – V. 46. – P. 231–254.

4. Zou L, Li Y., Hu Y., Wang Z., Yu Z. Frontal collision of two nonlinear internal solitary waves in a stratified fluid // Applied Ocean Research. – 2020. – V. 104. – P. 102334.

5. Adcroft A., Campin J.-M. MITgcm User Manual. Cambridge: MIT. 2011. P. 455.

6. Дерюгин Ю. Н., Козелков А. С., Зеленский Д. К., Глазунов В. А., Полищук С. Н., Ялозо А. В., Емельянов А. Б. Программный модуль пакета программ «ЛОГОС» для моделирования аэрогидродинамики и теплопереноса (ЛОГОС-ТМП), версия 2017–2019.

7. Lamb K. G. Numerical experiments of internal wave generation by strong tidal flow across a finite amplitude bank edge // Journal of Geophysical Research: Oceans. – 1994. – V. 99. – No. C1. – P. 843–864.

8. George L. Mellor Users guide for a three-dimensional, primitive equation, numerical ocean model Program in Atmospheric and Oceanic Sciences Princeton University // Princeton, 2002 - NJ 08544 – 0710.

9. Berntsen J. Users guide for a modesplit σ -coordinate numerical ocean model. 2004. P. 51.

10. Hirt C. W., Nichols B. D. Volume of fluid (VOF) method for the dynamics of free boundaries // J. Comput. Phys. – 1981. – V. 39. – P. 201–225.

11. Marshall J. S., Adcroft A. J., Hill C., Perelman L., Heisey C. A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers // J. Geophys. Res. – 1997. – V. 102. № C3. – P. 5753–5766.

12. Marshall J. S., Hill C., Perelman L., Adcroft A. J. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling // J. Geophys. Res. – 1997. – V. 102. № C3. – P. 5733–5752.

13. Козелков А. С., Курулин В. В., Лашкин С. В., Шагалиев Р. М., Ялозо А. В. Исследование потенциала суперкомпьютеров для масштабируемого численного моделирования задач гидродинамики в индустриальных приложениях // Журнал вычислительной математики и математической физики. – 2016. – Т. 56. № 8. – С. 1524–1535.

14. Grue J. Generation, propagation, and breaking of internal solitary waves // Chaos. – 2005. – V. 15. № 3. – P. 1–14.

15. Chen C.-Y., Hsu J.R.-C., Chen C.-W., Chen H.-H., Kuo C.-F., Cheng M.-H. Generation of Internal Solitary Wave by Gravity Collapse // J. Mar. Sci. Technol. – 2007. – V. 15. № 1. – P. 1–7.

DYNAMIC EFFECTS OF INTENSIVE INTERNAL WAVES IN A TWO-LAYER LIQUID: LABORATORY EXPERIMENTS AND NUMERICAL SIMULATION

L. V. Lobovikova[1], P. V. Lobovikov[1], O. E. Kurkina[1], A. A. Kurkin[1], A. S. Kozelkov[2]

[1] Nizhny Novgorod State Technical University n.a. R.E. Alekseev, Nizhny Novgorod
[2] Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The results of laboratory experiments on the study of the dynamics and structure of physical fields during the propagation of long waves at the interface in a two-layer liquid are described. Much attention is paid to the characteristics of the pressure field at the bottom and vertical walls of the pool. Mathematical models (LOGOS, MITgcm) are adapted and calibrated for calculations, quasi-identical to the laboratory one. Numerical experiments are carried out with various configurations of the initial state of the liquid. The results of numerical simulation are in qualitative and quantitative agreement with the results of laboratory experiments. However, the model built within the LOGOS package has a higher accuracy in reproducing the indicated physical processes, especially for the dynamic characteristics of field waves, in particular, pressure. The results of the work suggest that the LOGOS-based model can be further used for detailed modeling of the propagation of internal waves in a stratified fluid and for identifying various flow regimes and assessing their dynamic efficiency, both in laboratory conditions and on the sea shelf.

Key words: Laboratory experiments, numerical modeling, internal waves, two-layer fluid, MITgcm numerical model, LOGOS software package.

УДК 004.4
DOI 10.53403/9785951505071_2022_304

РЕАЛИЗАЦИЯ МЕХАНИЗМА ОТОБРАЖЕНИЯ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ОБЪЕМНОГО РЕНДЕРИНГА В ПАРАЛЛЕЛЬНОМ РЕЖИМЕ

В. В Ломтев, В. В. Жирнов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе представлен механизм отображения результатов моделирования с использованием объемного рендеринга [1] в параллельной системе постобработки ScientificView [2], предназначенной для визуализации результатов моделирования различных физических процессов и экспериментальных данных, представляемых в виде сеточной структуры или набора точек (частиц, молекул, кластеров).

Ключевые слова: ScientificView, объемный рендеринг, параллельный режим

Определение объемного рендеринга

Объемный рендеринг – технология компьютерной графики, которая позволяет представить распределение величины в объеме в виде облака пыли или тумана. В качестве исходных данных для отображения выступают результаты 3D-моделирования, представленные набором дискретных ячеек (сетка) с определенным значением величины в пределах каждой дискретной ячейки. Помимо цветовой интерпретации величины (как это делается в поверхностном полигональном случае), в данной технологии также используются соответствующие уровням величины значения полупрозрачности. Оперируя цветовой интерпретацией значений величины и соответствующими значениями полупрозрачности, пользователь может исключать определенные значения из анализа, или выделять их относительно других, как показано на рис. 1.

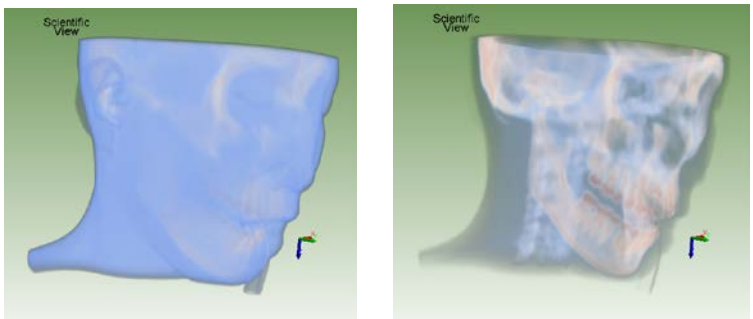


Рис. 1. Демонстрация работы технологии объемного рендеринга

Особенности реализации объемного рендеринга

Рендеринг объемных данных значительно отличается от рендеринга данных в классическом полигональном представлении. Современные видеокарты больше адаптированы к полигональной графике и способны обрабатывать десятки миллионов полигонов в секунду. И даже, когда речь идет

о смешивании цветов для поддержки полупрозрачных полигональных моделей, для каждого пикселя экрана идет обработка лишь небольшого числа фрагментов (некое значение цвета и глубины пикселя, которое претендует стать итоговым), в зависимости от количества используемых слоев глубины. При объемном рендеринге для каждого пикселя экрана необходимо обработать данные всех ячеек, находящиеся под позицией пикселя (все ячейки, которые проецируются в данный пиксель), что является ресурсозатратной задачей не только для видеокарты (в силу сильной ограниченности памяти), но и для центрального процессора, так как современные результаты моделирования могут содержать миллиарды ячеек. Поэтому, зачастую, исходные данные аппроксимируют на менее подробные сетки, состоящие из кубиков (вокселей), как показано на рис. 2. Обычно выбор размерности такой воксельной сетки в системах постпроцессинга производится пользователем в зависимости от ожидаемой степени соответствия исходным данным.

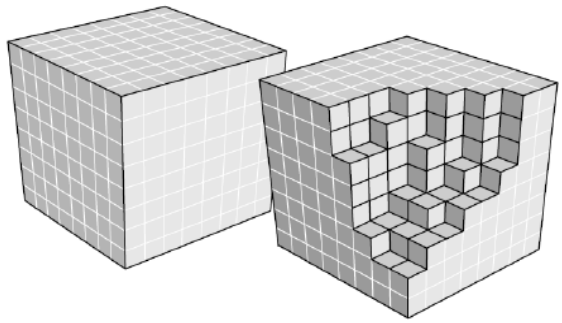


Рис. 2. Демонстрация воксельной сетки

Немаловажным фактором при отображении объемных данных является время формирования кадра. Даже на аппроксимированных сетках центральный процессор не сможет обеспечить скорость отображения, достаточную для интерактивного взаимодействия пользователя со сценой (далее – интерактивного взаимодействия), поскольку для каждого непустого пикселя необходимо найти и обработать все воксели, которые проецируются в данный пиксель. Использование ресурсов видеокарты позволит обеспечить высокую скорость формирования изображений в объемном представлении.

В параллельной системе постобработки ScientificView основным инструментом для визуализации как двумерных, так и трехмерных данных является графическая библиотека OpenGL [3]. Графическое ядро подсистемы визуализации ScientificView построено на базе API библиотеки OpenGL. При разработке универсальной концепции реализации объемного рендеринга решено было использовать только механизмы данной библиотеки и ее расширений, так как она является открытой, кроссплатформенной и поддерживается большинством из известных семейств видеокарт. К тому же, использование воксельного представления очень удобно для размещения объемных данных (матричная позиция вокселя и значение в нем) в памяти видеокарты. Такое представление хорошо согласуется с организацией 3D-текстур. Благодаря автоматической интерполяции, характерной для данного вида текстур, разработчик автоматически способен получить значение в любой точке объемных данных, даже если это положение не совпадает с центром вокселя.

В реализации объемного рендеринга используются механизмы соответствия цвета и коэффициента полупрозрачности значениям величины, распределенным по трехмерной сетке. В качестве таких механизмов удобно использовать одномерные текстуры. При этом для цветовой интерпретации используется текстура, заполняемая RGB значениями (слагаемые компоненты итогового цвета – красный, зеленый, синий), а для интерпретации коэффициента полупрозрачности используется текстура, заполняемая значениями компонента A (альфа). В системах визуализации задание вышеописанных соответствий является мощным инструментом для управления отображением исследуемых величин, так как пользователю предоставляется возможность скрыть определенные значения, сделав их полностью прозрачными (A = 0), или смешать цвета так, чтобы выделить определенные зоны данных, задавая контрастный цвет интересующим значениям с использованием коэффициента полупрозрачности наиболее близким к единице.

Процесс поиска вокселей, которые проецируются в определенный пиксель, заменяется на сбор значений величин, встречающихся при прохождении луча через 3D-текстуру в направлении от минимальной глубины к максимальной в позиции пикселя. При этом от шага прохождения вдоль луча зависит скорость формирования общего кадра. Шаг прохождения вдоль луча называют величиной сэмплирования, которая составляет характерный размер вокселя. Зачастую, с целью ускорения отображения при интерактивных взаимодействиях со сценой, в системах визуализации величину сэмплирования

увеличивают на определенный множитель. При этом прохождение по лучу проходит быстрее. По окончании интерактивного взаимодействия величина сэмплирования восстанавливается. Операции прохождения вдоль луча и смешения цветов реализуются во фрагментном шейдере, при этом в шейдер обязательно должны быть поданы в качестве uniform переменных: идентификатор 3D-текстуры объемных данных, идентификаторы 1D-текстур интерпретации цвета и компонента A, величина сэмплирования.

Концепция и алгоритм реализации объемного рендеринга в параллельной системе постобработки ScientificView

Таким образом, концептуально, формирование изображений с помощью объемного рендеринга можно разбить на три основных этапа:

– подготовка данных (не влияет на скорость интерактивного взаимодействия);

– предварительные операции отображения;

– трассировка лучей.

На этапе подготовки данных:

– исходные данные аппроксимируем на воксельную сетку, указанной пользователем размерности и вычисляем величину сэмплирования;

– при помощи API OpenGL на видеокарте выделяем память под 3D-текстуру и заполняем ее согласно распределению величины по воксельной сетке;

– при помощи API OpenGL на видеокарте выделяем память под 1D-текстуру цветовой интерпретации величины и заполняем ее согласно заданным пользователем уровням;

– при помощи API OpenGL на видеокарте выделяем память под 1D-текстуру соответствия величине компонента A и заполняем текстуру согласно заданным пользователем уровням.

На этапе предварительных операций отображения:

– находим минимальные и максимальные глубины для каждого пикселя. Для этого производим полигональное отображение (без цветовой интерпретации) данных для объемного анализа с включенным тестом глубины. В первый проход определяем минимальные глубины с функцией теста глубины GL_LESS, во второй проход определяем максимальные глубины с функцией теста глубины GL_GATHER. Результирующие буферы глубины сохраняем на видеоадаптере в виде текстур;

– включаем использование фрагментного шейдера.

При этом внутри фрагментного шейдера для каждого пикселя (этап трассировки лучей):

– определяем при помощи преобразований наблюдения модели и проекции 3D-координаты старта и окончания луча (по позиции пикселя и данным из текстур минимальных и максимальных глубин), при этом пропускаем пустые зоны, где значение минимальной глубины равно единице;

– проходим по лучу согласно величине сэмплирования, при этом каждую новую 3D-точку переводим в нормированные координаты текстур (в текстурах координаты нормируются в интервале от нуля до единицы);

– для каждой новой точки вдоль луча определяем величину из 3D-текстуры;

– согласно величине, определяем цвет и компонент A из соответствующих текстур, производим смешение цветов;

– при выходе луча из 3D-текстуры, заканчиваем обработку пикселя и устанавливаем для него цвет и компонент A согласно накопленным значениям.

В случае запуска системы ScientificView в режиме клиент-сервер, обработка данных и формирование изображений выполняются на стороне сервера. При этом сервер может работать как в последовательном, так и в параллельном многопроцессорном режиме с распределенной памятью. В последнем случае трассировка лучей проводится на специально выделенном управляющем процессоре, оснащенном видеокартой.

На управляющем процессоре проводится операция по сбору значений для итоговой воксельной сетки. С этой целью ячеечные величины исходных данных зачитываются с перехлестом на рабочих

процессорах. Это позволяет для каждого рабочего процессора определить воксели, которые не вносят вклад в итоговую сетку (в данные воксели попали все ячейки с перехлестного слоя), и воксели для которых формируется дополнительная информация для поиска итоговой величины (воксели, в которые попали как свои ячейки, так и ячейки с перехлестного слоя). В качестве дополнительной информации выступает определенное количество кортежей «расстояние от центра вокселя до центра ячейки – значение величины, соответствующее этому расстоянию».

После формирования воксельных представлений на рабочих процессорах, управляющий процессор запускает операцию попарного объединения среди рабочих, с динамическим распределением нагрузки. При этом в спорных вокселях происходит определение наиближайших кортежей из дополнительной информации. В результате итоговые значения для общей воксельной сетки концентрируются на управляющем процессоре, после чего запускается операция нахождения значений величины в спорных вокселях, исходя из определенного количества значений расстояний и соответствующих им значений величины.

С целью сохранения формы исходных данных, на управляющем процессоре запускается две операции объединения кадра для формирования минимальных и максимальных глубин. Данные глубины, согласно концепции, сохраняются в виде текстур на видеокарте для дальнейшего использования во фрагментном шейдере. В результате выполнения вышеописанных операций, на управляющем процессоре сосредотачивается вся необходимая информация для формирования представления с использованием объемного рендеринга.

Заключение

Возможность представления результатов моделирования при помощи технологии объемного рендеринга уже включена в последовательную версию постпроцессора ScientificView и доступна пользователям. В настоящее время автором ведутся работы по апробации механизма в параллельном режиме на реальных задачах пользователей.

Литература

1. Leven J., Corso J., Cohen J. Interactive visualization of unstructured grids using hierarchical 3D textures // IEEE 2002 Symposium on Volume Visualization and Graphics. − 2002. − Pp. 37–44.

2. Потехин А. Л., Логинов И. В., Лопаткин А. И., Ломтев В. В и др. Технология удаленной обработки результатов расчетов сверхбольшого объема // Журнал информационные технологии в проектировании и производстве. – 2017. № 2 (166). − С. 33–41.

3. Ричард С. Райт мл., Бенджамин Липчак. OpenGL. – М.: Суперкнига, 3-е издание. 2006.

IMPLEMENTATION OF THE SIMULATION RESULT REPRESENTATION MECHANISM USING VOLUME RENDERING IN PARALLEL MODE

V. V. Lomtev, V. V. Zhirnov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents the mechanism of representing simulation results with the use of the volume rendering procedure [1] in the parallel postprocessing system ScientificView [2] developed for viewing various physical processes and experimental data, which are represented as a mesh structure, or a set of points (particles, molecules, clusters).

Key words: ScientificView, volume rendering, parallel mode.

УДК 539.4
DOI 10.53403/9785951505071_2022_308

АЛГОРИТМ ДИНАМИЧЕСКОЙ БАЛАНСИРОВКИ НАГРУЗКИ ДЛЯ АДАПТИВНЫХ СЕТОК

Р. В. Муратов

Всероссийский научно-исследовательский институт автоматики им. Н. Л. Духова, Москва

Использование статичных равномерных сеток является нецелесообразным во многих практических задачах. В частности, это справедливо для многомасштабных задач, задач с расчетом кумулятивных процессов и других. Распространенным приемом в таких случаях является использование локально адаптивных сеток. Однако использование сеточной адаптации приводит к новым проблемам при реализации параллельных вычислений. Так, изменение числа ячеек и топологии сетки в течение расчета создает существенный дисбаланс в распределении вычислительной нагрузки между процессами, что не позволяет проводить эффективный параллельный счет. В данной статье предлагается диффузный алгоритм динамической балансировки нагрузки, применимый к адаптивным полярным сеткам. Алгоритм тестируется на модельной задаче, которая имитирует реальные вычисления на адаптивной сетке с различной нагрузкой на ячейки. В серии численных экспериментов демонстрируется масштабируемость метода порядка 70 и 90% для различных модификаций предложенного алгоритма.

Ключевые слова: динамическая балансировка нагрузки, сеточная адаптация.

Введение

В связи с широким распространением высокопроизводительных параллельных вычислительных систем, все более актуальной становится проблема их эффективного использования. Одним из аспектов данной проблемы является эффективное планирование и распределение задач внутри распределенной вычислительной системы с целью оптимального использования вычислительных ресурсов и сокращения времени счета. На практике часто возникают ситуации, когда часть вычислительных ресурсов простаивает, в то время как часть ресурсов перегружена.

Для оптимального использования ресурсов требуется применять методы динамической балансировки нагрузки, которые непосредственно во время счета выравнивают нагрузку между вычислительными потоками. Часто проблемы дисбаланса нагрузки возникают из-за существенной неоднородности исполняемой задачи. В рамках масштабных вычислений с использованием сеточных методов, дисбаланс может быть обусловлен различием вычислительной нагрузки в различных ячейках, к примеру, такая ситуация наблюдается, если в различных ячейках используются коды с различной вычислительной сложностью. Также сильная неоднородность и дисбаланс возникают при сильном изменении топологии сетки в течение вычислений. Такая ситуация наблюдается при использовании сеточной адаптации [1], в этом случае динамическая балансировка нагрузки обязательна.

В данной статье рассматривается задача балансировки нагрузки на иерархической адаптивной сетке. Полная постановка задачи приводится в разделе 2, алгоритм балансировки рассматривается в 3 главе, глава 4 содержит результаты моделирования для тестовой задачи и оценки производительности для предложенного алгоритма балансировки.

Постановка задачи

Рассмотрим двумерную неструктурированную конформную сетку, состоящую из четырехугольных элементов и заполняющую вычислительную область. Под конформной сеткой в данном случае подразумевается сетка, в которой смежные ячейки имеют общую грань, т. е. отсутствуют «висящие» вершины. Данную сетку будем называть базовой, а составляющие ее ячейки — базовыми ячейками. В течение расчета ячейки могут разбиваться на четыре составные части, которые называются дочерними ячейками. Также допускается обратная операция, при которой четыре ячейки, имеющие общую родительскую ячейку, сливаются в одну. Операции адаптации (огрубление и разбиение) могут выполняться как на стадии инициализации начальных данных, так и в ходе расчета. Сетка после проведения процедуры адаптации уже не является конформной, она имеет иерархическую структуру, которую можно описать как лес двумерных деревьев.

Единственное ограничение, которое накладывается на сетку при выполнении адаптации, — необходимость поддерживать баланс 1:2 между соседними ячейками. Другими словами, уровни адаптации смежных ячеек не могут отличаться более чем на единицу. На рис. 1 показаны примеры некорректной (слева) и правильной (справа) сеток. Серым кругом на рис. 1. обозначена область, которая требует максимальной адаптации, а красным пунктиром на некорректной сетке обозначены грани ячеек, на которых не соблюдается баланс 1:2. Ячейки сетки максимального уровня называются листовыми ячейками, они и образуют расчетную область задачи.

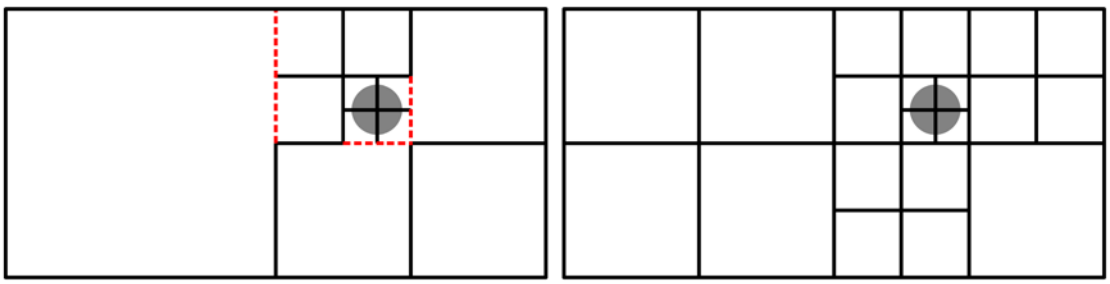


Рис. 1. Примеры правильной (справа) и некорректной (слева) сеток

Для использования возможностей распределенных вычислительных систем, необходимо разделить полученную сетку на несколько частей и распределить эти части между вычислительными потоками. Декомпозиция сетки может быть выполнена один раз в начале вычислений, если проводится расчет на статичной сетке, но такой подход не подходит при работе с адаптивными сетками, поскольку операции добавления и удаления ячеек могут создать существенный дисбаланс нагрузки между процессами. Таким образом, мы приходим к необходимости динамического изменения декомпозиции области в ходе расчета, то есть к необходимости динамической балансировки нагрузки в задачах на адаптивных сетках.

Методы декомпозиции можно разделить на два класса: глобальные и диффузные. Глобальные методы декомпозиции основываются на полной структуре сетки. Одними из лучших методов являются глобальные алгоритмы, основанные на декомпозиции графа [2]. Данные методы могут не только разделить сетку на части с равным числом ячеек, но также минимизировать длину границ между различными подобластями, что позволяет снизить время на коммуникации между процессами. Другим популярным глобальным методом является рекурсивная координатная бисекция [3], для работы алгоритма необходимо знать только центры ячеек, а связность ячеек не играет роли. Глобальные методы декомпозиции хорошо работают для решения задачи статической декомпозиции сетки, но имеют существенные недостатки применительно к задаче динамической балансировки нагрузки. Основной недостаток заключается в том, что глобальные алгоритмы начинают строить декомпозицию каждый раз «с нуля», из-за чего они являются довольно медленными. Также в случаях, когда топология сетки интенсивно меняется в течение расчета, может потребоваться большое число обменных операций: не исключаются случаи, когда в результате новой декомпозиции все ячейки

с одного процесса перемещаются на другой. Также глобальная балансировка не способна учесть сфокусированную нагрузку, что приводит к осцилляциям границ декомпозиции вокруг сфокусированной нагрузки.

Озвученные проблемы не актуальны для диффузных методов балансировки нагрузки. Диффузные алгоритмы основаны на пересылке части нагрузки между процессами в направлении, противоположном градиенту нагрузки (что имитирует закон теплопроводности Фурье) [4]. Применительно к сеточным методам это означает пересылку небольшой доли ячеек между соседними процессами в ходе проведения операции балансировки. Таким образом, декомпозиция не сильно изменяется на этапе балансировки и требуется незначительный обмен ячейками между процессами. Диффузные алгоритмы также позволяют найти хорошо сбалансированную декомпозицию, хотя для этого требуется некоторое число итераций. Интересный вариант диффузного алгоритма рассматривается в статье [5], где в расчетах методом SPH для декомпозиции области используется диаграмма Воронного. Однако в наших приложениях диаграммы Воронного не показали таких хороших результатов, поэтому в данной работе мы предлагаем вариант диффузного алгоритма, который основывается на рекурсивной координатной бисекции.

Описание алгоритма балансировки

Опишем алгоритм на примере двумерной расчетной области, которая имеет форму части кольца

$$K = \{(r, \varphi): R_{in} \le r \le R_{out}, 0 \le \varphi < \alpha\},$$

где внутренний радиус R_{in} может обращаться в ноль. Предположим, что число процессов P является произведением двух чисел $P = P_\varphi \times P_r$. Для разбиения расчетной области мы будем использовать двумерную декомпозицию: P_φ будет числом процессов вдоль угла, а P_r будет числом процессов вдоль радиального направления.

Первоначально область разбивается радиальными линиями под углами Φ_i, где $i = \overline{i, P_\varphi}$ на P_φ частей. Следующим шагом каждый из полученных секторов разбивается на P_r частей дугами, которые имеют радиусы $R_{i,j}$ ($i = \overline{1, P_\varphi}, j = \overline{0, P_r}$). Важно отметить, что набор дуг может отличаться для каждого сектора. Пример подобной декомпозиции приведен на рис. 2. В ходе балансировки радиальные линии и дуги окружностей могут смещаться, что показано стрелками на рисунке.

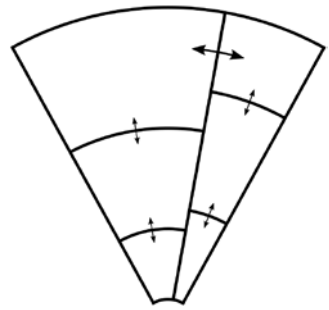


Рис. 2. Декомпозиция на шесть областей

Граничные значения всегда соответствуют границам области

$$\Phi_0 = 0,\ \Phi_{P_\varphi} = \alpha,\ R_{i,0} = R_{in},\ R_{i,P_r} = R_{out},\ i = \overline{1, P_\varphi}.$$

Таким образом, каждый процесс имеет по два индекса $i = \overline{1, P_\varphi}$, $j = \overline{0, P_r}$. Начальное разбиение не имеет значения, поскольку за несколько итераций балансировки алгоритм приводит к сбалансированной декомпозиции, когда число ячеек на каждом процессе примерно равно. Мы рассмотрим также две опции распределения ячеек по процессам. В первом случае предполагается, что листовая ячейка принадлежит некоторому процессу (i, j), если центр соответствующей базовой ячейки расположен внутри области (i, j). Согласно второму методу, мы относим листовую ячейку к процессу (i, j), если она лежит непосредственно в области. Первый способ гарантирует, что все листовые ячейки, имеющие одного родителя, будут расположены на одном процессе. Первый способ будем называть декомпозицией по базовым ячейкам, второй способ – декомпозицией по листовым ячейкам.

Теперь перейдем непосредственно к рассмотрению алгоритма динамической балансировки. Пусть на каждом процессе (i, j) определена нагрузка $w_{i,j}$. Нагрузка может определяться как число

ячеек на процессе, как среднее процессорное время расчета за последние K итераций или другими способами. Далее введем нагрузку на i-ый сектор:

$$W_i = \sum_{j=1}^{P_r} w_{i,j}\,, \qquad i = \overline{1, P_\varphi}.$$

В ходе балансировки сначала рассчитывается смещение радиальных линий по формуле

$$\Delta\Phi_i = D_i^\varphi v_i^\varphi, \qquad v_i^\varphi = \frac{W_{i+1} - W_i}{W_{i+1} + W_i}, \qquad i = \overline{1, P_\varphi - 1}.$$

где v^φ характеризует дисбаланс между смежными секторами и принимает значения от –1 до 1, а параметр D^φ ограничивает движение радиальных линий. В качестве D^φ используется значение в одну десятую ширины смежных секторов

$$D_i^\varphi = 0.1 \max(\Phi_{i+1} - \Phi_i, \Phi_i - \Phi_{i-1})\,, \qquad i = \overline{1, P_\varphi - 1}.$$

Аналогичные формулы используются для смещений дуг внутри секторов

$$\Delta R_{i,j} = D_{i,j}^r v_{i,j}^r, \qquad v_{i,j}^r = \frac{w_{i,j+1} - w_{i,j}}{w_{i,j+1} + w_{i,j}}, \qquad i = \overline{1, P_\varphi}, \qquad j = \overline{1, P_r - 1},$$
$$D_{i,j}^r = 0.1 \max\left(R_{i,j+1} - R_{i,j}, R_{i,j} - R_{i,j-1}\right), \qquad i = \overline{1, P_\varphi}, \qquad j = \overline{1, P_r - 1}.$$

На последнем шаге обновляются значения

$$\Phi_i := \Phi_i + \Delta\Phi_i, \qquad R_{i,j} := R_{i,j} + \Delta R_{i,j}.$$

Аналогичным образом можно производить балансировку в другом порядке: сначала сдвигать дуги окружностей, а затем радиальные линии независимо в каждом кольце. Также алгоритм применим к двумерным прямоугольным областям, в этом случае разбиение производится вдоль координатных осей. Более того, алгоритм может применяться к произвольным областям, для этого необходимо вписать расчетную область в прямоугольник. Также для сложных областей можно выбирать другую систему ортогональных направлений.

Одним из недостатков предложенного алгоритма является необходимость подбора параметров, для которых выполняется соотношение $P = P_\varphi \times P_r$. Однако, технику балансировки можно легко модифицировать и обобщить для возможности разбивать область на произвольное число процессов. Предположим, что помимо общего числа процессов P задается также число столбцов P_x, если значения не кратны, тогда каждый столбец разбивается можно разбить на $\lfloor P/P_x \rfloor$ или $\lfloor P/P_x \rfloor + 1$ частей таким образом, чтобы общее число областей равнялось P (здесь $\lfloor . \rfloor$ означает целую часть).

Пример разбиения прямоугольной сетки на 7 процессов в три столбца приведен на рис. 3. Естественно, теперь необходимо модифицировать формулы для смещения границ столбцов. Изменения коснутся только формулы для расчета нагрузки на столбец.
Новая формула

$$W_i = \frac{1}{P_{y,i}} \sum_{j=1}^{P_{y,i}} w_{i,j}\,, \qquad i = \overline{1, P_x}$$

учитывает также число процессов $P_{y,i}$ в i-ом столбце. Если в каждом столбце содержится одинаковое число процессов, тогда не имеет значения, какую формулу использовать: итоговая формула для смещений одинакова, поскольку множитель $P_{y,i}$ сокращается при вычислении относительного дисбаланса.

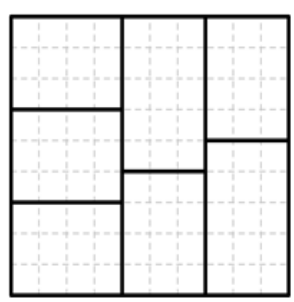


Рис. 3. Разбиение прямоугольной области на 7 процессов

Тестирование алгоритма и производительность

Данный алгоритм балансировки тестируется на модельной задаче. Адаптация и нагрузка вводятся искусственным образом, но за основу взята реальная задача. Базовая сетка представляет собой полярную сетку в полукольце $R_{in} = 10, R_{out} = 10, 0 \le \varphi \le \pi$ с ячейками, приближенными к квадратам, то есть *aspect ratio* ячеек около единицы. Узлы сетки задаются по следующим формулам

$$\varphi_i = \frac{i\alpha}{N_\varphi},\ i = \overline{0, N_\varphi},\ r_j = R_{in}\left(1 + \frac{\alpha}{N_\varphi}\right)^j,\ j = \overline{0, N_r}.$$

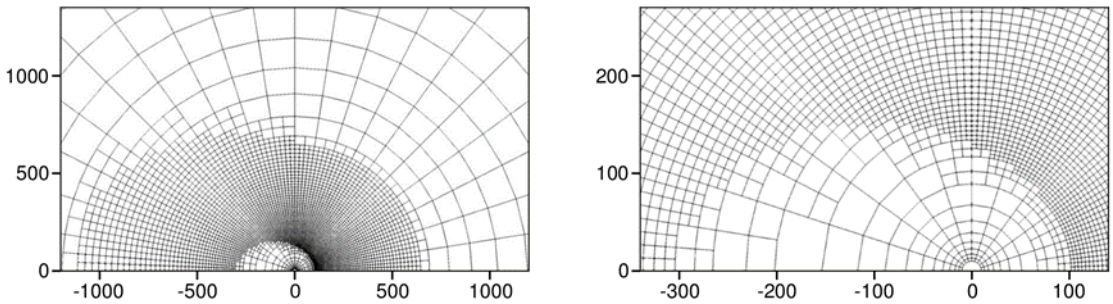


Рис. 4. Центральная часть сетки, разделенная на три области с различными уровнями адаптации. На правом рисунке показано приближение около центра координат

Сторона ячеек у внешней границы примерно в 10^3 раз больше, чем у ячеек в центре области (во столько раз различаются радиусы внутренней и внешней частей). Начальное разрешение по углу $N_\varphi^{base} = 80$, разрешение в радиальном направлении $N_r^{base} = 180$. Предполагается, что расчетная область разделена на три части двумя эллипсами:

$$\left(\frac{x+100}{200}\right)^2 + \left(\frac{y}{150}\right)^2 = 1,\ \left(\frac{x+250}{900}\right)^2 + \left(\frac{y}{700}\right)^2 = 1.$$

Внутренняя часть не адаптируется, внешняя часть имеет первый уровень адаптации, а промежуточный слой имеет третий уровень адаптации. Пример сетки с аналогичным распределением уровней приведен на рис. 4.

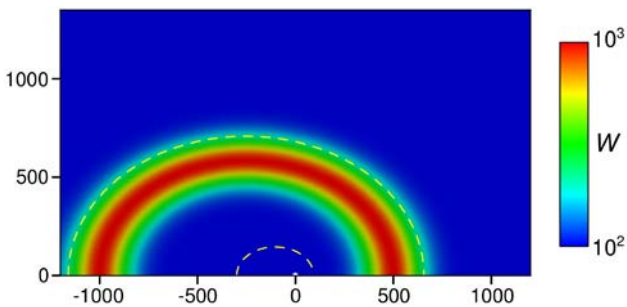


Рис. 5. Распределение нагрузки W в области. Желтой штриховой линией показаны границы области с максимальным уровнем адаптации

После процедуры адаптации сетка содержит около 244 тысяч ячеек, из которых 6 тысяч во внутренней части, 21 тысяча во внешней части, и 217 тысяч или 89 % всех ячеек расположены в промежуточной части с хорошим разрешением. Помимо существенно неоднородной сетки, мы добавляем также искусственную нагрузку к каждой ячейке. Предполагается, что кольцевой слой ячеек, которые лежат на внешней границе кольца с максимальным разрешением, имеет нагрузку в десять раз превышающую нагрузку на остальные ячейки. Используется распределение нагрузки на ячейки, заданное формулой

$$W(x,y) = 10^2 + 10^3 \exp\left(-25\left(\sqrt{\left(\frac{x+250}{750}\right)^2 + \left(\frac{y}{580}\right)^2 - 1}\right)\right).$$

Для моделирования нагрузки каждая ячейка считает функцию синуса в цикле от нуля до целой части $W(x,y)$, где (x,y) являются координатами центра ячейки. Распределение нагрузки показано цветом на рис. 5.

Предполагается, что моделируемый физический процесс протекает достаточно медленно, поэтому значительные изменения топологии сетки и распределения нагрузки занимают множество временных шагов. Таким образом, мы можем использовать статическую сетку и статическую нагрузку для тестирования нашего алгоритма. Нашей целью является показать, что алгоритм позволяет достигнуть устойчивой декомпозиции сетки на процессы за конечное число итераций балансировки, а также показать, что полученное разбиение является хорошо сбалансированным.

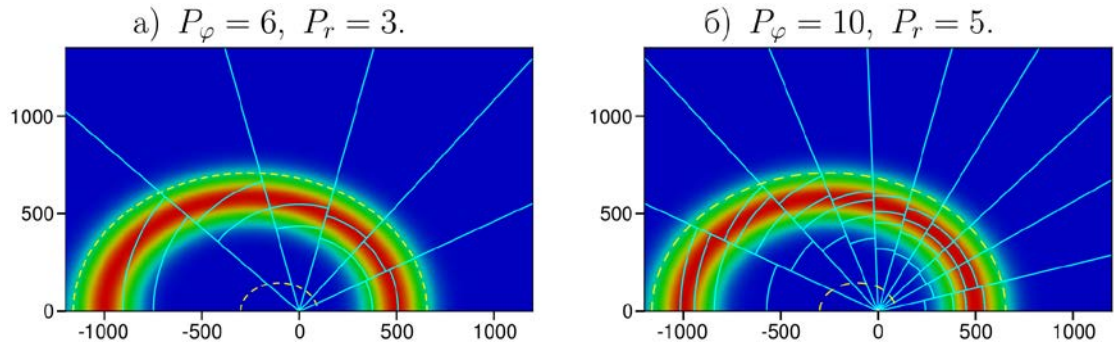


Рис. 6. Равновесное разбиение для двух наборов параметров P_φ и P_r

В начальный момент устанавливается равномерное распределение радиальных линий и дуг

$$\Phi_i = \frac{i\alpha}{P_\varphi},\ R_{i,j} = R_{in} + \frac{j(R_{out}-R_{in})}{N_r}.$$

После этого начинаются итерации балансировки нагрузки. Сначала необходимо уравновесить число ячеек на каждом процессе, для этого алгоритм балансировки нагрузки применяется сотню раз с использованием числа ячеек в качестве нагрузки. После этого начинается модельный расчет, в котором используется искусственная нагрузка. На каждом временном шаге процессы обмениваются фиктивными данными граничных ячеек для моделирования коммуникации между процессами, которая присутствует в реальных задачах. Данные вычисления выполняются на протяжении 10 тысяч итераций с применением алгоритма балансировки каждую двадцатую итерацию.

Десяти тысяч итераций более чем достаточно для получения устойчивого разбиения сетки по процессам. Следующие 10 тысяч шагов используются для сравнения результатов балансировки, по ним рассчитывается среднее процессорное время для выполнения одного шага по времени.

Было проведено две серии экспериментов с различными значениями параметров P_φ и P_r. Используемые параметры приведены в табл. 1. В первой серии экспериментов использовался метод разбиения по базовым ячейкам, в другой серии использовалось разбиение по листовым ячейкам. Устойчивые разбиения для $P_\varphi = 10$, $P_r = 5$ и для $P_\varphi = 6$, $P_r = 3$ показаны на рис. 6. Для других наборов параметров P_φ и P_r распределения аналогичные. Как и ранее, цветом на изображениях показана искусственная нагрузка от 10^2 до 10^3, а границы области с максимальным разрешением показаны желтыми штриховыми линиями. Сплошные цианновые линии обозначают границы между процессами. Представленные распределения рис. 6) соответствуют методу разбиения по базовым ячейкам, разбиение по листовым ячейкам дает визуально неотличимую конфигурацию. Однако такое небольшое различие сильно сказывается на производительности

Таблица 1

Параметры декомпозиции, используемые при тестировании

P	1	2	6	8	9	12	16	18	20	24	28	30	32	36	40	48	50
P_φ	1	2	3	4	3	4	4	6	5	8	7	6	8	9	8	8	10
P_r	1	1	2	2	3	3	4	3	4	3	4	5	4	4	5	6	5

Изменение числа вычислительных узлов при расчете фиксированной задачи означает, что тестируется сильная масштабируемость. Введем метрику для измерения ускорения:

$$S = \frac{T_1}{T_P},$$

где T_P является временем осуществления 10 тысяч итераций на P вычислительных узлах. В общем случае, значение T_P может варьироваться при одном P и различных параметрах декомпозиции P_φ и P_r. Здесь мы пренебрегаем этим и используем исключительно параметры, представленные в табл. 1. Производительность двух методов показана на рис. 7. В обоих случаях масштабируемость близка к линейной. Как и ожидалось, производительность и масштабируемость метода с разбиением листовых ячеек лучше, чем с разбиением по базовым ячейкам. Среднее процессорное время одной итерации около 3 с для единичного узла, при использовании 50 вычислительных узлов среднее время одной итерации равно 0.085 и 0.066 с для метода разбиения по базовым ячейкам и по листовым соответственно. Таким образом, мы получаем масштабируемость методов на 50 узлов 71 и 91 %.

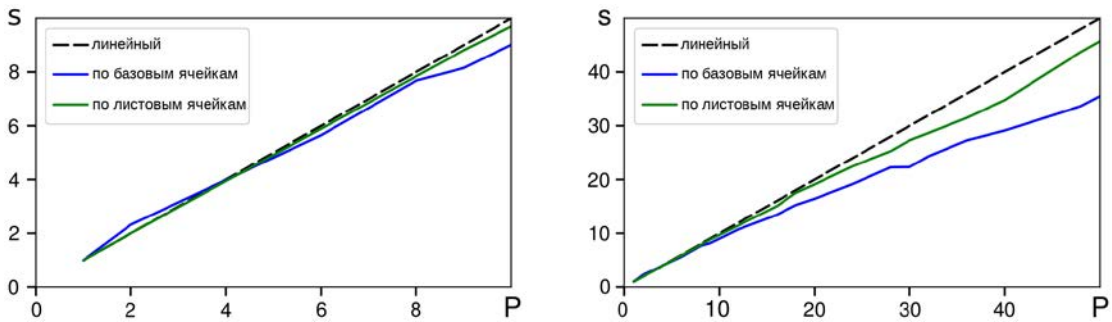


Рис. 7. Графики производительности для двух методов, а также линейный график для сравнения. Рисунок слева демонстрирует масштабируемость при использовании до 10 узлов, рисунок справа – до 50 узлов

Заключение

В статье предлагается метод динамической балансировки нагрузки, разработанный специально для приложений с высоким дисбалансом нагрузки, созданным использованием адаптивных сеток. Предложенный алгоритм показал свою эффективность на модельной задаче, включающей сильную неоднородность сетки и высокую неоднородность нагрузки на узлах. В двух сериях тестов производительности было показано, что на 50 вычислительных узлах достигается ускорение в 36 и 46 раз для различных опций балансировки.

Благодарности

Работа выполнена при финансовой поддержке гранта Российского научного фонда, проект № 21-71-00102.

Литература

1. Lan Z., Taylor V. E., Bryan G. Dynamic load balancing of SAMR applications on distributed systems // Scientific Programming. 2002. Vol. 10, № 2. P. 319–328.

2. Walshaw C., Cross M., Everett M. G. Parallel dynamic graph partitioning for adaptive unstructured meshes // Journal of Parallel and Distributed Computing. 1997. Vol. 47, № 2. P. 102–108.

3. Simon H. D., Teng S.-H. How good is recursive bisection? // SIAM Journal of Scientific Computing. 1997. Vol. 18, № 5. P. 1436–1445.

4. Cybenko G. Dynamic load balancing for distributed memory multiprocessors // Journal of Parallel and Distributed Computing. 1989. Vol. 7. P. 279–301.

5. Egorova M. S., Dyachkov S. A., Parshikov A. N., Zhakhovsky V. V. Parallel SPH modeling using dynamic domain decomposition and load balancing displacement of Voronoi subdomains // Computer Physics Communications. 2019. Vol. 234. P. 112–125.

DYNAMIC LOAD BALANCING ALGORITHM FOR ADAPTIVE MESHES

R. V. Muratov

Dukhov Automatics Research Institute, Moscow

Using of fixed uniform meshes may happen to be impractical for various tasks. For example, it takes place in multiscale problems, simulations of cumulative processes and many other problems. A common solution in these cases is to use an adaptive mesh refinement (AMR) method. However, using of AMR leads to new problem in parallel computation, changing the number of cells creates significant load imbalance between processes and does not allow to work parallel codes efficiently. In this paper we propose a diffusive dynamic load balancing algorithm for an adaptive polar mesh. We test the algorithm on a model problem that simulates calculations on dynamically changing mesh with varying computational work in cells. We consider two modifications of the proposed algorithm and obtain a scalability of application about 70 and 90 % respectively in series of numerical experiments with the model problem.

Key words: dynamic load balancing, adaptive mesh refinement.

УДК 004.4
DOI 10.53403/9785951505071_2022_316

АРХИТЕКТУРА МОДУЛЬНОЙ ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ ЛОГОС

А. Г. Надуев, А. Д. Черевань, Д. А. Кожаев

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Модульная интеграционная платформа (ЛОГОС-МИП) является программным комплексом для интеграции расчетных модулей умеющих моделировать задачи газодинамики, гидродинамики, термодинамики, механики твердого тела и т. д., с целью проведения связанного и сопряженного расчета задач мультифизики, параметрических и оптимизационных исследований.

В докладе представлена общая концепция ЛОГОС-МИП, назначения компонент ЛОГОС-МИП и принципы их взаимодействия, а так же уровни интеграции расчетных модулей в ЛОГОС-МИП.

Ключевые слова: ЛОГОС-МИП, ЛОГОС-Платформа, интеграция расчетных модулей, связанный расчет, сопряженный расчет.

Назначение и цели модульной интеграционной платформы ЛОГОС

Назначением модульной интеграционной платформы ЛОГОС («ЛОГОС-Платформа») является объединение расчетных модулей суперкомпьютерного моделирования и систем инженерного анализа в рамках пакета программ ЛОГОС, для подготовки и расчета комплексных мультидисциплинарных задач и проведения параметрических и оптимизационных исследований.

При проектировании «ЛОГОС-Платформа» разработчики ставили перед собой следующие цели:

- Обеспечение сопряженного и связанного мультифизичного моделирования на высокопараллельных вычислительных системах;
- Обеспечение возможности проведения параметрических и оптимизационных исследований;
- Создание унифицированной среды разработки для разработчиков ЛОГОС и разработчиков отраслевых и авторских расчетных модулей;
- Объединение расчетных модулей пакета программ ЛОГОС с авторскими и отраслевыми расчетными модулями в единой графической оболочке интегрированной платформы.

Состав модульной интеграционной платформы ЛОГОС

Для обеспечения поставленных целей в состав «ЛОГОС-Платформа» входят следующие компоненты, выполняющие определенные задачи:

- Интегратор – графический интерфейс конфигурирования модулей МИП;
- Обработчик сценария – программная среда выполнения расчетного задания (сценария);
- Модули обработчика сценария – модули, реализующие логику этапов расчетного сценария;
- Модуль сопряжения – модуль подключения механизмов МИП к основным расчетным модулям;
- Подключаемые расчетные модули – модули, подключаемые к основным расчетным модулям и расширяющие их функциональные возможности;

• Транспортный уровень – настраиваемая среда, обеспечивающая взаимодействие между основными расчетными модулями в процессе связанного расчета мультидисциплинарных задач;

• Обрабатывающие функции (ОФ) – компоненты для настройки транспортного уровня;

• Файловый контейнер обмена данными – модуль взаимодействия основных расчетных модулей в процессе поэтапного расчета мультидисциплинарных задач.

Уровни интеграции в модульную интеграционную платформу ЛОГОС

При необходимости внедрения сторонних расчетных модулей в «ЛОГОС-Платформа» следует учитывать возможность следующих уровней интеграции, которые наследуют друг от друга необходимые действия по мере повышения уровня:

1. Интеграция расчетного модуля в Интегратор – этот уровень не требует изменения исходного кода интегрируемого расчетного модуля. Должен быть подготовлен SPEC файл в формате Интегратора, в котором содержится информация об объектах проекта расчетного модуля и правилах их редактирования во внешних утилитах (например, САПР, препроцессоры и т. д.);

2. Использование механизма пользовательских функций (ПФ) – начиная с этого уровня интеграции, требуется изменение исходного кода интегрируемого расчетного модуля:

   a. Интеграция библиотеки модуля сопряжения;

   b. Реализация коннекторов ПФ в расчетном модуле;

   c. Реализация сервисной функции, предоставляющей доступ к коннекторам ПФ.

SPEC файл должен содержать информацию о коннекторах ПФ расчетного модуля. Отдельно, должны быть разработаны динамические библиотеки с ПФ и SPEC файлы к ним;

3. Взаимодействие расчетного модуля с подключаемыми расчетными модулями – в расчетном модуле должны быть реализованы:

   a. Точки передачи управления модулю сопряжения;

   b. Сервисные функции, предоставляющие доступ к данным расчетного модуля.

SPEC файл должен содержать информацию о точках передачи управления и сервисных функциях. Отдельно, должны быть разработаны динамические библиотеки с подключаемыми расчетными модулями и SPEC файлы к ним;

4. Взаимодействие расчетного модуля в связанном расчете мультидисциплинарной задачи с использованием транспортного уровня – должны быть разработаны подключаемый расчетный модуль осуществляющий взаимодействие с транспортным уровнем и динамические библиотеки с ОФ, а так же и SPEC файлы к ним.

Схема взаимодействия компонент «ЛОГОС-Платформа» при всех уровнях интеграции представлена на рис. 1.

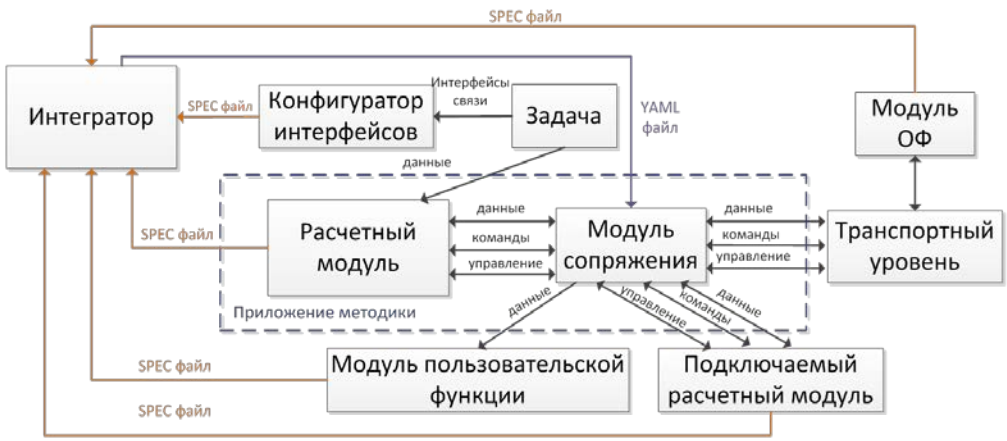


Рис. 1. Схема взаимодействия компонент «ЛОГОС-Платформа» при всех уровнях интеграции

Интегратор модульной интеграционной платформы ЛОГОС

Интегратор это приложение, предоставляющее графический интерфейс пользователя и обеспечивающее:

- Динамическую интеграцию расчетных модулей с использованием SPEC файлов;
- Подготовку задач расчетных модулей с возможностью вызова соответствующих САПР, препроцессоров, конвертеров как внешних утилит;
- Настройку механизма ПФ отраслевых методик, для использования ПФ при расчете задач;
- Объединение задач расчетных модулей в связанную мультидисциплинарную задачу с выбором интерфейсов взаимодействия и возможностью их настройки с использованием схемного редактора;
- Подготовку сценариев расчета задач оптимизационных и параметрических исследований с использованием схемного редактора;
- Запуск обработчика сценария для выполнения подготовленных сценариев с контролем хода расчета сценария;
- Анализ полученных результатов с возможностью вызова постпроцессоров как внешних утилит.

Пример графического интерфейса Интегратора во время расчета связанной задачи на локальном компьютере представлен на рис. 2.

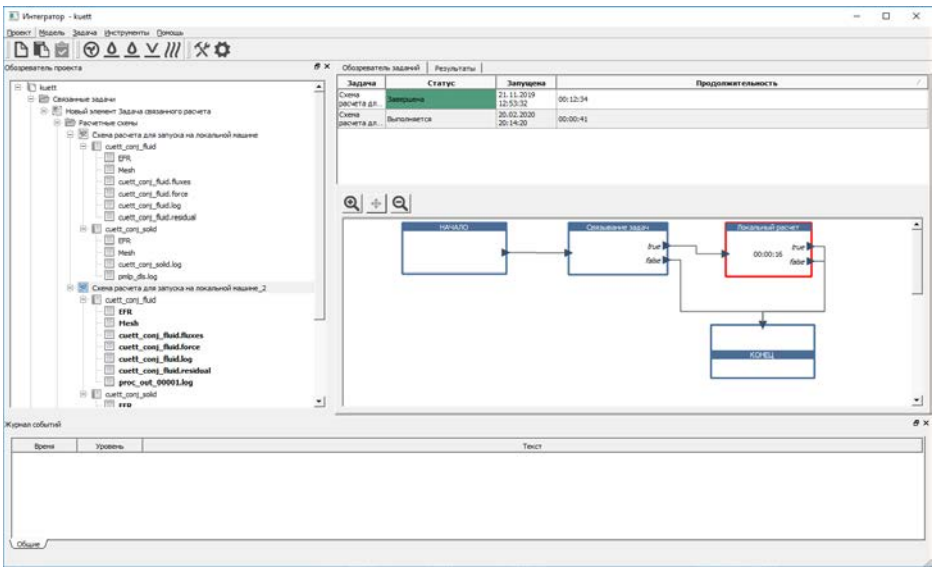


Рис. 2. Пример графического интерфейса Интегратора во время расчета связанной задачи

Обработчик сценария модульной интеграционной платформы ЛОГОС и его модули

Обработчик сценария модульной интеграционной платформы ЛОГОС и его модули предназначены для выполнения расчетных заданий (сценариев) подготовленных в схемном редакторе Интегратора.

В процессе исполнения расчетного задания обработчик сценария производит передачу управления своим модулям в соответствии с логикой сценария, а те, в свою очередь, реализуют логику отдельных этапов, вызывая для этого внешние программы и используя возможности языка программирования Python [1];

Интегратор в процессе расчета сценария обработчиком, взаимодействует с его БД и журналом событий, отображает ход расчета, предоставляет пользователю интерфейс для оценки результатов расчета и возможность досрочной остановки расчета сценария.

Пример сценария одного шага параметрического исследования, подготовленный в схемном редакторе Интегратора представлен на рис. 3.

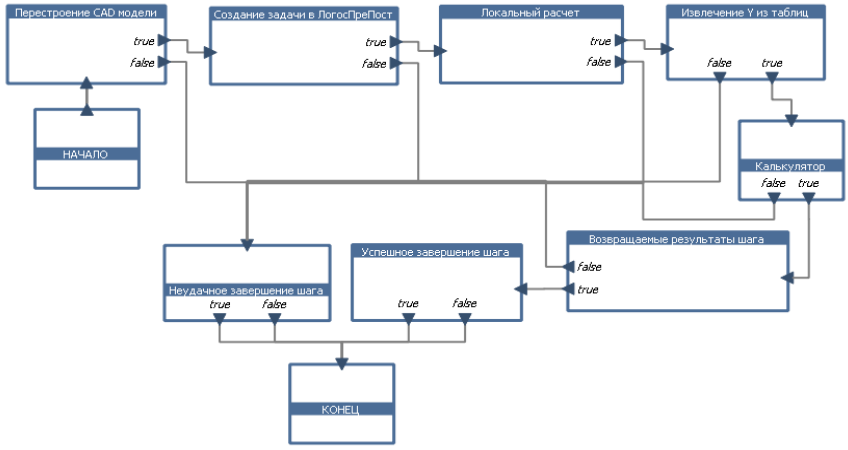


Рис. 3. Пример сценария одного шага параметрического исследования, подготовленный в схемном редакторе Интегратора

Модуль сопряжения модульной интеграционной платформы ЛОГОС

Модуль сопряжения является статической библиотекой, которая должна быть интегрирована в расчетный модуль, для обеспечения взаимодействия с другими компонентами ЛОГОС-Платформа.

Расчетный модуль взаимодействует с модулем сопряжения путем вызова его функций с целью выполнения следующих задач:

- Инициализация модуля сопряжения, во время которой происходит: считывание конфигурационного файла, загрузка динамических библиотек подключаемых расчетных модулей и ПФ и настройка механизмов их вызова;
- Создание локального MPI [2] коммуникатора для обменов данными между MPI процессами расчетного модуля вне обменов через транспортный уровень;
- Передача управления модулю сопряжения в определенных точках выполнения расчета задачи («начало расчета», «конец расчета», «начало модельного шага», «конец модельного шага» и т. д.). После которой модуль передачи делегирует управление подключаемым расчетным модулям, зарегистрированным в этой точке, а те в свою очередь рассчитывают свою часть задачи, используя сервисные функции расчетного модуля, для обмена данными с его внутренними структурами.

Схема взаимодействия расчетного модуля с компонентами «ЛОГОС-Платформа» через модуль сопряжения представлена на рис. 4.

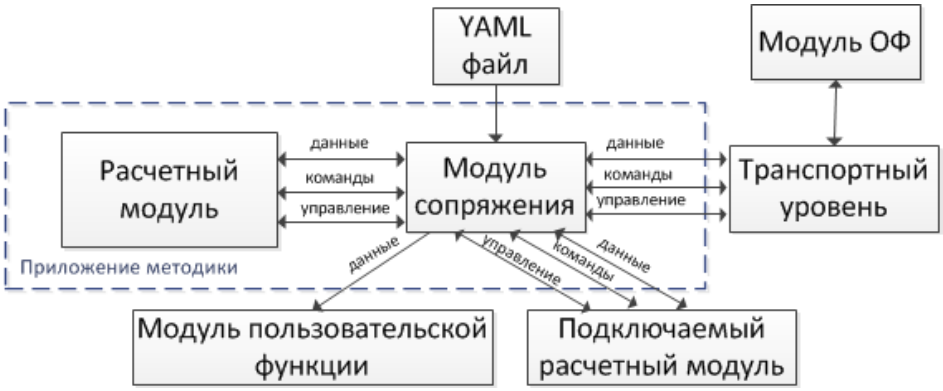


Рис. 4. Схема взаимодействия расчетного модуля с компонентами «ЛОГОС-Платформа» через модуль сопряжения

Подключаемые расчетные модули модульной интеграционной платформы ЛОГОС и его модули

Подключаемые расчетные модули расширяют функциональные возможности расчетных модулей, при этом разработка подключаемых расчетных модулей может осуществляться независимыми командами разработчиков, так как подключаемые расчетные модули являются автономными компонентами, а взаимодействие между расчетным модулем и подключаемыми расчетными модулями ведется посредством сервисных функций, предоставляющих доступ к данным расчетного модуля.

Набор подключаемых расчетных модулей и стратегия их подключения настраиваются в Интеграторе и могут меняться в зависимости от решаемой задачи.

Схема взаимодействия расчетного модуля с подключаемыми расчетными модулями через модуль сопряжения представлена на рис. 5.

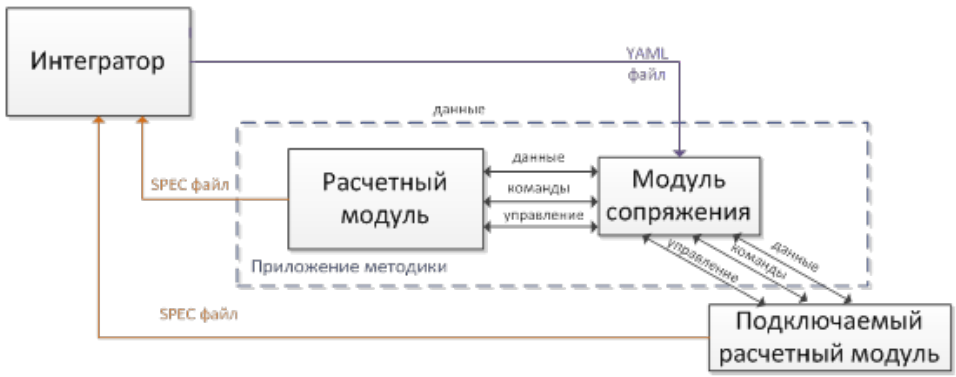


Рис. 5. Схема взаимодействия расчетного модуля с подключаемыми расчетными модулями через модуль сопряжения

Транспортный уровень и обрабатывающие функции интеграционной платформы ЛОГОС

Транспортный уровень является программной средой, обеспечивающей взаимодействие между основными расчетными модулями в процессе связанного расчета мультидисциплинарных задач.

Интерфейсы транспортного уровня, динамически настраиваемые с помощью обрабатывающих функций (ОФ) и при необходимости подключения новых расчетных модулей пользователи в схемном редакторе Интегратора должны произвести настройку их взаимодействий с использованием существующих или вновь разработанных ОФ (среди которых, например, могут быть конверторы, интерполяторы и т. д.).

Взаимодействие между расчетными модулями с помощью транспортного уровня может осуществляться с использованием протоколов MPI или TCP/IP [3], т е. расчетные модули могут рассчитывать мультидисциплинарную задачу, как на общем расчетном поле супер-ЭВМ, так и с подключением к такому расчету расчетных модулей работающих на других компьютерах, под другими операционными системами. При этом взаимодействие может осуществляться с теоретически неограниченным количеством расчетных модулей по неограниченному количеству интерфейсов.

Схема транспортного уровня для взаимодействия расчетного модуля с неограниченным количеством расчетных модулей по неограниченному количеству интерфейсов представлена на рис. 6.

Схема взаимодействия расчетных модулей по интерфейсу транспортного уровня с использованием ОФ представлена на рис. 7.

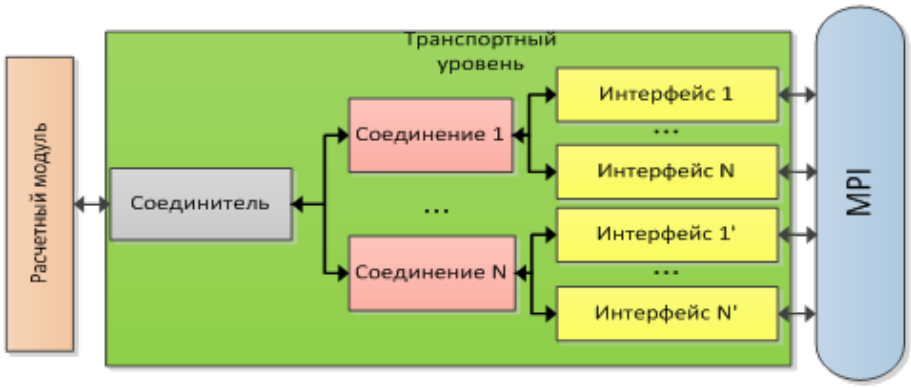


Рис. 6. Схема транспортного уровня для взаимодействия расчетного модуля с неограниченным количеством расчетных модулей по неограниченному количеству интерфейсов

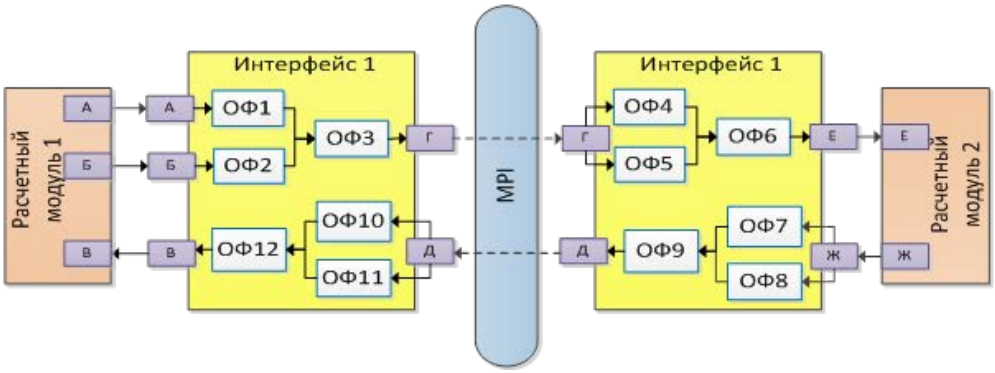


Рис. 7. Схема взаимодействия расчетных модулей по интерфейсу транспортного уровня с использованием ОФ

Файловый контейнер обмена данными интеграционной платформы ЛОГОС

Файловый контейнер обмена данными - модуль взаимодействия расчетных модулей в процессе поэтапного расчета мультидисциплинарных задач, базирующийся на возможностях библиотеки ЕФР;

Библиотека ЕФР предоставляет:

- Единый функциональный интерфейс для коллективной работы с файловыми данными – данные расчетной математической сетки, распределенные между счетными процессорами, могут сохраняться в общий файловый контейнер, при этом загрузка данных возможна на любое количество счетных процессоров;
- Единый формат представления расчетных данных – формат является абстрактным, что позволяет подстраивать его для использования различными расчетными модулями и продолжать счет в рамках одного расчетного модуля или передачу данных в другой расчетный модуль.

Для реализации механизма поэтапного расчета мультидисциплинарных задач используются два подключаемых модуля:

- Сериализатор – извлекает с помощью сервисных функций расчетного модуля данные, необходимые для передачи и сохраняет их в файловом контейнере ЕФР;
- Десериализатор – восстанавливает данные, сохраненные в файловом контейнере ЕФР, интерполирует их на математическую сетку и передает их в сервисные функции расчетного модуля.

Схема использования файлового контейнера для обмена данными в процессе поэтапного расчета мультидисциплинарных задач представлена на рис. 8.

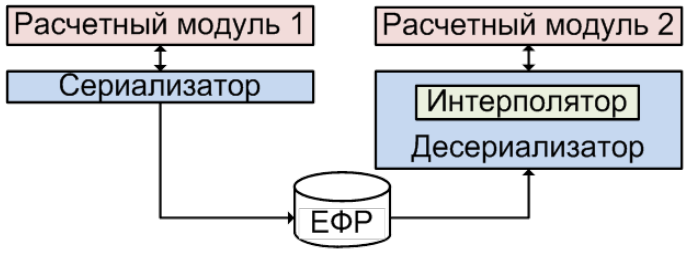


Рис. 8. Схема использования файлового контейнера для обмена данными
в процессе поэтапного расчета мультидисциплинарных задач

Литература

1. Язык программирования Python [Электронный ресурс]: Python Software Foundation. – Virginia, USA, [2001-2017] – Режим доступа: https://docs.python.org/3/index.html.
2. MPI Forum [Электронный ресурс]. Режим доступа http://www.mpi-forum.org.
3. Internet Standards [Электронный ресурс]. Режим доступа https://ietf.org/standards.

ARCHITECTURE OF THE “LOGOS” MODULAR INTEGRATION PLATFORM

A. G. Naduev, A. D. Cherevan, D. A. Kozhaev

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The Modular Integration Platform LOGOS-MIP is a set of programs integrating gas dynamics, hydrodynamics, thermodynamics, solid mechanics, and other computational modules to perform coupled and conjugate multiphysics simulations and parametric and optimization analysis.

The paper presents a general concept of LOGOS-MIP, functionality of its components and principles of their interaction, and levels of integration of the computational modules in LOGOS-MIP.

Key words: LOGOS-MIP, Logos Platform, integration of computational modules, coupled simulation, conjugate simulation.

УДК 519.6
DOI 10.53403/9785951505071_2022_323

ФУНКЦИОНАЛЬНЫЙ БЛОК ГЕНЕРАЦИИ ОБЪЕМНЫХ НЕСТРУКТУРИРОВАННЫХ СЕТОК НА ОСНОВЕ РЕГУЛЯРНОГО ДРОБНО-АДАПТИВНОГО ШАБЛОНА В МЕТОДИКЕ ТИМ

В. А. Никитин

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе приведено описание функционального блока, позволяющего генерировать объемную неструктурированную сетку на основе регулярного дробно-адаптивного шаблона. Модуль может использоваться как на этапе подготовки задачи к расчету, так и во время счета задачи для перестроения счетной сетки, динамически адаптированной под решение задачи. Приведено описание алгоритма работы генератора. Также приведен результат внедрения модуля в методику ТИМ.

Ключевые слова: генератор объемной сетки, неструктурированные сетки, дробно-адаптивный шаблон, методика ТИМ.

При математическом моделировании физических процессов методом конечных разностей генераторы сеток играют ключевую роль. Они используются как для подготовки сетки для задачи на старте, так и в ходе проведения расчета, перестраивая искаженную сетку, на которой дальнейший счет невозможен либо ее использование приводит к возрастанию времени проведения расчета.

Методика ТИМ[1] предназначена для решения многомерных нестационарных задач механики сплошной среды на неструктурированных подвижных сетках произвольного вида. По этой методике проводятся сложные расчеты в трехмерной постановке в лагранжевых переменных, связанных с вихревыми и струйными течениями. Такие расчеты приводят к большим деформациям вещества, что в свою очередь искажает сетку в трех направлениях (рис. 1).

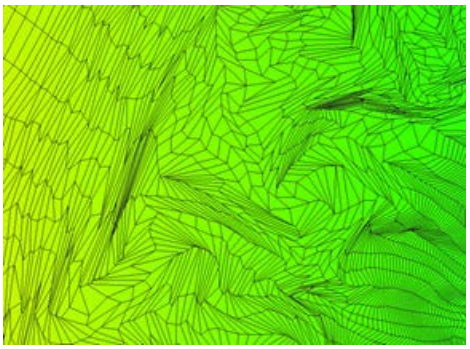


Рис. 1. Искаженная сетка

Для корректировки сетки на протяжении всего расчета в методике ТИМ реализован широкий набор алгоритмов, которые позволяют проводить расчеты с автоматическим контролем состояния и поддержания качества сетки. Но все они работают локально и при большом количестве исправляемых элементов заметно увеличивают время счета задачи. В таких случаях целесообразнее построить новую сетку со сгущением в локальных зонах (там, где это необходимо для корректного описания решения) и выполнить на нее переинтерполяцию физических величин.

Анализ существующих в математическом отделении РФЯЦ-ВНИИЭФ генераторов неструктурированных сеток, использующих аналогичный подход [2–3], показал, что для внедрения их в методику ТИМ необходимо провести работу по отчуждению функционала от программ, в которые эти генераторы внедрены. Оценка работ по отчуждению оказалась сравнимой с реализацией генератора с нуля. Так же нет уверенности, что работу имеющихся генераторов возможно реализовать в параллельном режиме с использованием интерфейса MPI.

Для решения описанной проблемы для методики ТИМ создан модуль, позволяющий генерировать объемную неструктурированную сетку, как в ручном режиме с заданием требуемых параметров при подготовке сетки для проведения расчета, так и в автоматическом, когда подготовка параметров генерации сетки проводится без участия пользователя на основе данных об исходной сетке.

Модуль генерации объемной неструктурированной сетки на основе регулярного дробно-адаптивного шаблона позволяет создавать сетку, состоящую из многогранных ячеек, имеющих форму параллелепипедов. Каждая из шести сторон ячейки может быть представлена одной или четырьмя гранями, в зависимости от условий формирования соседних ячеек. Области сгущения сетки задаются с помощью набора контрольных объемов или на основе данных об исходной сетке. Контрольный объем представляет собой геометрическую фигуру в пространстве, внутри которой указан требуемый размер, предъявляемый к генерируемым ячейкам. Реализовано шесть типов контрольных объемов:

• параллелепипед (рис. 2);

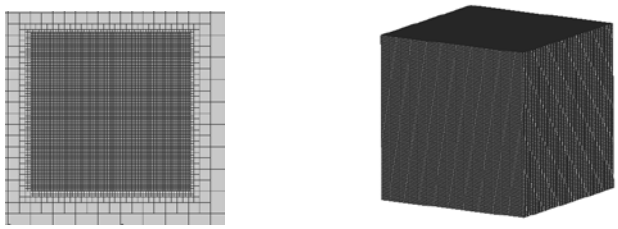


Рис. 2. Построение объема в виде параллелепипеда

• цилиндр (рис. 3);

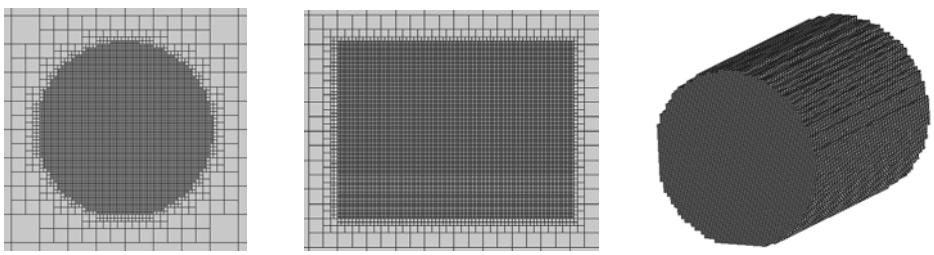


Рис. 3. Построение объема в виде цилиндра

• усеченный конус (рис. 4);

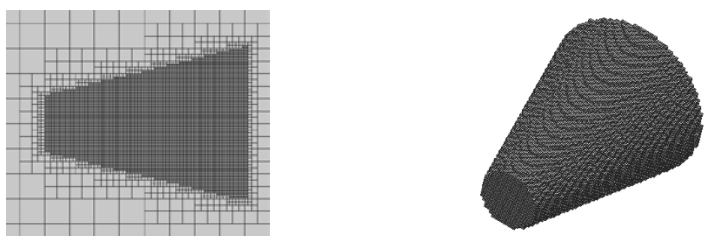


Рис. 4. Построение объема в виде конуса

• сфера (рис. 5);

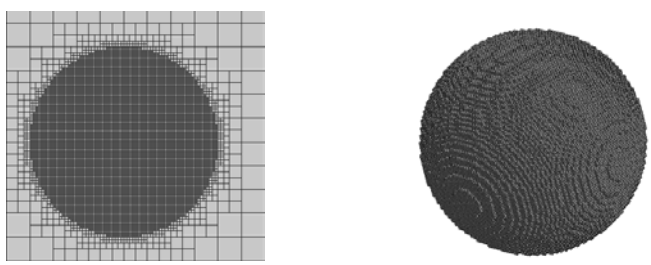


Рис. 5. Построение объема в виде сферы

• труба (рис. 6);

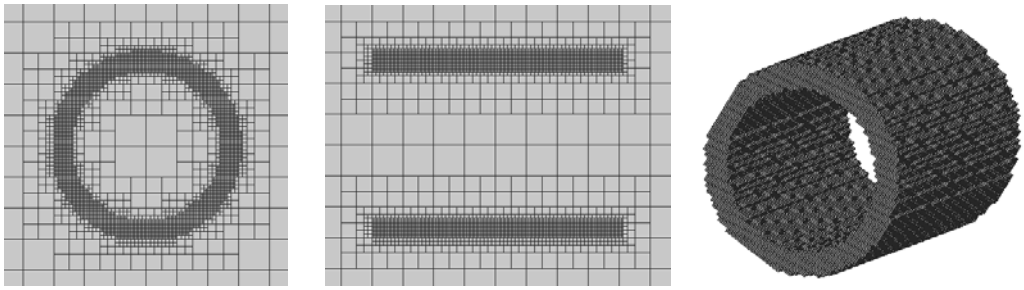


Рис. 6. Построение объема в виде трубы

• комбинированный контрольный объем (рис. 7).

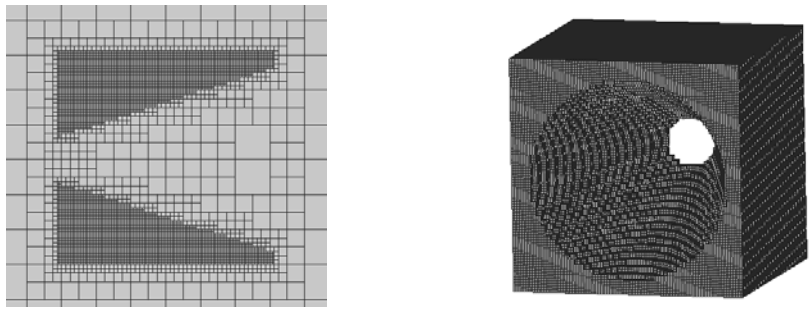


Рис. 7. Построение объема в виде комбинации объемов

При генерации сетки допускается любое количество и сочетание контрольных объемов (рис. 8).

Возможно пересечение и наложение контрольных объемов. В этом случае в пространстве пересечения алгоритм работы модуля будет стремиться к созданию ячеек с наименьшим требуемым размером из пересекаемых контрольных объемов.

Для работы модуля необходимо задать следующую информацию:

1) Габарит, в котором необходимо построить сетку. Габарит указывается заданием минимальной и максимальной точки пространства.

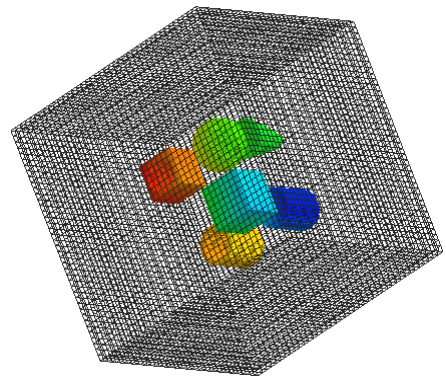


Рис. 8. Пример построения различных контрольных объемов

2) Базовый размер ячейки – требуемая длина ребра ячейки первоначальной сетки. То есть, сетки с которой начинается работа модуля.

3) Номера граничных условий – номера, которые заносятся в информацию о гранях, лежащих на внешних поверхностях генерируемой сетки.

4) Набор контрольных объемов, которые задают область пространства со своей геометрией и требуемой в этой области размером ячеек:

• Параллелепипед – задается как габарит, т. е. минимальной и максимальной точками пространства.

• Цилиндр – задается двумя точками пространства, формирующих ось цилиндра, и радиусом основания.

• Конус – задается двумя точками пространства, формирующих ось конуса, и двумя радиусами оснований.

• Сфера – задается координатами центра и радиусом.

• Труба – задается двумя точками пространства, формирующих ось трубы, и внешним и внутренним радиусами основания.

• Комбинированный контрольный объем – задается двумя типами контрольных объемов. Внутри первого заданного объема строится адаптивная сетка с требуемым размером до тех пор, пока не встретится второй заданный объем. Адаптивная сетка внутри второго контрольного объема не строится.

Генерация объемной сетки разбивается на два этапа: генерация трехмерной регулярной сетки с возможностью вставки адаптивной сетки в ячейки и преобразование регулярной сетки в нерегулярный формат на этапе записи файла-разреза в формате ЕФР [4].

Этап генерации трехмерной регулярной сетки разбивается на несколько шагов. На первом шаге формируется матричная сетка по заданным параметрам. Выбор такого формата представления сетки обусловлен простотой поиска соседей для элементов сетки (узлов и ячеек), а также минимальным набором дополнительной информации для доступа к элементам.

Следующим шагом проводится анализ пересечения каждой ячейки матричной сетки с набором заданных контрольных объемов. Если ячейка имеет пересечение с одним из объемов, проводится анализ на необходимость дробления этой ячейки. То есть, происходит сравнение размера текущей ячейки с требуемым размером в контрольном объеме. Если ячейка проходит по критерию, то для нее вызывается процедура дробления. Как такового дробления не происходит, т. е. матричная сетка никаких изменения не претерпевает. Вместо этого формируется новая, аналогичная исходной, матричная сетка, габаритами и местоположением в пространстве совпадающая с обрабатываемой ячейкой. Размерность сетки 2×2×2 ячеек. В ячейку записывается ссылка на новую сетку. Нумерация узлов ячейки согласовывается с крайними узлами новой сетки. Для новой сетки также формируется информация о соседних ячейках исходной сетки и аналогичных матричных сетках в этих ячейках

После обработки всех ячеек исходной сетки формируется набор матричных сеток, который образует адаптивность первого уровня. Далее анализируются ячейки из этого набора сеток на пересечение с контрольными объемами. Если есть ячейка, удовлетворяющая критериям дробления, то для нее, аналогично описанному выше методу, формируется матричная сетка. Таким образом формируется набор матричных сеток второго уровня адаптивности.

Третий и последующий уровни адаптивности формируются аналогично. Количество уровней адаптивности не ограничено (рис. 9).

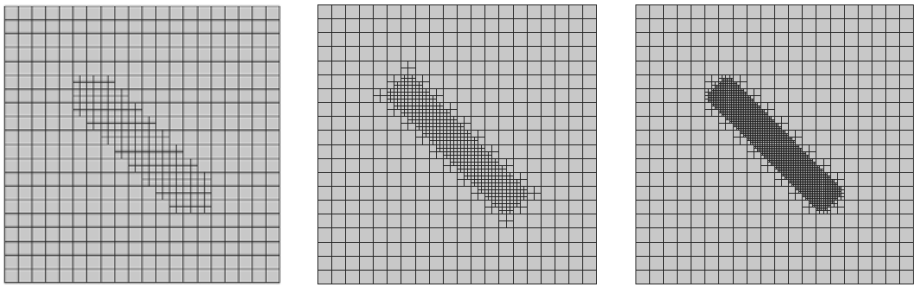


Рис. 9. Формирование первых трех уровней адаптивности

После построения всех уровней происходит сглаживание переходов между уровнями адаптивности. Для этого анализируются все «не подробленные» ячейки регулярной адаптивной сетки. Для ячейки формируется список соседей всех уровней, которые соседствуют с обрабатываемой через грань. Анализируются уровни адаптивности каждой ячейки из этого списка.

Если в списке есть ячейка с уровнем адаптивности превышающий уровень обрабатываемой ячейки больше чем на один, то обрабатываемая ячейка дробится аналогично описанному выше методу. Такой анализ проводится до тех пор, пока разница между «не подробленными» ячейками не будет превышать один уровень адаптивности (рис. 10).

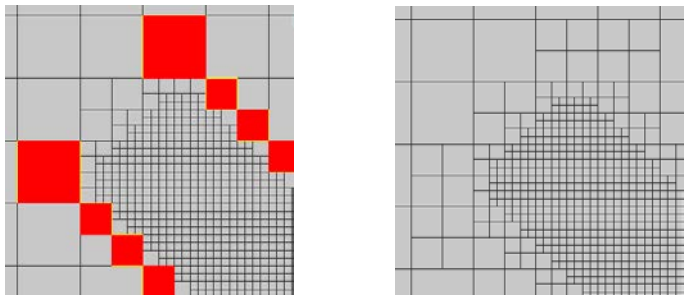


Рис. 10. Формирование первых трех уровней адаптивности

В результате образуется объемная сетка в матричном адаптивном представлении (рис. 11)

Формирование неструктурированной объемной сетки происходит в формате ЕФР на стадии записи файла-разреза. Для приведения регулярной адаптивной сетки к требуемому формату необходимо сформировать информацию о гранях – номера разделяемых ячеек и список узлов. Для ячейки формируется список граней, из которых она состоит.

На первом шаге проводится нумерация всех «не подробленных» ячеек всех уровней адаптивности. Таким образом, при формировании информации о гранях в поля номеров разделяемых ячеек сразу пишется конечная информация.

При формировании списка узлов для грани необходимо соблюдать правильный порядок следования узлов. Согласно формату ЕФР узлы грани должны идти в порядке против хода часовой стрелки, если смотреть на эту грань из первой разделяемой ячейки.

Обработка сетки начинается с последнего уровня адаптивности. Таким образом ячейки меньшего размера формируют набор граней, относящийся к одной из сторон более крупной ячейки. Это позволяет экономить время на вычислениях соседства между ячейками и определением количества граней для конкретной ячейки (рис. 12).

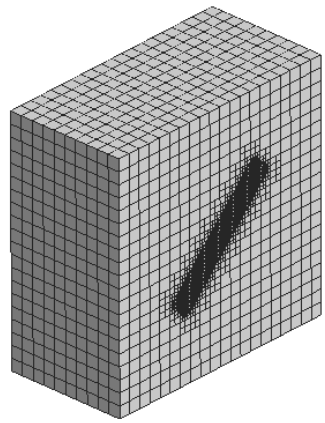


Рис. 11. Результат генерации матричной адаптивной сетки

Так как преобразование происходит с матричной сетки, то при формировании узлов координаты вычисляются через номера этих узлов в сетке и смещение относительно минимальной точки габарита сетки.

Сформированная сетка записывается в файл формата ЕФР-3.

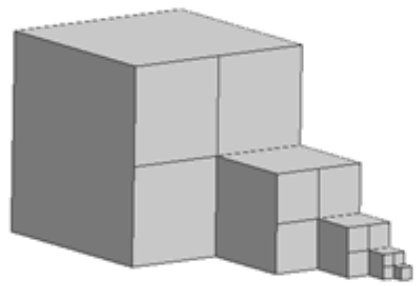


Рис. 12. Формирование ячеек

На текущий момент основной задачей модуля генерации неструктурированной трехмерной сетки, – это создание новой сетки на основе данных о старой деформированной сетке в методике ТИМ-3D. В ходе счета задачи алгоритмы, анализирующие качество сетки, дают сигнал о необходимости перестроения счетной сетки. Производится подготовка параметров для генерации новой сетки: вычисляются габариты ячеек старой сетки и определяется размер для новых ячеек. Для каждой ячейки перестраиваемой сетки формируется контрольный объем «Параллелепипед», который эквивалентен габариту ячейки. В качестве требуемого размера задается исходный размер ячейки. Набор контрольных объемов передается в модуль генерации сетки. Так же в модуль передаются габариты задачи и базовый размер ячейки исходной сетки.

После окончания работы генератора результат сохраняется в файл формата ЕФР-3, который передается в методику ТИМ-3D для загрузки и переинтерполяции величин со старой сетки на новую (рис. 13).

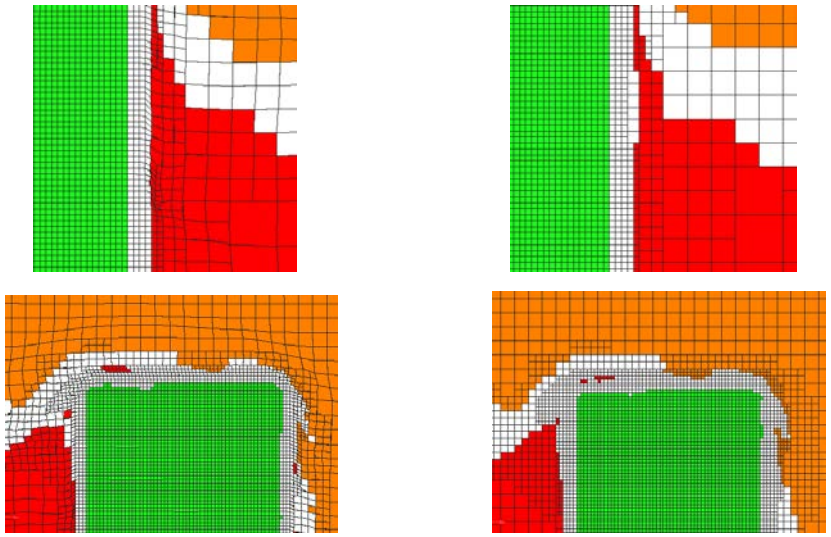


Рис. 13. Фрагмент перестроения сетки (слева – перестраиваемая, справа – новая)

Все операции по генерации новой сетки проводятся полностью в автоматическом режиме без остановки счета задачи.

Для оценки работоспособности модуля было проведено тестирование на различных задачах с помощью методики ТИМ-3D.

В первой задаче тестирование проводилось на ПЭВМ с 16 ГБ оперативной памяти и процессором Intel Core i7 2.93 GHz. Параметры перестраиваемой сетки: 2466883 ячеек, 3 уровня адаптивности. При работе генератора исходная сетка состояла из 129600 ячеек. В ходе построения адаптивных уровней прибавилось еще 2689344 ячеек. Из них 523252 первого уровня, 835308 – второго и 1412000 – третьего. Время построения матричной адаптивной сетки около 10 с. Время конвертации в неструктурированный формат – около 8 с. Время записи итоговой сетки в файл формата ЕФР – около 18,5 с. Общее время работы модуля – 37,2 с. Затраченный объем оперативной памяти не превышает 2,5 ГБ.

Во второй задаче тестирование проводилось на ЭВМ с 54 ГБ оперативной памяти. Цель теста – анализ зависимости времени работы модуля от количества генерируемых ячеек. Построение сетки проводилось, начиная с 60 млн ячеек с пятью уровнями адаптивности. Анализ результатов показал, что время построения линейно зависит от числа ячеек (рис. 14).

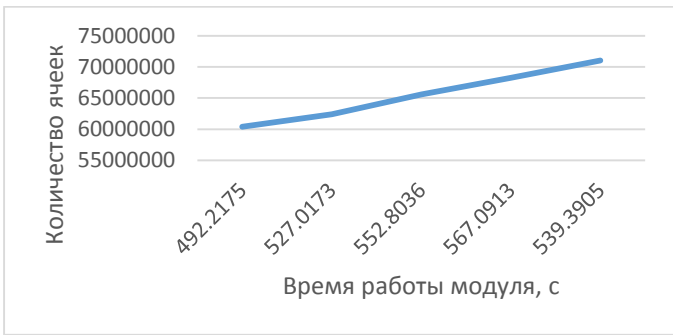


Рис. 14. График зависимости времени работы модуля от количества ячеек

Заключение

Разработанный функциональный блок расширил возможности методики ТИМ, позволяя генерировать объемную неструктурированную сетку, как в ручном и автоматическом режимах. В автоматическом режиме методика ТИМ получила возможность проводить сложные расчеты с перестройкой всей счетной сетки без остановки счета задачи.

В ближайших планах реализовать работу функционального блока в параллельном режиме с использованием интерфейса MPI.

Литература

1. Соколов С. С., Панов А. И., Воропинов А. А., и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 2. С. 37–52.

2. Смолкина Д. Н., Борисенко О. Н., Черенкова М. В., Гиниятуллина А. Г., Кузьменко М. В., Чухманов Н. В., Потехина Е. В., Попова Н. В., Турусов М. Р. Автоматический генератор неструктурированных многогранных сеток в препроцессоре пакета программ "ЛОГОС" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 2. С. 25–39.

3. Потехина Е. В., Ребров С. В., Логинов Д. В. Методы расчета трехмерных разностных сеток и распределения частиц в препроцессоре 3D-РНД // Труды международной научно-технической конференции «Компьютерное моделирование 2009» – 2009. С. 158–164.

4. Олесницкая К. К., Антипин И. А., Петрова М. А. Коллективный доступ к файловым данным на вычислительных системах с различной архитектурой в библиотеке ЕФР // XVII Международная конференция «Супервычисления и математическое моделирование»: Сборник докладов. Саров, 15–19 октября 2018.

A MODULE IN THE *TIM* CODE FOR GENERATION OF 3D UNSTRUCTURED GRIDS ON THE BASE OF A REGULAR ADAPTIVE REFINEMENT PATTERN

V. A. Nikitin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes a module that allows generating 3D unstructured grids on the base of a regular adaptive refinement pattern. This module can be used both in the stage of preparing data to solve the problem and in the stage of reconstructing the computational grid dynamically adapted to the problem solution. The generator operation algorithm is described. The result of the module incorporation into the TIM code is given.

Key words: a 3D grid generator, unstructured grids, an adaptive refinement pattern, the TIM code.

УДК 004.4
DOI 10.53403/9785951505071_2022_330

ПРОГРАММНЫЙ МОДУЛЬ ГЕНЕРАЦИИ ЗАМКНУТОЙ ПОВЕРХНОСТНОЙ ТРИАНГУЛЯЦИОННОЙ СЕТКИ В ПАКЕТЕ ПРОГРАММ ЛОГОС

В. А. Никитин, А. В. Шурыгин, И. Г. Новиков, А. В. Егоров, С. С. Соколов, А. И. Панов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе приведено описание программного модуля, интегрированного в состав «ЛОГОС-ПреПост», который позволяет генерировать замкнутую поверхностную триангуляционную сетку на основе метода вакуумной упаковки. Описаны этапы формирования поверхностной сетки. Приведено описание подходов, используемых на каждом этапе формирования сетки.

Ключевые слова: генератор поверхностной сетки, замкнутая поверхностная триангуляционная сетка, метод вакуумной упаковки, пакет программ ЛОГОС.

В настоящее время разрабатывается пакет программ инженерного анализа ЛОГОС [1], предназначенный для подготовки, проведения и анализа результатов компьютерного моделирования задач прочности и аэро-гидродинамики с использованием супер-ЭВМ с массовым параллелизмом. Для подготовки начальных данных для расчетных методик, используется интегрированный в состав ЛОГОС пре-постпроцессор «ЛОГОС-ПреПост» [2].

При подготовке математических моделей задач аэродинамики и гидродинамики используется генератор многогранных дискретных моделей [3], входящий в состав «ЛОГОС-ПреПост». В качестве входных данных генератор использует переданную ему замкнутую треугольную поверхностную сетку. К поверхностной сетке предъявляется ряд требований, связанных с отсутствием дефектов таких как: наложения, пересечения треугольников, «дыр», висящих треугольников и вершин, малых двугранных углов между соседними треугольниками, вырожденных треугольников.

Поверхностная сетка в «ЛОГОС-ПреПост» может быть получена из нескольких источников. Во-первых, уже существующая поверхностная сетка может быть импортирована в «ЛОГОС-ПреПост» из сторонних форматов (*.stl, *.vtk и др.). Во-вторых, поверхностная сетка может быть сгенерирована для исходной геометрической модели в параметрическом представлении. И в-третьих, поверхностная сетка может быть получена на этапе перестроения поверхностной сетки при генерации объемной сетки. Сам генератор поверхностной сетки [4], используемой в «ЛОГОС-ПреПост», может работать с исходной поверхностной триангуляционной сеткой. Поэтому, крайне важно, чтобы эта исходная сетка была корректной, т. е. не имела дефектов и была связанной. В противном случае дальнейшее построение объемной сетки становится невозможным.

Устранение обнаруженных дефектов в «ЛОГОС-ПреПост» может быть проведено как в автоматическом, так и в ручном режимах. Для обнаружения дефектов в геометрических моделях «ЛОГОС-ПреПост» предоставляет специальные средства диагностики. Во многих случаях, тем не менее, автоматическое исправление проблем невозможно или приводит к нежелательным побочным эффектам. В таких случаях у пользователя остается только вариант с ручным исправлением обнаруженных дефектов. Однако при большом количестве дефектов исправление их вручную является трудоемким

и, как следствие, длительным процессом. Таким образом, существует высокая потребность в развитии средств автоматической подготовки корректной замкнутой триангуляционной сетки для последующего перестроения поверхностных и построения объемных сеток. Развитие таких средств позволит существенно сократить время подготовки 3D-моделей к построению объемных сеток и тем самым сократить общее время подготовки и проведения расчетов.

Разработчики сторонних систем инженерного анализа также столкнулись с вышеуказанными проблемами. Разработанный ими подход к решению проблемы включает предоставление пользователю средств генерации замкнутых поверхностных сеток (аппроксимирующих исходную поверхность снаружи или изнутри) по исходной поверхностной сетке, допуская наличие в исходной сетке любых несогласованностей (зазоров, нахлестов, пересечений и т. п.). Метод генерации такой замкнутой сетки получил название «термоупаковки» (shrinkwrap), также широко употребляется название «генератор вакуумной упаковки», а также сокращенное название «враппер» [5–9].

В рамках «ЛОГОС-ПреПост» реализован модуль генерации замкнутой поверхностной триангуляционной сетки методом «вакуумной упаковки», позволяющий упростить подготовку поверхностных сеток для дальнейшей подготовки задачи.

Разработчики программного модуля выделяют следующие этапы работы модуля:

1. Предобработка исходной поверхности.
2. Генерация замкнутой поверхности вблизи исходной поверхности.
3. Проецирование замкнутой поверхности на исходную поверхность.
4. Постобработка результирующей поверхности.

Предобработка исходной поверхности

Исходная поверхностная сетка передается в модуль как набор несвязанных треугольников. Чтобы извлечь из поверхности максимальное количество информации все треугольники сшивают в единую поверхность. При наличии дефектов в описании треугольников исходная поверхность может быть незамкнутой. Возможен вариант получения набора несвязанных между собой поверхностей. При формировании связанной поверхности все треугольники приводятся к единому порядку обхода вершин, а также производится удаление вырожденных треугольников, для которых направление нормали не определено. В том числе производится анализ на наличие характерных особенностей исходной геометрии. Под характерными особенностями подразумевается набор линий, формируемый из ребер, в которых разделяемые ими треугольники находятся под углом друг к другу. Значение угла между нормалями треугольников, при котором формируются характерные линии, регулируется пользователем через параметры. Так же к характерным ребрам относятся ребра, принадлежащие только одному треугольнику. Еще один набор характерных особенностей формируется из линий самопересечения поверхности. На рис. 1 приведен набор характерных линий

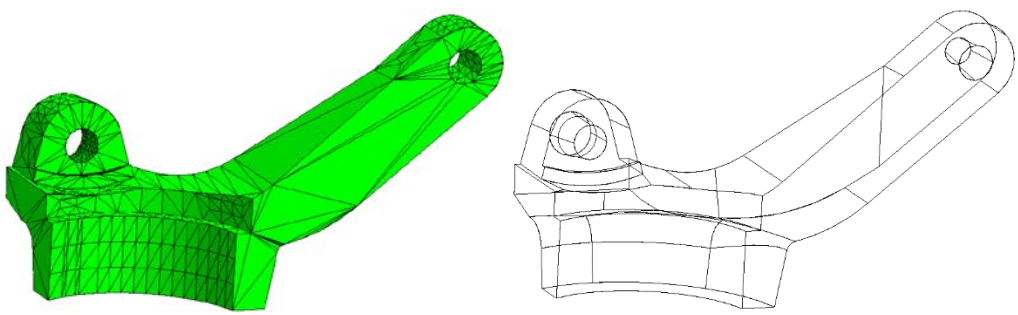


Рис. 1. Исходная поверхность (слева) и набор характерных линий (справа)

Информация о характерных особенностях необходима для лучшей аппроксимации замкнутой поверхности на этапе проецирования.

Генерация замкнутой поверхности вблизи исходной поверхности

Генерация замкнутой поверхности состоит из нескольких шагов.

На первом шаге происходит генерация объемной сетки на основе дробно-адаптивного шаблона, согласно заданным пользователем параметров. Основным параметром для генерации объемной сетки является базовый размер, который задает длину стороны ячейки сетки. Размеры объемной сетки превышают габариты исходной поверхности в каждом направлении. Это сделано для правильного формирования объемов, разделяемых исходной поверхностью. В ходе генерации выделяются ячейки объемной сетки, через которые прошла исходная поверхность. Дробление ячеек происходит по совокупности нескольких критериев: количество попавших треугольников в объемную ячейку, максимальное отклонение нормалей к треугольникам и наличие характерных особенностей.

При построении дробности в объемной сетке разработчиками было опробовано два подхода.

Первый подход основывался на преобразовании матричной сетки в нерегулярный формат. В данном случае все ячейки объемной сетки, через которые не прошла исходная поверхность, удаляются. Затем оставшиеся ячейки при необходимости дробятся и пустые удаляются. Количество уровней дробления ячеек не ограничено. Такой подход позволяет исключить из обработки пустые ячейки и не хранить их в памяти (рис. 2).

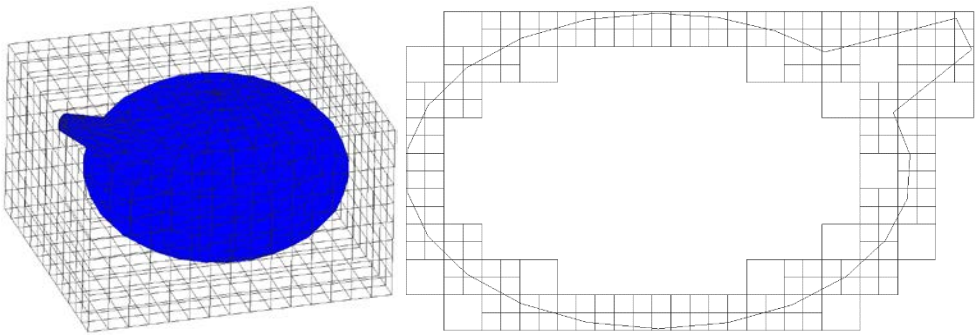


Рис. 2. Объемная сетка (слева) и она же после дроблений и удаления пустых ячеек (справа, в сечении)

Минусами такого подхода являются: сложность топологии нерегулярной сетки приводит к хранению дополнительной информации, что увеличивает время на ее обработку и приводит к дополнительному расходу оперативной памяти.

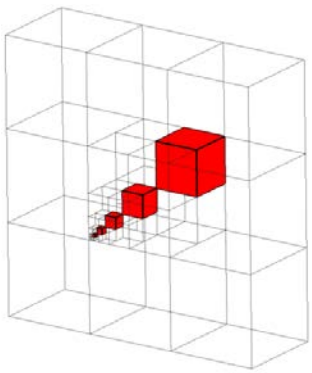


Рис. 3. Построение уровней дробности

Второй подход основан на использовании матричной объемной сетки на всех уровнях дробления ячеек. При таком подходе в ячейку, которая удовлетворяет критерию дробления, встраивается аналогичная независимая матричная сетка с размерностью 2×2×2 ячеек (рис. 3).

Для элемента сетки (ячейка или узел) получение информация о соседних элементах выполняется через его индексы. Такой подход сокращает объем информации хранящейся в памяти в 2 раза, чтов свою очередь приводит к увеличению скорости генерации замкнутой поверхности более чем в 2 раза.

Второй подход так же позволяет расширить возможности модуля для определения замкнутости исходной поверхности и возможности сформировать замкнутую триангуляционную поверхность из любого заданного пользователем объема. Реализация новых инструментов анализа на основе второго подхода возможна за счет хранения информации по всему габариту исходной геометрии.

В обоих подходах в зависимости от указанного пользователем объема (внешнего или наибольшего внутреннего) на основе объемной сетки формируется внешняя или внутренняя замкнутая поверхность (рис. 4).

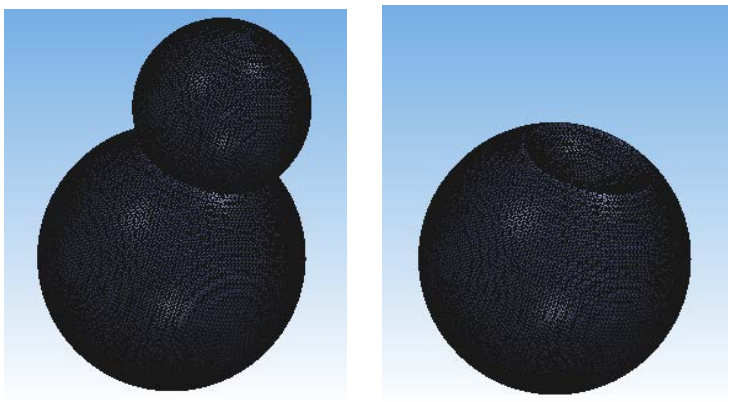


Рис. 4. Замкнутые поверхности, построенные с внешней стороны (слева) и с внутренней стороны (справа)

Генератор дробно-адаптивной сетки имеет ряд инструментов, позволяющий пользователю более детально прописать некоторые особенности исходной поверхности. К таким инструментам относятся контрольные объемы и функция предотвращения контакта между заданными поверхностями. Под контрольными объемами подразумевается набор геометрических примитивов в виде блока, конуса, сферы и цилиндра. Основная идея использования контрольных объемов состоит в том, что пользователь имеет возможность задавать необходимый размер ячеек в объеме, ограниченном одной из геометрических фигур (рис. 5).

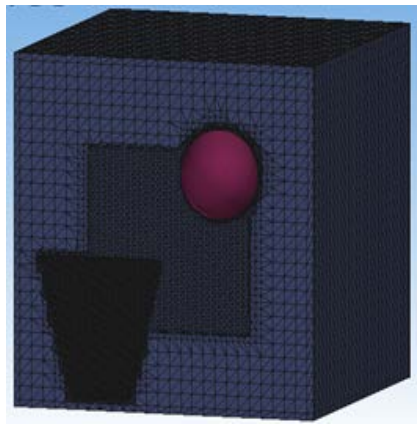


Рис. 5. Использование контрольных объемов

Предотвращение контакта используется, когда ячейки сетки заполняют пространство между близкорасположенными поверхностями исходной геометрии. В таком случае размеры ячеек итоговой поверхностной сетки превышают размер щели, заполняя ее, образуют контакт между близкими поверхностями. Функция позволяет при задании достаточно грубых базовых размеров ячеек не допускать контакта между близкими поверхностями (рис. 6).

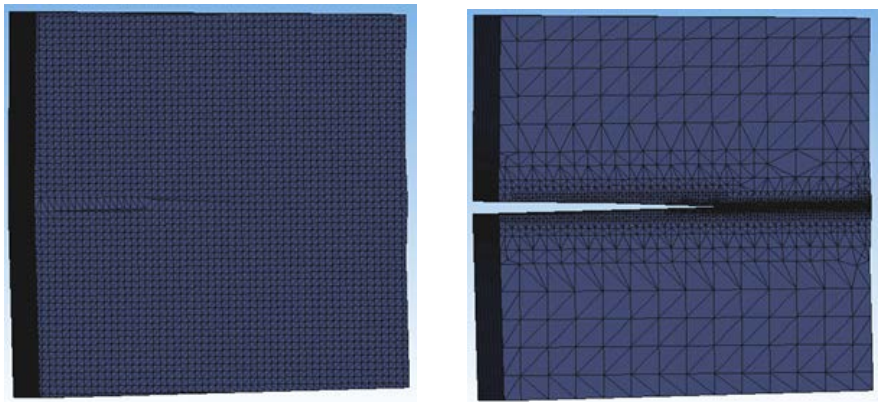


Рис. 6. Итоговая сетка без (слева) и с (справа) использованием функции предотвращения контакта

Результатом работы данного этапа является замкнутая трехмерная поверхность, состоящая преимущественно из четырехугольников (рис. 7).

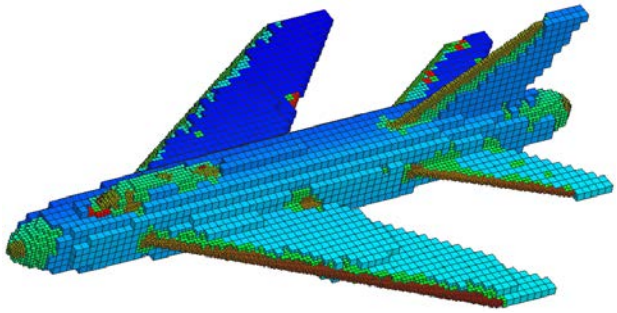


Рис. 7. Внешняя замкнутая поверхность

Проецирование замкнутой поверхности на исходную поверхность

Работа модуля на этапе проецирования замкнутой поверхности состоит из нескольких шагов.

На первом шаге для каждого узла замкнутой поверхности ищется ближайший треугольник исходной поверхности. На треугольнике вычисляется ближайшая точка к проецируемому узлу. Узел проецируется в найденную точку на треугольнике (рис. 8).

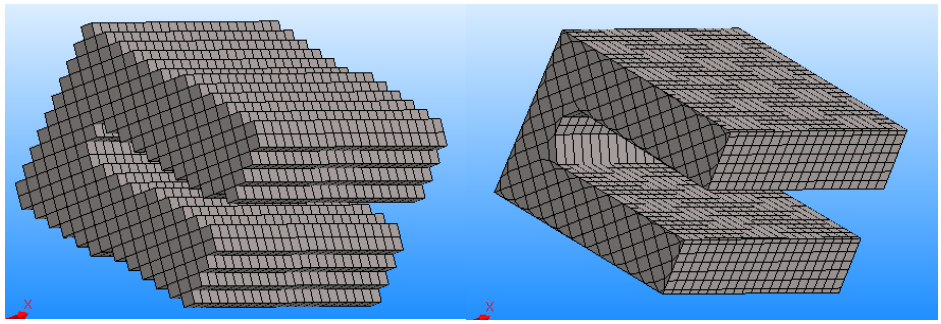


Рис. 8. Замкнутая поверхность до проецирования (слева) и после (справа)

После проецирования все узлы замкнутой поверхности принадлежат исходной поверхности, тем самым достигается уменьшение ошибки аппроксимации. Но полученная поверхность может иметь ряд самопересечений и самоналожений (рис. 9).

Для устранения полученных дефектов поверхности применяются удаление вырожденных многоугольников и локальное сглаживание сетки. В результате изначальное местоположение спроецированных узлов может измениться. Для таких узлов проводится уточнение их положения относительно исходной поверхности, чтобы не нарушить аппроксимацию (рис. 10).

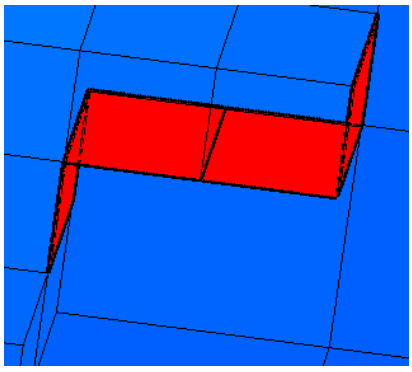


Рис. 9. Подворачивание многоугольников под соседние многоугольники (выделены красным)

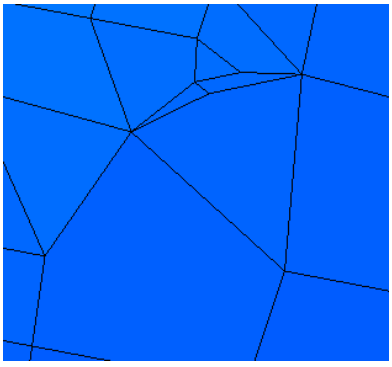


Рис. 10. Исправления сетки после проекции

Следующим шагом для получения качественной аппроксимации является учет характерных особенностей. После проекции и устранения дефектов некоторые углы исходной геометрии остаются скошенными (рис. 11).

При устранении подобных дефектов проводится анализ ребер многоугольной сетки на предмет их расположения относительно исходной геометрии. Если ребро попало под исходную поверхность, то происходит поиск точки на ближайшей характерной кривой для прописывания особенности геометрии. Если такая точка найдена, то один из узлов рассматриваемого ребра переносится в эту точку. В результате замкнутая многоугольная сетка более качественно аппроксимирует исходную геометрию (рис. 12).

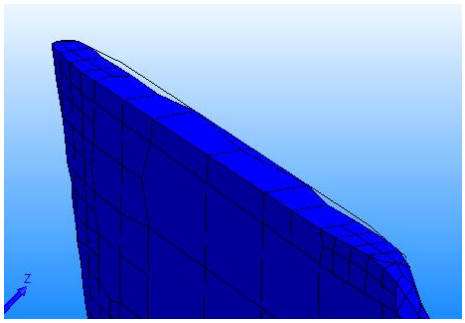


Рис. 11. Скошенные углы на замкнутой поверхности (линией выделена характерная особенность)

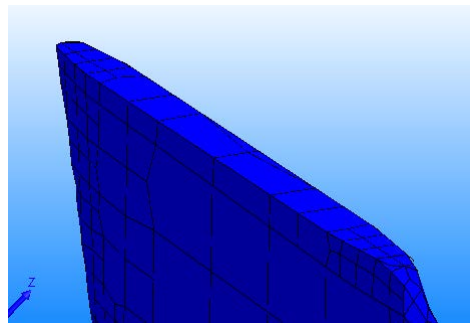


Рис. 12. Учет «внешних» характерных особенностей

Далее проводится учет характерных особенностей, находящихся в вогнутых областях исходной геометрии. Для поиска таких особенностей снова анализируются ребра многоугольной сетки, но рассматриваются только ребра, лежащие с внешней стороны от исходной геометрии. Такие ребра обрабатываются путем вставки на обрабатываемое ребро дополнительного узла, который в свою очередь разбивает ребро на два. Данный узел переносится в найденную точку на характерной особенности (рис. 13).

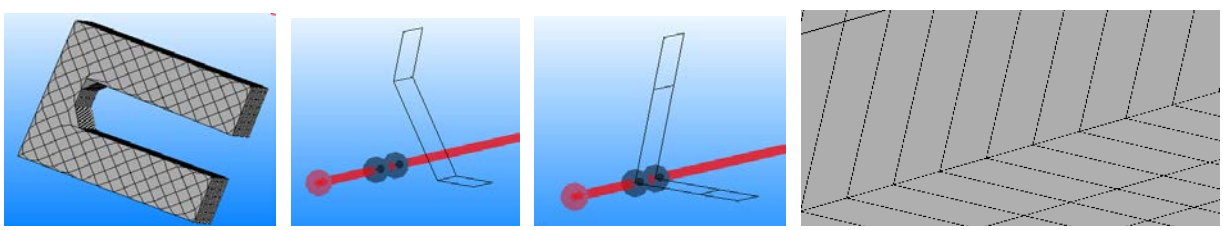


Рис. 13. Учет «внутренних» характерных особенностей

Заключительным шагом проекции является прописывание углов геометрии. Для этого анализируются ребра, у которых узлы расположены на характерных особенностях. Если у анализируемого ребра узлы лежат на разных характерных линиях, то происходит поиск угловой точки. В случае, когда такая точка есть, происходит перенос одного из узлов ребра в эту точку (рис. 14)

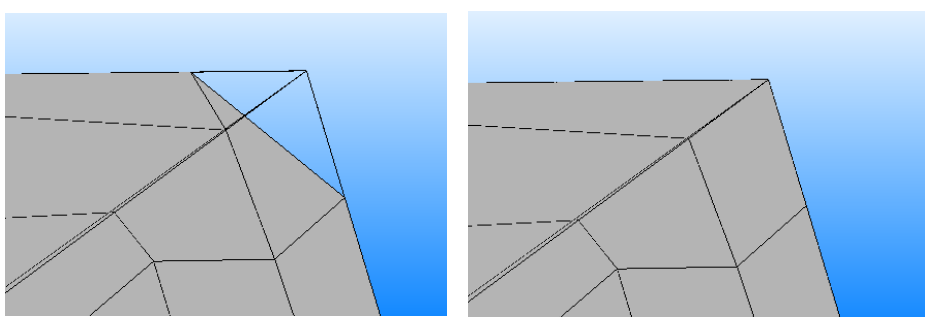


Рис. 14. Прописывание углов геометрии

После проецирования узлов на характерные особенности замкнутая многоугольная поверхность более приближена к исходной поверхности.

Постобработка результирующей поверхности

На этом этапе происходит преобразование многоугольной поверхностной сетки в триангуляционную поверхностную сетку. При триангуляции учитывается кривизна исходной поверхности так, чтобы новые ребра имели минимальное отклонение от исходной поверхности (рис. 15)

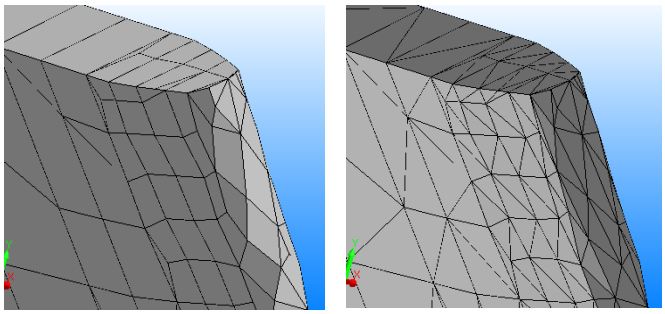


Рис. 15. Триангуляция многоугольной поверхности

После триангуляции сетки, проводится анализ ее качества на наличие самопересекающихся треугольников, треугольников малой площади, ребер не характерного размера, либо изначальная геометрия недостаточно точно описывается (ошибка аппроксимации углов, не сохранение характерных линий). В первом случае наличие проблем в замкнутой поверхностной сетке не позволяет применить алгоритм построения тетраэдризации, поскольку для этого необходимо наличие замкнутой не самопересекающейся поверхностной триангуляции. В другом случае наличие проблем отражается на качестве представления изначальной геометрии.

Для выявления набора треугольников неудовлетворительного качества проводится анализ построенной поверхностной сетки. Все выявленные в ходе анализа треугольники подвергаются перестройке. В настоящее время в модуле реализовано 2 подхода: перестроение с использованием информации об исходной поверхностной сетке и перестроение без использования информации об исходной поверхностной сетке.

В первом подходе для устранения проблемных участков в построенной триангуляции реализован альтернативный метод построения поверхностной сетки, который работает только в некоторой выделенной области.

По граничным узлам данной области строится контур, внутри которого далее будет построена триангуляция, а также формируется набор треугольников исходной поверхности, по которому данная новая триангуляция будет строиться («подложка»). Помимо координат граничных узлов задается порядок обхода данного контура, а также координаты векторов внешних нормалей в каждой точке контура.

Идея описываемого метода состоит в выделении на подложке почти плоских участков. Каждый почти плоский участок проецируется на плоскость, строится его двумерная триангуляция, которая затем проецируется обратно на поверхность. После этого построенные треугольники запоминаются, а контур уменьшается. Затем рассматривается следующий почти плоский участок, пока контур не станет пустым. Поскольку выделяемые участки поверхности являются почти плоскими, то искажения геометрии области незначительны.

Стоит отметить, что рассматриваемый метод применим к достаточно сложным геометриям, например, на рис. 16 представлена окончательная триангуляция контура, полученная в процессе выделения 22 плоских участков.

Второй подход по перестроению локальных зон поверхностной триангуляционной сетки состоит из нескольких блоков и применяется для исправления сетки, полученной после неуспешной работы первого подхода при помощи замены сетки во фрагменте.

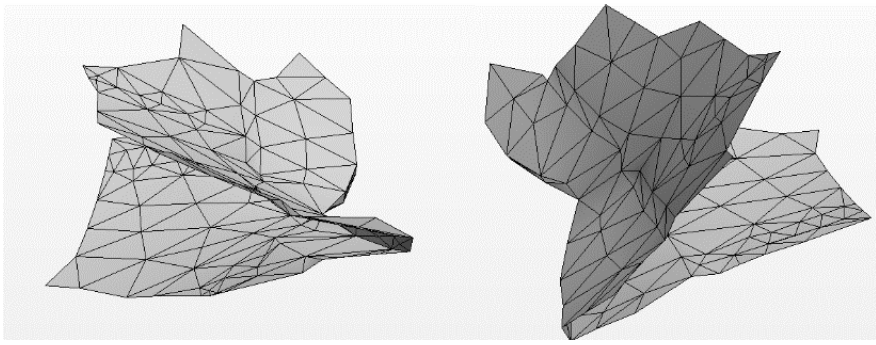


Рис. 16. Триангуляция сложного контура

Суть состоит в том, что в случае, если получаемая сетка не удовлетворяет заданным критериям качества или не может быть построена в выбранном фрагменте задачи, ограниченном некоторым контуром, производится перестроение ее в данном контуре, на основе известного метода [10] триангуляции полигонов «отсечение уха» - от контура в каждый момент времени отсекается «ухо» в виде треугольника, состоящего из трех подряд идущих вершин контура. Данным методом можно произвести триангуляцию любого выпуклого или невыпуклого полигона на плоскости, однако в пространстве, триангуляция пространственного полигона сопряжена с рядом трудностей, связанных, во-первых, с невозможностью определения пересечения вводимых в триангуляцию новых треугольников с уже имеющимися, во-вторых, с невозможностью учесть кривизну получаемой поверхности (рис. 17)



Рис. 17. Выделенный контур, с удаленными ячейками внутри него (слева)
и триангуляция выделенного полигона (справа)

Отрезание «уха» производится и с учетом характерной длины ребер треугольников, которые были в выделенном контуре до его перестроения. В этом случае, в триангуляцию добавляются не только новые ребра, но и новые узлы. После добавления нового узла, имеющиеся ячейки дробятся на треугольники (рис. 18).

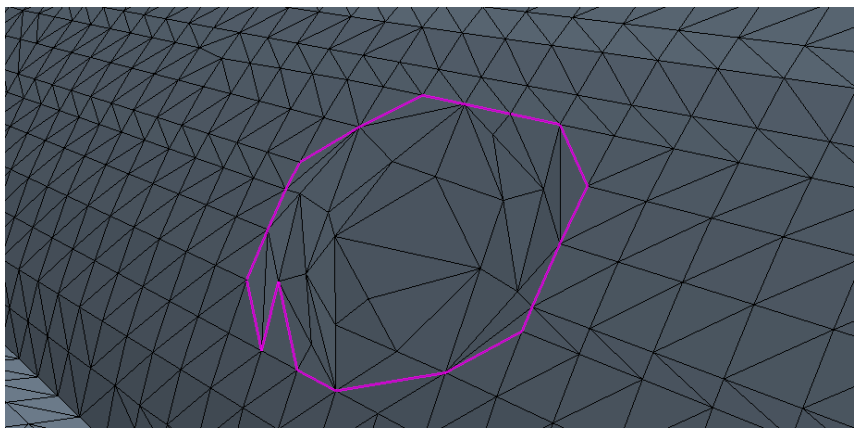


Рис.18. Выделенный фрагмент, заполненный сеткой с добавлением новых узлов

На довольно сложных контурах построить сетку без существенной потери описываемой геометрии с таким подходом невозможно, так как нет дополнительных данных о характерных кривых. В этом случае могут происходить случаи, рассмотренные на рис. 19 слева. Однако так как этот способ перестроения является дополнительным, то такие ситуации на большинстве геометрий не происходят, либо их количество минимально, и потеря качества минимизирована (рис. 19).

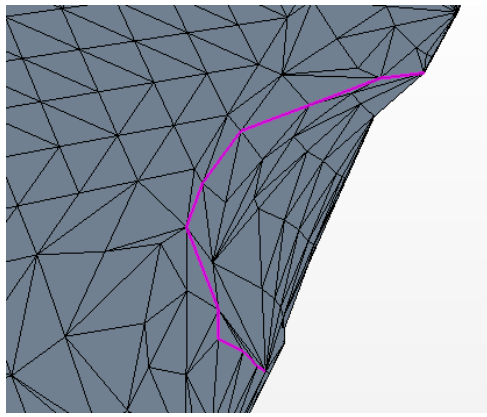


Рис. 19. Фрагмент с потерей качества аппроксимации
исходной поверхности по алгоритмам отрезания «уха»

После окончания последнего этапа модуля результирующая триангуляционная поверхностная сетка передается в пре-постпроцессор «ЛОГОС-ПреПост» в качестве основы для работы поверхностного или объемного генераторов.

Литература

1. Погосян М. А., Савельевских Е. П., Шагалиев Р. М., Козелков А. С., Стрелец Д. Ю., Рябов А. А., Корнев А. В., Дерюгин Ю. Н., Спиридонов В. Ф., Циберев К. В. Применение отечественных суперкомпьютерных технологий для создания перспективных образцов авиационной техники // Журнал ВАНТ, сер. Математическое моделирование физических процессов, 2013, вып.2, стр. 3–17.

2. Фархутдинов В. Ф., Тарасов В. И., Соловьев А. Н., Борисенко О. Н., Лазарев В. В., Логинов Д. В., Аверина Н. С., Марунин А. В., Гриднев А. И., Федякина М. Г., Кузнецов М. Г., Черенкова М. В., Фролова Е. А., Лукичев А. Н., Смолкина Д. Н., Купалова А. Г., Кузьменко М. В., Ховрин Н. А., Сергеева А. С., Попова Н. В., Баканова Т. Ю. Обзор возможностей по обработке геометрических и построению сеточных моделей в ЛОГОС-Препост // Труды XIV Международной конференции «Супервычисления и математическое моделирование». Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2013. С. 585–592.

3. Смолкина Д. Н., Борисенко О. Н., Черенкова М. В., Гиниятуллина А. Г., Кузьменко М. В., Чухманов Н. В., Потехина Е. В., Попова Н. В., Турусов М. Р. Автоматический генератор неструктурированных многогранных сеток в препроцессоре пакета программ "ЛОГОС" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 2. С. 25–39.

4. Лукичев А. Н., Цалко Т. В., Панкратов Д. М., Логинов Д. В., Белова А. И., Моськина Е. О. Особенности построения триангуляционных сеток на поверхностях в аналитическом и фасеточном представлении // Сборник докладов 15 научно-технической конференции «Молодежь в науке». Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2017. С. 73–78

5. Leif P. Kobbelt, Jens Vorsatz, Ulf Labsik Hans-Peter Seidel. A Shrink Wrapping Approach to Remeshing Polygonal Surfaces. 1999. EUROGRAPHICS ’99, vol 18, number 3

6. Wei Yuan. United States Patent Application Publication. Methods for generating digital or visual representations of a closed tessellated surface geometry. 2007.

7. Zhao J., Ledouxa H., Stoter J. Automatic repair of citygml LOD2 buildings using shrink-wrapping. Volume II-2/W1, ISPRS 8th 3DGeoInfo Conference & WG II/2 Workshop, 27–29 November 2013, Istanbul, Turkey

8. Marcel Campen, Leif Kobbelt. Eurographics 2012 Tutorial. A Practical Guide to polygon mesh repairing.

9. Erling Eklund, Y. K. Lee, H. Vardhan Mesh, H. Ghazialam. ANSYS Inc. Generation of Large Size Industrial CFD Applications using a Cartesian Grid based Shrink Wrap approach. 2007.

10. (1993) «Slicing an ear using prune-and-search». Pattern Recognition Letters 14 (9): 719–722. DOI: 10.1016/0167-8655(93)90141-

A PROGRAM MODULE FOR GENERATION OF A CLOSED SURFACE TRIANGULAR GRID IN "LOGOS" SOFTWARE PACKAGE

V. A. Nikitin, A. V. Shurygin, I. G. Novikov, A. V. Egorov, S. S. Sokolov, A. I. Panov

Russian Federal Nuclear Center –
All-Russia Scientific Research Institute of Experimental Physics, Sarov

The paper presents a program module of the LOGOS PrePost prepostprocessor that allows generating a closed surface triangular grid on the base of the vacuum packaging method. The surface grid generation stages are described. The approach used in each grid generation stage is discussed.

Key words: surface grid generator, the vacuum packaging method, a closed surface triangular grid, the LOGOS software package.

УДК 004.4
DOI 10.53403/9785951505071_2022_340

МОДУЛЬНАЯ ИНТЕГРАЦИОННАЯ ПЛАТФОРМА ЛОГОС. ТЕХНОЛОГИЯ ПОДГОТОВКИ И ВЫПОЛНЕНИЯ МУЛЬТИФИЗИЧНОГО МОДЕЛИРОВАНИЯ НА ОСНОВЕ СВЯЗАННО-СОПРЯЖЕННОГО РАСЧЕТА

К. К. Олесницкая, А. В. Дьяков, Н. А. Побуринная, А. А. Тюндина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Модульная интеграционная платформа (ЛОГОС-МИП) – это единая среда для проведения сквозного мультифизичного моделирования на основе связанных, сопряженных и поэтапных расчетов, параметрических и оптимизационных исследований. Платформа включает в себя: программные средства интеграции; легковесные модули; транспортный уровень; обрабатывающие функции; файловый контейнер и специализированные библиотеки. Ключевой задачей программных средств является обмен данными между расчетными модулями при совместном моделировании. Обмен может осуществляться как через память, так и через файловый контейнер. А используемые специализированные библиотеки (такие как интерполяторы и конверторы) в процессе передачи данных позволяют выполнять их преобразование из одного формата представления в другой.

Ключевые слова: ЛОГОС-МИП, платформа, связанно-сопряженный расчет, мультифизичное моделирование, расчетные и легковесные модули, адаптер связи, сериализатор, десериализатор, менеджер шагов, интерполяторы.

Введение

ЛОГОС-МИП – это модульная интеграционная платформа (далее Платформа), которая представляет собой единую среду для проведения сквозного мультифизичного моделирования на основе связанных, сопряженных и поэтапных расчетов, параметрических и оптимизационных исследований.

Под мультифизичным моделированием понимается математическое моделирование физических процессов, при котором происходит объединение в совместном расчете различных областей физики: газодинамики, гидродинамики, термодинамики, механики твердого тела и т. п.

В докладе представлена технология сквозного мультифизичного моделирования на основе связанно-сопряженного расчета по программным модулям пакета программ ЛОГОС [1]. Под связанным расчетом понимается совместный расчет нескольких программных модулей, при котором выполняется двухсторонний обмен данными через память. Под сопряженным расчетом понимается совместный расчет нескольких программных модулей, при котором выполняется односторонняя передача данных через файловый контейнер.

Ключевые компоненты Платформы

По сути, Платформа является конструктором для цифрового моделирования, где детали – это программные модули, а соединительные элементы – это средства, которые позволяют объединять детали в единое целое.

Ключевыми компонентами технологии связанно-сопряженного расчета являются:

– Расчетные модули – программные модули численного моделирования физических процессов (например, ЛОГОС-Аэро (газодинамика), ЛОГОС-Тепло (термодинамика), ЛОГОС-ПА (механика твердого тела) и т. п.).

– Программные средства интеграции (например, модуль связи, точки передачи управления, сервисные функции, коннекторы, функции вызова и т. п.).

– Легковесные модули – это динамически подключаемые модули, которые работают только в связке с расчетными модулями, расширяя их функциональные возможности (например, Адаптеры связи, Сериализатор, Десериализатор, Менеджер временных шагов и т. п.).

– Транспортный уровень – это настраиваемая коммуникационная среда, которая обеспечивает обмен данными между расчетными модулями.

– Обрабатывающие функции – это программные модули, для обеспечения согласованности обменных данных Транспортного уровня.

– Файловый контейнер – это обменные данные расчетных модулей в формате единого файлового разреза (ЕФР) [2].

– Специализированные библиотеки (например, интерполяторы, конвертеры, библиотеки файлового доступа и т. п.).

Интеграция расчетного модуля в Платформу

Для интеграции расчетного модуля в Платформу необходимо выполнить следующее:

– Подключить модуль связи (статическую библиотеку ccf_bridge).

– Реализовать функцию получения сервисных функций root_service_function.

– Добавить вызов функции инициализации модуля связи ccf_bridge_init.

– Добавить вызов функции завершения работы с модулем связи ccf_bridge_finalize.

– Сформировать файл спецификаций.

Интеграция легковесного модуля в Платформу

Для интеграции легковесного модуля в Платформу необходимо выполнить следующее:

– Реализовать интерфейсную функцию получения коннекторов service_function_connector.

– Реализовать интерфейсную функцию вызова invoke.

– Собрать легковесный модуль как динамическую библиотеку.

– Описать интерфейс необходимых сервисных функций.

– Описать обрабатываемые точки передачи управления.

– Сформировать файл спецификаций, который содержит настройки подключения к расчетному модулю и шаблон настроек управления работой.

Поддержка легковесного модуля в расчетном модуле

Для поддержки легковесного модуля в расчетном модуле необходимо выполнить следующее:

– Добавить сервисные функции, необходимые легковесному модулю.

– Скорректировать функцию получения сервисных функций.

– Добавить вызов функции передачи управления модулю связи ccf_bridge_invoke для точек передачи управления, которые поддерживает легковесный модуль.

Теперь, при необходимости использования функциональных возможностей легковесного модуля, он может подключаться к расчетному модулю.

Подключение легковесных модулей к расчетному модулю

Подключение легковесных модулей к расчетному модулю осуществляет модуль связи на этапе инициализации. Без инициализации модуль связи работает в “холостом” режиме, то есть сразу возвращает управление расчетному модулю, а расчетный модуль работает в автономном режиме без использования легковесных модулей.

На рис. 1 представлена схема связи расчетного и легковесного модулей. Инициализация модуля связи выполняется при наличии конфигурационных файлов в директории исполняемого файла расчетного модуля. В процессе инициализации анализируются входные данные из конфигурационных файлов и, на основе информации о подключении легковесных модулей к расчетному модулю, выполняется следующее:

– Загрузка необходимых легковесных модулей.

– Связывание расчетного модуля с необходимыми легковесными модулями (на рис. 1 эта связь выделена сплошной стрелкой).

– Связывание необходимых легковесных модулей с расчетным модулем (на рис. 1 эта связь выделена пунктирной стрелкой).

– Формирование последовательности вызовов легковесных модулей для всех точек передачи управления модулю связи.

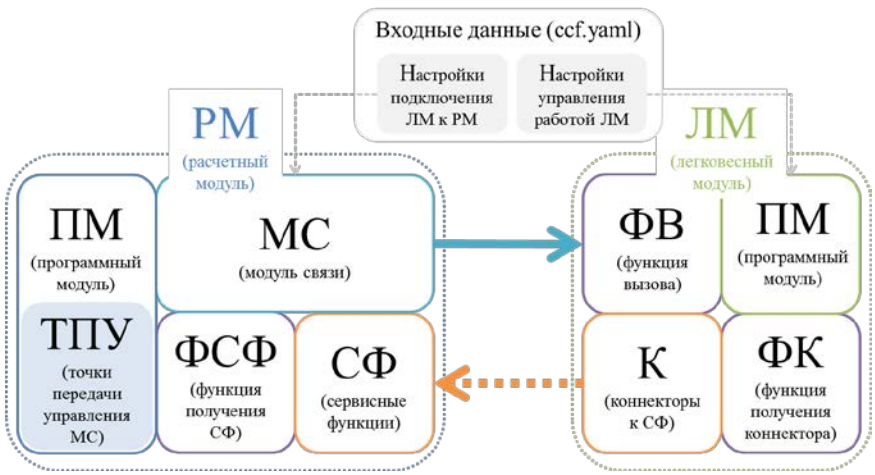


Рис. 1. Схема связи расчетного и легковесного модулей

В точке передачи управления модуль связи выполняет вызов необходимых легковесных модулей.

При завершении работы с модулем связи выполняется отключение легковесных модулей от расчетного модуля.

Общая схема связанно-сопряженного расчета

На рис. 2 представлена общая схема сквозного мультифизичного моделирования на основе связанно-сопряженного расчета с использованием расчетных модулей ЛОГОС-Аэро, ЛОГОС-Тепло и ЛОГОС-ПА.

Первым этапом моделирования выполняется связанный расчет с использованием расчетных модулей ЛОГОС-Аэро и ЛОГОС-Тепло, при котором на каждом расчетном шаге происходит обмен данными. В конце этапа расчетные модули сохраняют данные, необходимые для сопряжения с расчетным модулем ЛОГОС-ПА в файловый контейнер данных.

Вторым этапом моделирования выполняется передача данных из файлового контейнера в расчетный модуль ЛОГОС-ПА и сопряженный расчет.

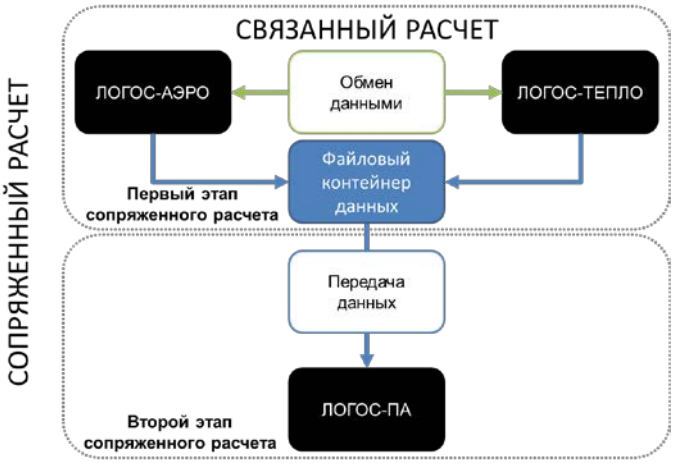


Рис. 2. Общая схема связанно-сопряженного расчета

Главной задачей обмена и передачи данных между расчетными модулями является их преобразование из специализированного формата представления данных одного расчетного модуля к специализированному формату представления данных другого расчетного модуля.

Схема связанно-сопряженного расчета на базе Платформы

Рассмотрим подход к решению этой задачи на базе Платформы. На рис. 3 представлен аналог схемы связанно-сопряженного расчета. Суть подхода заключается в подключении к расчетным модулям легковесных модулей.

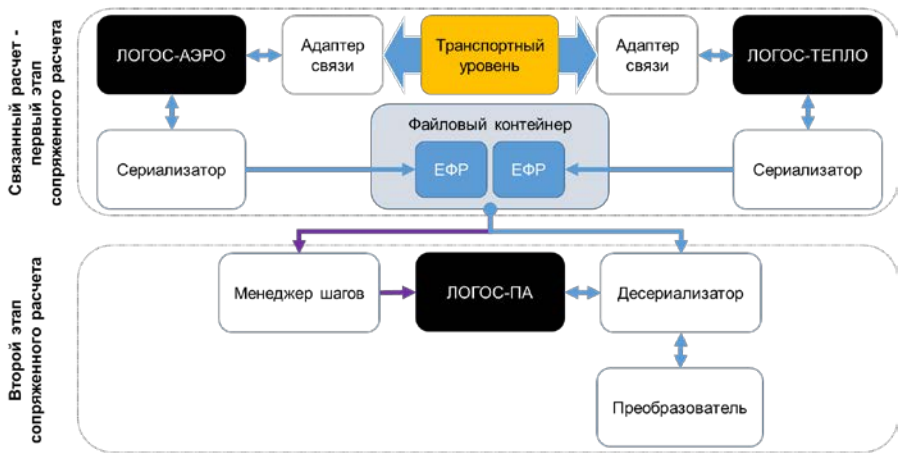


Рис. 3. Схема связанно-сопряженного расчета на базе Платформы

Первым этапом моделирования выполняется связанный расчет с использованием расчетных модулей ЛОГОС-Аэро и ЛОГОС-Тепло. С помощью Адаптеров связи, Транспортного уровня и обрабатывающих функций Платформа обеспечивает взаимодействие расчетных модулей во время счета, а именно межпроцессорный обмен данными по MPI между ЛОГОС-Аэро и ЛОГОС-Тепло. С помощью легковесного модуля Сериализатор Платформа обеспечивает сохранение расчетных данных ЛОГОС-Аэро и ЛОГОС-Тепло в файловый контейнер в формате единого файлового разреза ЕФР [2]. Файловый контейнер содержит информацию, необходимую для сопряжения с расчетным модулем ЛОГОС-ПА.

Вторым этапом моделирования выполняется сопряженный расчет с использованием расчетного модуля ЛОГОС-ПА. С помощью легковесных модулей Менеджер шагов и Десериализатор Платформа обеспечивает сопряжение расчетного модуля ЛОГОС-ПА с ЛОГОС-Аэро и ЛОГОС-Тепло.

Сопряжение осуществляется по расчетному времени. Менеджер шагов предоставляет ЛОГОС-ПА информацию о том, сколько расчетных шагов было рассчитано на первом этапе и на какие моменты времени были сохранены расчетные данные ЛОГОС-Аэро и ЛОГОС-Тепло в файловый контейнер. Десериализатор выполняет выборку необходимых данных из файлового контейнера и преобразование их к данным расчетного модуля ЛОГОС-ПА, далее расчетный модуль выполняет сопряженный расчет.

Ключевыми средствами обмена и передачи данных между расчетными модулями являются легковесные модули (Адаптер связи, Сериализатор, Десериализатор, Менеджер шагов) и Транспортный уровень.

Адаптер связи

Адаптер связи обеспечивает обмен данными между расчетным модулем и Транспортным уровнем согласно логике взаимодействия с другими расчетными модулями.

К основным этапам работы Адаптера относятся:

– Начальная инициализация.

– Обработка начала шага расчета.

– Обработка конца шага расчета.

– Завершение расчета.

На этапе начальной инициализации Адаптер связи получает из конфигурационного файла список интерфейсов обмена, инициализирует Транспортный уровень и менеджер времени для каждого активного интерфейса.

На этапе обработки начала шага расчета Адаптер связи выполняет следующее:

– Определяет активные интерфейсы.

– Получает из расчетного модуля обменные данные активных интерфейсов.

– Передает в Транспортный уровень обменные данные активных интерфейсов для отправки другим расчетным модулям.

– Передает в Транспортный уровень команду начала обмена.

На этапе обработки конца шага расчета Адаптер связи выполняет следующее:

– Определяет активные интерфейсы.

– Получает из Транспортного уровня обменные данные для активных интерфейсов от других расчетных модулей.

– Передает в расчетный модуль обменные данные для активных интерфейсов.

На этапе завершения расчета Адаптер связи передает в Транспортный уровень команду остановки процесса обмена.

Транспортный уровень и обрабатывающие функции

Транспортный уровень – это настраиваемая коммуникационная среда, которая обеспечивает обмен данными между расчетными модулями. На этапе инициализации Транспортного уровня для каждого расчетного модуля, участвующего в связанном расчете формируется структура объектов, которая представлена на рис. 4.

Узловым элементом структуры, через который расчетный модуль (через адаптер связи) взаимодействует с Транспортным уровнем, а далее с другими расчетными модулями (через их адаптеры связи), является Соединитель. Соединитель может содержать в себе несколько Соединений. Одно Соединение соответствует одному расчетному модулю, с которым текущий расчетный модуль осуществляет обмен данными. Таким образом, для каждой пары взаимодействующих расчетных модулей создается свое Соединение. Такой подход позволяет обмениваться данными одновременно более чем с двумя расчетными модулями, с использованием нескольких Интерфейсов связей для каждой пары Соединений.

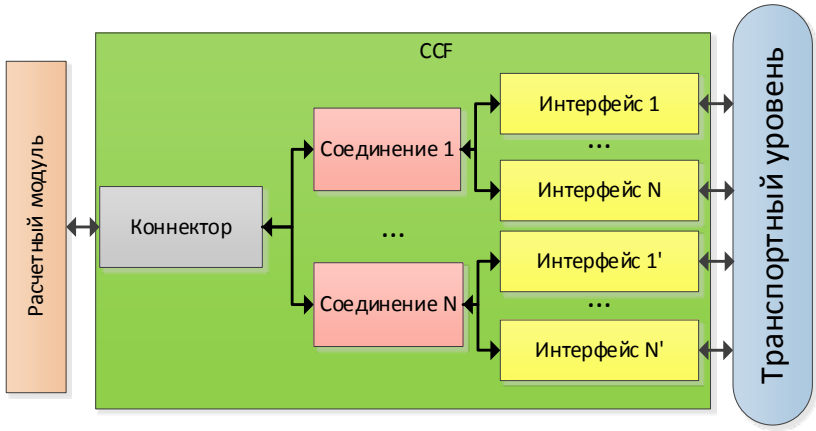


Рис. 4. Структура объектов транспортного уровня

Интерфейс является настраиваемым модулем, с помощью которого организуются цепочки обрабатывающих функций (ОФ) для обеспечения согласованности данных, передаваемых от одного расчетного модуля в другой. Общая схема Интерфейса представлена на рис. 5.

В качестве обрабатывающих функций выступают функции конверторов, интерполяторов и логгеров, которые конвертируют физические параметры, интерполируют сеточные величины с расчетной модели одного расчетного модуля на расчетную модель другого, и помогают анализу данных, проходящих через Транспортный уровень, соответственно.

Обрабатывающие функции располагаются в динамических библиотеках. Информация об именах динамических библиотек ОФ и последовательность их вызовов для каждого Интерфейса связи описывается в конфигурационном файле.

На рис. 6 представлен пример схемы взаимодействия объектов Интерфейса связи при связанном расчете двух расчетных модулей.

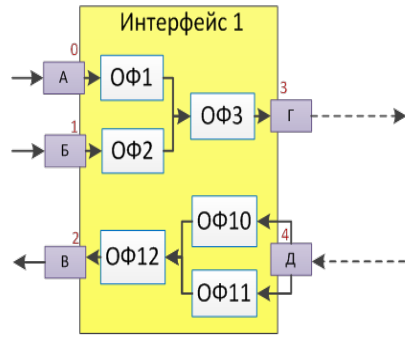


Рис. 5. Общая схема Интерфейса

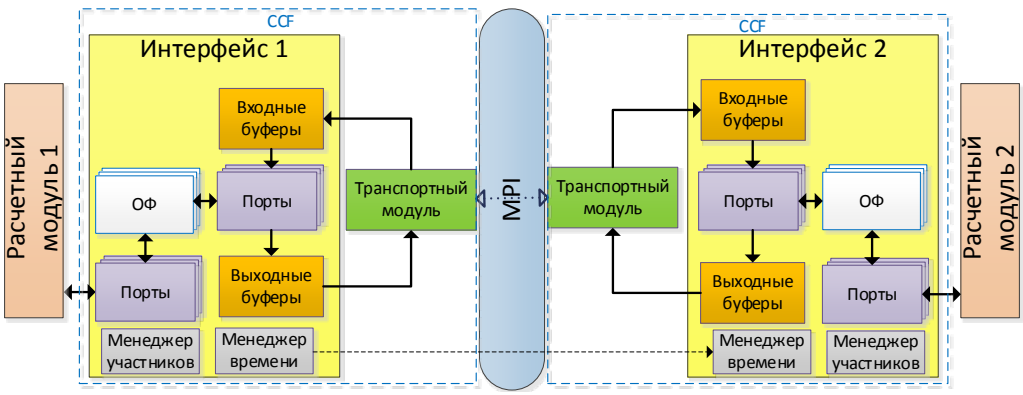


Рис. 6. Пример схемы взаимодействия объектов Интерфейса при связанном расчете двух расчетных модулей

Легковесный модуль Сериализатор

Легковесный модуль Сериализатор предназначен для чтения данных в специализированном формате из расчетного модуля, преобразования и записи данных в унифицированном формате в файловый контейнер в формате ЕФР.

К основным этапам работы Сериализатора относятся:

– Проверка шага расчетного модуля на соответствие стратегии работы.

– Выборка данных из расчетного модуля (например, вся или часть объемной сетки, вся или часть поверхностной сетки).

– Преобразование данных расчетного модуля (например, преобразование системы координат, преобразование системы единиц и т. п.).

– Запись данных в файловый контейнер средствами библиотеки ЕФР.

Преобразования системы координат

Подготовка расчетных моделей может осуществляться независимо для каждого расчетного модуля с учетом его специфических особенностей. И, следовательно, расчетные модели могут быть построены в различных системах координат и не совпадать геометрически, что приведет к возникновению проблем при передаче данных с одной расчетной модели на другую интерполяционными методами. Поэтому как в Сериализаторе, так и Десериализаторе поддержана предобработка расчетной модели источника, которая включает в себя следующие способы преобразований:

– Смещение вдоль оси OX.

– Смещение вдоль оси OY.

– Смещение вдоль оси OZ.

– Поворот вокруг оси OX.

– Поворот вокруг оси OY.

– Поворот вокруг оси OZ.

Если несогласованность расчетных моделей выявлена после первого этапа сопряженного расчета (получен файловый контейнер обменных данных), то преобразования систем координат можно настроить в Десериализаторе. Данная возможность не потребует повторного расчета, но преобразованная расчетная модель источника не сохраняется, и, следовательно, отсутствует возможность визуальной оценки, как корректности выполнения преобразования, так и корректности задания параметров преобразования.

Легковесный модуль Менеджер шагов

Легковесный модуль Менеджер шагов предназначен для сопряжения данных нескольких файловых контейнеров в формате ЕФР. Сопряжение может осуществляться как по номерам расчетных шагов, так и по времени расчетных шагов. В результате работы Менеджера шагов, расчетный модуль получает полную информацию о том, сколько в файловом контейнере временных шагов и на какие моменты времени.

Легковесный модуль Десериализатор

Легковесный модуль Десериализатор предназначен для чтения данных унифицированного формата из файлового контейнера в формате ЕФР и преобразования их в данные специализированного формата расчетного модуля.

К основным этапам работы Десериализатора относятся:

– Проверка шага расчетного модуля на соответствие стратегии работы.

– Выборка и чтение данных из файлового контейнера средствами библиотеки ЕФР.

– Преобразование данных файлового контейнера (например, преобразование системы координат, преобразование системы единиц и т. п.).

– Выборка данных из расчетного модуля (например, вся или часть объемной сетки, вся или часть поверхностной сетки).

– Преобразование данных файлового контейнера к данным расчетного модуля.

– Установка преобразованных данных в расчетный модуль.

Преобразование данных файлового контейнера к данным расчетного модуля

Основной задачей Десериализатора является преобразование данных файлового контейнера к данным расчетного модуля. На текущий момент поддержано несколько методов преобразования, которые отличаются целевым назначением:

– Чтение по локальным индексам (без интерполяции).
– Чтение по глобальным индексам (без интерполяции).
– С помощью поверхностного интерполятора FSI.
– С помощью универсального интерполятора POINTS.

Чтение по локальным индексам (без интерполяции)

Данный метод предназначен для продолжения счета в рамках расчетного модуля. Метод не является универсальным и может применяться, при условии, что расчетная модель и декомпозиция расчетной модели соответствуют расчетной модели и декомпозиции расчетной модели в обменных файлах.

Чтение по глобальным индексам (без интерполяции)

Данный метод предназначен для перезапуска счета в рамках расчетного модуля. Метод не является универсальным и может применяться, при условии, что расчетная модель соответствует расчетной модели в обменных файлах. Доступ к данным в обменных файлах осуществляется по глобальным индексам сеточных элементов, которые являются уникальными для конкретной расчетной модели.

С помощью поверхностного интерполятора FSI

Данный метод предназначен для передачи данных с поверхностной расчетной модели из одного расчетного модуля в другой. На текущий момент поддержано два способа интерполяции:

– Из центров ячеек сетки источника в центры ячеек сетки приемника (применяется метод обратных взвешенных расстояний для центральных точек).
– Из центров ячеек сетки источника в узлы ячеек сетки приемника (применяется метод взвешенных площадей пересечения поверхностей).

С помощью универсального интерполятора POINTS

Это интерполятор по точкам, соответственно источником и приемником являются множества точек. Для интерполяции данных используется метод обратных взвешенных расстояний. Чем точка источник ближе к точке приемнику, тем больше ее вклад (вес) в интерполируемое значение.

Вес точки источника – обратная функция расстояния во второй степени.

Универсальность интерполятора заключается в том, что нет ограничения на способы поиска точек в сетке источнике и сетке приемнике и нет ограничений на сетки источника и приемника (поверхностная или объемная). Таким образом, интерполятор подходит для работы, как с поверхностной, так и с объемной сеткой.

На текущий момент поддержаны следующие способы интерполяции:

– Из центров ячеек сетки источника в центры ячеек сетки приемника.
– Из центров ячеек сетки источника в узлы ячеек сетки приемника.
– Из узлов ячеек сетки источника в центры ячеек сетки приемника.
– Из узлов ячеек сетки источника в узлы ячеек сетки приемника.

Подготовка задачи связанно-сопряженного расчета в Интеграторе

Интегратор – это графический интерфейс пользователя, который позволяет настраивать одиночные, связанные, сопряженные задачи, обеспечивает локальный и удаленный запуски задач с контролем хода расчета и возможностью анализа полученных результатов.

Подготовка задачи связанно-сопряженного расчета в общем виде выполняется в 4 этапа:

– Подготовка одиночных задач, которая включает в себя подключение и настройку легковесных модулей: Сериализатор, Десериализатор, Менеджер шагов).

– Подготовка связанной задачи, которая включает в себя: настройку общей схемы связанного расчета, связывание доступных обменных интерфейсов, конфигурирование связанных обменных интерфейсов.

– Подготовка сопряженной задачи;

– Создание и конфигурирование схемы расчета.

Для запуска задачи нужно выделить схему расчета и из контекстного меню выбрать пункт меню «Запустить расчет».

Опыт практического применения

Отработка связанно-сопряженного расчета на базе Платформы с использованием расчетных модулей ЛОГОС-Аэро, ЛОГОС-Тепло и ЛОГОС-ПА была выполнена на тестовой задаче о деформировании клапана под действием потока газа в канале. На рис. 7 представлена схема связанно-сопряженного расчета. На рис. 8 представлены исходные данные на втором этапе сопряженного расчета, результаты передачи данных интерполяционными методами с помощью легковесного модуля Десериализатор и результат расчета по расчетному модулю ЛОГОС-ПА.

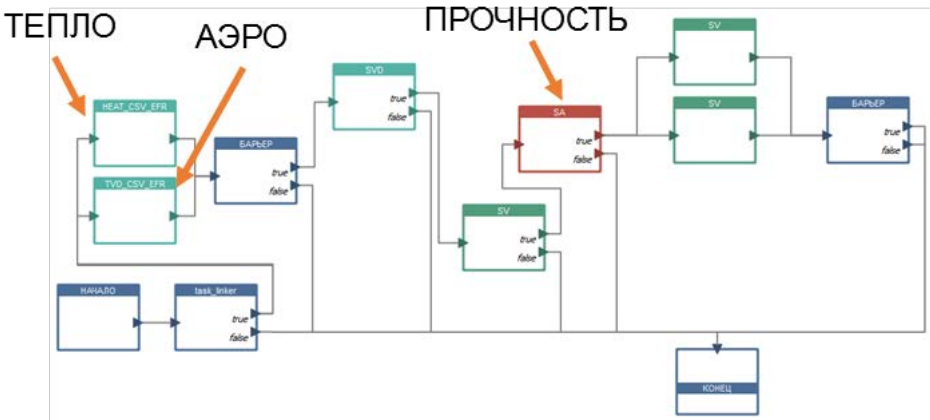


Рис. 7. Схема связанно-сопряженного расчета

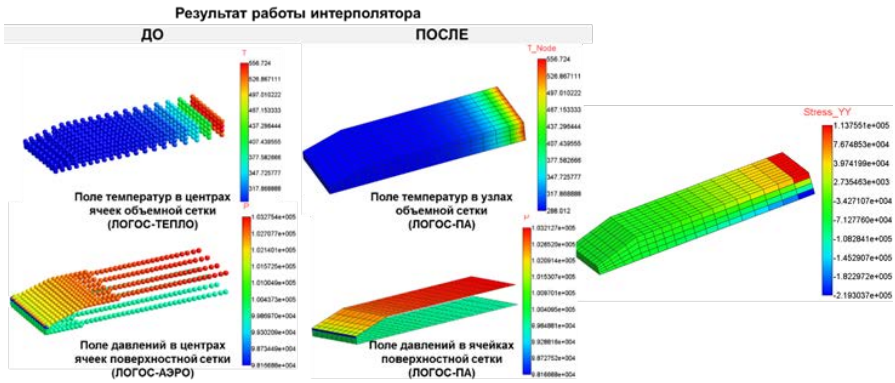


Рис. 8. Результаты связанно-сопряженного расчета тестовой задачи

Литература

1. Дерюгин Ю. Н., Козелков А. С., Спиридонов В. Ф., Циберев К. В., Шагалиев Р. М. Многофункциональный высокопараллельный пакет программ ЛОГОС для решения задач тепломассопереноса и прочности // Сборник тезисов докладов Санкт-Петербургского научного форума "Наука и общество", Санкт-Петербург, 2012.
2. Олесницкая К. К., Антипин И. А., Петрова М. А. Коллективный доступ к файловым данным на вычислительных системах с различной архитектурой в библиотеке ЕФР // XVII Международная конференция "Супервычисления и математическое моделирование": Сборник докладов. Саров, 15–19 октября, 2010. С. 392–400.

"LOGOS" MODULAR INTEGRATION PLATFORM. TECHNOLOGY FOR SETTING UP AND EXECUTING MULTIPHYSICS SIMULATIONS BASED ON COUPLED AND CONJUGATE CALCULATIONS

K. K. Olesnitskaya, A. V. Dyakov, N. A. Poburinnaya, A. A. Tyundina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

Modular integration platform (LOGOS-MIP) is an integrated environment for end-to-end multiphysics simulations based on coupled, conjugate and step-by-step calculations and parametric and optimization analysis. The platform includes: software-based integration tools; light-weight modules; transport layer; processing functions; file container and ad-hoc libraries. The key function of the software-based tools is to enable communication between computational modules during joint simulations. The communication can be performed both through the memory and through the file container. The ad-hoc libraries (like interpolators and converters) enable data format conversion during communication.

Key words: LOGOS-MIP, platform, coupled and conjugate simulation, multiphysics simulation, computational and lightweight modules, communication adapter, serializer, deserializer, step manager, interpolators.

УДК 519.6; 538.9; 67.02
DOI 10.53403/9785951505071_2022_350

ПРОГРАММНЫЙ КОМПЛЕКС «ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР»: ОТ КОНЦЕПЦИИ К РЕАЛИЗАЦИИ

В. В. Попов, А. Н. Быков, Ю. Н. Дерюгин, М. В. Ветчинников, Д. Ю. Дьянов, С. Н. Полищук, В. Г. Куделькин, Е. А. Синицин, А. Н. Бахаев, М. В. Журавлева, О. И. Бритова, М. А. Титов, А. В. Баранов, Л. В. Гамов, И. Г. Милешин, В. Ю. Кузнецов, П. А. Машенькин, Е. Ф. Голубкова, А. С. Данилов, М. В. Медведкина, А. А. Каякин, А. И. Ломайкин, О. В. Сапронова, Е. А. Веселова, А. Б. Емельянов, А. А. Лазарев

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В ФГУП «РФЯЦ-ВНИИЭФ» в 2019 году были начаты работы по созданию программного комплекса «Виртуальный 3D-принтер». Результатом работ в 2021 году должен стать комплекс программного обеспечения для моделирования физических процессов, протекающих при селективном лазерном плавлении (СЛП) с целью прогнозирования структуры, свойств материалов, а также получения изделий с заданными свойствами, и проведения топологической оптимизации изделий («Виртуальный 3D-принтер»). Данные работы направлены на создание отечественного аналога иностранного CAE программного обеспечения в области лазерных аддитивных технологий, которые появились на рынке в 2015–2019 годах.

В докладе приведены сведения об основных концептуальных и архитектурных решениях, выбранных при создании программного комплекса «Виртуальный 3D - принтер», приведено описание реализации функциональности программы в части работы с геометрией деталей с использованием топологической оптимизации и решетчатых структур, рассмотрены вопросы интеграции базы данных по свойствам материалов и различных расчетных модулей.

Ключевые слова: аддитивные технологии, селективное лазерное плавление, многоуровневое математическое моделирование, топологическая оптимизация, решетчатые структуры, генеративный дизайн, неявное представление геометрии, управление технологическим оборудованием.

Введение

В настоящее время огромный практический интерес в области лазерных аддитивных технологий представляет решение следующих задач:

– создание деталей сложных форм посредством генеративного дизайна с использованием топологической оптимизации и решетчатых структур, что позволяет более эффективно использовать материал (за счет этого достигается его экономия), создавать облегченные конструкции, уменьшать стоимость и время изготовления деталей;

– предсказание микро- и макросвойств изготавливаемых деталей (прежде всего механических), минимизация остаточных напряжений, короблений и деформаций, обеспечение уровня свойств на заданном уровне;

– выбор оптимальных режимов работы оборудования с целью обеспечения качества синтезируемых деталей, повышения скорости, снижения количества использованного порошка, минимизации брака при производстве.

Решение указанных задач возможно с использованием методов многоуровнего математического моделирования физико-химических процессов в лазерном аддитивном производстве, соединении методов математического моделирования с методами создания геометрии деталей, а также с методами управления технологическим оборудованием. В период 2015 – 2019 годов ряд крупных мировых компаний, специализирующихся на создании CAD/CAE/CAM программного обеспечения предложили первые версии своих программных продуктов для решения этих задач.

Во ФГУП «РФЯЦ-ВНИИЭФ» ведется разработка аналогичного программного обеспечения – «Виртуальный 3D-принтер».

Цель работы – создание комплекса программного обеспечения «Виртуальный 3D-принтер» для моделирования физических процессов, протекающих при селективном лазерном сплавлении с целью прогнозирования структуры, свойств материалов, а также получения изделий с заданными свойствами, и проведения топологической оптимизации изделий.

В докладе представлены: архитектура программного комплекса; некоторые структуры данных; алгоритмы моделирования на макро и мезоуровнях с указанием математических моделей, которые планируется использовать; методы интеграции модулей в единый программный комплекс; взаимосвязи модулей; приведены примеры реализации модулей блоков «Интегрирующая оболочка» и «Проектирование».

Структура программного комплекса «Виртуальный 3D-принтер»

Программный комплекс состоит из двадцати модулей, обеспечивающих различную функциональность и объединенных в четыре блока. Важным вопросом является интеграция модулей под единым удобным графическим интерфейсом пользователя.

Все модули можно разделить на две группы. Первая группа модулей это модули блоков «Интегрирующая оболочка», «Проектирование», «Производство», которые обеспечивают различную функциональность при работе с объектами геометрии (создание и редактирование, импорт/экспорт), подготовку деталей к печати или моделированию, включая топологическую оптимизацию, а также для просмотра результатов (рис. 1). Эти модули представляют собой единую программу, предоставляющую графический интерфейс пользователя создаваемого программного комплекса «Виртуальный 3D-принтер» и основой их является модуль «Интегрирующая оболочка». Вторая группа модулей это модули для топологической оптимизации и моделирования физических процессов на макроуровне (уровень детали или ее частей) и мезоуровне (уровень отдельной формируемой дорожки, включая ванну расплава и процессы кристаллизации, роста зерен, дендритов и ламелей). Эти модули в зависимости от необходимой детализации моделирования физических процессов, целевой платформы (настольные компьютеры, рабочие станции, суперкомпьютеры), операционной системы и ряда других факторов представляют собой совокупность исполняемых файлов, различных версий, взаимодействующих между собой посредством модульной интеграционной платформы (ЛОГОС-МИП) или посредством прямых вызовов с обменом данными через общую память или файлы обмена данными. На рис. 2 показана схема изложенного выше. Ниже отдельно опишем построение графической части программного комплекса (графического интерфейса) из различных модулей на базе модуля «Интегрирующая оболочка» и расчетной части программного комплекса на базе модулей блока «Моделирование».

Модули блоков «Интегрирующая оболочка», «Проектирование», «Производство»

В модуле «Интегрирующая оболочка» реализованы следующие идеи и подходы. Модуль построен как многооконное графическое приложение (рис. 1), в котором в каждом окне можно строить собственные детали или их части, производить подготовку детали к производству, выполнять топологическую оптимизацию или математическое моделирование. Общие элементы – панели кнопок и боковые панели – позволяют работать с объектами в текущем окне. Между окнами можно осуществлять обмен деталями и данными путем копирования и вставки. Формирование деталей и их

частей, задание этапов подготовки к производству и математическому моделированию в любом окне осуществляется путем создания последовательности операций (команд), каждая из которых задается собственным пользовательским диалогом, и после выполнения формирует собственный результат, который отображается в текущем окне и доступен для последующего выбора в других операциях (внутри диалоговых окон). Любую из предыдущих операций можно посмотреть, отредактировать и выполнить заново, что позволяет, в частности, при формировании сложных деталей поменять параметры отдельных операций и сгенерировать заново результат всех последующих операций, что приведет к учету сделанных изменений. Такая тактика построения деталей позволяет не только отслеживать цепочку построения деталей и иметь возможность перестроить детали с измененными параметрами, но и реализовать внутренний скриптовый язык описания последовательности построения деталей, а также использовать этот скриптовый язык для компактной записи деталей на диск в форме описания цепочки операций, приводящих к построению деталей (что отчасти реализовано в текущей версии).

Для представления объектов геометрии и для работы с ними используется несколько различных структур данных и соответствующие алгоритмы. Структуры данных позволяют работать с объектами, заданными в виде:

– фасеточной поверхности (совокупность фасетов, являющихся выпуклыми N-угольниками, для STL это трехугольные фасеты);

– балочного представления (совокупность узлов и соединяющих их ребер, с информацией о диаметре ребер и количестве граней);

– деревьев объектов неявной геометрии (совокупность объектов в виде дерева, описывающих построение деталей из множеств посредством логических операций, с возможностью изменения формы множеств в зависимости от дополнительных параметров, а также с определенной целевой функцией, значение которой для произвольной точки пространства позволяет определить принадлежность детали и расстояние до поверхности) [1, 2, 3, 4];

– структур с сеточной информацией для моделирования и (или) для топологической оптимизации (сетка и исходная или результирующая геометрия) [5];

– массивов из любых объектов четырех предыдущих представлений.

Наличие этих представлений геометрических объектов следует из требований, предъявляемых к функциональности программного комплекса. Визуализация любых геометрических объектов осуществляется в фасеточном или балочном представлении. Поскольку деревья неявной геометрии абстрактно описывают построение геометрических объектов, то для визуализации их в программе есть код для построения фасеточного представления деревьев неявной геометрии.

Операции, задаваемые в диалоговых окнах, используют и формируют тот или иной вид (представление) объектов геометрии, посредством алгоритмов, позволяющих выполнять действия над объектами геометрии. Здесь, на уровне кода выполняемого операциями, интегрирована функциональность отдельных модулей: «Генерация решеток» (функциональность для генерации различных типов решетчатых структур); «Заполнение геометрии решетками» (функциональность для использования различных типов решетчатых структур); «Сглаживатель» (функциональность для сглаживания результатов топологической оптимизации); «Производство» (функциональность подготовки деталей к печати и данных для моделирования процессов печати); «Интегрирующая оболочка» (функциональность для построения деталей); «УРС» (функциональность для доступа к базе данных по свойствам материалов, расчетным данным с мезоуровня и к технологическим картам).

На рис. 2–6 приведено схематическое изображение изложенного выше. На рис. 2 приведена схема деления всех модулей программного комплекса на две части – модули собранные в единый графический интерфейс с использованием модуля «Интегрирующая оболочка» и модули блока «Моделирования», отвечающие за моделирование различных процессов. На рис. 3 и 4 приведена детализация части схемы с рис. 2, показывающая схему модуля «Интегрирующая оболочка» и структуры хранения информации в дочернем окне об операциях и геометрических объектах, а также схема интеграции других модулей блоков «Проектирование» и «Производство». Пример задания последова-

тельности операций для построения сложной детали (образец для стандартных испытаний на растяжение со вписанной решетчатой структурой) приведен на рис. 5. На рис. 6 проиллюстрирована реализация идеи построения дерева описания геометрии в неявном виде, подробнее в статье [4].

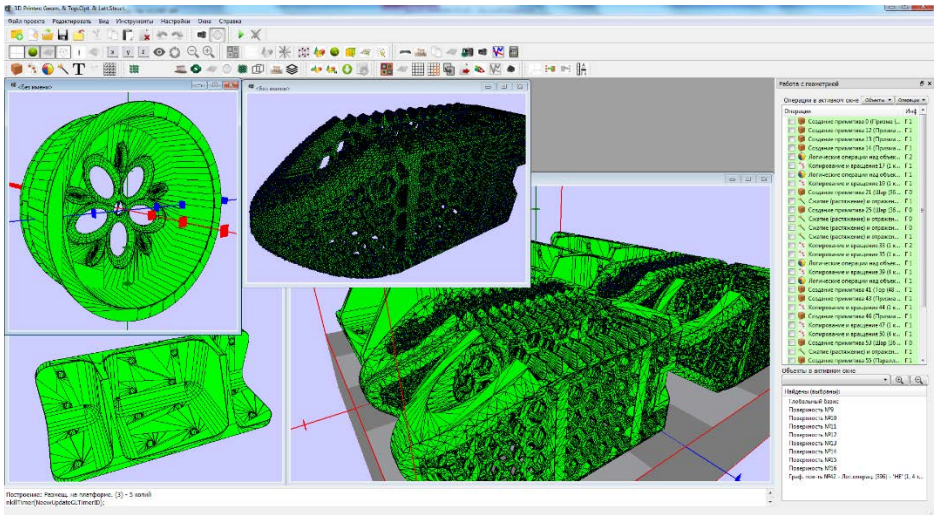


Рис. 1. Графический интерфейс комплекса программ (модуль «интегрирующая оболочка»), примеры построения и подготовки деталей к расчетам и печати



Рис. 2. Взаимосвязь модулей блока «Моделирование» и модулей интегрированных в единый графический интерфейс программного комплекса

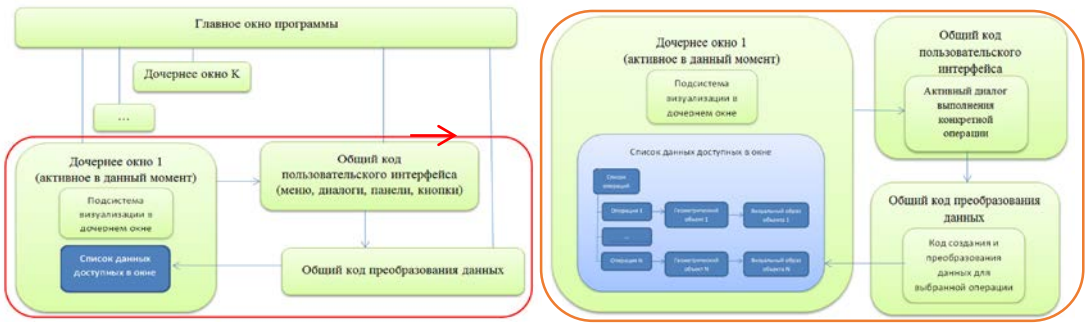


Рис. 3. Схема модуля «Интегрирующая оболочка» и структуры хранения информации в дочернем окне об операциях и геометрических объектах



Рис. 4. Схема интеграции других модулей блоков «Проектирование» и «Производство» в модуль «интегрирующая оболочка»

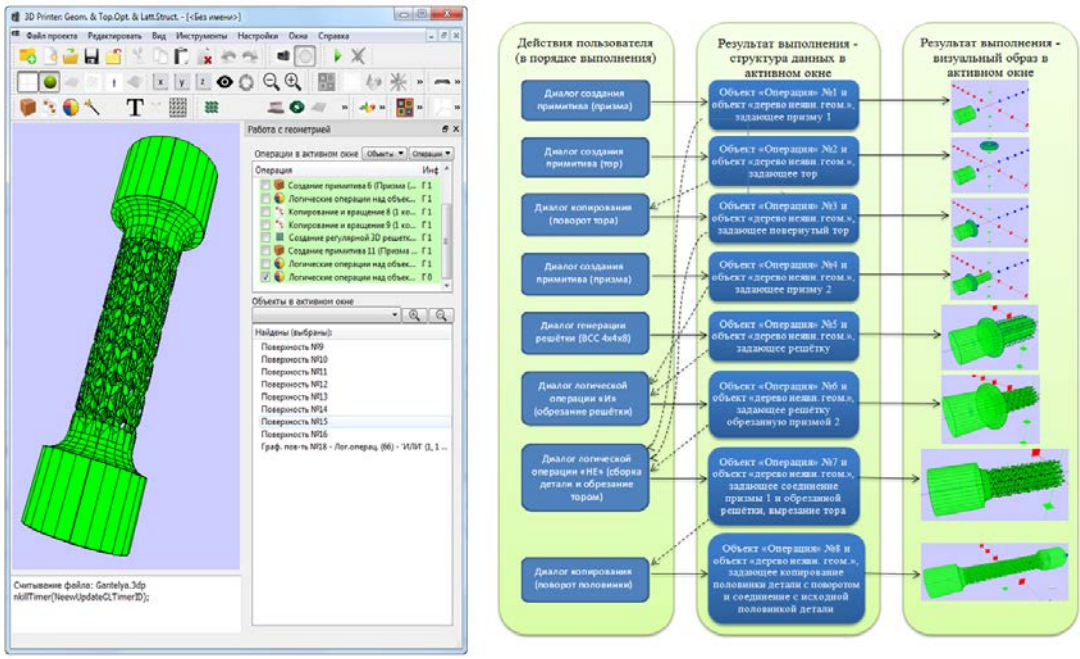


Рис. 5. Пример задания последовательности операций для построения сложной детали (образец для стандартных испытаний на растяжение со вписанной решетчатой структурой)

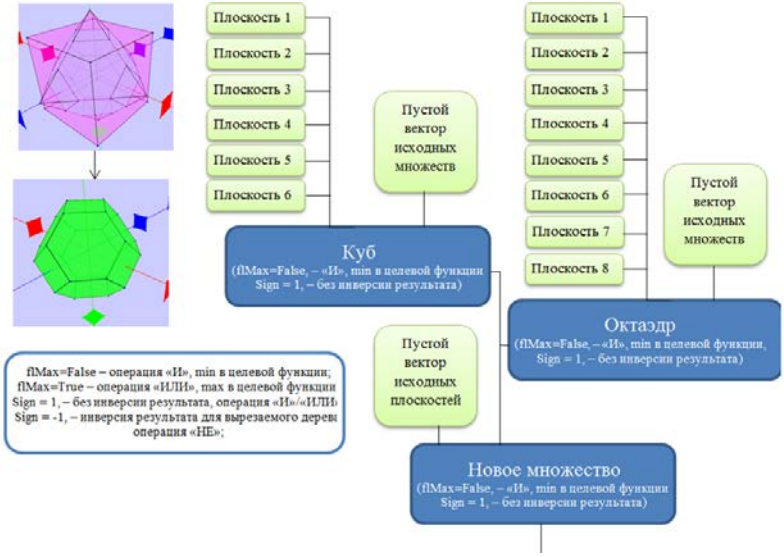


Рис. 6. проиллюстрирована реализация идеи построения дерева описания геометрии в неявном виде (подробнее в статье [4])

В настоящее время интегрированные в общий код модули блоков «Интегрирующая оболочка», «Проектирование», «Производство» позволяют следующее.

1) Создавать геометрические примитивы (8 типов) и строить сложные детали с помощью логических (булевых) операций, операций копирования и некоторых других.

2) Импортировать и экспортировать геометрию деталей в виде STL файлов.

3) Генерировать и использовать различные типы решетчатых структур в балочном или объемном представлении (44 типа трехмерных решеток вписанных в объемные ячейки разной формы; 9 типов двумерных решеток вписанных в плоские ячейки и вытянутые по третьему направлению; 6 типов TPMS решеток – самоподдерживающиеся поверхности суммарной нулевой кривизны и решетки производные от них; решетки со случайными параметрами и ориентацией балок; решетка с учетом градиента заданных величин).

4) Подготавливать данные, в том числе и сеточные (воксельная сетка, состоящая из множества ячеек в форме параллелепипеда), для математического моделирования (виртуальные испытания) и для топологической оптимизации. Выгружать в файлы и считывать эти данные.

5) Подготавливать созданные или загруженные детали к печати на аддитивном оборудовании с генерацией управляющего кода или данных для математического моделирования физических процессов при печати на макро и мезоуровнях.

6) Считывать данные после топологической оптимизации и строить по ним детали новой формы, в том числе в виде дерева неявной геометрии [5].

7) Выполнять запросы к базе данных по свойствам материалов (модуль «УРС» – уравнения состояния) с целью просмотра и использования в расчетах.

Пример использования реализованной функциональности этих блоков представлен на рис. 1.

Модули блока «Моделирование»

Все модули блока «Моделирование» можно разделить на три группы модулей (рис. 7).

1) Модули для моделирования на макроуровне (уровень детали и ее фрагментов): «Тепло макроуровень»; «Прочность макроуровень»; «Коробление»; «Остаточные напряжения».

2) Модули для моделирования на мезоуровне (уровень отдельной выплавляемой дорожки с возможной детализацией до зеренной и кристаллической структур): «Тепломассоперенос мезоуровень»;

«Прочность мезоуровень»; «Клеточные автоматы»; «Термомеханика»; «Микросегрегация»; «Микроструктура»; «Фазовые переходы и микроструктура».

3) Модули для получения и использования экспресс-моделей (зависимости для быстрого вычисления величин при использовании на макроуровне, которые заранее получены из сложных детальных трехмерных расчетов на мезоуровне): «Теплофизические свойства»; «Геометрия зоны сплавления»; «Усадка и дефекты». Модуль «УРС» для хранения расчетно-экспериментальных и литературных данных по свойствам материалов (модуль «УРС»).

Каждая указанная группа модулей обеспечивает расчеты физических процессов на определенном уровне. Каждый модуль содержит реализацию алгоритмов для моделирования отдельных физических процессов. Для моделирования совокупности физических процессов на определенном уровне (макро, мезо) создаются алгоритмы и их программные реализации (расчетные коды в виде исполняемых файлов) с использованием указанных модулей, для решения конкретных проблем лазерного аддитивного производства и для выбора оптимальных параметров печати и режимов работы оборудования.

Работа со счетными модулями организована посредством подготовки необходимых данных для расчетов при задании операций (команд) в графическом интерфейсе модуля «Интегрирующая оболочка» (рис. 8) и последующего вызова расчетных кодов (топологическая оптимизация, виртуальные испытания, расчеты на макроуровне или мезоуровне). Указанные расчетные коды в виде исполняемых файлов могут быть различных версий ввиду различной детализации процессов, реализации под различные платформы и операционные системы.

Для взаимодействия разнородных расчетных модулей между собой внутри расчетных кодов для моделирования процессов макроуровня или мезоуровня (модули созданные различными организациями соисполнителями проекта), в том числе при исполнении на разнородных сетевых ресурсах (в том числе на суперкомпьютерах и облачных ресурсах), предполагается использовать разрабатываемую во ФГУП «РФЯЦ-ВНИИЭФ» технологию «Модульная интеграционная платформа» (ЛОГОС-МИП) внутри локальных укрупненных расчетных модулей, отвечающих за моделирование совокупности процессов на данном уровне (макроуровне или мезоуровне). При этом для тестирования и демонстрации работоспособности на стадии разработки (для работы на локальном компьютере под ОС семейства Windows) для связи модулей используются прямые вызовы и обмен через общую память и файлы данных.

Отметим, что вызов указанных исполняемых файлов может осуществляться как из «Интегрирующей оболочки», например, при необходимости расчетным путем получить свойства формируемого материала в расчетах на мезоуровне, так и из других расчетных модулей, например, при необходимости во время расчета на макроуровне иметь какие-либо величины из расчета на мезоуровне. Вся совокупность модулей на рис. 7 и расчетные коды в синей рамке на рис. 8 представляют собой содержимое синего прямоугольника со схемы на рис. 2.

Алгоритм вычислений на макроуровне обеспечивает моделирование процессов печати с учетом послойного формирования деталей в объеме порошкового материала. Основной вычисляемый процесс это теплоперенос (модуль «Тепло макроуровень») в области, содержащей материалы с разными свойствами (порошок и сплавленный материал) с учетом взаимодействия лазерного излучения с веществом и граничными условиями, описывающими конвективный теплообмен и обмен излучением. Отличительной особенностью является изменение пористости порошка в зоне сплавления (спекания), что приводит к потребности выделять зоны, где сформировался сплавленный материал с совершенно иными свойствами, а также зависимость всех теплофизических свойств от температуры и пористости.

При выполнении моделирования на макроуровне необходимо периодически (реже чем моделируемое послойное формирование детали) моделировать напряженно-деформированное состояние (НДС) сплавленного твердого тела (модуль «Прочность макроуровень») с учетом термонапряжений и механических свойств полученных в зависимости от пористости и температурной предыстории (зависимость температуры и градиента температуры от времени), которая определяет формируемую микроструктуру (зеренная и кристаллическая структура). После вычисления НДС необходимо оценивать отклонение от исходной геометрии формируемых деталей (модуль «Коробление») и рассчитывать уровень остаточных напряжений с учетом релаксации напряжений (модуль «Остаточные напряжения»).

Указанная зависимость теплофизических свойств материалов от пористости и температуры содержится в виде полученных зависимостей (экспресс-моделей) в модуле «Теплофизические свойства». Фазовые переходы при плавлении и кристаллизации описываются особенностями в температурных зависимостях свойств, используемых на макроуровне. Полученные кинетические уравнения изменения пористости связанной с усадкой порошкового материала при взаимодействии лазерного излучения с веществом (тоже экспресс-модели) содержатся в модуле «Усадка и дефекты». В случае если для выбранных материалов есть литературные и экспериментальные данные, то они используются с помощью базы данных (БД) по свойствам материалов и модуля «УРС». Учет механических свойств в зависимости от пористости и температуры определяется посредством модуля «Теплофизические свойства», а учет полученной микроструктуры в зависимости от температурной предыстории осуществляется посредством данных из модуля «УРС», где в БД отдельно хранятся рассчитанные и оцифрованные KGT диаграммы (по фамилиям авторов: Курц–Джиованола–Тривиди). Эти диаграммы связывают температуры и скорости кристаллизации (зависят от градиентов температуры) с типами формируемой микроструктуры и значениями механических свойств. Расчет этих диаграмм производиться на мезоуровне один раз для заносимого в базу данных модуля «УРС» материала.

Модуль «Геометрия зоны сплавления» содержит зависимости (экспресс-модели), которые рассчитаны на мезоуровне и используются при подготовке детали к печати и позволяют ответить на вопросы о выборе ряда параметров печати, таких как толщина слоя порошка, расстояние между дорожками, удельная мощность и оценить качество поверхности и объемную скорость построения детали.

Моделирование на макроуровне должно позволить выбрать оптимально параметры печати, оценить качество изготовления деталей – величины деформаций и остаточных напряжений, уровень механических свойств, пористость, тип микроструктуры. Предсказать появление брака. В перспективе эти результаты могут быть использованы для оптимизации стратегии построения – треков и формируемой микроструктуры (уровня свойств). Входной информацией является заданная расчетная сетка, материалы (свойства порошковых и сплавленных материалов, различные экспресс-модели свойств), послойная временная последовательность треков построения с указанием удельной мощности лазеров. Выходные данные это поля температур, напряжений и деформаций, пористости, типа микроструктуры, различных свойств.

Алгоритм вычислений на мезоуровне обеспечивает моделирование процессов печати в единичном треке (дорожке) с учетом насыпания слоя порошка, взаимодействия лазерного излучения с веществом, формирования ванны расплава, испарения вещества, протекания процессов тепломассопереноса многокомпонентной жидкости в ванне расплава, кристаллизации, формирования микроструктуры с учетом термонапряжений и микросегрегации (компонентов и примесей). Основной вычисляемый процесс это тепломассоперенос (модуль «Тепломассоперенос мезоуровень») в порошковом слое с расчетом НДС в сплавленном твердом теле (модуль «Прочность мезоуровень»). Расчет формирования микроструктуры (зеренной и кристаллической) осуществляется с помощью модулей «Клеточные автоматы», «Термомеханика», «Микросегрегация», «Микроструктура» в случае больших вычислительных мощностей или с помощью модуля «Фазовые переходы и микроструктура» в случае если достаточно иметь менее точный и более быстрый расчет микроструктуры. В расчетах тепломассопереноса на мезоуровне также надо учитывать нелиненйные зависимости свойств материалов, которые содержаться в модуле «УРС».

Моделирование на мезоуровне должно позволить выбрать оптимальные параметры печати, которые приводят к формированию качественных треков (дорожек), смоделировать формирование микроструктуры из первопринципов, получить KGT диаграммы и экспресс-модели для наполнения модулей «Теплофизические свойства», «Геометрия зоны сплавления», «Усадка и дефекты», «УРС». Входной информацией является заданная геометрия насыпанного слоя порошка (специальный алгоритм), зависимости теплофизических свойств кристаллических материалов, параметры лазерного излучения (мощность, скорость лазера, диаметр пятна, распределение энергии в пятне), расчетная сетка или расстановка частиц (для метода SPH). Выходные данные это поля температур, концентраций, напряжений и деформаций, усредненные зависимости различных свойств и экспресс-модели, микроструктура.

Расчеты на мезоуровне носят в основном исследовательский характер и нужны для работы с новыми порошковыми материалами, отработки технологии изготовления деталей из новых материалов. Расчеты на макроуровне это типичные расчеты при отработке технологии печати конкретных деталей из хорошо исследованного материала.

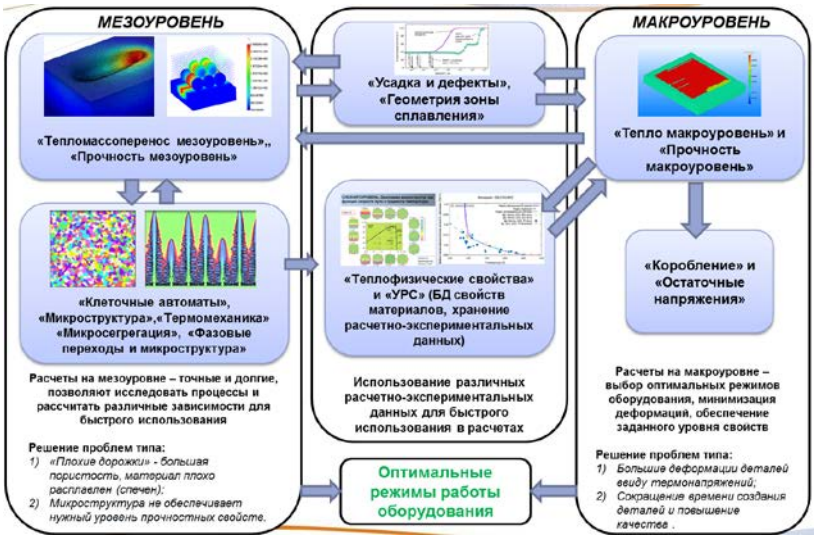


Рис. 7. Схема взаимосвязи всех расчетных модулей блока «Моделирование»

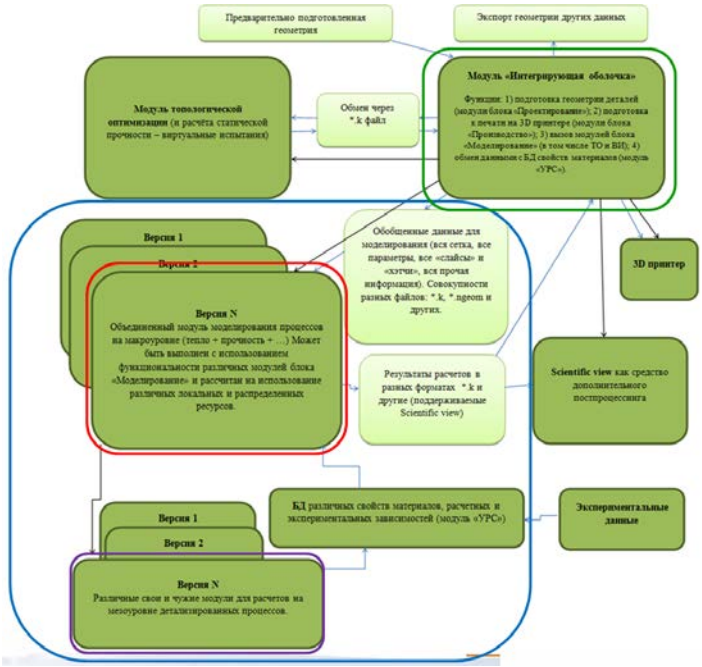


Рис. 8. Схема взаимодействия расчетных кодов для макро и мезоуровня с графическим интерфейсом комплекса программ

Сценарии работы в программном комплексе

Основные сценарии работы пользователей с программным комплексом «Виртуальный 3D-принтер» предполагаются следующие.

– Конструктор (дизайнер) – создание или загрузка геометрии деталей, редактирование и подготовка деталей к аддитивному производству с помощью топологической оптимизации и использования решетчатых структур, выполнение виртуальных испытаний на прочность.

– Ученый (исследователь) – многомасштабное математическое моделирование с целью поиска оптимальных режимов работы оборудования, технологических зависимостей для новых материалов, предсказания результатов изготовления деталей.

– Технолог – подготовка деталей к аддитивному производству, поиск и выбор оптимальных режимов работы оборудования, предсказание результатов изготовления деталей.

– Оператор – экспериментальное или серийное изготовление деталей.

На рис. 9 приведены схемы иллюстрирующие работы по данным сценариям.

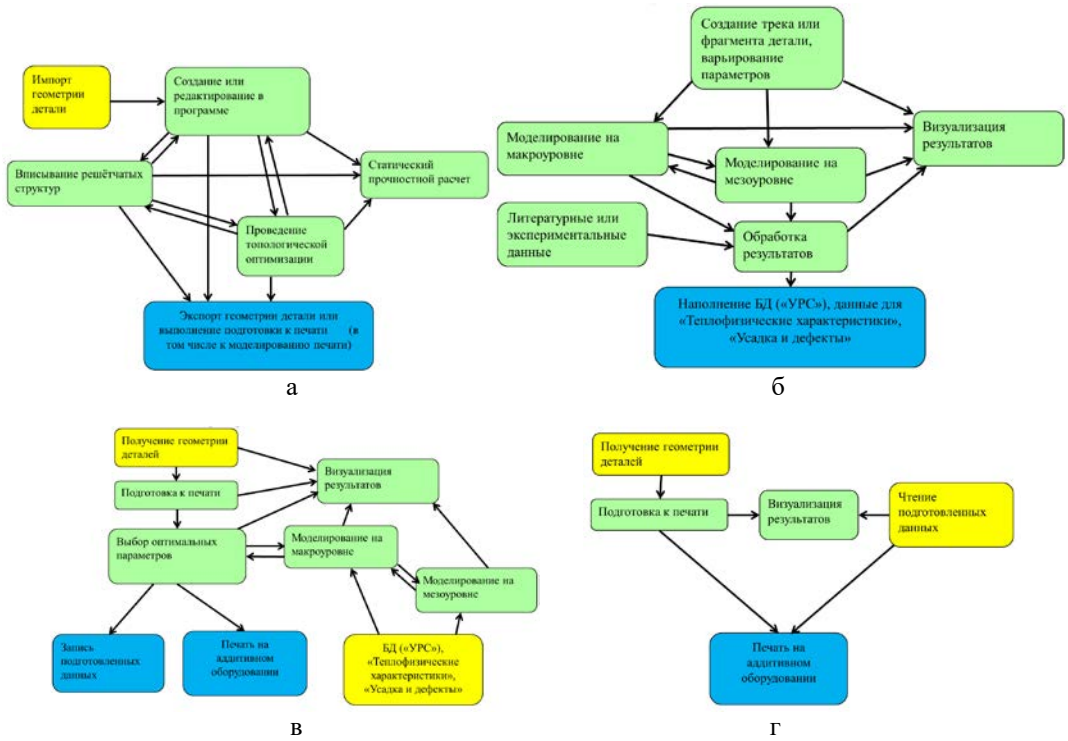


Рис. 9. Типичные сценарии работы различных пользователей в программном комплексе (а – конструктор, дизайнер, б – ученый, исследователь, в – технолог, г – оператор, желтым – входные данные, синим – выходные данные, зеленым – различные этапы работы, стрелками переходы между этапами работы)

Поскольку большое количество расчетных модулей блока «Моделирование» и их параметры позволяют выполнять очень разнообразные расчеты, то в основном в зависимости от сценариев работы будем ориентироваться на следующие основные сценарии расчетов в программном комплексе «Виртуальный 3D-принтер» (в скобках некоторые характеристики и особенности этих расчетов):

– грубый выбор режимов работы оборудования и грубая оценка результатов печати (низкая детализация, высокая скорость, персональный компьютер, только макроуровень, упрощенный теплоперенос и прочность, без детализации физических процессов);

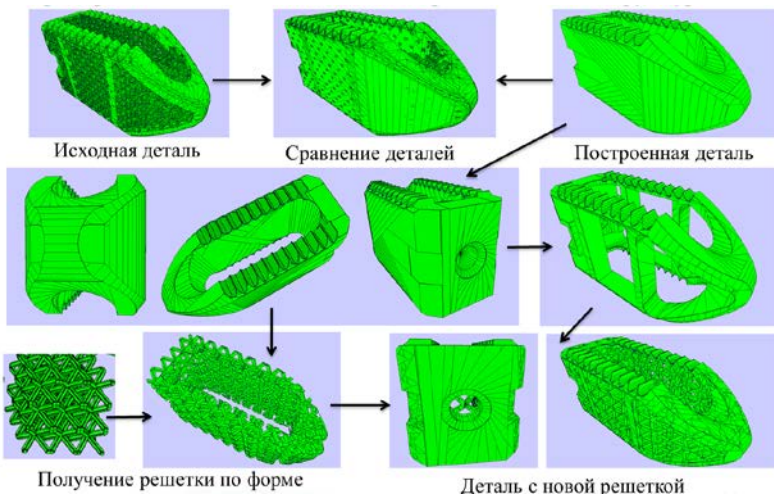
– типовой выбор режимов работы оборудования и типовая оценка результатов печати (средняя детализация, средняя скорость, персональный компьютер, только макроуровень, теплоперенос и прочность с детализацией физических процессов);

– точный выбор режимов работы оборудования и точная оценка результатов печати (высокая детализация, средняя скорость, суперкомпьютер, только макроуровень, теплоперенос и прочность с детализацией физических процессов);

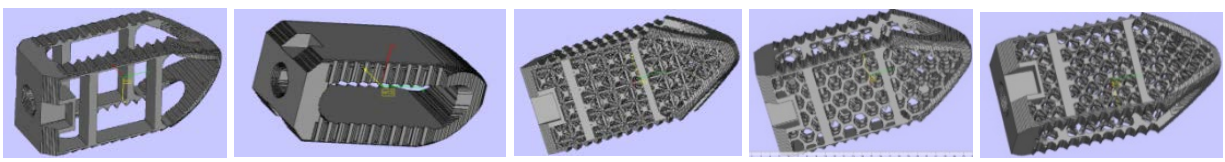
– высокоточный выбор режимов работы оборудования и высокоточная оценка результатов печати (высокая детализация, низкая скорость, суперкомпьютер, макроуровень и/или мезоуровень, тепломассоперенос, прочность, микроструктура, максимальная детализация физических процессов).

Примеры работы реализованных модулей

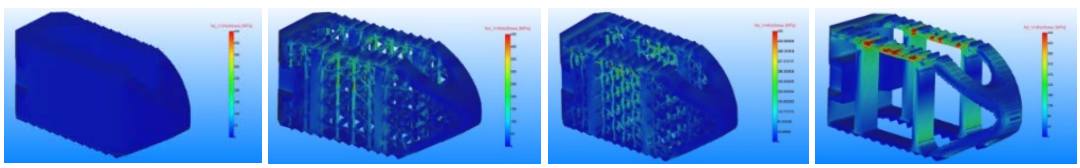
На рис. 10–11 приведены некоторые примеры работы модулей реализованных на первом этапе проекта в 2019 году. На рис. 10 показана последовательность создания геометрии сложной детали со вписанными внутрь различными типами решетчатых структур для последующего проведения виртуальных испытаний и результаты виртуальных испытаний на прочность. На рис. 11 показаны некоторые тестовые расчеты процессов тепломассопереноса на макро и мезоуровнях.



а



б



в

Рис. 10. Построение сложной детали «Спинной кейдж» в комплексе программ и последующие виртуальные испытания (а – схема построения; б – полученные варианты детали; в – некоторые результаты расчетов напряжений внутри деталей)

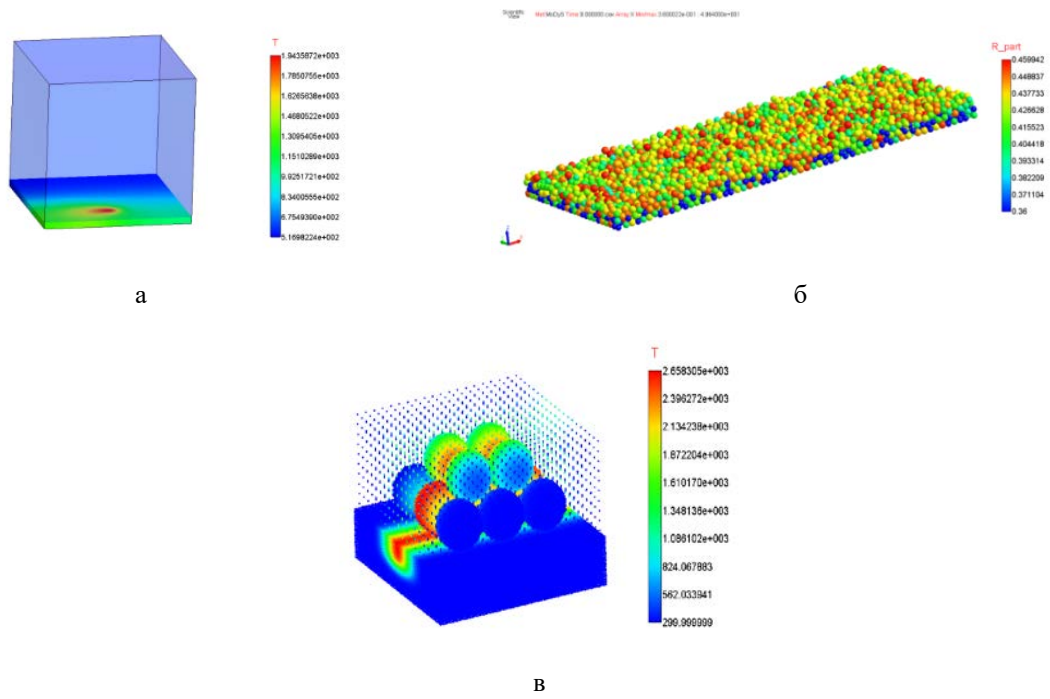


Рис. 11. Примеры тестовых расчетов (а – на макроуровне; б – насыпание слоя порошка; в – на мезоуровне)

Заключение

Во ФГУП «РФЯЦ-ВНИИЭФ» реализуется проект «Виртуальный 3D-принтер». Цель проекта – создание комплекса программного обеспечения, обеспечивающего функциональность и широкие возможности для оптимизации формы деталей на основе методов топологической оптимизации (бионический дизайн), в том числе и с использованием решетчатых структур, управления оборудованием для лазерного аддитивного производства и определения оптимальных параметров работы оборудования на основе многомасштабного математического моделирования.

Литература

1. Гибсон Я., Розен Д., Стакер Б. Технологии аддитивного производства. –М.: Техносфера, 2016. С. 656.

2. Рвачев В. Л. Методы алгебры логики в математической физике. –Киев: Наукова думка, 1974, С. 260.

3. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики. – СПб: БХВ - Петербург, 2003, С. 560.

4. Быков А. Н., Попов В. В., Синицин Е. А., Дьянов Д. Ю., Медведкина М. В. Программа для топологической оптимизации конструкций с использованием ячеистых (решетчатых) структур // Труды XVII Международной конференции «Супервычисления и математическое моделирование». Саров, 15–19 октября 2018, С. 114–123.

5. Дьянов Д. Ю., Медведкина М. В., Быков А. Н., Попов В. В. Методы топологической оптимизации в программном комплексе «3D PRINTER» // Математическое моделирование. 2019. Т. 31, № 7. С. 75–90.

SOFTWARE PACKAGE «VIRTUAL 3D PRINTER»: FROM THE CONCEPT TO ITS IMPLEMENTATION

V. V. Popov, A. N. Bykov, Yu. N. Deryugin, M. V. Vetchinnikov, D. Yu. Dyanov, S. N. Polishchuk, V. G. Kudelkin, E. A. Sinitsyn, A. N. Bakhaev, M. V. Zhuravlyeva, O. I. Britova, M. A. Titov, A. V. Baranov, L. V. Gamov, I. G. Mileshin, V. Yu. Kuznetsov, P. A. Mashenkin, E. F. Golubkova, A. S. Danilov, M. V. Medvedkina, A. A. Kayakin, A. I. Lomaykin, O. V. Sapronova, E. A. Veselova, A. B. Emelyanov, A. A. Lazarev

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

FSUE «RFNC-VNIIEF» started developing the «Virtual 3D printer» software package in 2019. The expected result of these efforts in 2021 is a software package («Virtual 3D printer») for the simulation of physical processes of the selective laser melting (SLM) to predict the structure and properties of materials, fabricate products with the specified properties, and perform their topology optimization. The objective of this work is to create a Russian analog to foreign software in the area of additive laser technologies, which entered the market in 2015-2019.

The report presents the main conceptual and architectural solutions for the software package «Virtual 3D printer», describes the software package functionality to manipulate with geometries of component parts using the topology optimization and lattice structures, and considers the issues of integrating the database of material properties and different compute modules.

Key words: additive technologies, selective laser melting, multilevel mathematical modeling, topology optimization, lattice structures, generative design, implicit representation of geometry, control of manufacturing equipment.

УДК 004.932
DOI 10.53403/9785951505071_2022_363

МУЛЬТИФРАКТАЛЬНОЕ КОМПЛЕКСИРОВАНИЕ МНОГОДИАПАЗОННЫХ РАДИОЛОКАЦИОННЫХ ИЗОБРАЖЕНИЙ

А. А. Потапов[1], *В. А. Кузнецов*[2], *Е. А. Аликулов*[2]

[1]Институт радиотехники и электроники им. В.А. Котельникова РАН, Москва
[2]Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина, Воронеж

В работе кратко рассмотрены существующие подходы к комплексированию радиолокационных изображений, полученных в разных частотных диапазонах. Предложен новый способ их комплексирования на основе фрактального подхода, в котором участок подстилающей поверхности анализируется на различных масштабах. Применение предложенного способа позволяет повысить информативность результирующего изображения.

Ключевые слова: радиолокационная станция с синтезированной апертурой, радиолокационное изображение, многодиапазонность, комплексирование.

Современные радиолокационные системы представляют собой сложные информационно-управляющие технические системы реального масштаба времени, включающие иерархическую пространственно-распределенную совокупность средств радиолокации, комплексов средств автоматизации, средств приема, обработки, отображения и передачи радиолокационных изображений (РЛИ), а также средств, технологий и субъектов управления, предназначенные для информационного обеспечения широкого круга военно-технических, научно-экспериментальных и социально-экономических задач [1]. Радиолокационные станции с синтезированной апертурой (РСА) как самостоятельные средства видовой воздушной разведки, осуществляющие непрерывное наблюдение земной (морской) поверхности и объектов не зависимо от времени суток и метеорологических условий играют все большую роль в информационном обеспечении боевых действий войск [2].

Традиционно в РСА используется сантиметровый диапазон радиоволн для пилотируемых воздушных судов и в последнее время начал использоваться миллиметровый диапазон волн для беспилотных летательных аппаратов. Однако следует иметь в виду, что диапазон радиоволн этим не ограничивается. Каждый из диапазонов ЭМВ имеет свои особенности по отношению к радиолокации земной поверхности и объектов и, в принципе, так или иначе может быть использован в целях радиолокационной воздушной разведки. В настоящее время ведутся интенсивные поиски оптимального для радиолокационного комплекса воздушной разведки набора частот радиоволн, который можно было бы объединить в многодиапазонную РСА. При этом учитываются как информационные, так и тактико-технические и технологические возможности разработки, производства и применения таких комплексов [2]. Один из проблемных вопросов на сегодняшний день – комплексная обработка РЛИ многодиапазонной РСА.

Результаты анализа современных решений в области комплексирования РЛИ [3] показали, что обработка изображений может производиться на нескольких уровнях: сигналов или данных (пикселей применительно к РЛИ), признаков и решений, а также их совокупности. Слияние изображений на уровне пикселей характеризуется высокими трудозатратами на их совмещение и низкой точностью – РЛИ, сформированные в различных частотных диапазонах, отличаются пространственным разрешением и полосой обзора. Разработчикам приходится решать задачи автоматического поиска

на всех совмещаемых РЛИ опорных точек, что не всегда возможно. Кроме того, проводимые по таким точкам аффинные преобразования обуславливают потерю информации, в частности, высока вероятность сглаживания так называемых «блестящих точек» какого-либо объекта. Одним из недостатков комплексирования РЛИ на уровнях признаков и решений является снижение достоверности дальнейшего дешифрирования вследствие перемножения вероятностей правильного обнаружения. Кроме того, одним из наиболее часто используемых при комплексировании подходов является снижение размерности пространства данных или признаков различными методами, что ведет к потере незначительной (со слов авторов этих методов) информации и, по нашему мнению, существенному снижению достоверности анализа вследствие нарушения принципа целостности сложной системы.

В данной работе предложен способ комплексирования РЛИ многодиапазонной РСА, в котором значение длины волны каждого канала РСА рассматривается как некоторая мера на определенном масштабе. Такой подход хорошо укладывается в положение о фрактальной размерности [4]. При этом, как показывают результаты исследований [3], в качестве меры следует рассматривать не длину волны или рабочую частоту, а разрешающую способность каналов РСА. Тогда с учетом линейных размеров типового объекта воздушной разведки, например, танка или бронетранспортера, появляется возможность построить график зависимости числа элементов изображения $N(\varepsilon)$, занимаемых объектом, от разрешения РСА в терминах фрактальной теории. Если рассматривать такой график с точки зрения классического способа расчета фрактальной размерности D согласно известному выражению

$$D = \lim_{\varepsilon \to 0} \frac{\ln(N(\varepsilon))}{\ln(1/\varepsilon)}, \qquad (1)$$

то разрешающая способность РСА – не что иное, как масштаб анализа ε. Значение фрактальной размерности будет определяться тангенсом угла наклона аппроксимирующей линии. Исходя из этого, чем ближе к аппроксимирующей функции реальные значения, тем точнее можно определить фрактальную размерность. Таким способом можно сформулировать требование к выбору разрешающей способности каждого канала многодиапазонной РСА.

В связи с тем, что при обеспечении одинаковых значений разрешающей способности РСА по дальности и азимуту (продольной дальности) требуется варьировать единственный параметр – интервал синтезирования, длина полосы обзора в каждом канале будет различной, что поясняется на рис. 1.

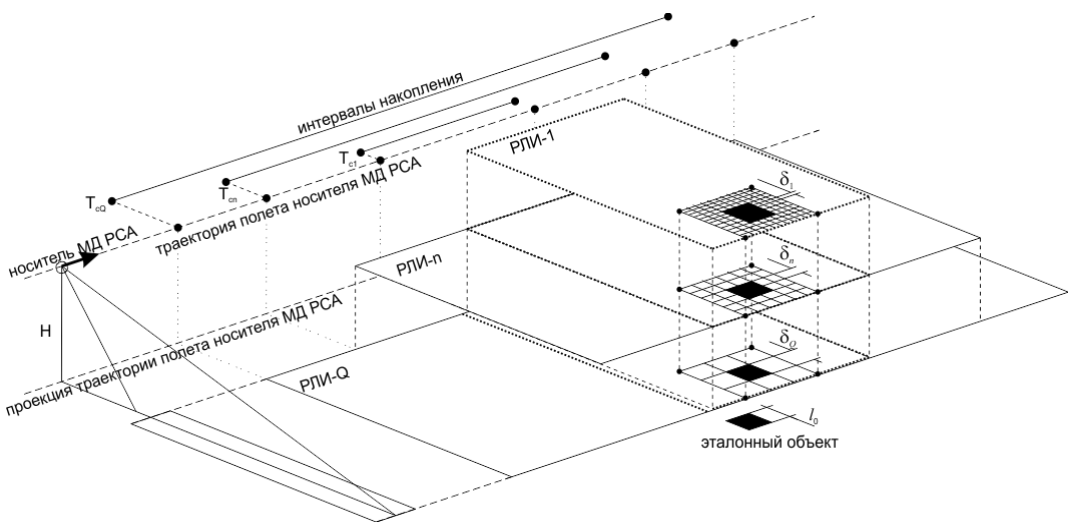


Рис. 1. Геометрия съемки многодиапазонной РСА

В работе предложено ранжировать исходные изображения в порядке убывания значений несущих частот или ширины спектра зондирующих сигналов, на которых они получены, а затем определять базовое изображение по критерию максимальной частоты зондирующего сигнала, как наиболее информативное.

Для каждого исходного *n*-го изображения необходимо вычислить масштаб ε_n согласно выражению

$$\varepsilon_n = \left\lceil \frac{w_n - 1}{2} \right\rceil, \tag{2}$$

где $\lceil * \rceil$ – операция округления к большему, *n* – порядковый номер ранжированного изображения, $w_n = \lceil l_0 / \delta_n \rceil = \lceil 2\Delta f_n l_0 / c \rceil$ – число элементов изображения, соответствующих линейному размеру l_0 некоторого эталонного объекта, *c* – абсолютная величина скорости распространения электромагнитных волн.

Участок сцены, соответствующий базовому изображению, согласно рис. 1, вырезается на остальных изображениях относительно середины интервалов накопления зондирующих сигналов T_{cn} (фазовых центров синтезированной апертуры антенн). Поскольку для обеспечения одинаковой дальности обнаружения в каждом частотном канале РСА требуется различная мощность излучения, а также в случае различной чувствительности приемников РСА, требуется выравнивание динамического диапазона РЛИ. Как показано в [5], нормировка динамического диапазона может быть осуществлена несколькими способами. С учетом вычисленных масштабов каждого РЛИ ε_n, их динамический диапазон может быть выравнен, например, согласно выражению

$$z_{\varepsilon n} = \frac{z_n - \min(z_n)}{\max(z_n) - \min(z_n)} + \varepsilon_n, \tag{3}$$

где $z_{\varepsilon n}$ – множество значений яркости элементов преобразованного *n*-го изображения с учетом масштаба, z_n – множество значений яркости элементов исходного *n*-го изображения, min (z_n) и max (z_n) – минимальное и максимальное значения яркости исходного *n*-го изображения соответственно.

Непосредственно комплексирование проводится путем формирования поля фрактальных размерностей $D(i,j)$, $i = 1 \ldots M$, $j = 1 \ldots N$, для этого в скользящем окне размером $W \times W$, где *W* – нечетное целое число, рассчитывается локальная мультифрактальная размерность L_q [6]

$$L_q = \frac{1}{|q|} \lim_{\varepsilon \to 0} \frac{\ln(I(q,\varepsilon))}{\ln(1/\varepsilon)}, \; при \; q \neq 0, \tag{4}$$

где *q* – порядок скейлингового момента, при $q = 1$ локальная мультифрактальная размерность соответствует фрактальной размерности, при $q \neq 1$ выявляются мультифрактальные свойства, которые также возможно использовать при комплексировании,

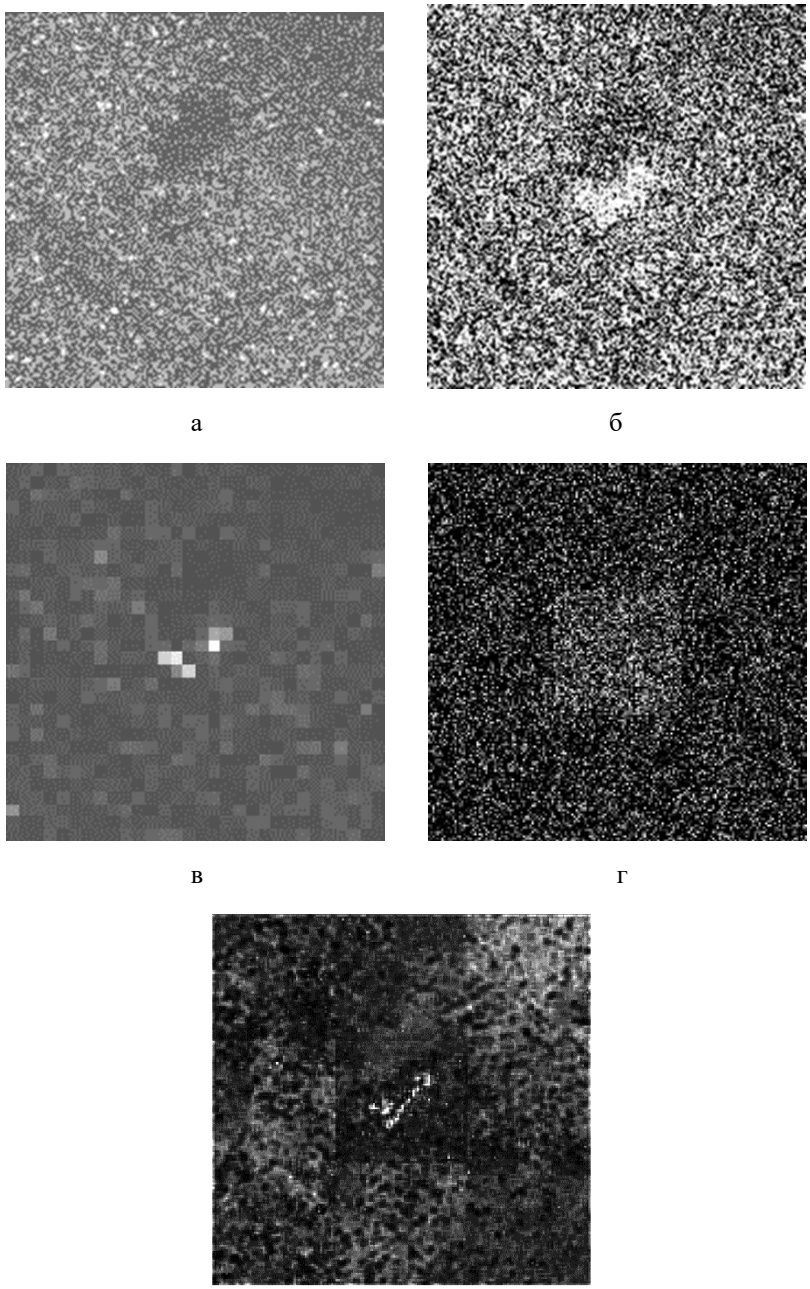
$I(q,\varepsilon) = \alpha \sum_i^W \sum_j^W \mu_\varepsilon(i,j)^q$ – обобщенная статистическая сумма, $\alpha = \dfrac{\sum_i^W \sum_j^W |z_\varepsilon(i,j) - z(i,j)|}{\varepsilon_n - \varepsilon_Б}$,

$\varepsilon_Б$ – масштаб базового изображения *z*, $\mu_\varepsilon(i,j) = \dfrac{|z_\varepsilon(i,j) - z(i,j)|}{\sum_i^W \sum_j^W |z_\varepsilon(i,j) - z(i,j)|}$.

Вычисленные таким способом значения L_q каждого центрального элемента скользящего окна запоминаются в соответствующем элементе двумерной матрицы $D(i,j)$, $i = 1 \ldots M$, $j = 1 \ldots N$,

а в случае $q \neq 1$ – в соответствующем элементе трехмерной матрицы $D(q,i,j)$, содержащей элементы исходных радиолокационных изображений одной и той же сцены, полученных одновременно в различных частотных диапазонах.

В качестве примера мультифрактального комплексирования РЛИ многодиапазонной РСА на рис. 2 приведены результаты обработки РЛИ одного и того же объекта, синтезированные на разных частотах с различным разрешением. Следует отметить, что, как видно из примера, все изображения отличаются динамическим диапазоном и сильно зашумлены, а на первом РЛИ объект замаскирован (видна только радиолокационная тень), поэтому его на изображении не видно.



а б в г д

Рис. 2. Пример мультифрактального комплексирования РЛИ: а – г исходные РЛИ;
д – результат фрактального комплексирования (яркости преобразованы для наглядности)

Однако благодаря существенным отличиям текстур РЛИ подстилающей поверхности и замаскированного объекта удается получить достоверный результат. Информативность каждого из исходных РЛИ, оцененная известным образом по энтропии, оказалась в среднем не менее чем на 37 % ниже, чем у результирующего, что подтверждает высокую эффективность предложенного способа.

Таким образом, в работе предложен новый способ комплексирования РЛИ многодиапазонной РСА, основанный, в отличие от известных, на одновременном вычислении скользящим окном по всем исходным РЛИ локальных мультифрактальных размерностей методом итерационных покрытий, и позволяющий существенно повысить информативность комплексирования в целом при значительном сокращении вычислительных затрат на обработку.

Литература

1. Ботов М. И., Вяхирев В. А. Основы теории радиолокационных систем и комплексов / Под общ. ред. Ботова М. И.–Красноярск: Сибирский федеральный университет, 2013.

2. Радиолокационные системы воздушной разведки, дешифрирование радиолокационных изображений: учебник для курсантов ВВИА имени профессора Жуковского Н. Е. / Под ред. Школьного Л. А.–М.: ВВИА, 2008.

3. Кузнецов В. А., Аликулов Е. А., Потоцкий А. Н. Состояние и перспективы комплексирования многомерных радиолокационных изображений // Сб. трудов XXVI Междунар. науч.-практ. конфер. «Радиолокация, навигация, связь». Т. 2. Воронеж: ВГУ, 2020. С. 181–188.

4. Потапов А. А. Фракталы в радиофизике и радиолокации: топология выборки. – М.: Университетская книга, 2005.

5. Гурьянов М. А. Преобразование динамического диапазона радиолокационных изображений для различных снимаемых поверхностей // Журнал радиоэлектроники.2014. № 12.

6. Yong Xia, Dagan Feng, Rongchun Zhao Morphology-Based Multifractal Estimation for Texture Segmentation // IEEE Transactions on Image Processing.2006. Vol. 15, №. 3. P. 614–623.

FRACTAL FUSION OF MULTI-BAND RADAR IMAGES

A. A. Potapov, V. A. Kuznetsov, E. A. Alikulov
[1] Kotelnikov Institute of Radioengineering and Electronics RAS, Moscow
[2] Zhukovsky – Gagarin Air Force Academy, Voronezh

The paper briefly discusses existing approaches to the radar images fusion obtained in different frequency bands. A new method of their fusion based on the fractal approach is proposed, in which a section of the underlying surface is analyzed at different scales. Application of the proposed method makes it possible to increase the information content of the resulting image.

Key words: synthesized aperture radar, radar image, multi-band, fusion.

УДК 004.932
DOI 10.53403/9785951505071_2022_368

ВОЗМОЖНОСТИ УСКОРЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЭФФЕКТА ВТОРИЧНОЙ МОДУЛЯЦИИ СИГНАЛА БОРТОВОЙ РАДИОЛОКАЦИОННОЙ СТАНЦИИ В СРЕДЕ MATLAB

А. А. Потапов [1], В. А. Кузнецов [2], Н. А. Меньших [3]

[1] Институт радиотехники и электроники им. В.А. Котельникова РАН, Москва
[2]Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю. А. Гагарина, Воронеж
[3]Военный инновационный технополис «ЭРА», Анапа

В работе кратко рассмотрена возможность оптимизации кода в среде MATLAB. Предложено два подхода для ускорения работы кода MATLAB. Применение предложенного способа позволяет ускорить вычисления при задействовании малого числа ядер процессора.

Ключевые слова: радиолокационная станция, имитационное моделирование, параллельные вычисления, вторичная доплеровская модуляция.

Среди современных прикладных пакетов математического моделерования наибольшее распространение для реализации исследовательских задач получил MATLAB, представляющий широкие возможности численного анализа. Для того чтобы обеспечить совместимость с существующими проектами, MATLAB обеспечивает возможность переводить код на другие языки программирования, например, Fortran или C [1]. Большое количество библиотек MATLAB позволяют моделировать явления самой разной природы, однако крупномасштабные численные расчеты в среде MATLAB предъявляют к аппаратным вычислительным средствам высокие требования. Пользователь сталкивается с одним из двух вариантов решения данной проблемы: либо произвести модернизацию аппаратных средств, либо оптимизировать алгоритмы вычислений. Первый способ имеет множество различных недостатков: во-первых, пользователь не всегда может быстро освоить вычислительную мощность в полной мере, во-вторых, имеет временный характер так как каждый год появляются новые технологии для увеличения производительности компьютера. Поскольку все потенциальные возможности MATLAB реализуются благодаря обширному набору библиотек и операций векторизации, существует другой способ для ускорения вычисления – это оптимизация кода для эффективного выполнения алгоритма.

Задача имитационного моделирования радиолокационной воздушной цели сама по себе является достаточно сложной в вычислительном плане, а моделирование эффекта вторичной модуляции сигнала бортовой радиолокационной станции, отраженного от вращающихся лопаток компрессора низкого давления турбореактивного двигателя воздушной цели в процессе ее движения – в особенности. В работе используется многолучевой подход к описанию распространения электромагнитной энергии с учетом ее многократного переотражения от элементов фюзеляжа, что занимает весьма значительное время и обуславливает целесообразность ускорения имитационного моделирования.

Целью данной работы является оценка времени имитационного моделирования эффекта вторичной модуляции радиолокационного сигнала при использовании центра обработки данных (ЦОД), а также требуемого количества вычислительных ресурсов.

Для ускорения работы кода MATLAB известно два подхода: предварительное размещение (preallocation) и векторизация (vectorization). С использованием предварительного размещения осуществляется инициализация массива окончательного размера, требуемого для вычислений. Предварительное размещение помогает избегать динамического изменения размера массивов, особенно если код содержит циклы for и while. Поскольку массивы в MATLAB находятся в непрерывных блоках памяти, повторное изменение размера массива требует от MATLAB дополнительных затрат времени на поиск большего блока памяти и последующего перемещения массива в этот блок памяти. Использование предварительного размещения массива позволяет избежать этих ненужных операций и сократить общее время выполнения вычислений.

Векторизация – это процесс перехода от использования циклов к матричным и векторным операциям. MATLAB позволяет пользователю использовать процессорно-оптимизированные библиотеки матричных и векторных вычислений. В результате можно увеличить производительность посредством векторизации кода [2]. Векторизованные вычисления MATLAB, использующие большие массивы, позволят пользователю ускорить вычислены с использованием графического процессора (GPU – graphics processing unit). В случаях, когда циклы for не могут быть векторизованы, для ускорения алгоритма используются параллельные циклы for (parfor) или генерация кода на языке программирования C.

В связи с развитием многопроцессорности и многоядерности, параллельные вычисления превратились из экзотики в повседневность, однако многие вычислительные среды, в частности, MATLAB, не поддерживают многопоточность по умолчанию и требуют дополнительных инструкций и модулей. Многоядерные параллельные вычисления производятся на основе MPI (Message Passing Interface) интерфейса, который реализует упрощенный переноса программы с кластера на кластер, не изменяя ни байта исходного кода. Параллельная программа должна эффективно использовать вычислительные мощности и коммуникационную среду.

Генерация кода на языке C предоставляет несколько иной подход к ускорению – вместо распределения вычислительной задачи по множеству ядер процессора производится анализ этой задачи и выделение наиболее часто используемых в ней функций. Функции языка MATLAB имеют формат M-функций. M-функции являются M-файлами, которые допускают наличие входных и выходных аргументов. Они работают с переменными в пределах собственной рабочей области, отличной от рабочей области системы MATLAB. Встроенные M-функции являются скомпилированными и выполняются достаточно быстро, однако, существует возможность создания MEX-функций, реализованных на языке C/C++, из существующих M-функций с помощью специального интерфейса, называемого MATLAB Coder. MEX-функции ведут себя подобно встроенным функциям MATLAB, но их разработка требует точно определять типы и размеры аргументов, а также учитывать различия между языками MATLAB и C.

Рассмотрим псевдо-алгоритм для расчета вторичной доплеровской модуляции (ВДМ) сложных воздушных целей, который состоит из следующих частей:

1. Загрузка модели воздушной цели и определение характеристик процесса радиолокации.

2. Расчет параметров для расчета ВДМ – таких как длина волны сигнала РЛС, количество лучей, направляемых от РЛС в сторону воздушной цели, мощность одного луча и т. д.

3. Установка дополнительных параметров расчета: параметров канала, а также алгоритма расчета ВДМ с учетом эффективной поверхности рассеяния (ЭПР) (например, плотность водяного пара в пределах канала и атмосферное давление).

4. Расчет ВДМ с учетом ЭПР.

Расчет ВДМ в указанном алгоритме проводится на основе 3D-модели, представленной в формате STL (stereolithography – стереолитография, так как изначально этот формат применялся в области трехмерной печати). Данный формат удобен тем, что его можно использовать для 3D-моделей во многих популярных системах автоматизированного проектирования (Solidworks, Autodesk Inventor, Blender). Формат STL описывает объект с помощью списка треугольных граней (фацетов) и нормалей к ним. Для каждого фацета и нормали указываются три числа с плавающей запятой, представляющие координаты в декартовой системе координат. При этом нормаль фасета должна быть единичным вектором, направленным от фацета наружу.

Отметим, что пункты 2 и 3 выполняются единожды для каждой трехмерной модели, а пункт 4 является составным и несет основную нагрузку на вычислительное устройство как в плане потребляемой памяти, так и в плане числа операций вычислений. Очевидно, что именно эта часть алгоритма заслуживает наибольшего внимания.

Оценка времени расчета вторичной доплеровской модуляции подразумевает зависимость этого времени от некоторых исходных данных. Рассматривая сложность и детализацию объекта, выраженную в количестве фацетов, оценим данную зависимость по нескольким экспериментам. Для этого проведена серия расчетов для STL-моделей различной сложности (с различным количеством фацетов) на различных процессорах с использованием последовательных и параллельных вычислений. В экспериментах использовались два типа процессора – Intel® Core ™ i7-6850K CPU @ 3.60GHz, установленный на штатном рабочем месте, а также Intel® Xeon® Gold 6150 CPU @ 2.70GHz, установленный на серверах центра обработки данных (ЦОД) Военного инновационного технополиса «ЭРА» [3]. Второй процессор имеет 18 ядер [4], что предоставляет возможность еще большего распараллеливания вычислений. Общее время расчета в каждом эксперименте разделено на суммарное количество расчетов ЭПР отдельного фацета и тем самым получено среднее время расчета для одного фацета. Результаты расчетов представлены в табл. 1 и на рис. 1.

Таблица 1

Время вычисления в секундах для различного количества фацетов

№	Тип процессора и количество используемых при расчетах ядер	Число фацет		
		35590	51364	64703
1	Intel® Core ™ i7-6850K CPU @ 3.60GHz – 1 (АРМ)	41	72	76
2	Intel® Core ™ i7-6850K CPU @ 3.60GHz – 6 (АРМ)	13	24	30
3	Intel® Xeon® Gold 6150 CPU @ 2.70GHz – 1 (ЦОД)	10	45	57
4	Intel® Xeon® Gold 6150 CPU @ 2.70GHz – 6 (ЦОД)	9	17	23

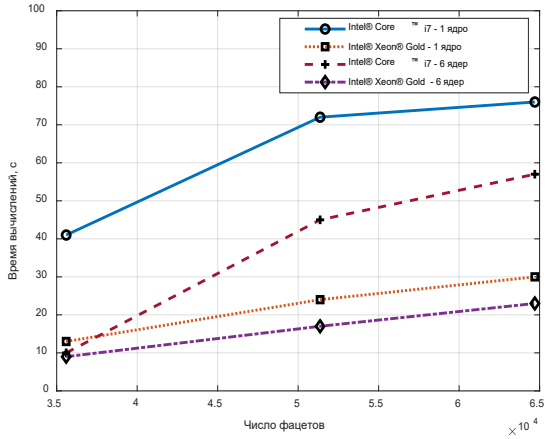


Рис. 1. Зависимость времени вычисления от количества фацетов модели на различных процессорах и их конфигурациях

Приведенные на рис. 1 графики демонстрируют, что эффективность параллельных вычислений возрастает с количеством фацетов. Этот результат вполне ожидаем. Вторым выводом является то, что эффективность параллельных вычислений, произведенных на различных процессорах отличается незначительно. Данный результат ставит под сомнение необходимость проведения расчетов с привлечением ЦОД.

Поскольку расчет ВДМ не выражается в виде четкой зависимости от количества фацетов – сказывается различие в количестве видимых фацетов с определенного ракурса и числа переотражений лучей – рассмотрим обобщенную структуру алгоритма с абстрактными «элементарными» в данном контексте операциями.

Рассмотрим обобщенную внутреннюю организацию расчета ВДМ:

```
for i=1:n0 %цикл по списку азимутальных углов
   %формирование сетки лучей
   for j=1:n1 %цикл по числу отсчетов на интервале наблюдения
      %выделение характеристик фацетов
      for k=1:n2 % цикл по числу излученных лучей
         while A %проверка условий пересечения лучей и фацетов
            %оценка пересечений
            %оценка переотражений лучей – учет потерь энергии
            if B %хотя бы одно пересечение имеется
               for z=1:n4 % цикл по пересеченным элементам модели
                  % расчет ЭПР видимого фацета с учетом поляризации
               end
            end
         end
      end
      % сбор отраженных лучей
   end
   % вычисление спектра ВДМ
   % построение спектрограммы
end
```

Проанализируем приведенную структуру для определения подходящего варианта ускорения имитационного моделирования эффекта вторичной модуляции:

1. Расчет ЭПР видимого фацета с учетом поляризации является абстрактной единицей приведенного алгоритма. Расчет ЭПР производится в M-функции, следовательно, возможно ускорение алгоритма путем преобразования рассматриваемой m-функции в MEX-функцию [5].

2. Цикл по числу излученных лучей содержит независимые итерации, а прохождение каждого луча оценивается независимо от других. Это позволяет говорить о возможности использования вместо обычного цикла for цикла parfor [6].

Воспользуемся приведенными особенностями алгоритма и преобразуем его с использованием MEX-функции и цикла parfor. Чтобы проанализировать эффективности каждого из изменений, проведем серию расчетов как с использованием MEX-функции, так и с использованием ее аналога m-функции, при этом варьируя количество используемых ядер в цикле parfor. Для упрощения расчеты будем проводить для одного азимутального угла на 127 отсчетах на интервале наблюдения и количестве излученных лучей, равном 289. Для сравнения эффективности совместного применения parfor и MEX-функций, проведем серию расчетов (по 10 на каждую модификацию) и сравним полученные результаты (табл. 2).

Таблица 2

Результаты экспериментальных оценок времени расчетов ВДМ при различных модификациях параметров алгоритма расчета

№	Использование parfor (количество используемых ядер процессора)	Использование MEX-функции	Усредненное время расчета, с
1	– (1)	–	3544.215506
2	+ (2)	–	2300.645179
3	+ (3)	–	1712.494566
4	+ (4)	–	1480.251042
5	+ (5)	–	1444.556580
6	+ (6)	–	1380.726564
7	– (1)	+	2221.899430
8	+ (2)	+	1508.524474
9	+ (3)	+	1225.291389
10	+ (4)	+	1114.454955
11	+ (5)	+	1096.561757
12	+ (6)	+	1052.225927

При параллелизации алгоритма вычисления проводились на ядрах процессора Intel Core i5-8400 CPU 2,800 ГГц. Количество используемых ядер варьировалось для анализа влияния MEX-функции на изменение времени расчета. Для наглядности и удобства анализа представим табличные данные в виде графика (рис. 2).

Легко заметить, что полученные зависимости имеют экспоненциальный характер, причем использование более 4 ядер в параллельных вычислениях уже не имеет такого принципиального значения в плане повышения скорости выполнения кода. Для оценки максимально достижимого параллелизма уместно воспользоваться законом Амдала, который описывает связь между ожидаемым ускорением параллельных реализаций алгоритма и последовательным алгоритмом в предположении, что решаемая задача –в данном случае исходные данные для моделирования, остается постоянной [7]. Допущение о фиксированном размере задачи является ключевым для закона Амдала – если количество обрабатываемых данных меняется с варьированием количества ядер, следует обратиться к другой зависимости, например, закону Густафсона [8].

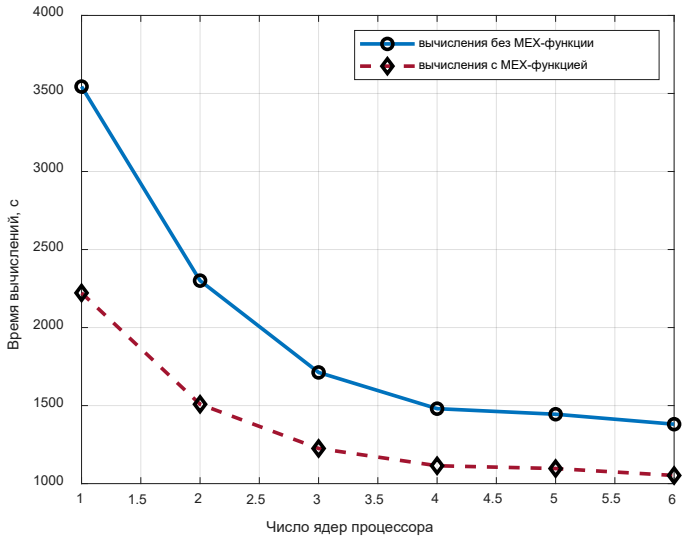


Рис. 2. Зависимость времени выполнения расчетов от количества используемых ядер

Введем некоторые необходимые для анализа величины. Ускорением параллельного алгоритма называется отношение:

$$S_n = \frac{T_1}{T_n} \tag{3}$$

где T_n – время вычисления задачи на n процессорах, T_1 – время выполнения однопоточной программы. Если $S_n > 1$, то параллельная версия алгоритма эффективна, в противном случае накладные расходы (издержки) реализации параллельной версии алгоритма чрезмерно велики. Закон Амдала описывается следующей математической формулой:

$$S_n = \frac{1}{\alpha + \frac{1-\alpha}{n}}, \tag{4}$$

где S_n показывает, во сколько раз можно ускорить вычисления (ускорение), n – количество процессоров (ядер), α – доля последовательно вычисляемого кода.

Закон Амдала, несмотря на то, что в нем не учитывается множество факторов, накладывает ограничения на максимально достижимую эффективность параллельного алгоритма. Оценим долю последова-

тельной части программы по закону Амдала и спрогнозируем выигрыш в производительности при использовании большего числа процессоров. Проведем две оценки доли последовательной части программы: с использованием MEX-функции (α_2) и без таковой (α_1). Рассчитаем для этого указанные доли для каждой параллельной конфигурации из таблицы 1 и выведем среднее значение для α_2 и α_1 (табл. 3).

Таблица 3

Расчетные значения последовательной доли программы расчета ВДМ

Кол-во используемых ядер	α_1	α_2
2	0.2983	0.3579
3	0.2248	0.3272
4	0.2235	0.3354
5	0.2595	0.3669
6	0.2675	0.3683
Среднее значение α	0.2547	0.3511

Полученные значения характеризуют эффект от применения MEX-функций – доля последовательно вычисляемого кода в среднем увеличивается в 1,3785 раз. Начиная с некоторого числа процессов, зависящего от доли последовательной части, ускорение будет меняться незначительно и дальнейшее распараллеливание не имеет смысла. Максимальное достижимое ускорение не может превысить в случае применения обычных m-функций $S_{max1} = 1/\alpha_1 = 3{,}9261$, а в случае применения MEX-функции – $S_{max2} = 1/\alpha_2 = 2{,}8482$. Заметим, что в первом случае при использовании 6 ядер достигнуто ускорение в 2,5669 раз – 65 % от теоретического максимума, а во втором случае при использовании 6 ядер достигнуто ускорение в 2,1116 – 74 % от теоретического максимума. Если рассматривать возможность расчета на процессоре ЦОД с использованием всех имеющихся 18 ядер, то получим значения ускорения $S_1 = 3.3772$ (86 % от теоретического максимума) и $S_2 = 2.5830$ (90 % от максимума). Оценивая полученные показатели эффективности, выраженные в виде доли полученного ускорения от теоретически возможного, получим вывод, что трехкратное увеличение количества ядер дает прирост ускорения всего на 16 %, поэтому выполнение параллельных вычислений на шести процессорах можно считать приемлемым и достаточным.

Таким образом, к программе расчета спектра ВДМ была применена стратегия ускорения вычислений в среде MATLAB, основной упор в которой ставился на совместном применении параллельных циклов parfor и MEX-функций. Результаты проведенных экспериментов показали, что значительная доля возможного ускорения достигается при умеренном количестве используемых ядер процессора, следовательно, применение процессора ЦОД не имеет ключевого значения. В дальнейшем планируется анализ эффективности применяемых структур данных и перевод вычислений с процессора на GPU. Подобные преобразования будут призваны открыть новые возможности для векторизации и распараллеливания вычислительных задач в среде MATLAB.

Литература

1. Lopez, C. MATLAB Numerical Calculations / C. Lopez.–New York: Apress, 2014.–336 p.

2. Chapra, S. Applied Numerical Methods with MATLAB® for Engineers and Scientists: Fourth Edition / S. Chapra.–New York: McGraw-Hill Education, 2017.–720 p.

3. ЭРА вычислений на новом уровне: СуперЭВМ расширила исследовательские возможности технополиса Минобороны [электронный ресурс] Режим доступа: https://www.era-tehnopolis.ru/news/mass-media/era-vychisleniy-na-novom-urovne/ (дата обращения: 02.01.21.)

4. Процессор Intel® Xeon® 6150 класса Gold [электронный ресурс] Режим доступа: https://ark.intel.com/content/www/ru/ru/ark/products/120490/intel-xeon-gold-6150-processor-24–75m-cache–2–70- ghz.html (дата обращения: 02.01.21.)

5. MATLAB Coder Documentation: Code Generation [электронный ресурс] Режим доступа:https://www.mathworks.com/help/coder/code-generation.html (дата обращения: 02.01.21.)

6. Parallel Computing Toolbox Documentation: Parallel for-Loops (parfor) [электронный ресурс] Режим доступа: https://www.mathworks.com/help/parallel-computing/parallel-for-loops-parfor.html?s_tid=CRUX_lftnav (дата обращения: 02.01.21.)

7. Белова, И. М. Параллельное программирование: учебное пособие / Белова И. М., Рассказов А. А.–М.: МГИУ, 2012. –101 с.

8. Sun, X.-H. Reevaluating Amdahl's law in the multicore era / X.-H. Sun, Y. Chen // J. J. Parallel and Distributed Computing.–2010.–№ 70.–pp. 183–188.

CAPABILITIES OF ACCELERATING THE SIMULATION OF THE SECONDARY SIGNAL MODULATION EFFECT OF ON-BOARD RADAR IN MATLAB ENVIRONMENT

A. A. Potapov[1], V. A. Kuznetsov[2], N. A. Menshikh[3]

[1]Kotelnikov Institute of Radioengineering and Electronics RAS, Moscow
[2]Zhukovsky – Gagarin Air Force Academy, Voronezh
[3]Military Innovative Technopolis "ERA", Anapa

The paper briefly describes code optimization capabilities in MATLAB environment. Two approaches are proposed to speed up the MATLAB code operation. The use of the proposed method allows accelerating computations with a small number of enabled processor cores.

Key words: radar, simulation, parallel computations, secondary Doppler modulation.

УДК 004.4
DOI 10.53403/9785951505071_2022_375

ПРЕПРОЦЕССОР ПАКЕТА ПРОГРАММ ЛОГОС. ПОДГОТОВКА НАЧАЛЬНЫХ ДАННЫХ ДЛЯ МОДЕЛИРОВАНИЯ ЗАДАЧ ПРОЧНОСТНОГО АНАЛИЗА С ИСПОЛЬЗОВАНИЕМ СОЛИД ЭЛЕМЕНТОВ

А. Л. Потехин, А. Н. Соловьев

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В работе отражены основные возможности по подготовке в препостпроцессоре ЛОГОС[1] начальных данных для проведения инженерного анализа задач прочности. Описаны средства по обработке геометрической модели (такие как импорт из САПР, редактирование геометрических моделей, включая операции упрощения), применение различных средств сеткогенерации, как на геометрических гранях, так и в объемных телах. Рассказывается о средствах прямой генерации сеток для случаев, в которых исходные геометрические модели используются в виде вспомогательных элементов, а пользователь осуществляет генерацию и редактирование расчетных сеток прямыми методами.

Ключевые слова: ЛОГОС, ЛОГОС-Препост, препроцессор, препостпроцессор, математическая модель, сеткогенерация, генерация сеток, расчетная сетка.

Введение

В мире не существует полностью автоматических средств подготовки начальных данных для моделирования задач прочности с использованием метода конечных элементов (МКЭ). Как следствие, для подготовки математических моделей используется развитое профильное программное обеспечение (ABAQUS [2], ANSYS [3], HyperMesh [4] и др.), которое может содержать себе сотни и даже тысячи различных операций.

При подготовке дискретных моделей (расчетных сеток) в инженерных пакетах преобладает использование двух разных подходов:

• подготовка сеток для «оценочных» расчетов с использованием автоматических алгоритмов, таких как генерация сеток из тетраэдров;

• существенное редактирование геометрических моделей и применение различных стилей генерации сеток в разных фрагментах для получения высококачественных моделей для получения результатов моделирования с высокой точностью.

Как правило, оба метода используются в различных пропорциях, в зависимости от класса задачи, сложности геометрии и опыта конкретного инженера.

Трудозатраты на подготовку одной математической модели для сложных исходных конструкторских сборок вполне могут составлять единицы человеко-лет. А для особо сложного конструктива с учетом высоких требований, предъявляемых к расчетным сеткам – десятки человеко-лет. При этом непосредственно время математического расчета, как правило, составляет незначительную долю времени относительно общего времени, затрачиваемого на подготовку математической модели, проведение и анализ результатов расчетов. Наибольшее время, как правило, занимает этап подготовки расчетных сеток, который может занимать до 80 % от общего времени расчета.

Поэтому, на рынке инженерных программных продуктов выигрывают те продукты, которые предоставляют максимум автоматизации всего процесса подготовки и проведения расчета.

При подготовке математических моделей для решения задач прочности можно выделить следующие ключевые этапы (рис. 1):

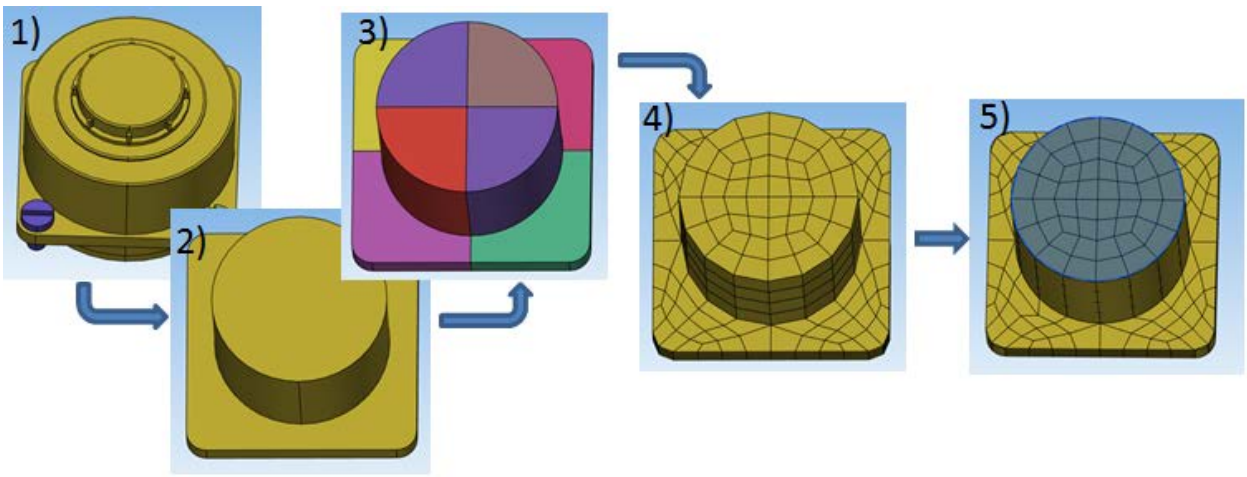


Рис. 1. Ключевые этапы подготовки математических моделей

1) первичная обработка геометрических моделей:

• трансляция геометрических моделей из файлов обменного (нативных или нейтральных текстовых) формата;

• проверка корректности загруженных геометрических элементов;

• анализ качества геометрических моделей с точки зрения наличия проблем, которые будут препятствовать построению качественных расчетных сеток;

• исправление/устранение проблем геометрических моделей.

2) идеализация/упрощение геометрических моделей, в частности, удаление характерных особенностей, снижение числа тел/граней/ребер, снижение размерности геометрических элементов;

3) редактирование геометрических моделей (в частности, подготовка моделей к сеткогенерации);

4) генерация расчетных сеток автоматическими и «прямыми» методами, операции редактирования расчетных сеток;

5) задание граничных и начальных условий, параметров моделирования и расчета.

При разработке препостпроцессора пакета программ ЛОГОС (далее ЛОГОС-ПП) реализовывались средства генерации сеток для обоих рассмотренных выше подходов. Очевидно, что для получения наиболее корректных и точных результатов инженерного анализа необходимым условием является обеспечение высокого качества сеточных моделей, и в данной работе акцент сделан именно на этом направлении.

В данной работе пойдет речь об этапах редактирования геометрических моделей и подготовки расчетных сеток, являющихся составной частью сквозной технологии проведения инженерных расчетов задач прочности на основе объемных элементов с использованием МКЭ.

Подготовка начальных данных в ЛОГОС-ПП

Редактирование геометрических моделей

Реализованные в ЛОГОС-ПП функциональные возможности по сфере применения условно можно разделить на следующие направления:

• анализ качества геометрических моделей, устранение проблем геометрических моделей;

• упрощение геометрических моделей;
• редактирование геометрических моделей (подготовка к сеткогенерации);
• генерация расчетных сеток автоматическими и «прямыми» методами;
• операции с ячейками/узлами расчетных сеток.

Как правило, в начале работы инженер загружает геометрическую модель из файла обменного формата, сохраненного из внешнего САПР. ЛОГОС-ПП использует математические алгоритмы 3D-моделирования, реализованные в программной библиотеке геометрического ядра C3D [5], разрабатываемой компанией С3ДЛабз. На этапе после загрузки геометрических моделей возможны различные коллизии, связанные, как с ошибками проектирования в САПР, так и с разным представлением одного конструктива в различных геометрических ядрах.

С целью недопущения подобных ошибок на более поздние этапы формирования математической модели в ЛОГОС-ПП реализованы средства анализа качества моделей для проверки наличия потенциальных проблем и пригодности моделей к генерации сеток. После активации пользователем диалога анализа качества он выбирает интересующие его критерии из перечня, при необходимости изменяет параметры критериев, заданные по умолчанию и запускает анализ, после чего проводит визуальную обработку результатов анализа. На рис. 2 представлен вид диалога задания параметров анализа качества, и вид тела с подсвеченными проблемными гранями, у которых наблюдается самопересечение контура, так называемые X-грани.

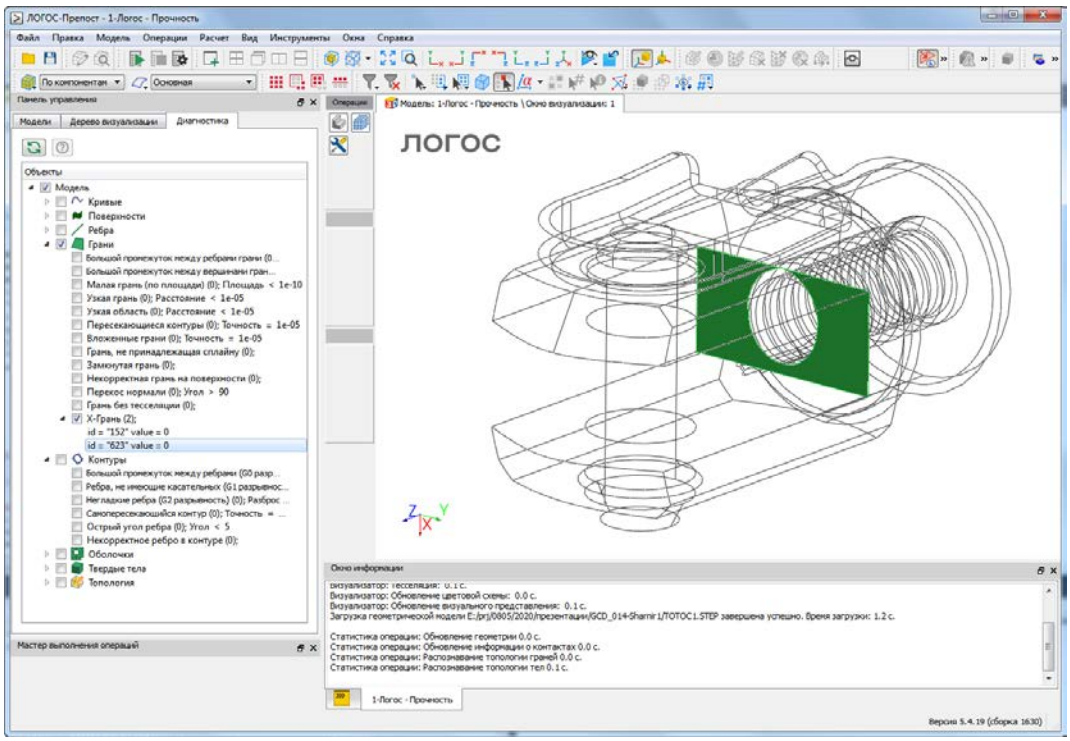


Рис. 2. Результаты анализа качества геометрической модели «Соединение с хомутом», загруженной с сайта GrabCAD [6]

Далее требуется оценить необходимый уровень детализации геометрической модели. Дело в том, что, как правило, конструкторская сборка содержит модель реального изделия с излишним с точки зрения компьютерного моделирования уровнем детализации. Различные мелкие конструктивные элементы вроде пазов, мелких отверстий, резьбы и т. д. могут не влиять на итог моделирования, но существенно осложнить этап подготовки сеточной модели и увеличить время моделирования в целом. Для упрощения геометрических моделей в рамках ЛОГОС-ПП реализованы следующие возможности: удаление отверстий, фасок, углублений, пазов; объединение ребер/граней, удаление характерных особенностей малого размера; построение срединных поверхностей. На рис. 3 приведен пример работы операции удаления характерных особенностей, а именно фаски и скругления.

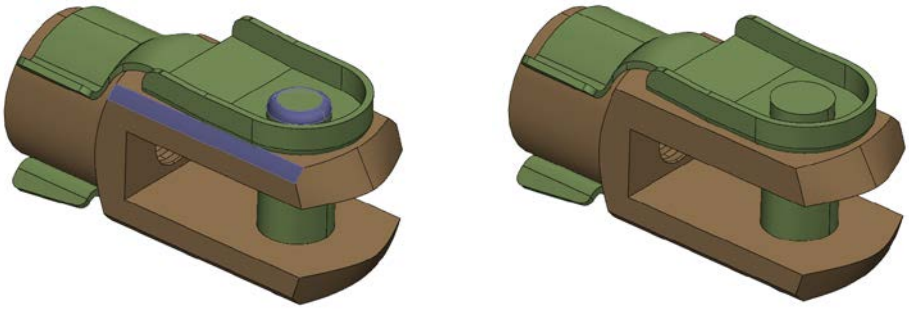


Рис. 3. Выбранные для удаления фаска и скругление (слева) и результат работы операции (справа)

Вместе с тем в ЛОГОС-ПП реализованы средства по явной замене части геометрических элементов на их более простые по форме аналоги: удаление тел и поверхностей, создание геометрических тел/граней/ребер. На рис. 4 приведена исходная конструкторская модель и модель с упрощенными телами подсборки «Шасси».

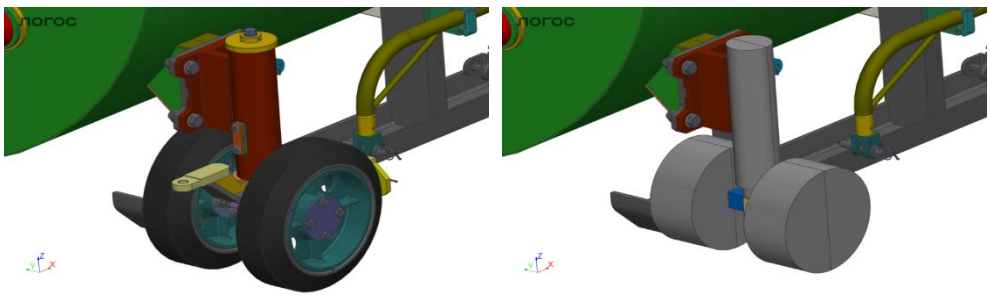


Рис. 4. Исходная конструкторская модель (слева) и упрощение ее подсборки «Шасси» (справа)

Для редактирования геометрических элементов в ЛОГОС-ПП реализовано более пятидесяти операций создания/редактирование тел/граней/ребер.

При решении повседневных инженерных задач их геометрические модели содержат элементы сложной формы. Генерация сеток автоматическими методами в таких объектах, как правило, не дает приемлемого качества расчетных сеток. Для повышения качества сеток используется разбиение тел сложной формы на набор тел простой формы с последующим построением сеток с использованием метода отображения или протягивания. На рис. 5 приведен пример отделения от тела сложной формы частей с обеспечением возможности дальнейшего построения в части новых тел сетки с использованием метода протягивания.

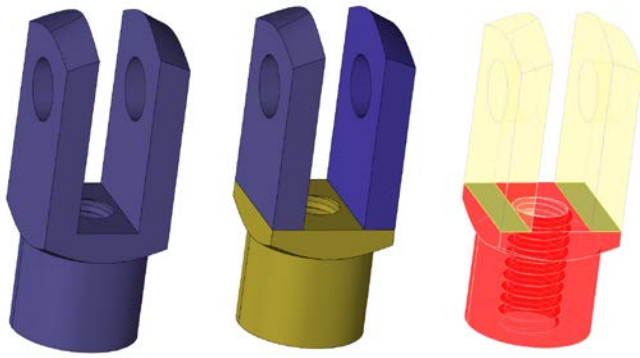


Рис. 5. Слева показано исходное тело; в центре три тела, полученные после отделения проушин заданной плоскостью, справа – желтым отображены тела, для которых возможно построение сетки методом протягивания

Средства пользовательского интерфейса

Кроме широкого перечня функциональных возможностей по редактированию геометрических элементов и генерации расчетных сеток в ЛОГОС-ПП (да и в других препроцессорах тоже) ключевыми являются возможности, обеспечивающие удобство работы пользователя. К таким возможностям относятся следующие средства:

- навигации и просмотра модели;
- выборочного отображения элементов модели;
- схемы визуализации и раскраски элементов модели;
- задания параметров интерактивного отбора элементов модели;
- работы с стандартными и пользовательскими видами
- набор типовых оконных элементов (виджетов) при формировании диалоговых окон задания параметров операций
- набор типовых элементов области визуализации (манипуляторов) для интерактивного задания параметров операций

1) Навигаторы: компоненты, сборка, группы

Навигаторы: компоненты, сборка, группы являются диалоговыми окнами, предоставляющими различные возможности группировки элементов геометрии для возможности выполнения массовых операций с объектами – скрытие/показ в области визуализации, задание цвета и т. п. (рис. 6).

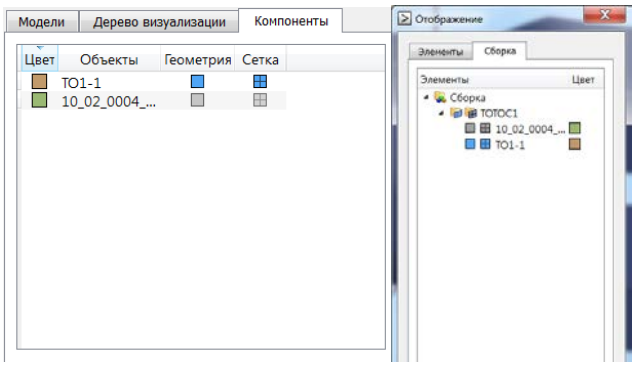


Рис. 6. Примеры навигаторов ЛОГОС-ПП

2) Выборочное отображение

Возможность выборочного отображения реализована в виде диалогового окна, предоставляющего возможности по управлению видимостью объектов по их типу, например, можно отобразить только элементы сетки – шестигранники и т. п. (рис. 7).

3) Схемы «раскраски» геометрических элементов

Схемы «раскраски» геометрических элементов позволяют различным образом представить геометрию/сетку в области визуализации, на рис. 8 показан выпадающий список с различными режимами раскраски геометрических элементов. Например, в режиме раскраски «Геометрические контакты» геометрическая модель отображается полупрозрачной, а контактные грани тел раскрашены желтым цветом, данная схема используется для наглядного представления контактных областей тел (второе тело справа на рис. 8).

4) Схемы отображения

Схемы отображения позволяют визуализировать геометрию/сетку различными способами – каркасом из ребер, заливкой, заливкой и реберным каркасом. Панель управления, предоставляющая возможность выбора между несколькими схемами отображения, представлена на рис. 9.

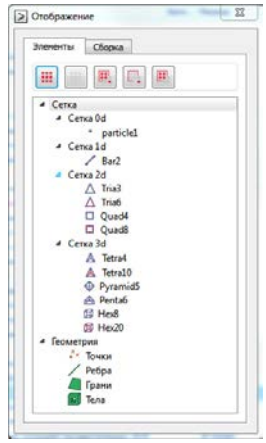


Рис. 7. Диалог управления типами отображаемых элементов

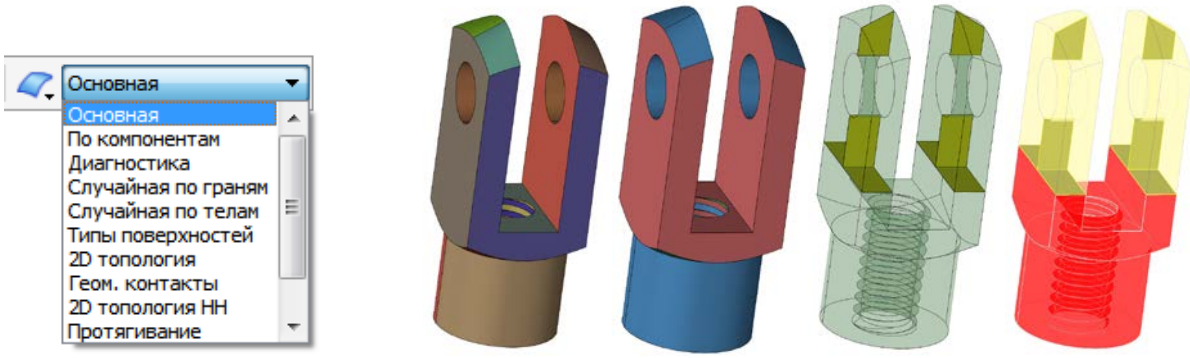


Рис. 8. Различные схемы раскраски геометрических элементов



Рис. 9. Панель управления схемами отображения

5) Области выделения и режимы отбора элементов

Области выделения и режимы отбора элементов позволяют выбрать тип отбираемых объектов в области визуализации, а также способ их отбора (точка, прямоугольник, круг, многогранник). Данные возможности позволяют эффективно отбирать объекты, как сеточные, таки геометрические, в области визуализации в интерактивном режиме. Элементы управления для отбора представлены на рис. 10.

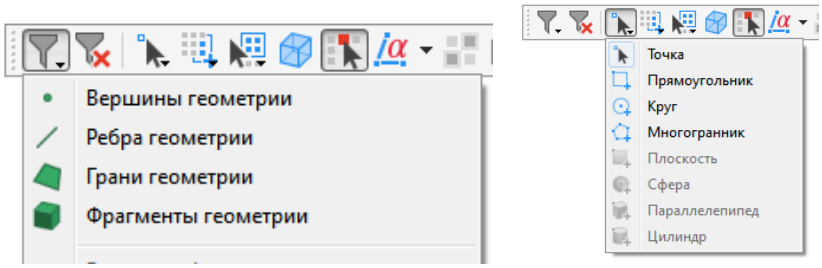


Рис. 10. Элементы управления для отбора элементов

6) Стандартные и пользовательские виды

Стандартные и пользовательские виды позволяют расположить модель одним из стандартных способов, а также запомнить/восстановить пользовательский вид. Управление видом показано на рис. 11.

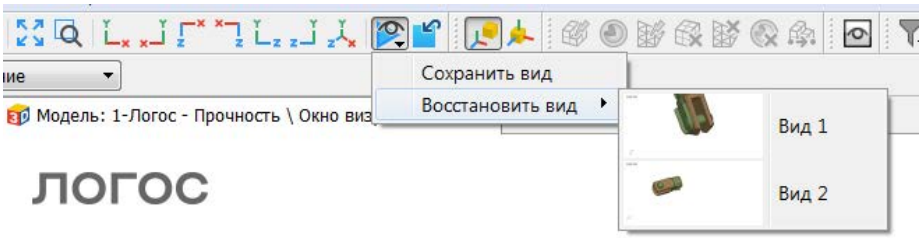


Рис 11. Элементы управления для отбора элементов

7) Типовые элементы диалогов задания параметров операций

Типовые элементы диалогов задания параметров операций реализованы в виде однотипных виджетов (отбор сеточных/геометрических объектов, задание параметров точек, векторов и т. п.),

располагаемых в схлопываемых блоках на диалоговом окне. Однотипные виджеты позволяют унифицировать интерфейс пользователя во всех операциях. На рис. 12 представлен интерфейс операции для разрезания тел плоскостью, содержащий виджеты для выбора тел, направления и точек.

8) Типовые манипуляторы в области отображения модели

Манипуляторы в области визуализации обеспечивают возможность как интерактивного задания параметров операций, а также позволяют облегчить пользователю визуальную интерпретацию заданных параметров. Например, на рис. 13 отображен манипулятор, позволяющий пользователю задавать параметры положения плоскости в 3D-пространстве.

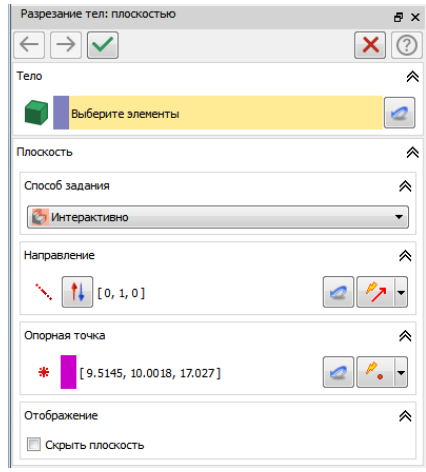


Рис. 12. Интерфейс операции для разрезания тел плоскостью

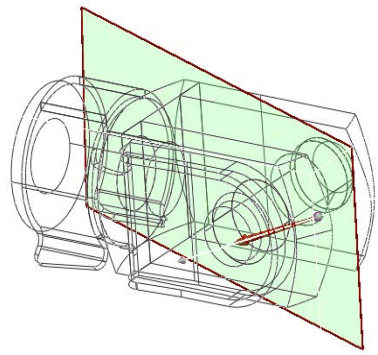


Рис. 13. Отображение манипулятора для задания положения плоскости

Сеткогенерация по геометрическим объектам

При построении сетки по геометрическому объекту перед инженером стоит задача получить сетку максимального качества, согласованную по узлам и хорошо аппроксимирующую исходную геометрическую модель. Для решения этой задачи в ЛОГОС-ПП имеются различные виды генераторов, а также механизмы определения наиболее подходящего метода, с предустановленными параметрами, поддержкой возможности согласования сеток с сетками соседних тел/граней и т. д.

При построении поддерживаются методы генерации на гранях треугольных, преимущественно-четырехугольных, полностью четырехугольных, а также блочно-регулярных и специальных сеток.

При построении объемных сеток в телах поддерживаются методы генерации тетраэдральных, преимущественно-шестигранных, сеток протягиванием и специальных сеток.

Генерация поверхностных сеток на гранях

При построении сеток непосредственно на геометрических гранях пользователь может выбрать режим сеткогенерации вручную, либо использовать режим автоматического выбора. На рис. 14 показано применение различных режимов к одной и той же геометрической грани. Работа части из этих режимов опирается на математические алгоритмы, реализованные в программной библиотеке CM2 MeshTools [7].

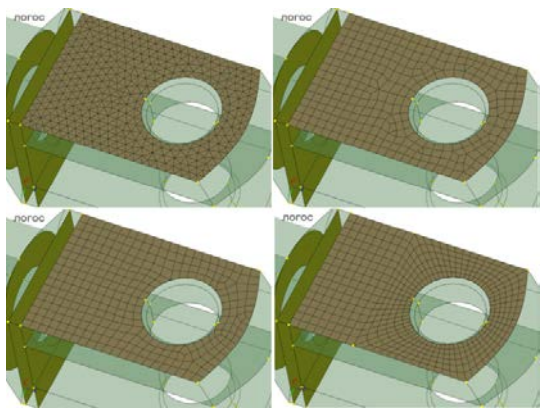


Рис. 14. Пример построения треугольной, преимущественно-четырехугольной, полностью четырехугольной и блочно-регулярной сеток

Сеткогенерация блочно-регулярных сеток на гранях

При генерации сеток с использованием блочно-регулярного метода первым этапом выполняется анализ формы граней на предмет из соответствия одному из нескольких предопределенных шаблонов. Если это соответствие обнаружено, значит, блочно-регулярная сетка может быть построена.

При анализе формы геометрических граней важно не количество геометрических вершин, а углы между касательными геометрических смежных ребер. На рис. 15 приведен перечень шаблонов, поддержка которых реализована в ЛОГОС-ПП, тем самым позволяя строить для подобных граней блочно-регулярные сетки.

Далее задаются разбиения на геометрических ребрах, учитывающие глобальные параметры и ограничения при генерации сеток данным методом. Опционально пользователь может изменить разбиения на отдельных геометрических ребрах, после чего определяется наиболее подходящий шаблон «разбиения» граней на регулярные согласованные блоки и в каждом блоке выполняется построение сетки с использованием блочного генератора [8]. На рис. 16 показаны примеры разбиения на примитивные блоки исходных граней, относящихся к различным шаблонам формы и результирующие сетки, сгенерированные для этих граней.

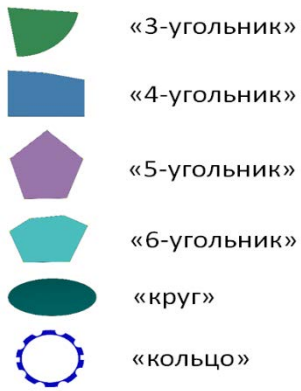


Рис. 15. Перечень шаблонов форм граней для генерации блочно-регулярных сеток

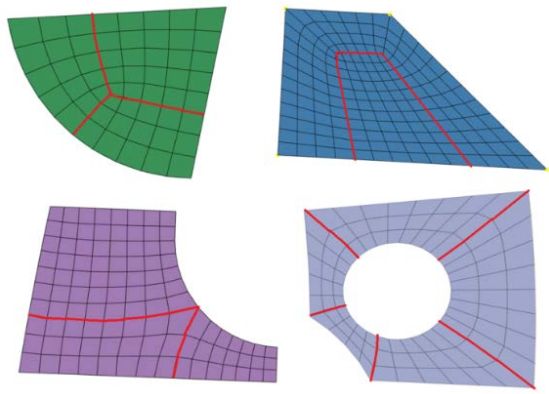


Рис. 16. Примеры построения блочно-регулярных сеток. Границы блоков показаны красным цветом

Генерация переходных слоев в поверхностных сетках

На практике среди множества типов геометрических граней часто встречаются кольца или их части. Исходя из стремления построить на таких гранях сетку из четырехугольных ячеек наиболее естественным видится применение простого генератора регулярных сеток (рис. 17).

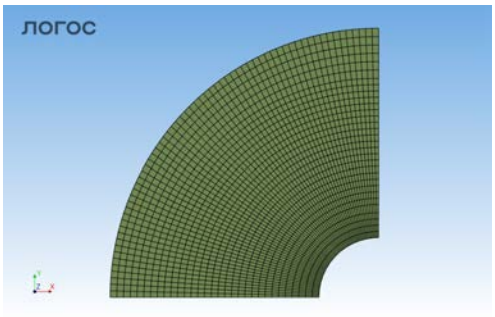


Рис. 17. Применение простого регулярного генератора

Итоговый размер сеточных элементов вблизи центра кольца значительно меньше размера на его большем радиусе, что приводит к общему увеличению числа элементов и, в итоге, к существенному увеличению времени моделирования. Для устранения данных недостатков используются специализированные алгоритмы формирования переходных слоев, направленные на сохранение размера сеточного элемента вдоль радиальной составляющей колец, как показано на рис. 18. Для этого грань разрезается на несколько частей, каждая из которых также является кольцом. Затем, начиная с периферии кольца, начинается построение сетки с использованием регулярного генератора. При этом для получения максимально качественного переходного слоя рекомендуется использовать число ячеек на границах согласования кратным друг другу.

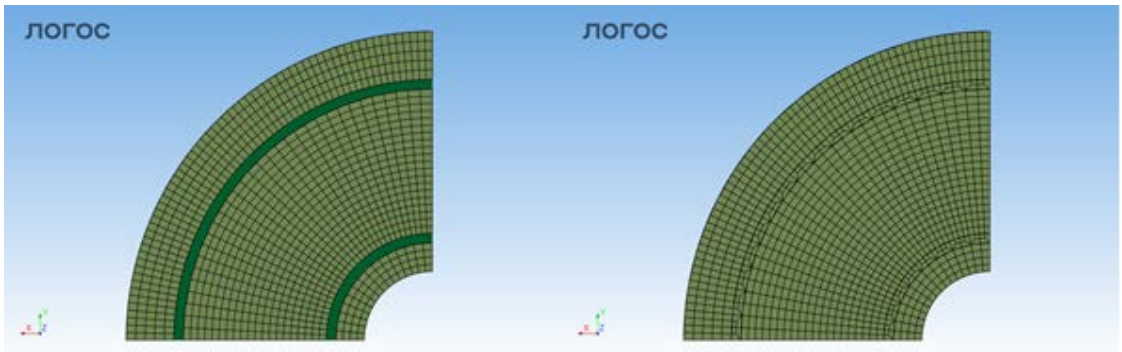


Рис. 18. Применение переходных слоев. Слева – исходные данные, справа – результат

Наконец, происходит согласование блоков посредством генерации на оставшихся гранях переходных слоев. Переходный слой состоит из последовательности повторяющихся блоков. На текущий момент используются два основных типа блоков «1 к 3» и двойной «1 к 2» (рис. 19).

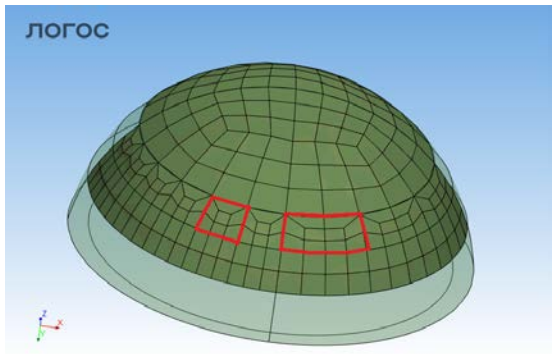


Рис. 19. Поддерживаемые шаблоны для переходных слоев

Генерация объемных сеток в телах методом протягивания

Построение объемных сеток протягиванием значительно упрощает и ускоряет процесс построения расчетных сеток. Данный метод позволяет строить 3D-сетки в телах, являющихся «топологическими призмами» с четырехугольными боковыми гранями. Инженер задает «стартовые» грани, на которых построена поверхностная сетка (рис. 20). Сетка строится протягиванием от набора «стартовых» граней к «финишной» (рис. 21). Также реализован алгоритм автоматического определения стартовых граней.

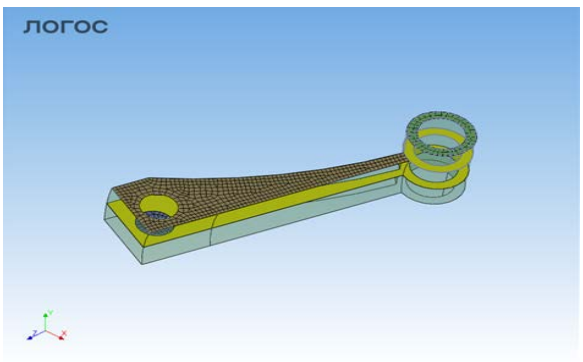


Рис. 20. Исходные данные для построения сеток протягиванием. Количество слоев протягивания может быть задано вручную

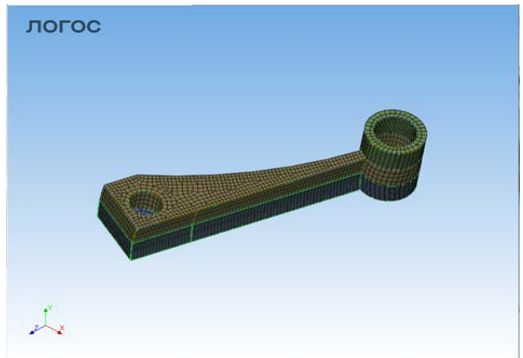


Рис. 21. Результат построения сеток протягиванием

Допускается наличие регулярной сетки на «боковых» гранях. В этом случае строится 3D-сетка, согласованная с сетками с «боковых» граней. Также учитываются сетки с соседних тел, если грани выбранного и соседнего тел являются контактными.

Сеткогенерация тетраэдрных сеток в телах

Тетраэдрная объемная сетка может быть построена практически в произвольном геометрическом объеме, что сильно расширяет область применения данной процедуры.

Реализованная в ЛОГОС-ПП процедура генерации тетраэдрной сетки допускает наличие сеток на гранях выбранного геометрического тела, при этом строится объемная сетка, согласованная с поверхностной. Также учитываются сетки с соседних тел, если грани выбранного и соседнего тел являются онтактными. В случае четырехугольных поверхностных элементов на грани – строится переходный слой пирамид.

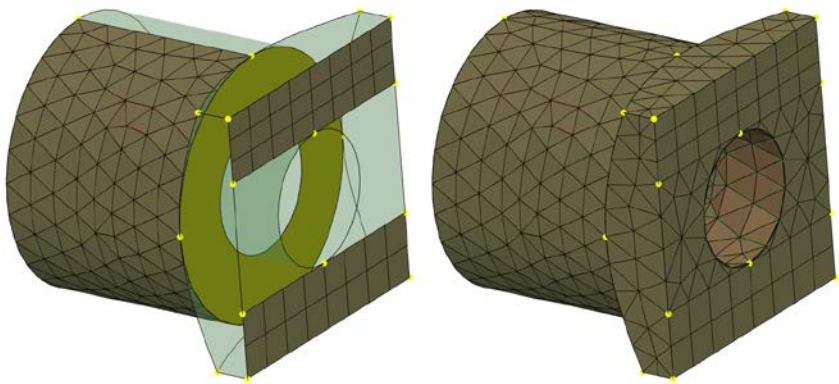


Рис. 22. Начальные данные для генерации тетраэдрных сеток. На трех гранях объема задана поверхностная сетка (слева). Итоговая сетка из тетраэдров и пирамид (справа)

«Прямая» генерация поверхностных сеток прямыми методами

На фрагментах сложной формы, где невозможно построить сетку протягиванием, упростить геометрическое тело или тело отсутствует, используются методы «прямой» генерации сеток. Такие как:

- извлечение поверхностной сетки (из набора объемных элементов);
- сетка по линейчатой поверхности;
- экструзия/протягивание;
- создание ячеек офсетом.

После сеткогенерации любого типа (по геометрическим объектам или «прямыми» методами) возможно редактирование сеточных элементов следующими алгоритмами:

- разбиение ячеек по шаблону;
- дробление слоя ячеек;
- объединение ячеек;
- редактирование/согласование направления нормалей;
- изменение направления «обхода» ячеек;
- создание/удаление узлов/ячеек;
- сшивка/расшивка узлов;
- поиск дубликатов/вырожденных ячеек;
- перемещение/проекция узлов;
- удаление сетки на геометрии.

Для контроля качества построенных сеток и ее оптимизации реализованы операции:

- анализ качества;
- сглаживание сеточных элементов.

Заключение

В ЛОГОС-ПП реализованы базовые средства генерации поверхностных и объемных сеток, как по элементам геометрических моделей, так и методами «прямой» сеткогенерации, что позволяет готовить сеточные модели для дальнейшего моделирования прочности в отчужденном режиме с использованием препостпроцессора ЛОГОС.

Литература

1. Пакет программ инженерного анализа и суперкомпьютерного моделирования [Electronic resource]–Mode of access http://logos.vniief.ru;

2. Simulia ABAQUS [Electronic resource] – Mode of access http://www.3ds.com/ru/produkty-i-uslugi/simulia/produkty/abaqus;

3. ANSYS Workbench [Electronic resource] – Mode of access http://ansys.com/Products/Platform;

4. Altair HyperMesh [Electronic resource] – Mode of access http://www.altairhyperworks/product/HyperMesh;

5. Геометрическое ядро C3D [Electronic resource] – Mode of access https://c3dlabs.com/ru/products/c3d-kernel/;

6. GrabCAD Library. The largest online community of professional designers, engineers, manufacturers and student [Electronic resource]–Mode of access https://grabCAD.com/clevis-joint/;

7. CM2 MeshTools Suite [Electronic resource] – Mode of access http://computing-objects.com/cm2-meshtools-suite/

8. Лазарев В. В. Распараллеливание и оптимизация построения блочных сеток в препроцессоре ЛОГОС // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 1. С. 54–63.

PREPROCESSOR OF “LOGOS” SOFTWARE PACKAGE: INITIAL DATA PRE-PROCESSING FOR SIMULATION OF THE STRENGTH ANALYSIS PROBLEMS USING SOLID ELEMENTS

A. L. Potekhin, A. N. Solovev

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents the LOGOS prepostprocessor [1] functionality to prepare data for the engineering strength analysis. The geometric model processing tools for data importing from CAD system, geometric model editing, including simplification operations, and the use of various tools for the mesh generation on geometric faces and three-dimensional bodies are described. Also, the paper describes tools for the direct generation of meshes in cases, when the original geometric models are used as auxiliary means and the user performs the generation and editing of a computational mesh with the direct methods.

Key words: LOGOS, Logos Prepost, preprocessor, postprocessor, a mathematical model, mesh generation, a computational mesh.

УДК 51
DOI 10.53403/9785951505071_2022_386

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ КОЛЕБАНИЙ СТВОЛА С УЧЕТОМ ТЕПЛООБМЕНА ПРИ СТРЕЛЬБЕ ОЧЕРЕДЯМИ

И. Г. Русяк, В. Г. Суфиянов, С. А. Королев, Д. А. Клюкин

Ижевский государственный технический университет им. Калашникова М. Т., Ижевск

Актуальной проблемой проектирования автоматических пушек является повышение точности и кучности стрельбы. Эти показатели в значительной степени зависят от деформации и колебаний ствола при стрельбе очередями, которые, в свою очередь, вызваны давлением пороховых газов и взаимодействием снаряда со стволом. Математическое моделирование данных процессов позволяет оценить влияние учитываемых факторов на начальные условия движения снаряда на внешнебаллистической траектории. В данной работе рассматривается математическая модель ствола автоматической пушки в одномерном постановке, в соответствии с теорией упругих продольно-поперечных колебаний стержней.

Ключевые слова: математическое моделирование, теплообмен, стрельба очередями.

Динамика ствола в одномерной постановке описывается моделью консольной балки [1], которая представляет собой толстостенную цилиндрическую трубу переменного сечения с технологическими отклонениями, получаемыми при изготовлении ствола, например, технологической непрямолинейностью линии центров канала ствола.

Еще одним фактором, оказывающим существенное влияние на колебания ствола, особенно в процессе стрельбы очередями, является его нагрев при взаимодействии с пороховыми газами. Изучению вопроса нагрева ствола [2] посвящены работы советских ученых Лебедева Н. И., Панковича П. Ф., Ильюшина А. А. и др., которые решали задачи расчета нестационарных температурных полей, термоупругих напряжений в стволах и тепловых напряжений за пределами упругости. Изучение процессов теплопроводности представляет значительный практический интерес по следующим причинам:

– увеличение мощности артиллерийских орудий приводит к увеличению нагрева ствола, что может значительно уменьшить их живучесть;

– возникающие перепады температур при выстреле являются источниками дополнительных напряжений;

– нагретый ствол оказывает воздействие на функционал ведущего пояска снаряда, что влияет на кучность стрельбы;

– на кучность стрельбы также влияет изменение характеристик металла, а именно его жесткость, что влечет к увеличению динамических и статических изгибов в стволе.

Целью проведения исследования является математическое моделирование продольно-поперечных колебаний ствола в процессе стрельбы очередями и анализ влияния разностенности и нагрева ствола, с учетом зависимости упругих свойств материала ствола от температуры, на размах поперечных отклонений дульного среза и углов вылета снаряда из ствола.

1. Математическая модель колебаний ствола

Рассмотрим дифференциальное уравнение в частных производных для колебаний ствола орудия в направлении оси Ox [3]:

$$\rho F \frac{\partial^2 u}{\partial t^2} = -\rho F g \sin\varphi + q_x + \frac{\partial}{\partial x}\left(F\sigma^x + 2\nu S p\right) - p\frac{\partial S}{\partial x}, \tag{1}$$

где ρ – плотность материала ствола; $F = F(x)$ – переменная площадь сечения ствола; $u = u(x,t)$ – величина продольной деформации; g – сила тяжести; φ – угол возвышения ствола; $q_x = q_x(x,t)$ – распределенные продольные силы; σ^x – напряжение в сечении x; ν – коэффициент Пуассона; $S = S(x)$ – переменная площадь сечения канала ствола; $p = p(x,t)$ – распределение давления внутри канала ствола.

Напряжение в сечении σ^x определяется по формуле:

$$\sigma^x = E\frac{\partial u}{\partial x} - \frac{\alpha E}{F}\int_F T df, \tag{2}$$

где E – модуль упругости; α – коэффициент линейного теплового расширения.

При определении динамики движения снаряда и распределения давления применялся термодинамический подход [4], система уравнений внутренней баллистики при этом решалась методом Рунге – Кутта 4-го порядка [5]. Для периода последействия решалась задача истечения газов со звуковой скоростью.

Рассмотрим уравнения поперечных колебаний в плоскостях Oxy и Oxz [5]:

$$\begin{aligned}\rho F \frac{\partial^2 v}{\partial t^2} = -\rho F g\sin\varphi + q_y + \frac{\partial}{\partial x}\left(\left(F\sigma^x - pS\right)\frac{\partial(v+v_{00})}{\partial x}\right) - \\ - \frac{\partial^2}{\partial x^2}\left(EJ_z\frac{\partial^2 v}{\partial x^2}\right) - \frac{\partial^2}{\partial x^2}\left(\alpha E\int_F T y df\right),\end{aligned} \tag{3}$$

$$\begin{aligned}\rho F \frac{\partial^2 w}{\partial t^2} = q_z + \frac{\partial}{\partial x}\left(\left(F\sigma^x - pS\right)\frac{\partial(w+w_{00})}{\partial x}\right) - \\ - \frac{\partial^2}{\partial x^2}\left(EJ_y\frac{\partial^2 w}{\partial x^2}\right) - \frac{\partial^2}{\partial x^2}\left(\alpha E\int_F T z df\right),\end{aligned} \tag{4}$$

где $v = v(x,t)$, $w = w(x,t)$ – величины поперечной деформации в вертикальной Oxy и горизонтальной Oxz плоскостях; J_z, J_y – моменты инерции сечений по соответствующим осям; v_{00}, w_{00} – разность технологических отклонений от координат центра тяжести сечения канала по осям Oy и Oz; $q_y = q_y(x,t)$, $q_z = q_z(x,t)$ – распределенные поперечные силы.

Технологические отклонения центров внутренних сечений по осям Oy и Oz представляют собой функции от координаты x: $y = y(x)$, $z = z(x)$.

Граничное условие для казенного среза представляет собой условие закрепления, а для дульного среза – отсутствие закрепления.

В качестве начальных условий к задачам (1), (3) и (4) принимаются решения одномерных уравнений статического прогиба упругого стержня под действием силы тяжести.

Задача теплопроводности рассматривалась в осесимметричной постановке. Распределение температур в поперечном сечении ствола определялось из решения одномерной задачи:

$$\frac{\partial\left(c\rho T(r,t)\right)}{\partial t}=\frac{1}{r}\frac{\partial}{\partial r}\left(r\lambda\frac{\partial T(r,t)}{\partial r}\right), \tag{5}$$

где $T(r,t)$ – величина теплового поля; c – теплоемкость материала ствола, λ – теплопроводность материала ствола; r – радиус, $r\in[r_0,r_1]$; r_0 – радиус канала ствола; r_1 – внешний радиус ствола.

Начальное условие:

$$T(r,0)=T_{\text{в}},$$

где $T_{\text{в}}$ – температура воздуха.

Граничные условия [3]:

$$-\lambda\left.\frac{\partial T(r,t)}{\partial r}\right|_{r=r_0}=a_{\text{г}}\cdot\left(T\big|_{r=r_0}-T_{\text{г}}(t)\right),$$

$$-\lambda\left.\frac{\partial T(r,t)}{\partial r}\right|_{r=r_1}=a_{\text{в}}\cdot\left(T_{\text{в}}-T\big|_{r=r_1}\right),$$

$$a_{\text{г}}=0.023\cdot\text{Re}^{0.8}\cdot\text{Pr}^{0.43}\cdot\lambda_{\text{г}}/d\ ,$$

где $a_{\text{г}}$, $a_{\text{в}}$ – коэффициенты теплоотдачи газопороховой смеси и воздуха соответственно; $T_{\text{г}}(t)$ – температура газопороховой смеси; $\lambda_{\text{г}}$ – теплопроводность газопроховой смеси; Re – число Рейнольдса; Pr – число Прандтля.

Разностная аппроксимация дифференциальных уравнений проводилась интегро-интерполяционным методом [6], позволяющим находить решения для моделей с негладкими и разрывными коэффициентами.

2. Результаты исследования влияния теплообмена на колебания ствола

Проведем моделирование колебаний ствола 30 мм автоматической пушки с нарезным стволом, с учетом упругих и массовых сил, давления пороховых газов, взаимодействия снаряда со стволом и теплообменом с газопороховой смесью. Угол возвышения ствола принимался равным $\varphi=5°$.

При расчете внутренней баллистики пушки максимальное давление внутри ствола составило $p_{\max}=370$ МПа, скорость снаряда на дульном срезе – $v_{\text{сн}}=890$ м/с, время выстрела – $t_{\text{д}}=4{,}5$ мс. В расчетах рассматривалась автоматическая стрельба очередью из 8 снарядов с периодом $t_{\text{пер}}=75$ мс [7]. Материал ствола имеет следующие характеристики: $\rho=7800$ кг/м^3, $c=567$ Дж/(кг·К), $\lambda=32$ Вт/(м·К), зависимость модуля Юнга от температуры выбрана для стали 38ХН3МФА [7] и представлена на рис. 1. Общее время стрельбы составило $\tau=0{,}6$ с.

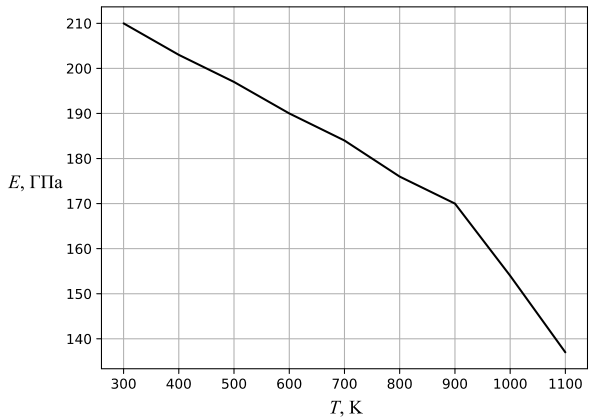


Рис. 1. Зависимость модуля упругости от температуры для материала ствола

Распределение температуры по поверхности ствола в момент окончания стрельбы $\tau = 0{,}6$ с представлено на рис. 2.

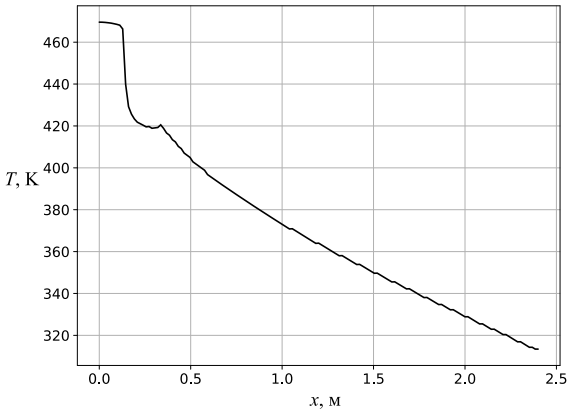


Рис. 2. Распределение температуры на поверхности ствола
в момент времени $\tau = 0{,}6$ с

В виду скоротечности процесса выстрела, успевает прогреваться лишь внутренняя поверхность ствола, температура же снаружи после очереди из 8 выстрелов превышает температуру окружающей среды на 17 градусов.

Функция технологических отклонений имеет вид:

$$g(x) = A\left(1 - \cos\left(\frac{2\pi}{L}x\right)\right), \tag{6}$$

где L – длина ствола; A – амплитуда отклонений.

В качестве оцениваемых параметров рассматривались размахи значений отклонения дульного среза в вертикальной и горизонтальных плоскостях $0xy$ и Oxz :

$$R_y = \max_{t\in[0,\tau]} v(l,t) - \min_{t\in[0,\tau]} v(l,t), \tag{7}$$

$$R_z = \max_{t\in[0,\tau]} w(l,t) - \min_{t\in[0,\tau]} w(l,t). \tag{8}$$

Примеры колебаний дульного среза вариантов 2 и 3 (см. табл. 1) представлены на рис. 3.

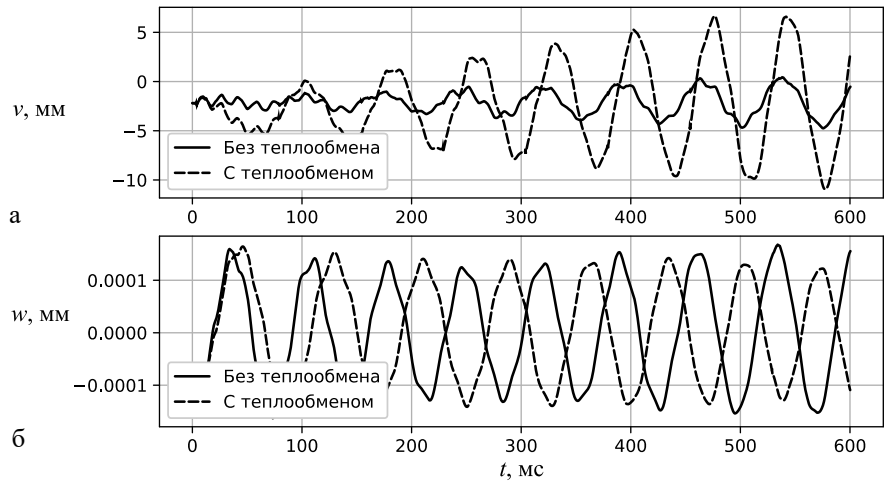


Рис. 3. Поперечные колебания дульного среза в вертикальной (*а*) и горизонтальной (*б*) плоскостях

Как видно из рис. 3, теплообмен может значительно увеличивать амплитуду колебаний ствола. Это связано с изменениями в характеристиках материала в зависимости от температуры, в данном случае – модуля упругости. В горизонтальной плоскости влияние тепловых воздействий имеет несколько иной вид, амплитуда остается прежней, но при этом увеличивается период колебаний. Колебания оценивались по величине размаха максимальных отклонений угла бросания снаряда от угла возвышения

$$R_\gamma = \max_{t\in[0,\tau]}\gamma(t) - \min_{t\in[0,\tau]}\gamma(t) \tag{9}$$

и размаха максимального отклонения угла вылета снаряда из дульного среза в плоскости Oxz

$$R_\psi = \max_{t\in[0,\tau]}\psi(t) - \min_{t\in[0,\tau]}\psi(t) \tag{10}$$

Динамика изменения отклонения угла бросания снаряда от угла возвышения в плоскости Oxy и отклонения угла вылета снаряда в плоскости Oxz представлена на рис. 4.

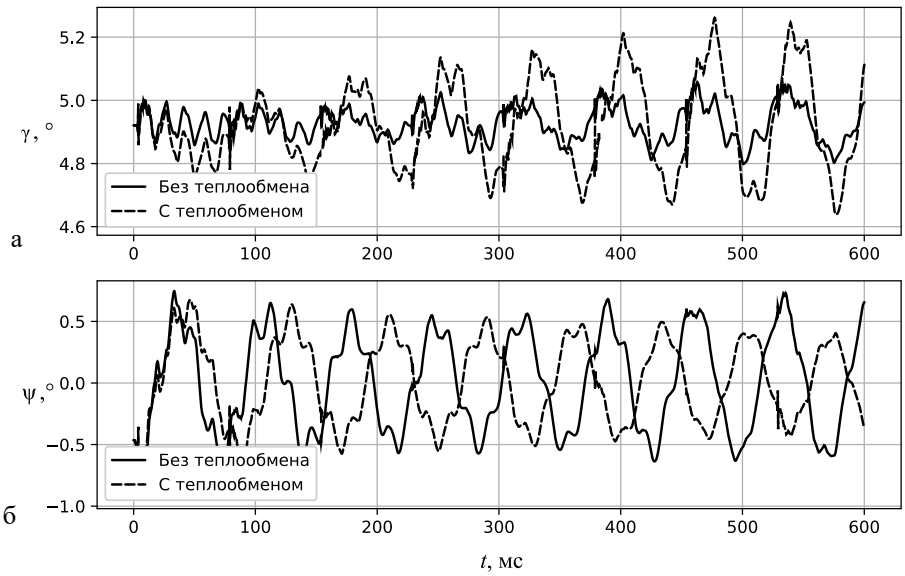


Рис. 4. Отклонения углов вылета снаряда в горизонтальной (*а*) и вертикальной (*б*) плоскостях

Результаты моделирования колебаний ствола с различными параметрами ($A = A_y$ – в вертикальной плоскости, $A = A_z$ – в горизонтальной плоскости) задания функций (6) технологических отклонений центров внутренних сечений $y = y(x)$ и $z = z(x)$ по осям Oy и Oz, соответственно, а также с учетом и без учета теплообмена представлены в табл. 1.

Таблица 1

Влияние теплообмена на размах колебаний ствола

№	Входные данные		С учетом теплообмена	R_y, мм	R_z, мм	R_γ, °	R_ψ, °
	A_y, мм	A_z, мм					
0	–	–	Нет	6,9	0,0	0,23	0,00
1	–	–	Да	7,3	0,0	0,31	0,00
2	1,0	–	Нет	5,1	0,0	0,26	0,00
3	1,0	–	Да	17,6	0,0	0,66	0,00
4	–	1,0	Нет	7,3	12,5	0,29	0,55
5	–	1,0	Да	6,9	10,9	0,31	0,41
6	1,0	1,0	Нет	3,5	5,0	0,18	0,26
7	1,0	1,0	Да	10,5	4,1	0,45	0,21

Как видно из табл. 1, теплообмен между газопороховой смесью и стволом существенно влияет на размахи отклонений дульного среза R_y, R_z и углов R_γ, R_ψ в процессе стрельбы очередями из автоматической пушки.

Заключение

Таким образом, в результате проведения работы реализована математическая модель продольно-поперечных колебаний ствола в одномерной постановке с учетом теплообмена пороховых газов со стволом. В математической модели учитываются динамика внутрибаллистического давления и движение снаряда внутри канала ствола, а также технологические отклонения линий центров ствола. На основе математического моделирования показано, что тепловая составляющая оказывает существенное влияние на размах отклонений дульного среза и углы вылета снаряда при стрельбе очередями.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-01-00072.

Литература

1. Тимошенко С. П. Прочность и колебания элементов конструкций. –М.: Наука. 1975.–С. 704.

2. Орлов Б. В., Ларман Э. К., Маликов В. Г. Устройство и проектирование стволов артиллерийских орудий. –М.: Машиностроение, 1976. – С. 432.

3. Хоменко Ю. П., Ищенко А. Н., Касимов В. З. Математическое моделирование внутрибаллистических процессов в ствольных системах. –Новосибирск: Издательство СО РАН, 1999. – С. 256.

4. Русяк И. Г., Липанов А. М., Ушаков В. М. Физические основы и газовая динамика горения порохов в артиллерийских системах. – М. Ижевск: Институт компьютерных исследований, 2016. С. 456.

5. Самарский А. А., Гулин А. В. Численные методы. – М.: Наука, 1989. – С. 432.

6. Липанов А. М., Русяк И. Г., Суфиянов В. Г. Исследование влияния колебаний ствола на угол вылета снаряда при выстреле // Вестник Томского государственного университета. Математика и механика. 2020. № 68. С. 80–94.

7. Марочник сталей и сплавов / Сорокин В. Г., Волосникова А. В., Вяткин С. А. и др.; Под общ. ред. Сорокина В. Г. –М.: Машиностроение, 1989. – С. 640.

MATHEMATICAL MODELING OF BARREL VIBRATIONS WITH REGARD TO HEAT EXCHANGE IN SERIAL FIRE

I. G. Rusyak, V. G. Sufiyanov, S. A. Korolyev, D. A. Klyukin

Kalashnikov Izhevsk State Technical University, Izhevsk

The urgent problem of designing automatic guns is the need to increase the accuracy of fire and density of hits. These metrics depend, to a significant extent, on the gun barrel strain and vibrations in serial fire, which, in turn, are caused by the pressure of explosive gases and the projectile-barrel interaction. The mathematical modeling of these processes allows estimating the effect of the factors above on the initial condition of the projectile motion along external ballistic trajectory. The paper considers a 1D mathematical model of a gun barrel according to theory of elastic longitudinal and transverse vibrations of rods.

Key words: mathematical modeling, heat exchange, serial fire.

УДК 004.4
DOI 10.53403/9785951505071_2022_392

РЕШЕНИЕ ЗАДАЧ АЭРОДИНАМИКИ С ПОДВИЖНЫМИ ГРАНИЦАМИ В ПАКЕТЕ ПРОГРАММ ЛОГОС

А. В. Саразов, А. С. Козелков, Д. К. Зеленский, Р. Н. Жучков, Т. В. Резвова

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Доклад посвящен вопросам численного моделирования задач аэродинамики с подвижными границами. В работе приводятся основные подходы расчета обтекания подвижных объектов в пакете программ ЛОГОС. Рассмотрена методика расчета на деформирующихся сетках с сохранением топологии связей. Обсуждаются достоинства и недостатки данного подхода. Дополнительно приводится технология расчета задач аэродинамики на сетках с перекрытиями. Приводятся основные этапы технологии, введены ключевые термины и правила расчета. Дополнительно обсуждается класс задач с примерами, где необходимо применять вышеуказанные подходы совместно. В качестве демонстрации работоспособности и применимости реализованных методик расчета приводятся решения характерных задач авиационной промышленности, имеющих экспериментальные данные.

Ключевые слова: пакет программ ЛОГОС, технология расчета на сетках с перекрытием, деформирующиеся сетки с сохранением топологии связи, обледенение, NACA0012, FSI, AGARD 445.6, HIRENASD, AGARD 23.

Введение

В настоящее время в области численной аэродинамики растет потребность в решении задач, связанных как с движением самих обтекаемых тел, так и их отдельных частей. В качестве примеров можно привести такие практически важные задачи, как движение органов управления летательного аппарата, задача катапультирования грузов в присутствии носителя, задачи деформации крыла под действием набегающего потока. До широкого внедрения современной вычислительной техники большинство результатов данного класса задач было получено на основе предположений, позволяющих полностью или частично линеаризовать исходные уравнения. В линейной постановке были выявлены основные особенности задач взаимодействия, разработаны точные и приближенные методы решения. Однако, решение в линейной постановке далеко не всегда удовлетворяет требованиям практики. В реальных процессах в газе возникают сложные волновые эффекты, часто сопровождаемые проявлением физически и геометрически нелинейных свойств среды.

Эффективным подходом качественного описания процессов аэродинамики является конечно-объемная методика, которая требует доработки для возможности численного моделирования течения с подвижными границами. В качестве возможных вариантов адаптации конечно-объемной методики для возможности моделирования течений с подвижными границами можно выделить подходы, реализованные в пакете программ ЛОГОС: деформация расчетной сетки с сохранением топологии связей и подход, основанный на сетках с перекрытиями. Выбор того или иного подхода для численного моделирования определяется непосредственно постановкой задачи, и в подавляющем большинстве случаев он очевиден. Однако, существуют задачи, в которых нельзя отдать предпочтение лишь одной из альтернатив ввиду невозможности полного описания физических процессов с использованием только одного подхода.

В работе приводится обзор реализованных физико-математических моделей в рамках пакета программ ЛОГОС для задач численной газовой динамики с подвижными элементами конструкции. Приводится решение характерных задач авиационной промышленности.

1. Математическая модель

1.1. Адаптация базовых уравнений

Нестационарное течение вязкого сжимаемого газа описываются базовыми уравнениями Навье – Стокса:

$$\int_{\Omega} \frac{\partial W}{\partial t} dV + \oint_{\partial\Omega} (F - G) dS = \int_{\Omega} H dV \tag{1}$$

где t – физическое время; W – вектор консервативных переменных; F – вектор конвективных потоков; G – вектор диффузионных потоков; H – правая часть уравнений.

В том случае, когда одна или несколько границ расчетной области изменяют свое положение, уравнение (1) не может быть использовано, так как движение границ вносит ошибки при дискретизации, что влечет за собой неконсервативность численной схемы. Для корректного описания характеристик течения необходимо адаптировать исходные уравнения Навье – Стокса посредством учета вклада движения граней контрольного объема при расчете потоков.

Базовый закон можно модифицировать, используя дифференциальное тождество:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV = \int_{\Omega(t)} \frac{\partial W}{\partial t} dV + \int_{\partial\Omega(t)} W \left(\vec{\dot{x}} \cdot \vec{n} \right) dS \tag{2}$$

где $\vec{\dot{x}}$ – скорость движения грани контрольного объема.

После подстановки (2) в (1) система Навье – Стокса примет вид:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV + \int_{\partial\Omega(t)} \left(F(W) - \vec{\dot{x}} W \right) \cdot \vec{n} dS - \int_{\partial\Omega(t)} \left(G(W) \right) \cdot \vec{n} dS = \int_{\Omega(t)} H dV \tag{3}$$

Как видно, первое слагаемое отражает не только изменение консервативных переменных во времени, но и скорость изменения контрольного объема. Также следует отметить, что при расчете конвективных потоков необходимо учитывать скорость движения грани. При расчете на подвижных сетках, как и для случая статических сеток, на границе задаются условия непротекания и прилипания. Скорость движения грани определяется на основе закона геометрической консервативности [1]. В том случае, если все границы неподвижны, то из уравнения (3) следует базовый закон Навье – Стокса.

1.2. Численная схема

Численный метод расчета нестационарной задачи состоит из предположения, что течение потока в каждый момент времени рассматривается как результат установления некоторого стационарного процесса. Поэтому вместо решения уравнений Навье – Стокса вида (3) рассматриваются модифицированные уравнения:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV + \frac{\partial}{\partial \tau} \int_{\Omega(t)} W dV + \oint_{\partial\Omega(t)} \left(F - \vec{\dot{x}} W - G \right) dS = \int_{\Omega(t)} H dV \tag{4}$$

где τ – псевдовремя.

Уравнение (4) допускает формальную дискретизацию по первым двум слагаемым. Схема первого порядка точности по времени уравнения (4) имеет вид:

$$\left[\frac{V^{n+1}}{\Delta\tau}+\frac{2V^{n+1}-V^{n}}{\Delta t}\right]\Delta W^{k+1}+\oint\limits_{\partial\Omega(t)}\left[F-\vec{x}W-G\right]dS=\int\limits_{\Omega(t)}HdV-\frac{1}{\Delta t}\left(W^{k}\left(2V^{n+1}-V^{n}\right)-W^{n}V^{n+1}\right) \quad (5)$$

где V^{i} – объем ячейки на моменты t_i.

Для расчета конвективных потоков могут использоваться различные схемы аппроксимации, такие как Roe, AUSMP, метод Годунова.

2. Методика деформирующихся сеток

Помимо методических аспектов важную роль играет алгоритм, учитывающий изменение исходной расчетной сетки. Одним из подходов является методика деформирующихся сеток с сохранением топологии связей, позволяющая успешно проводить численное моделирование процессов в случае несущественных деформаций тела. В данном случае расчетная сетка изменяется под требуемое положение границ путем обновления лишь координат узлов модели.

В пакете программ ЛОГОС реализован алгоритм деформации сетки IDW (Inverse Distance Weighting) [2]. Данный подход не требует решения системы уравнений для вычисления нового положения узлов и прост в реализации. Суть метода IDW состоит в том, что для вычисления перемещений внутренних узлов используется интерполяционная функция смещения граничных вершин. Базовый подход использует функцию интерполяции вида:

$$s(x)=\frac{\sum w_j(x)s_j}{\sum w_j(x)} \quad (6)$$

$$w_j(x)=\left\|x-x_{b_j}\right\|^{p},\ p<0 \quad (7)$$

где $w_j(x)$ – есть весовая функция; s_j – перемещение узла на границе.

В случае вырождения или сильной деформации данный подход не применим, и требуются альтернативные методы решения, например, сетки с перекрытиями.

3. Методика расчета на сетках с перекрытиями

Методика расчета на сетках с перекрытиями [3] позволяет заменить методику расчета на деформирующихся сетках в тех случаях, когда деформации носят серьезный характер. Данный подход предполагает построение независимых сеток для каждого тела в отдельности, в дальнейшем объединяя их в одну общую сетку с перекрытиями. Каждая сетка учитывает форму и геометрические особенности только своего тела, что позволяет упростить процесс построения расчетной геометрии.

В случае использования перекрывающихся сеток расчет производится на двух и более топологически несвязанных между собой дискретных моделях. Центральной задачей при создании расчетной методики является сопряжение этих областей. Ключевые этапы – это определение несчетных областей и построение интерполяционного шаблона для взаимодействия несвязанных сеточных регионов (рис. 1).

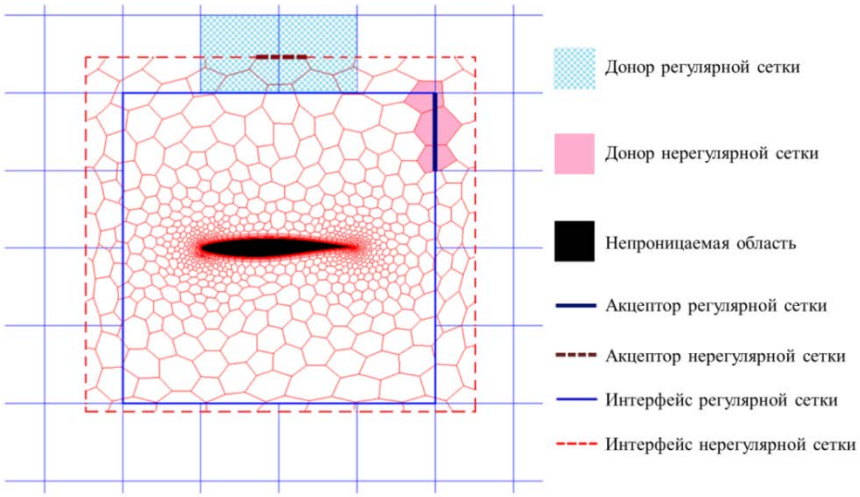


Рис. 1. Пример сетки с перекрытием

Ниже приведены важные определения:

– интерфейс – открытая граница дискретной модели, значение полей на которой формируется на основании данных из окрестных ячеек;

– акцептор – грань интерфейса;

– доноры – ячейки, расчетные поля которых используются для восстановления информации на интерфейсах;

– операция «сечение» – метод определения пересечения поверхностей одной из сеток с ячейками другой сетки в пространстве.

Также существует несколько базовых правил расчета задач на сетках с перекрытиями:

– к основным типам граничных условий добавляется специализированное условие – интерфейс;

– непроницаемые границы, такие как стенка, не допускают проникновение сквозь нее потока для любой сетки;

– все ячейки, попавшие внутрь замкнутого объема, ограниченного непроницаемой границей, должны быть исключены из счетного процесса;

– поиск несчетных областей проводится таким образом, чтобы минимизировать количество счетных ячеек расчетной области.

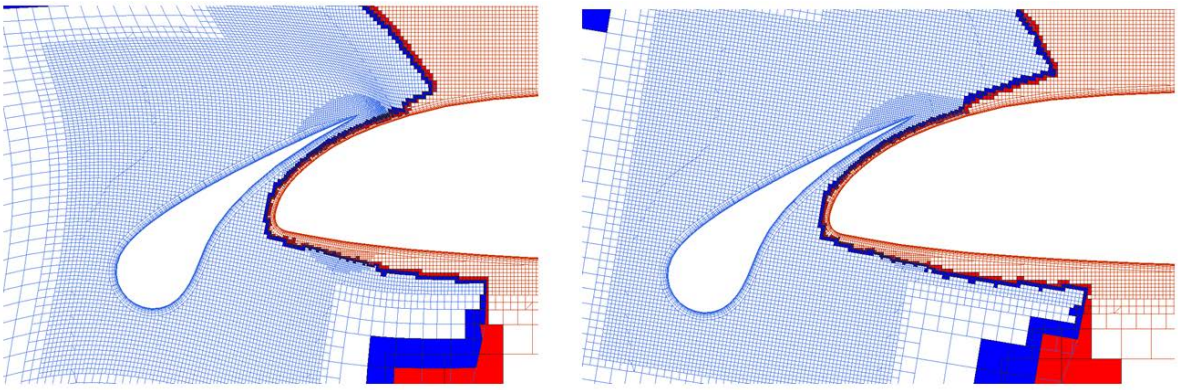
С развитием технологии перекрывающихся сеток проводились исследования, направленные на повышение качества получаемых результатов с помощью алгоритмов интерполяции высокой точности. Также предпринимались попытки построения консервативных схем расчета. С вычислительной точки зрения подходы, гарантирующие консервативность, весьма дорогостоящи, и применение их для произвольных нерегулярных сеток затруднительно. Следует отметить, что в современных инженерных методиках, как правило, для повышения точности используют качественные алгоритмы интерполяции. Поэтому при реализации технологии расчета на перекрывающихся сетках был выбран алгебраический подход сопряжения счетных областей. Данный выбор также был обусловлен тем, что предлагаемые подходы позволяют использовать базовые возможности пакета программ ЛОГОС, без каких-либо ограничений. В качестве алгоритма восстановления полей на гранях-акцепторах можно применять различные подходы: обратно взвешенная интерполяция, метод наименьших квадратов, трилинейная интерполяция и прочие методы.

4. Комплексный подход

Несмотря на достоинства каждой из рассмотренных методик, существуют задачи, которые не могут быть решены с использованием лишь одной из них. В качестве примера можно рассмотреть связанную задачу аэродинамики и прочности с подвижными элементами, также примером может служить задача обледенения на сетках с перекрытиями. В данном случае операция деформирования

применяется только на отдельно взятых сетках, которые в дальнейшем используются в методике расчета на сетках с перекрытиями.

На рис. 2 представлен интерполяционный шаблон модельной задачи с использованием методики деформации сетки и шаблон, полученный штатным методом.



а б

Рис. 2. Сравнение интерполяционных шаблонов, а – с использованием операции деформации, б – без использования операции деформации

Как видно из представленных рисунков, шаблоны интерполяции для двух подходов различны, когда положения профилей предкрылка и крыла идентичны.

5. Численные эксперименты

5.1. Расчет нестационарных АДХ профиля NACA 0012

Проводится численное моделирование нестационарного обтекания симметричного профиля NACA 0012 (рис. 3) турбулентным потоком сжимаемого вязкого газа при колебаниях профиля по углу атаки.

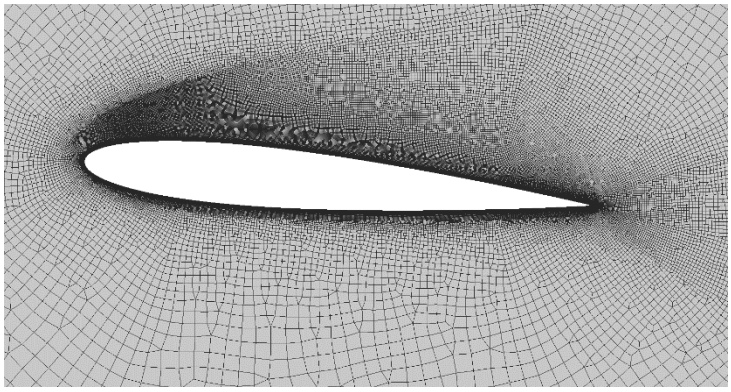


Рис. 3. Расчетная геометрия вблизи профиля

Расчетная сетка состоит из произвольных многогранников с призматическими слоями. Количество ячеек расчетной области 132199, количество узлов 263544.

Расчет обтекания профиля при его периодических колебаниях по углу атаки проводился с использованием методики деформирующихся сеток с сохранением топологии. Условия набегающего потока при моделировании дозвукового обтекания профиля соответствуют условиям аэродинамического эксперимента [4].

Скорость набегающего потока соответствовала числу Маха = 0.283 и числу Рейнольдса Re = 3.45e+6, при стандартном атмосферном давлении и температуре 25 ºС. Размер хорды профиля крыла равен 0.542 м.

Угловое движение профиля задавалось законом изменения его угла атаки за счет его вращения относительно оси, расположенной на средней линии на расстоянии 1/4 хорды от носика профиля.

В качестве количественной оценки в численном эксперименте определялись зависимости аэродинамических коэффициентов лобового сопротивления и подъемной силы от угла атаки в процессе колебаний профиля (рис. 4, 5).

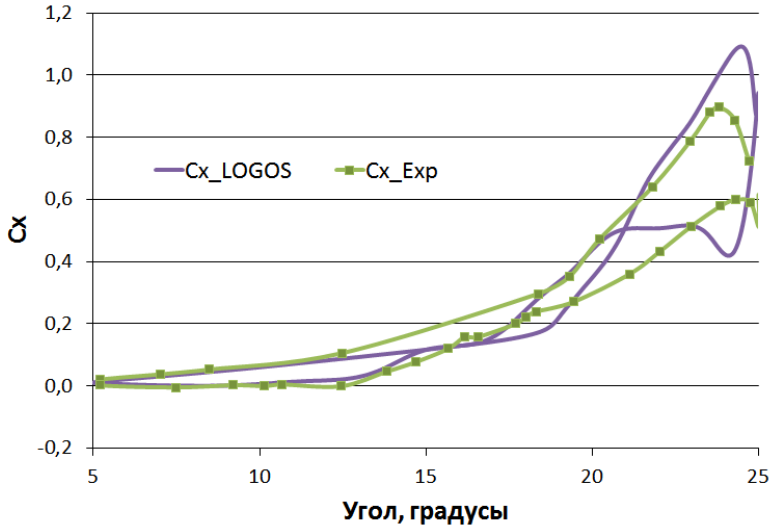


Рис. 4. Зависимость коэффициента лобового сопротивления от угла атаки

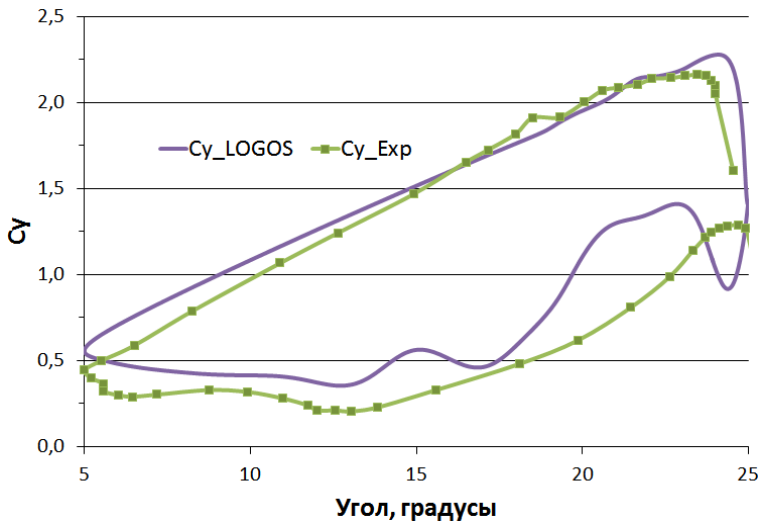


Рис. 5. Зависимость коэффициента подъемной силы от угла атаки

Как видно из представленных графиков, результаты моделирования в пакете программ ЛОГОС согласуются с экспериментом.

5.2. Задача отделения подвесного груза от крыла летательного аппарата

В качестве демонстрации работоспособности методики расчета на сетках с перекрытиями рассматривается задача нестационарного отделения подвесного груза в трансзвуковом потоке вязкого газа [5] (рис. 6).

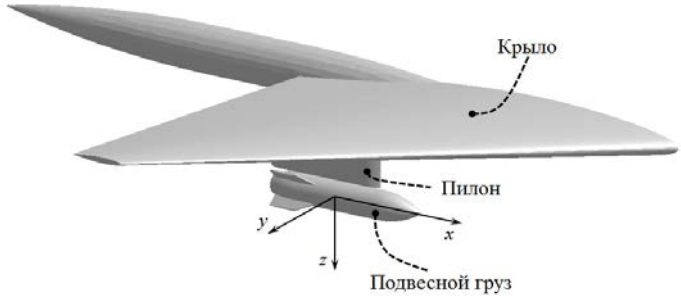


Рис. 6. Расчетная модель

Для решения задачи была сгенерирована сетка с перекрытиями, состоящая из 2 доменов. Общее количество ячеек ~2 млн. На груз на протяжении всего численного эксперимента задачи действовали аэродинамические нагрузки и сила тяжести; задавались масса, компоненты тензора инерции:

$$m = 907.1803 \ кг, \ M_{xx} = 27.12 \ м^2 \cdot кг, \ M_{yy} = M_{zz} = 488.1 \ м^2 \cdot кг.$$

Для моделирования принимался однородный поток вязкого газа с числом Маха $M_\infty = 0.95$ при значении числа Рейнольдса $Re = 1.8 \times 10^6$. Статическое давление и температура набегающего потока брались равным 36040 Па и 236 К соответственно. Шаг по времени равнялся 0.002 с. В задаче оценивается положение центра массы и изменение углов поворота груза с течением времени.

На рис. 7, 8 представлено сравнение результатов компьютерного моделирования и экспериментальных данных.

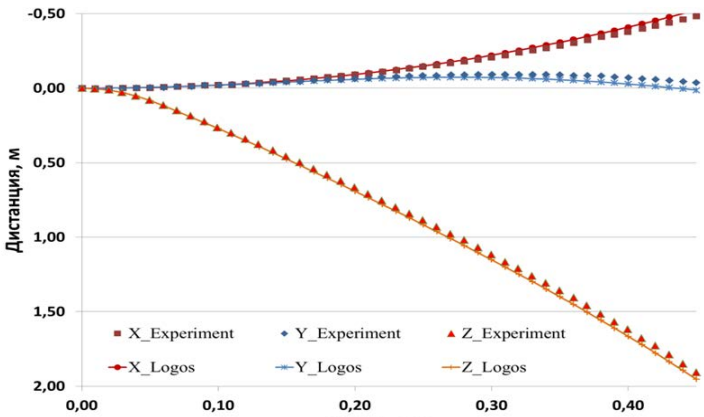


Рис. 7. Изменение координат центра масс в зависимости от времени

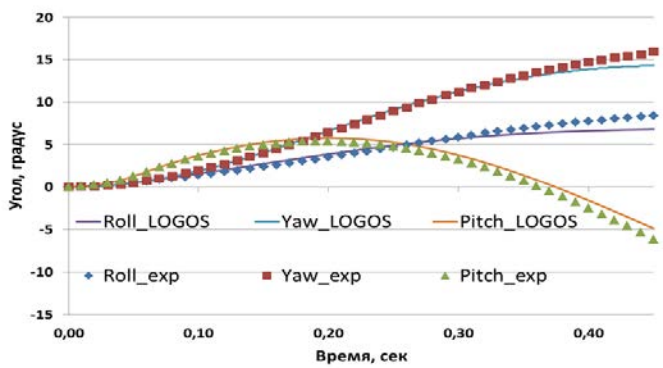


Рис. 8. Изменение углов курса, крена, тангажа в зависимости от времени

Как видно из представленных графиков, результаты расчета хорошо согласуются с экспериментом.

5.3. Задача аэроупругого равновесия крыла HIRENASD

Проводится расчет аэроупругих характеристик модели крыла коммерческого самолета в условиях трансзвукового потока при числе Рейнольдса 23.5e + 6 и аэродинамических нагрузках, соответствующих реальным условиям полета большого транспортного самолета.

Данный расчет проводится, используя технологию связанного счета. Течение вязкого сжимаемого газа моделируется с помощью ЛОГОС-Аэрогидромеханика, деформации крыла под действием набегающего потока рассчитываются модулем ЛОГОС-Прочность [6]. Между программами организовано взаимодействие – обмен граничными условиями:

– ЛОГОС-Аэрогидромеханика принимает от модуля ЛОГОС-Прочность перемещения узлов на поверхности;

– ЛОГОС-Прочность принимает от модуля ЛОГОС-Аэрогидромеханика поле давления и поле касательных напряжений.

Расчет потока проводится в стационарной постановке, а деформации рассчитываются статическим прочностным решателем. В качестве интерфейса связи между прочностным модулем и модулем аэродинамики используется поверхность крыла. Деформация сетки крыла происходит сразу после обмена данными между счетными модулями.

Для описания материала в расчетах использовалась модель упругого материала с параметрами: плотность $7920\ кг/м^3$, модуль упругости 1.813e11 Па, коэффициент Пуассона 0.33.

Выбранные аэродинамическая и прочностная геометрии и условия набегающего потока при моделировании трансзвукового обтекания соответствовали условиям аэродинамического эксперимента [7]:

– $M_\infty = 0.8$ – число Маха;

– $\alpha=2^o$ – угол атаки;

– $T_\infty=204$ K – температура набегающего потока;

– $P_\infty=201000$ Па – статическое давление набегающего потока;

– $V_\infty = 229.06$ м/с - скорость набегающего потока;

– $\mathrm{Re}=23.5e+6$ – число Рейнольдса.

На рис. 9 приведено распределение давления на поверхности деформированного крыла и фюзеляжа.

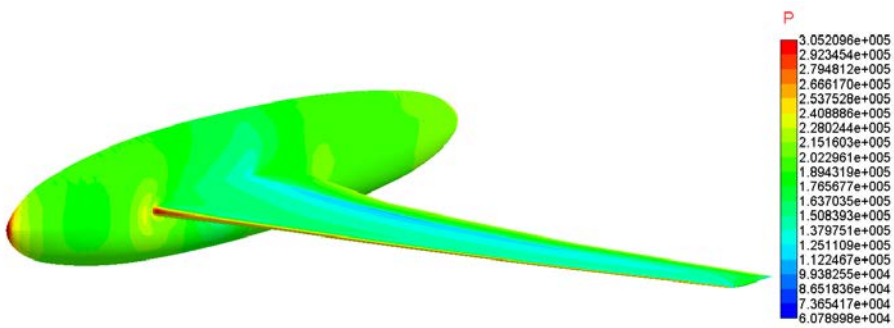


Рис. 9. Распределение поля давления

На рис. 10 приведено поле напряжений по Мизесу в прочностной части расчета для крыла в состоянии аэроупругого равновесия.

В численном эксперименте оценивались коэффициенты Cl и Cd, коэффициент Cp в нескольких сечениях крыла, а также деформации вдоль размаха крыла в состоянии аэроупругого равновесия. На рис. 11 представлено сравнение полученного коэффициента давления с экспериментальными данными для угла атаки 2º.

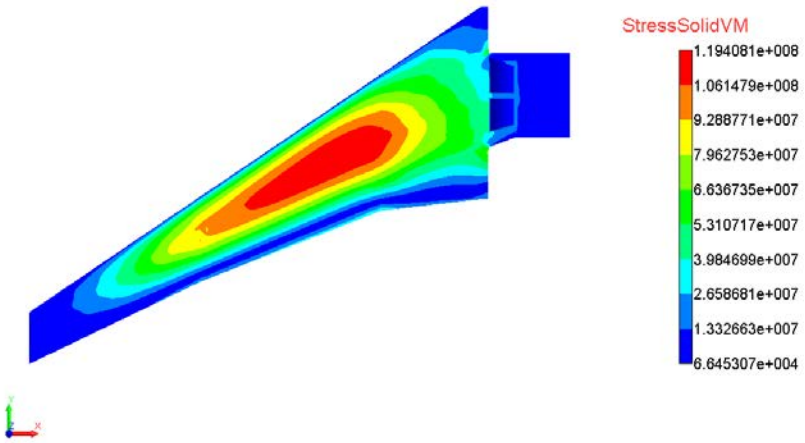


Рис. 10. Поле напряжений по Мизесу

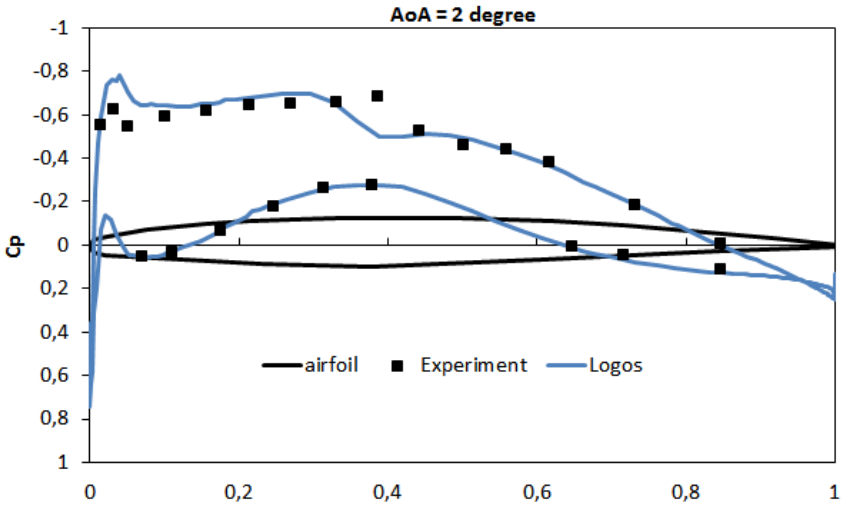


Рис. 11. График Ср по сечению

Как видно из представленного графика, результаты моделирования хорошо согласуются с экспериментальными данными.

В табл. 1 представлено сравнение с экспериментальными данными интегральных характеристик, а также максимального смещения крыла, полученных в численном эксперименте.

Таблица 1

Сравнение интегральных характеристик

	Эксперимент	ЛОГОС
Cl	0.358	0.361
Cd	0.0115	0.0113
dS (м)	0.0306	0.0319

Представленные результаты показывают хорошее согласие с экспериментом.

5.4. Расчет флаттера крыла AGARD 445.6

Решается задача о деформировании крыла AGARD 445.6 в потоке вязкого газа [8]–[10]. Течение вязкого сжимаемого газа моделируется с помощью ЛОГОС-Аэрогидромеханика, деформации крыла под действием набегающего потока рассчитываются модулем ЛОГОС-Прочность. Связанный расчет проводится в нестационарной постановке в модуле ЛОГОС-Аэрогидромеханика и явным динамическим решателем ЛОГОС-Прочность. В качестве интерфейса связи между прочностным модулем и модулем аэродинамики используется поверхность крыла. Деформация сетки крыла происходит сразу после обмена данными между счетными модулями.

Геометрия аэродинамической модели представляет собой крыло, представленное на рис. 12.

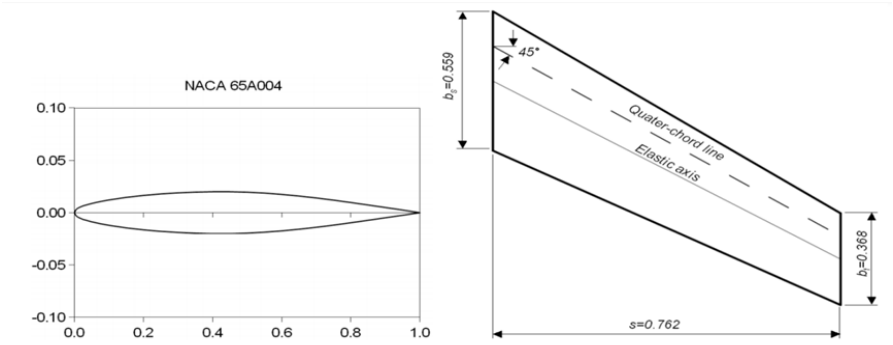


Рис. 12. Профиль и проекция крыла AGARD 445.6

В зависимости от заданных условий набегающего потока колебания крыла могут либо затухать, либо усиливаться. Последняя ситуация расценивается как наличие флаттера. Для локализации границы флаттера необходимо проанализировать серию подобных расчетов. Одним из методов определения границы флаттера является метод варьирования опорного давления при постоянной скорости звука.

Для описания поведения материала в расчетах использовалась модель ортотропного материала. Направления материала заданы вдоль размаха крыла и вдоль хорды. Плотность материала $\rho = 381.98$ кг/м^3, модуль упругости в продольном направлении $E_1 = 3.15$ e+9 Па, модуль упругости в поперечном направлении $E_2 = 0.42$ e+9 Па, коэффициент Пуассона в продольном направлении и в поперечном направлении в плоскости крыла ν=0.041, в поперечном направлении перпендикулярно плоскости крыла $\nu = 0.31$, модуль сдвига в каждой плоскости $G = 0.412$ e+9 Па.

Локализация границы флаттера крыла проводилась для чисел Маха 0.499 – 1.141 при нулевом угле атаки. Газодинамические параметры подбирались таким образом, чтобы сохранить скорость звука при варьировании давления.

Для количественной характеристики флаттера используются такие величины, как скоростной индекс флаттера (Flutter Speed Index, FSI) и отношение частот (Frequency Ration, FR). На рис. 13, 14 представлены графики коэффициентов в зависимости от числа Маха.

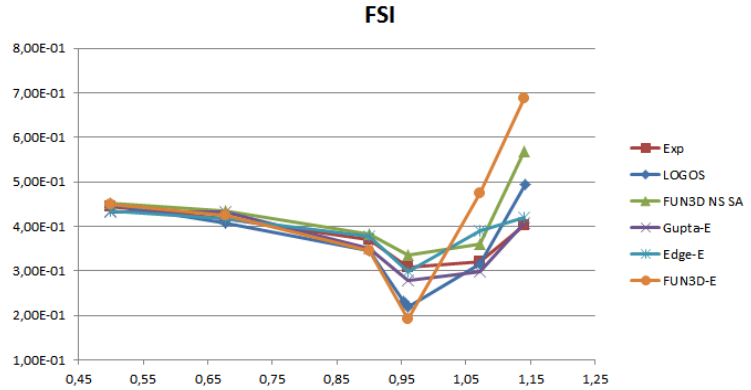


Рис. 13. Сравнение зависимостей коэффициента FSI от числа Маха

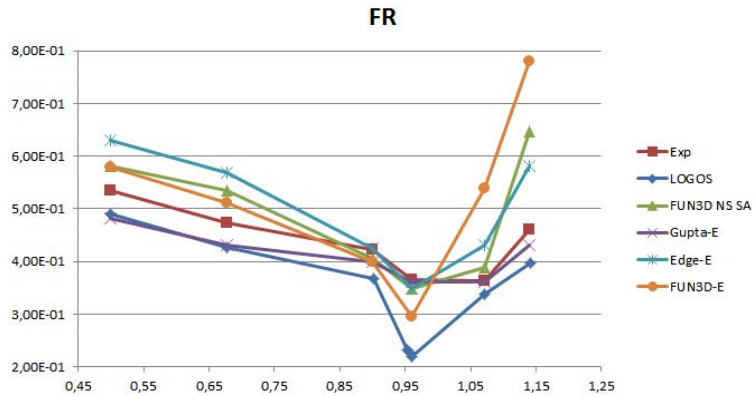
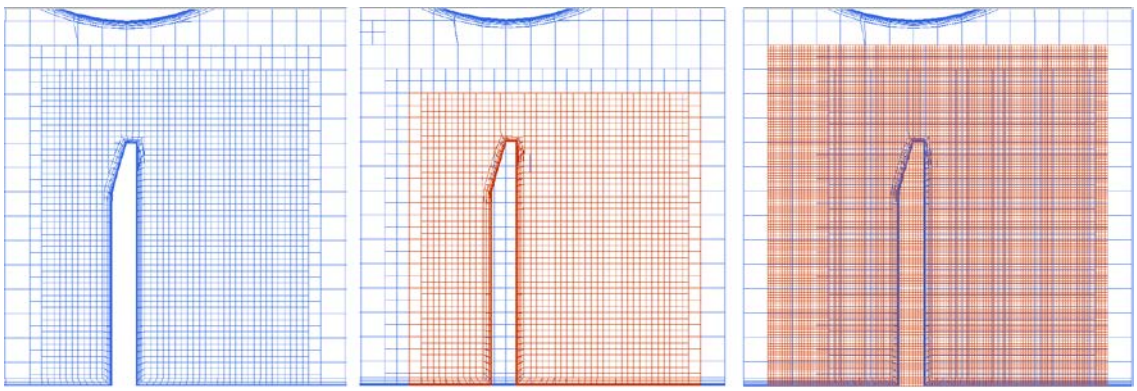


Рис. 14. Сравнение зависимостей коэффициента FR от числа Маха

Как видно из представленных графиков, результаты моделирования хорошо согласуются с экспериментальными данными.

5.5. Деформация клапана в канале

Решается связанная задача о деформировании клапана под действием потока газа в канале. В качестве интерфейса связи между прочностным модулем и модулем аэродинамики используется поверхность клапана. Деформация сетки клапана происходит сразу после обмена данными между счетными модулями. Моделирование проводится с использованием 3 сеток: сетка без перекрытий, эквивалентная сетка с перекрытиями и подробная сетка с перекрытиями (рис. 15).



а б в

Рис. 15. Расчетные сетки вблизи клапана, а – сетка без перекрытий, б – эквивалентная сетка с перекрытиями, в – подробная сетка с перекрытиями

Для моделирования обтекания клапана взяты параметры сжимаемого газа при следующих условиях:

– $P_{Tot}^{in} = 102325$ Па – полное давление на входе;

– $P_{Tot}^{out} = 101325$ Па – полное давление на выходе;

– $T_{Tot}^{in} = 288$ K – полная температура на входе;

– $T_{Stat}^{out} = 288$ K – статическая температура на выходе.

Для клапана используется модель упругопластического материала с кинематическим упрочнением. Плотность материала $\rho = 3500$ кг/м^3, модуль Юнга $E = 5e+7$ Па, коэффициент Пуассона $\nu = 0.3$, предел текучести $SigY = 1e+10$ Па. Клапан жестко закреплен в своем основании.

На рис. 16 показано перемещение контрольной точки на клапане с течением времени.

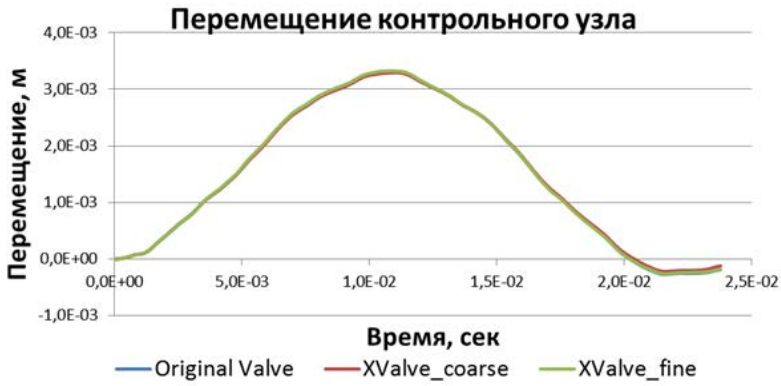


Рис. 16. Зависимость перемещения контрольного узла от времени

Как видно из представленных графиков, результаты моделирования с использованием сеток с перекрытиями хорошо согласуются с результатами, полученными на штатной сетке.

5.6. Образование инея на профиле NACA0012

Проводится численное моделирование процесса образования льда в виде инея на профиле NACA 0012 [11] в результате обтекания турбулентным потоком сжимаемого вязкого газа.

Для решения задачи используется сетка с перекрытиями, содержащая 2 несвязанных региона. Количество ячеек ~50000. На рис. 17 представлен фрагмент расчетной сетки.

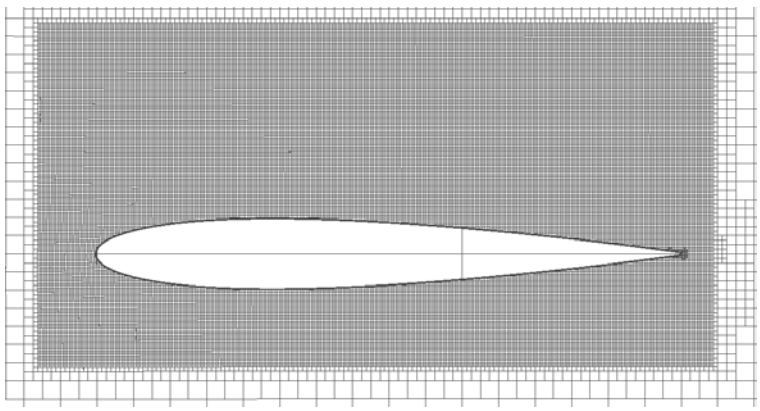


Рис. 17. Сетка с перекрытиями

Условия набегающего потока при моделировании обтекания профиля соответствуют условиям аэродинамического эксперимента [11]:

– M_∞ = 0.184893 – число Маха;

– α=4º – угол атаки;

– T_∞ =245.35 K – температура набегающего потока;

– P_∞ =95610 Па – статическое давление набегающего потока;

– V_∞ = 229.06 м/с - скорость набегающего потока;

– α = 1.303178e-6 – объемная доля жидкой фазы;

– D_{part} = 2.0e-5 м – осредненный диаметр капель;

– длительность моделирования 480 секунд.

Для описания турбулентных характеристик используются модель SA.

На рис. 18 представлено сравнение толщины льда, полученной в ходе численного расчета на сетке с перекрытиями, с экспериментальными данными и данными, полученными в ходе расчета на сетке без перекрытий.

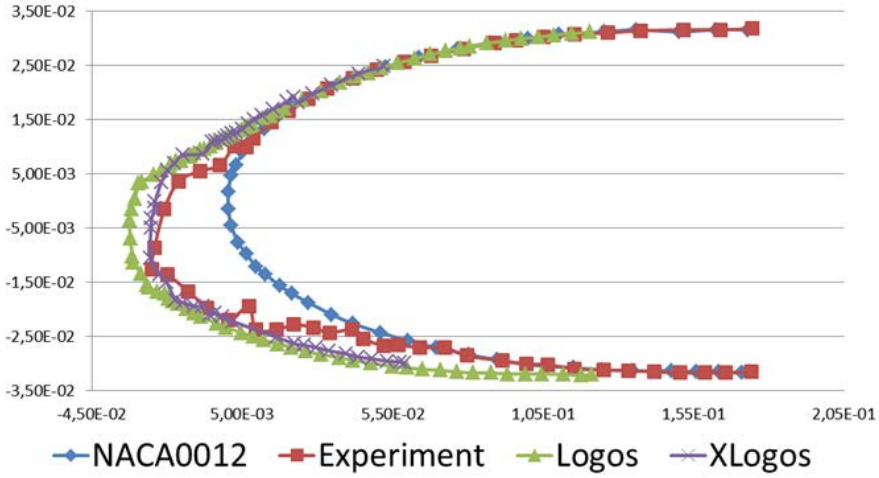


Рис. 18. Сравнение ледяного нароста

Как видно из представленного графика, форма ледяного нароста согласуется с экспериментальными данными.

Заключение

Работа посвящена вопросам решения задач численной газовой динамики с изменяющимися во времени граничными условиями. Приводятся адаптированные численные схемы решения данного класса задач на основе конечно-объемной методики. Рассмотрены подходы к решению задач аэродинамики с подвижными элементами конструкции, реализованные в пакете программ ЛОГОС.

В работе приводятся основные выражения алгоритма деформации сетки с сохранением топологии. Обсуждается сфера применимости данной методики и определен класс задач. Для более существенных деформаций рассматривается методика расчета на сетках с перекрытиями. Описываются основные положения методики расчета на сетках с перекрытиями.

В качестве иллюстрации работоспособности внедренных алгоритмов и численных схем рассмотрены характерные задачи авиационной промышленности.

Литература

1. Lesoinne M., Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations // Computer methods in applied mechanics and engineering 134 (1996) 71–90.

2. Luke E., Collins E., Blades E. A fast mesh deformation method using explicit interpolation, Journal of Computational Physics 231 (2012) 586–601.

3. Дерюгин Ю. Н., Саразов А. В., Жучков Р. Н. Особенности построения методики расчета на сетках типа «Химера» для неструктурированных сеток // Математическое моделирование, 2017, т. 29, №2, С. 106–118.

4. Shrewsbury G., Sancar L. Dynamic Stall of Circulation Control Airfoils // AIAA Paper. 1990. № 90-0573. 9 p.

5. Fox J. H. Generic Wing, Pylon, and Moving Finned Store, Verification and Validation Data for Computational Unsteady Aerodynamics // RTO-TR-26, October 2000, St. Joseph Ottawa/Hill, Canada.

6. Авдеев П. А, Александрова О. Л., Артемова Е. О., Афанасьев В. А., Барабанов Р. А., Борляев В. В., Дьянов Д. Ю., Дяченко И. А., Казанцев А. В., Корсакова Е. И., Косарим С. С., Медведкина М. В., Морозов С. В., Наумов А. О., Присташ М. М., Разваров Д. И., Резвова Т. В., Спиридонов В. Ф., Стародубов С. В., Тагирова И. Ю., Филимонкин Е. А., Циберев К. В., Челаков А. А., Шувалова Е. В. Обзор возможностей моделирования задач прочности с использованием пакета программ ЛОГОС // XV Международный семинар «Супервычисления и математическое моделирование»: Сборник тезисов. Саров, 2014.

7. Reimer L., Boucke A., Ballmann J., Behr M. Computational Analysis of High Reynolds Number Aero-Structural Dynamics (HIRENASD) Experiments.//IFASD-2009-130, International Forum on Aeroelasticity and Structural Dynamics. – Seattle, WA, June 21–25, 2009.

8. AGARD STANDARD AEROELASTIC CONFIGURATIONS FOR DYNAMIC RESPONSE I-WING 445.6: AGARD Report № 765/ Interdisciplinary Research Office NASA Langley Research Center Hampton; performer: Yates Carson E., Jr., -VA 23665-5225, USA, 1988. –P.1-83.

9. Ballmann J., Dafnis A., Korsch H., Buxel C., Reimerdes H.-G., Brakhage K.-H., Olivier H., Braun C., Baars A., Boucke A. Experimental Analysis of High Reynolds Number Aero-Structural Dynamics in ETW// 46th AIAA Aerospace Sciences Meeting and Exhibit, Paper № 841.- Reno, January 7–10, 2008.–PP.15.

10. Dynamic Aeroelastic Simulation of the AGARD 445.6 Wing using Edge: Tech. Rep. FOI–R–2259–SE/ FOI; performer: Pahlavanloo P. –Stockholm, April 2007.

11. Ozgen S., Canibek M. Ice accretion simulation on multi-element airfoils using extended Messinger model. // Heat Mass Transfer. 2008, Springer-Verlag.

SIMULATION OF AERODYNAMICS PROBLEMS WITH MOVING BOUNDARIES USING THE “LOGOS” SOFTWARE PACKAGE

A. V. Sarazov, A. S. Kozelkov, D. K. Zelenskiy, R. N. Zhuchkov, T. V. Rezvova

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper considers the issues of numerically simulating the aerodynamics problems with moving boundaries. Main approaches to the simulation of flow around moving objects using the Logos software package are presented. The procedure of simulating on deformable meshes with preservation of the communication topology and advantages and disadvantages of such approach are discussed. Besides, the simulation procedure for aerodynamics problems using meshes with overlaps is presented, its main stages are described, key terms and calculation rules are introduced. The paper discusses a class of problems with examples of the necessary employment in common of the approaches above. To demonstrate the efficiency and applicability of the implemented simulation techniques, solutions are given for typical aircraft industry problems, for which experimental data is available.

Key words: the LOGOS software package, the procedure of simulations on meshes with overlaps, deformable meshes with preservation of communication topology, icing, NACA0012, FSI, AGARD 445.6, HIRENASD, AGARD 23.

УДК 519.6
DOI 10.53403/9785951505071_2022_406

РЕЗУЛЬТАТЫ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ БРОНЕБОЙНОГО ДЕЙСТВИЯ УДАРНИКА НА ПРЕГРАДУ ПО МЕТОДИКЕ ТИМ

А. В. Свидинский, С. С. Соколов, И. Г. Новиков, А. В. Самодолов, Т. Н. Половникова

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В данной работе представлены результаты численного моделирования бронебойного действия ударника на преграду по методике ТИМ. Показано сравнение полученных результатов трехмерных расчетов, проведенных с использованием смешанных ячеек и без их использования с результатами двумерных расчетов без использования смешанных ячеек и с имеющимися экспериментальными данными.

Ключевые слова: трехмерные и двумерные расчеты, численное моделирование, методика тим, ударник, преграда, смешанные ячейки.

Полномасштабные опыты для экспериментальной проверки бронепробития при варьировании всех параметров (скорость, материал преграды, толщина преграды, угол подхода и угол атаки ударника) делают эту работу чрезвычайно затратной. Поэтому для обоснования выбора того или иного ударника и скорости подхода к преграде и для предварительной оценки величины бронепробития целесообразно использование численного моделирования. В ряде случаев расчет таких задач возможно свести к двумерной постановке, однако рассматривая углы подхода ударника к преграде отличные от нуля, задача становится трехмерной.

Для численного моделирования используется лагранжевая методика ТИМ. Численное моделирование на подвижной сетке с явно выделенными лагранжевыми границами между веществами сопряжено с рядом трудностей: сетка может искажаться в трех направлениях, возникают вихревые и струйные движения. Все это может приводить к существенному искажению сетки и как следствие этого – к невозможности дальнейшего численного моделирования на ней. Необходимо останавливать расчет для перестроения сетки вручную. Подход с использованием смешанных ячеек значительно повышает автоматизацию проведения трехмерных расчетов и позволяет применять аппарат перестроения сетки без остановки расчета. Наиболее актуально данный вопрос встает при проведении трехмерных расчетов конструкций, связанных со сложными вихревыми и струйными течениями. Появление смешанных ячеек возможно только на границах раздела веществ после перестроения сетки, сам расчет между этапами пересчета величин проводится в лагранжевом приближении без допущения перетекания веществ между ячейками. На данный момент в методике ТИМ реализован аппарат поддержания качества сетки, который основан на автоматическом выделении фрагмента, в котором сетка подвержена сильным деформациям, с последующим изменением сетки в этом фрагменте и переинтерполяции величин, реализован аппарат глобального пересчета величин на определенные этапы по достижению критериев качества сетки, вырабатываемых в блоке локальной перестройки сетки в автоматическом режиме.

В данной работе представлено применение подхода с использованием смешанных ячеек и автоматического перестроения сетки в процессе счета на примере задачи о пробивании преграды. Приведено сравнение результатов расчетов с использованием смешанных ячеек и без их использования. Воздействие ударника на преграду характеризуется углом его подлета к преграде, поэтому численное моделирование в ряде случаев представляет строго трехмерную задачу. Численное моделирование

проведено для бронебойного действия цилиндрического ударника диаметром 1 см и длиной 26 см из тяжелого сплава (ВНЖ–сплав вольфрама, никеля и железа) при скорости проникания от 0,6 до 1,3 км/с в преграду из брони средней твердости по лагранжевой методике ТИМ [1, 2]. Преграда представляла собой незакрепленную квадратную плиту со стороной 40 см, расположенную таким образом, чтобы вектор скорости ударника из центра масс проходил через центр плиты.

В начале рассмотрим сравнение расчетов в двумерной и трехмерной постановке на примере расчета лобового удара на полное пробитие со смешанными ячейками и без. Далее рассмотрим существенно трехмерную задачу.

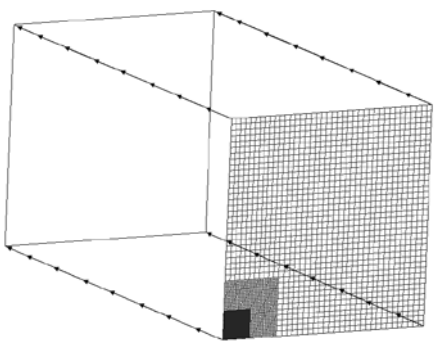
Рассматривается вариант расчета пробития ударником преграды с углом подхода 0° и углом атаки 0° со скоростью 0,6 км/с, толщина преграды составляла 2,5 см. Начальная геометрия представлена на рис. 1.



Рис. 1. Начальная геометрия

Расчеты выполнялись в двумерной и трехмерной геометрии. В трехмерном случае были рассмотрены несколько вариантов построения счетной сетки, учитывающие особенности проникания ударника в преграду с измельчением сетки в области пробития и в самом ударнике. Изначально была построена двумерная геометрия [3, 4] и затем путем ее трансляции получена трехмерная.

Первый вариант построения трехмерной геометрии



Второй вариант построения трехмерной геометрии

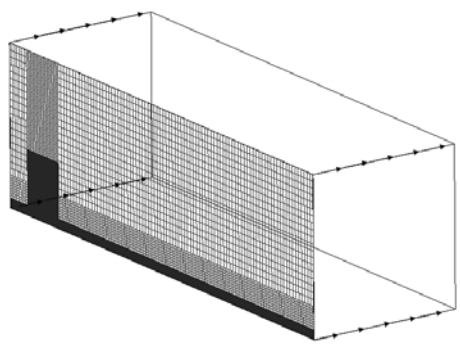
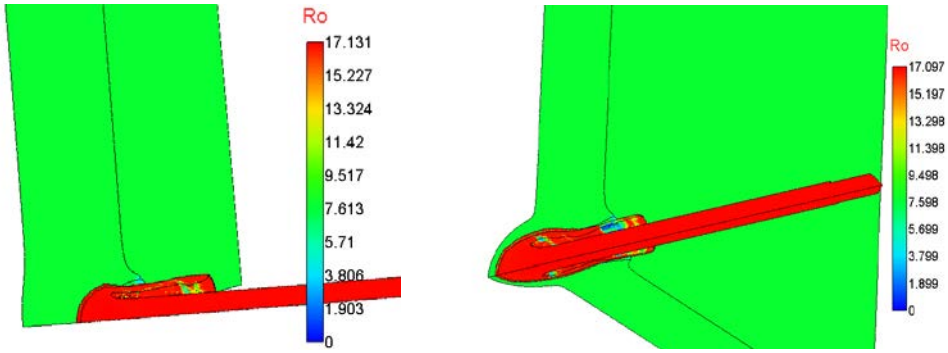


Рис. 2. Трехмерная геометрия

Количество ячеек в задаче составляло около 5,8 млн в первом варианте и 19,6 млн во втором. Характерный размер ячеек в ударнике – 0,03 см. Пространство вокруг ударника и преграды заполнялось воздухом.

Распределение плотностей в расчете, проведенном без использования смешанных ячеек, на моменты времени 100 и 300 мкс представлены на рис. 3.

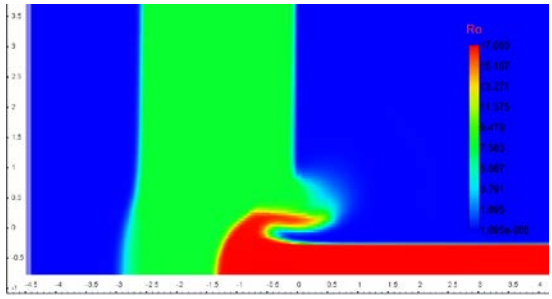


Момент времени 100 мкс Момент времени 300 мкс

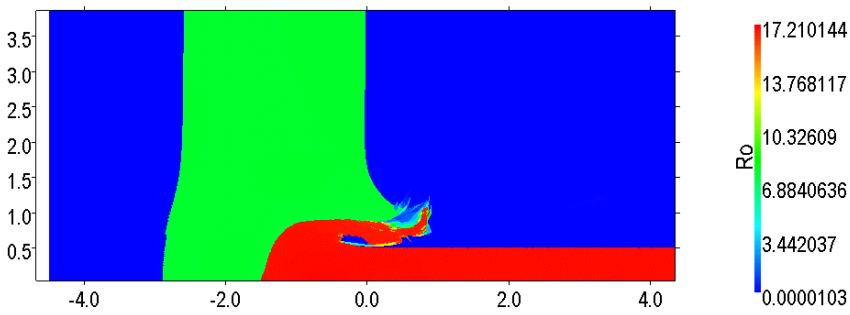
Рис. 3. Трехмерный расчет без смешанных ячеек

Расчет данной задачи характерен тем, что ударник, имея начальную скорость 0,6 км/с, пробивает преграду толщиной 2,5 см. Расчет трехмерной задачи сопровождался перестроением счетной сетки. Каждое перестроение связано с остановкой счета, построением новой сетки, переинтерполяцией величин. С внедрением возможности использования смешанных ячеек во время счета и автоматической переинтерполяцией величин с последующим выходом на счет было решено опробовать данный механизм и применить его для численного моделирования подобных задач. Отработка такого подхода производилась при расчетах как на двумерной, так и на трехмерной геометрии. Для этого было построено две начальные сетки (описанные выше). Результаты расчетов приведены ниже.

Двумерные расчеты проведены на сетке аналогичной трехмерной без смешанных ячеек. На рис. 4 – 6 приведено распределение поля давления (ГПа) и поля плотности (г/см^3) в расчетах.

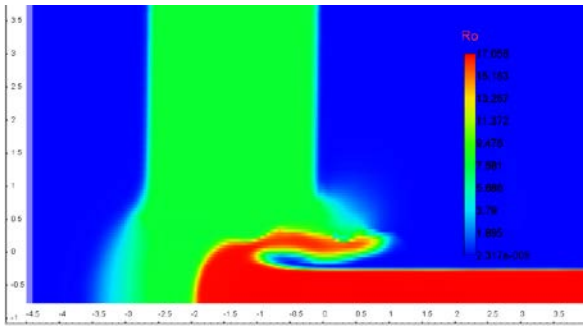


3D-расчет со смешанными ячейками

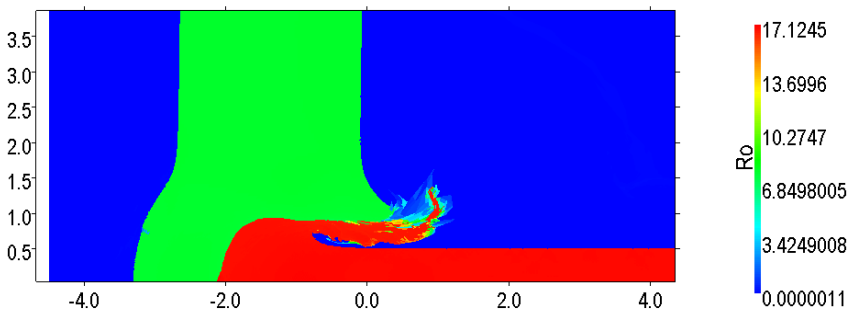


2D-расчет без смешанных ячеек

Рис. 4. Распределение поля плотности на момент времени 100 мкс, г/см^3

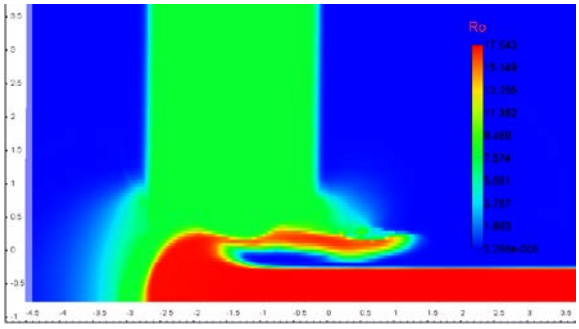


3D-расчет со смешанными ячейками

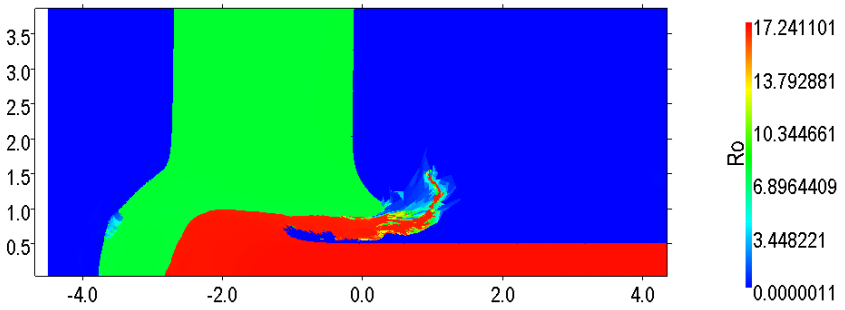


2D-расчет без смешанных ячеек

Рис. 5. Распределение поля плотности на момент времени 150 мкс, г/см^3



3D-расчет со смешанными ячейками



2D-расчет без смешанных ячеек

Рис. 6. Распределение поля плотности на момент времени 200 мкс, г/см^3

Сравнение результатов двумерного и трехмерного расчетов приведено на рис. 7. На данном рисунке граница веществ двумерного расчета нанесена на распределение плотности в трехмерном расчете.

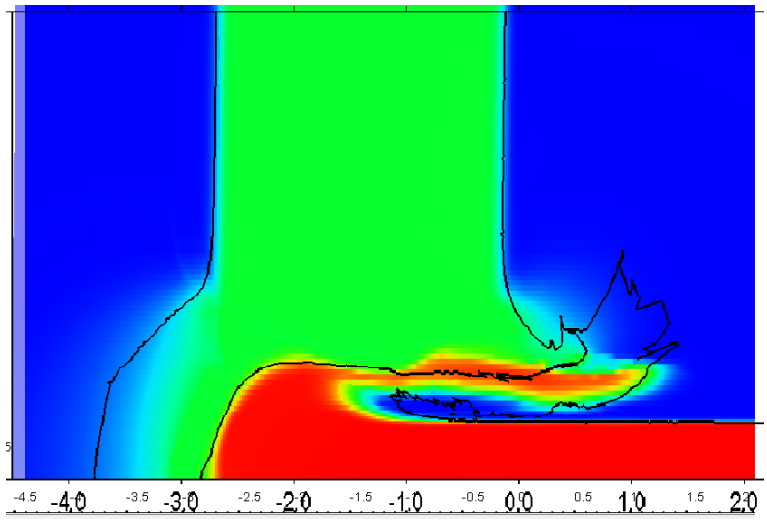


Рис. 7. Сравнение двумерного и трехмерного расчета. Распределение поля плотности на момент времени 200 мкс, г/см^3

Далее приведем график проникновения ударника в преграду от времени.

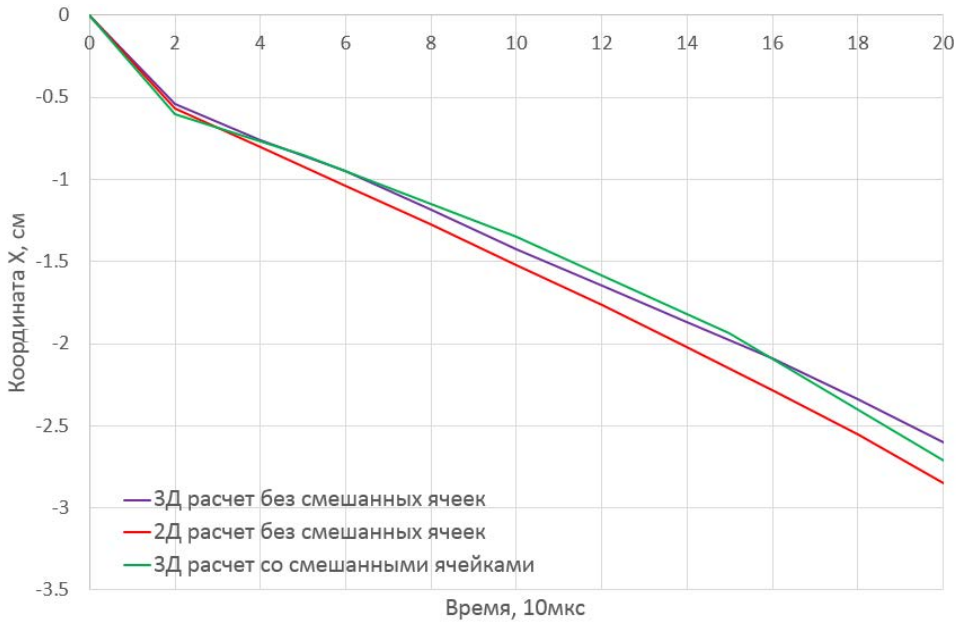


Рис. 8. График координаты X носика ударника от времени

На момент времени 200 мкс отличия в глубине проникания ударника в преграду между трехмерными расчетами составило 4,5 %. Отличие на тот же момент времени трехмерного расчета со смешанными ячейками и аналогичного по сетке двумерного расчета составило 5 %. Трехмерный расчет *без смешанных* ячеек проведен с ручным вмешательством пользователя, на некоторые моменты времени была перестроена вся сетка в ударнике либо в преграде. Трехмерный расчет *со смешанными* ячейками проведен полностью в автоматическом режиме. Перестроение сетки происходило через определенные моменты времени, исходя из критериев локальной перестройки сетки. Двумерный расчет проведен так же в автоматическом режиме без смешанных ячеек с применением поддержания качества счетной сетки.

На рис. 9 приведен фрагмент трехмерной задачи вблизи проникания ударника в преграду.

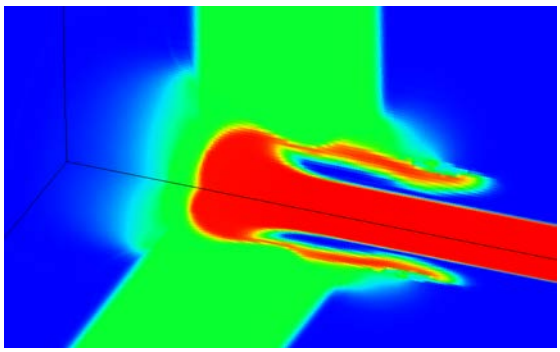
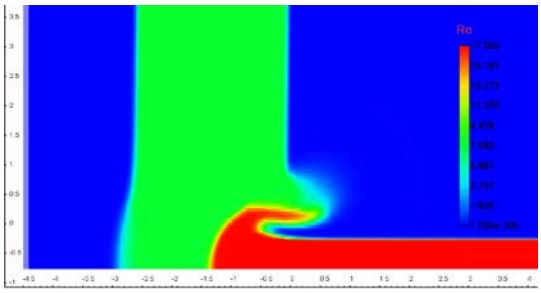


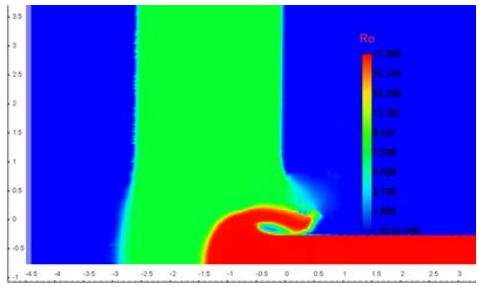
Рис. 9. Распределение поля плотности в трехмерном расчете
на момент времени 200 мкс, г/см^3

Расчет трехмерной задачи с применением переинтерполяции всей счетной области проведен полностью в автоматическом режиме без вмешательства пользователя.

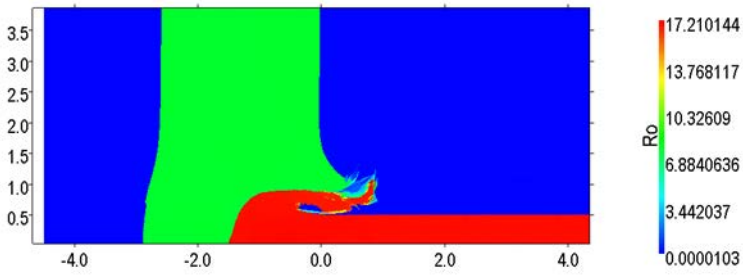
Результаты расчета второго варианта геометрии, в котором более детально прописана преграда в области проникания ударника приведен на рис. 10.



3D-расчет (первый вариант сетки)



3D-расчет (второй вариант сетки)



2D-расчет без смешанных ячеек

Рис. 10. Распределение поля плотности на момент времени 100 мкс, г/см^3

Отличие глубины проникания в трехмерных расчетах составило 3,5 %. И в двумерных, и в трехмерных расчетах получено полное пробитие преграды.

Далее рассмотрим результаты численного моделирования трехмерной постановки задачи, когда ударник подлетает под углом к преграде. Проведены расчеты в однообластной постановке с учетом смешанных ячеек и расчеты в многообластной постановке с учетом контактного взаимодействия.

Начальная геометрия строилась в пакете программ ЛОГОС-ПреПост [5], исходя из особенностей геометрии с необходимыми зонами измельчения. На рис. 11 и 12 представлена начальная геометрия для многообластной постановки задачи и для однообластной, соответственно.

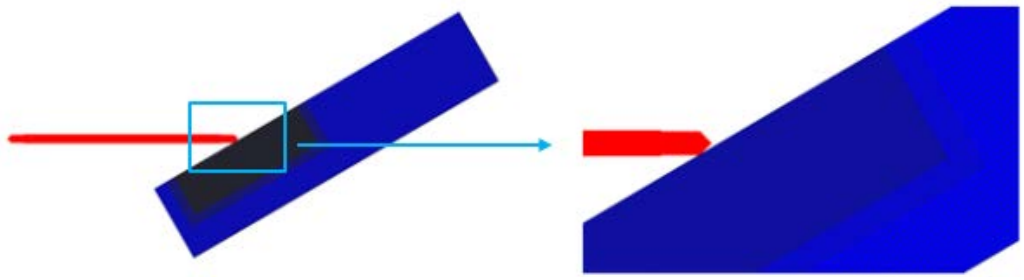


Рис. 11. Начальная геометрия для многообластной постановки задачи

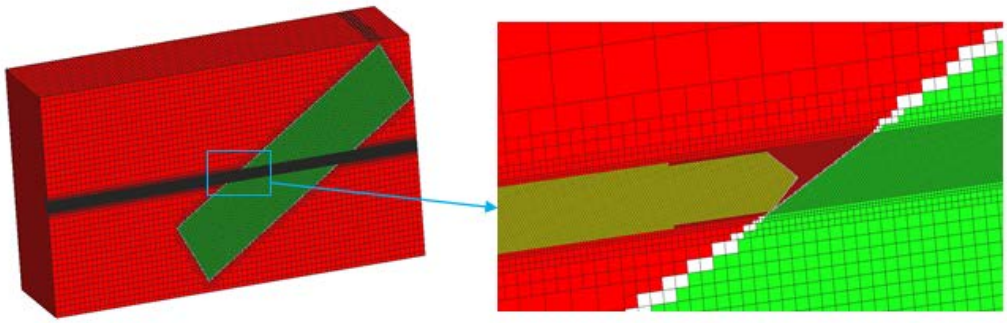


Рис. 12. Начальная геометрия для однообластной постановки задачи

В процессе счета для расчетов с использованием смешанных ячеек при сильной деформации сетки, когда ее качество не удовлетворяет определенным критериям методики локальной перестройки сетки выполняется переинтерполяция величин всей счетной области на новую сетку без остановки счета.

Для расчета на пробитие преграды с углом атаки ударника 0° угла подхода 60°, толщиной преграды 9 см и скоростью ударника 1,3 км/с график зависимости глубины проникания носика ударника от времени для трехмерных расчетов (со смешанными ячейками и без) представлен на рис. 13. На момент времени 130 мкс распределение поля плотностей для расчетов со смешанными ячейками и без представлено на рис. 14 и 15, соответственно. Получено полное пробитие преграды при проведении расчетов и со смешанными ячейками и без.

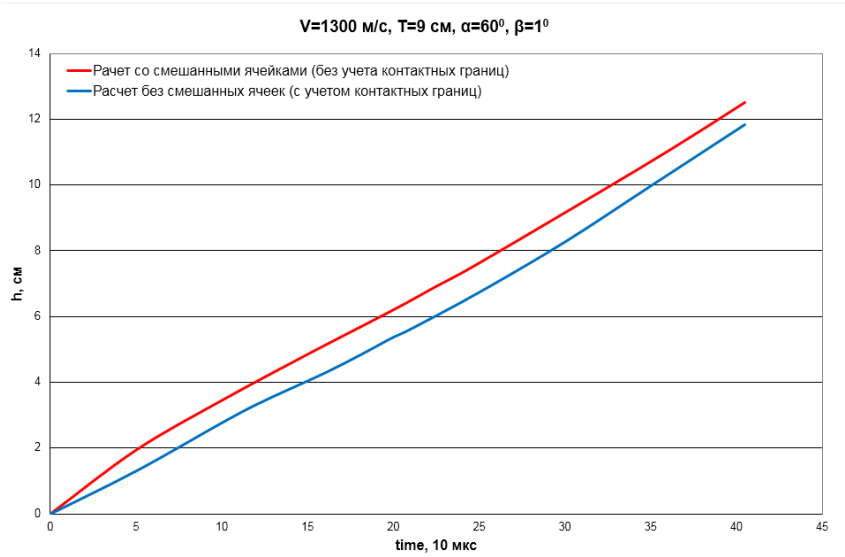


Рис. 13. График зависимости глубины проникания носика ударника от времени, см

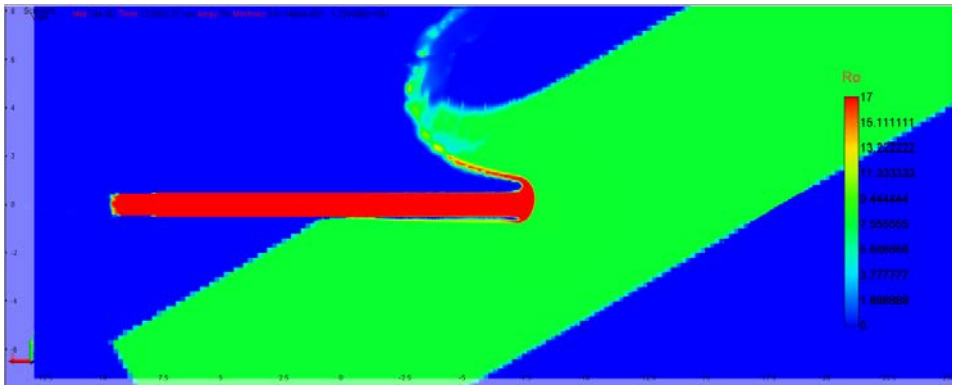


Рис. 14. Распределение поля плотностей для трехмерного расчета со смешанными ячейками на момент времени 130 мкс, г/см^3

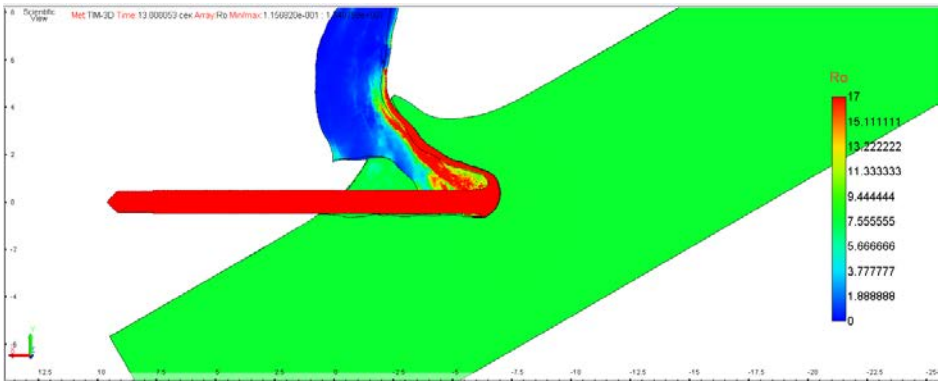


Рис. 15. Распределение поля плотностей для трехмерного расчета без смешанных ячеек на момент времени 130 мкс, г/см^3

Для расчета с углом атаки ударника 1° угла подхода 0°, толщиной преграды 20 см и скоростью ударника 1,3 км/с график зависимости глубины проникания носика ударника от времени для трехмерных расчетов (со смешанными ячейками и без) представлено на рис. 16. На момент времени 250 мкс распределения поля плотностей для расчетов со смешанными ячейками и без представлены на рис. 17 и 18, соответсвенно.

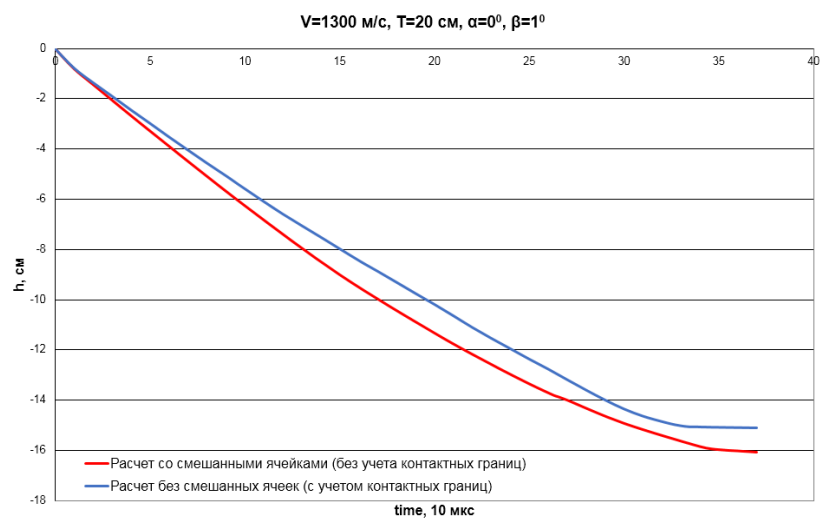


Рис. 16. График зависимости глубины проникания носика ударника от времени, см

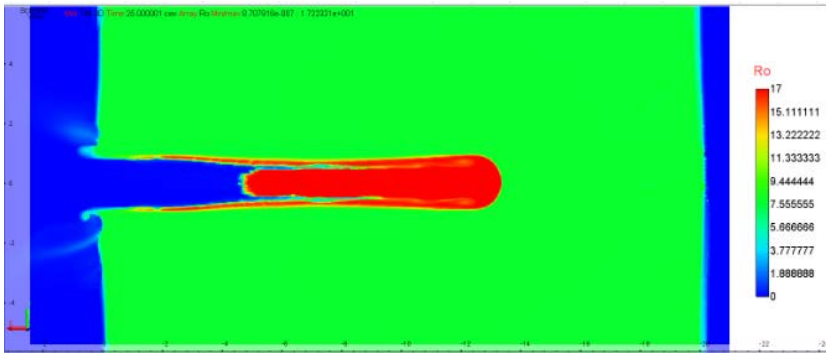


Рис. 17. Распределение поля плотностей для трехмерного расчета со смешанными ячейками на момент времени 250 мкс, г/см^3

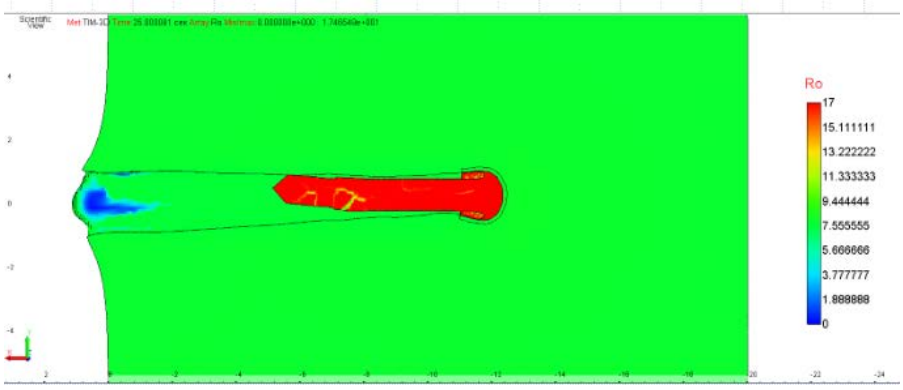


Рис. 18. Распределение поля плотностей для трехмерного расчета без смешанных ячеек на момент времени 250 мкс, г/см^3

Разница в глубине бронепробития в расчетах, проведенных в многообластной и однообластной постановках, составила примерно 6 % за счет того, что расчеты в многообластной постановке проводились с учетом выделенных контактных границ между веществами, а в однообластной постановке без учета.

Сравнение глубин бронепробития в зависимости от скорости встречи ударника с преградой, полученные в расчетах, с экспериментальными данными [6, 7] представлены на рис. 19. На данном рисунке λ обозначает удлинение ударника (отношение длины ударника к его диаметру). Разница по глубине бронепробития в расчетах, проведенных со смешанными ячейками и без, составляет 6 %.

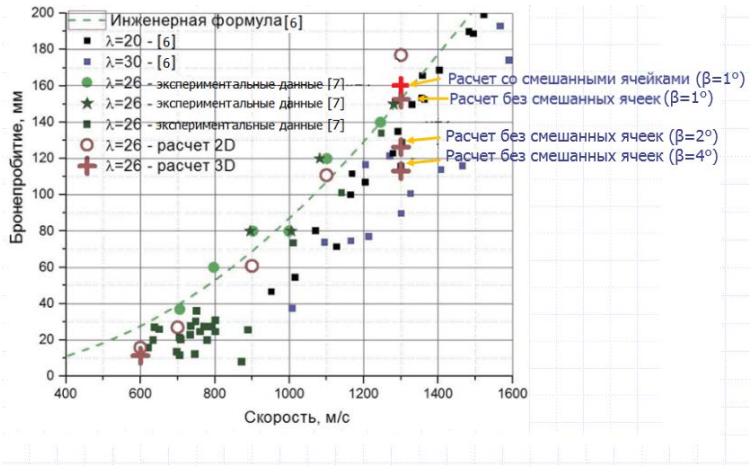


Рис. 19. График зависимости бронепробития от скорости ударника

Выводы

Существует два подхода к проведению трехмерного численного моделирования.

Во-первых, расчет во многообластной постановке в лагранжевом приближении с учетом контактного взаимодействия между областями с учетом скольжения. Для поддержания приемлемой для счета сетки на протяжении всего расчета задачи применяется перестроение сетки в пакете программ ЛОГОС-ПреПост с сохранением характерных размеров ячеек в веществах с последующей переинтерполяцией величин со старой сетки на новую во время остановки расчета. В этом подходе имеется зависимость от качества построенной сетки и частоты перестроения, т. е. зависимость от «человеческого фактора».

Во-вторых, расчет с использованием смешанных ячеек, с частичным или полным отказом от лагранжевых контактных границ между веществами. Здесь возможно два варианта: расчет во многообластной постановке с использованием смешанных ячеек в отдельно взятой математической области (при сильных деформациях необходима остановка расчета для проведения перестроения сетки с последующей переинтерполяцией величин в отдельно взятой области) и расчет в однообластной постановке с использованием смешанных ячеек с применением автоматической корректировки сетки в локальных областях и с применением пересчета величин с деформированной сетки на новую без остановки счета (либо на стационарную сетку, либо на динамически адаптированную под решение сетку). В этом случае появление смешанных ячеек возможно только на границах раздела веществ после локальной корректировки или полного перестроения всей сетки. Между этапами пересчета величин расчет проводится в лагранжевом приближении без допущения перетекания веществ между ячейками.

Подход к расчету задач с использованием смешанных ячеек позволяет исключить влияние «человеческого фактора» на результаты и значительно сократить сроки проведения расчетов без существенного отличия в получаемых результатах от экспериментальных данных и результатов расчета без смешанных ячеек с учетом контактного взаимодействия между различными материалами.

В проведенных двумерных и трехмерных расчетах наблюдается хорошее качественное согласие. Отличие глубины проникания ударника составило 3–6 % в зависимости от применяемого подхода счета. Полученные результаты коррелируют с двумерными расчетами и известными экспериментальными данными.

С помощью методики ТИМ и разработанного подхода с использованием смешанных ячеек для проведения такого класса расчетов можно довольно просто и быстро получить оценки по бронепробитию преград при вариации скорости, материала преграды, толщины преграды, угла подхода и угла атаки, формы ударника и тем самым минимизировать затраты на дорогостоящие эксперименты.

Литература

1. Соколов С. С., Панов А. И., Воропинов А. А., и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 2. С. 37–52.

2. Соколов С. С., Воропинов А. А., Новиков И. Г., и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29–43.

3. Панов А. И., Шурыгин А. В. Некоторые алгоритмы построения неструктурированных многоугольных сеток для методики ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2015. Вып. 1. С. 56–65.

4. Панов А. И., Шурыгин А. В. Использование различных преобразований счетной сетки в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2015. Вып. 2. С. 78–87.

5. Фархутдинов В. Ф., Тарасов В. И., Соловьев А. Н., Борисенко О. Н., Лазарев В. В., Логинов Д. В., Аверина Н. С., Марунин А. В., Гриднев А. И., Федякина М. Г., Кузнецов М. Г., Черенкова М. В., Фролова Е. А., Лукичев А. Н., Смолкина Д. Н., Купалова А. Г., Кузьменко М. В., Ховрин Н. А., Сергеева А. С., Попова Н. В., Баканова Т. Ю. Обзор возможностей по обработке геометрических и построению сеточных моделей в ЛОГОС-Препост // Труды XIV Международной конференции «Супервычисления и математическое моделирование». Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2013. С. 585–592.

6. Charles E. Anderson, Jr. Bruce L. Morris, David L. Littlefield. A Penetration Mechanics Database. SwRI Report 3593/CO1 (отчет для агентства DARPA, США), 1992.

7. Половникова Т. Н., Свидинский А. В., Соколов С. С., Новиков И. Г., Самодолов А. В. и др. «Трехмерное численное моделирование бронебойного действия удлиненного ударника в методике ТИМ-3D» // Доклад на 47-я НТК "Проектирование систем", 4–6 февраля 2020 г, МГТУ им. Н. Э. Баумана , Москва.

THE TIM CODE RESULTS OF NUMERICAL SIMULATION OF IMPACTOR'S ARMOUR-PIERCING EFFECT ON A BARRIER

A. V. Svidinskiy, S. S. Sokolov, I. G. Novikov, A. V. Samodolov, T. N. Polovnikova

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents results of the numerical simulation of the armour-piercing effect of an impactor on a barrier obtained by the TIM code. Results of 3D calculations with the use of mixed cells and without them are compared with results of 2D calculations without mixed cells and available data of experiments.

Key words: three-dimensional and two-dimensional calculations, numerical simulation, the TIM code, an impactor, a barrier, mixed cells.

УДК 004.3
DOI 10.53403/9785951505071_2022_417

ЭТАПЫ ПОДГОТОВКИ ДЕТАЛИ К ПЕЧАТИ В ПРОГРАММЕ “ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР”

Е. А. Синицин, О. И. Бритова, В. В. Попов, А. Н. Бахаев, Л. В. Гамов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В рамках проекта «Разработка комплекса программного обеспечения для моделирования физических процессов, протекающих при селективном лазерном сплавлении с целью прогнозирования структуры, свойств материалов, а также получения изделий с заданными свойствами, и проведения топологической оптимизации изделий» в ФГУП «РФЯЦ-ВНИИЭФ» с 2019 года разрабатывается программный комплекс «Виртуальный 3D-принтер». Один из модулей данного программного комплекса – модуль подготовки деталей к печати на аддитивном оборудовании.

В данном докладе рассказывается о программной реализации этапов подготовки деталей к 3D-печати, осуществляемой по технологии селективного лазерного сплавления (SLS, SLM). Основные этапы [1], о реализации которых пойдет речь в докладе, являются:

– компоновка (расположение) деталей на платформе и задание настроек;
– генерация поддержек;
– нарезание деталей на слои «слайсинг»;
– заполнение слоев штриховкой «хэтчинг»;
– генерация управляющего кода.

Ключевые слова: аддитивные технологии, селективное лазерное спекание, селективное лазерное плавление, «виртуальный 3D-принтер», компоновка деталей на платформе, генерация поддержек, нарезание деталей на слои «слайсинг», заполнение слоев штриховкой «хэтчинг», генерация управляющего кода.

Введение

В настоящее время в мире промышленно-развитые страны активно развивают аддитивные технологии [1]. Большой интерес представляет использование лазерных аддитивных технологий (в частности технология селективного лазерного сплавления) с целью получения объемных металлических деталей из порошков различных металлов и сплавов. Особенностью и основной трудностью данных технологий является выбор оптимальных режимов работы оборудования и обеспечение заданного качества изготавливаемых деталей (прежде всего речь об уровне механических свойств, остаточных напряжений и деформаций).

Во ФГУП РФЯЦ-ВНИИЭФ с 2019 года разрабатывается комплекс программного обеспечения для моделирования физических процессов, протекающих при селективном лазерном сплавлении с целью прогнозирования структуры, свойств материалов, а также получения изделий с заданными свойствами, и для проведения топологической оптимизации изделий – «Виртуальный 3D-принтер». Предполагается, что базовая версия создаваемой программы будет обеспечивать:

– создание и редактирование деталей;

– оптимизацию формы деталей на основе методов топологической оптимизации (бионический дизайн), в том числе с использованием решетчатых структур;

– подготовку деталей к печати;

– выбор оптимальных параметров работы оборудования;

– прогнозирование свойств деталей (микроструктура и механические свойства) на базе многомасштабного математического моделирования (на макро- и мезоуровнях).

Использование программного комплекса должно способствовать более широкому использованию аддитивных технологий в промышленности, сокращению времени внедрения технологии для выпуска новых деталей и получению высококачественных изделий.

Важной частью данного программного комплекса «Виртуальный 3D-принтер» является модуль подготовки деталей к печати, который также используется для подготовки начальных данных для моделирования физических процессов на макроуровне. Данные модуль тесно интегрирован с другими модулями в единую программу с пользовательским графическим интерфейсом. Процесс подготовки деталей к печати состоит из нескольких стандартных этапов [1], которые могут быть заданы последовательностью операций. Выполнение каждой операции (этапа) в программе задается в своем пользовательском диалоге. Подробное описание функциональности каждого этапа приведено ниже.

Компоновка (размещение) деталей на платформе

Качество напечатанного изделия напрямую зависит от правильности расположения детали на платформе 3D-принтера, что задается при подготовке детали к трехмерной печати. Расположение детали наряду с другими факторами влияет: на прочность получаемого изделия, на качество получаемой поверхности, на время изготовления и количество используемого материала (изготовление поддержек). Из статьи [2] известно, что прочностные характеристики изделий при нагружении вдоль направления выращивания детали (укладки слоев) намного выше, чем при нагружении в перпендикулярных направлениях. Поэтому расположить деталь надо таким образом, чтобы в ответственных местах (особенно в местах сечений с малой площадью) нагрузка шла именно вдоль укладки слоев.

Реализованная в программном комплексе «Виртуальный 3D-принтер» операция «Компоновка деталей на платформе» позволяет выбирать аддитивное оборудование, на котором будет производиться печать, задавать настройки (областей печати, лазеров, материалов, режимов предпечатной подготовки и печати), регулировать положение и ориентацию деталей на платформе, размещать несколько экземпляров деталей (если это необходимо).

На рис. 1 продемонстрирована работа данной операции. Красным цветом показаны выбранные в диалоге детали для размещения на платформе, а зеленым – детали, уже размещенные на платформе.

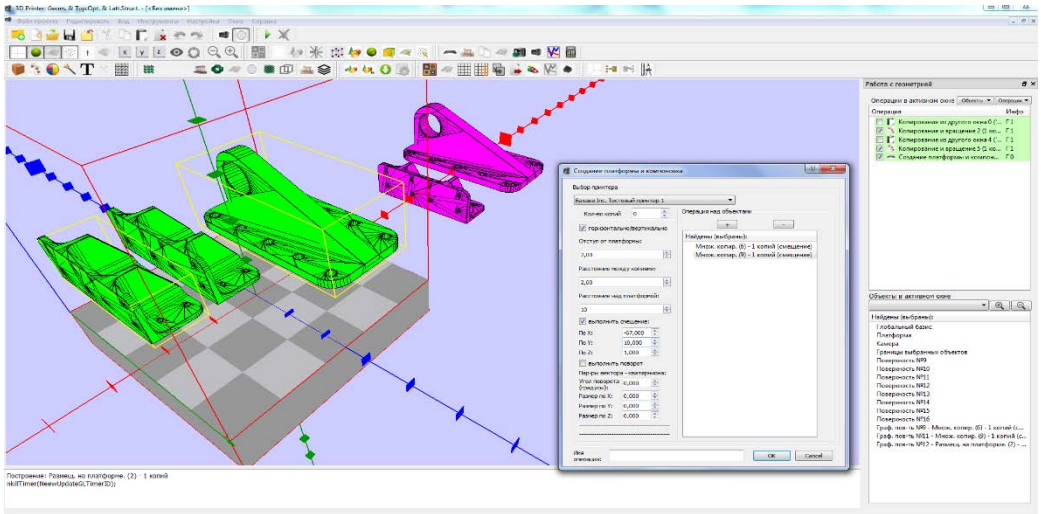
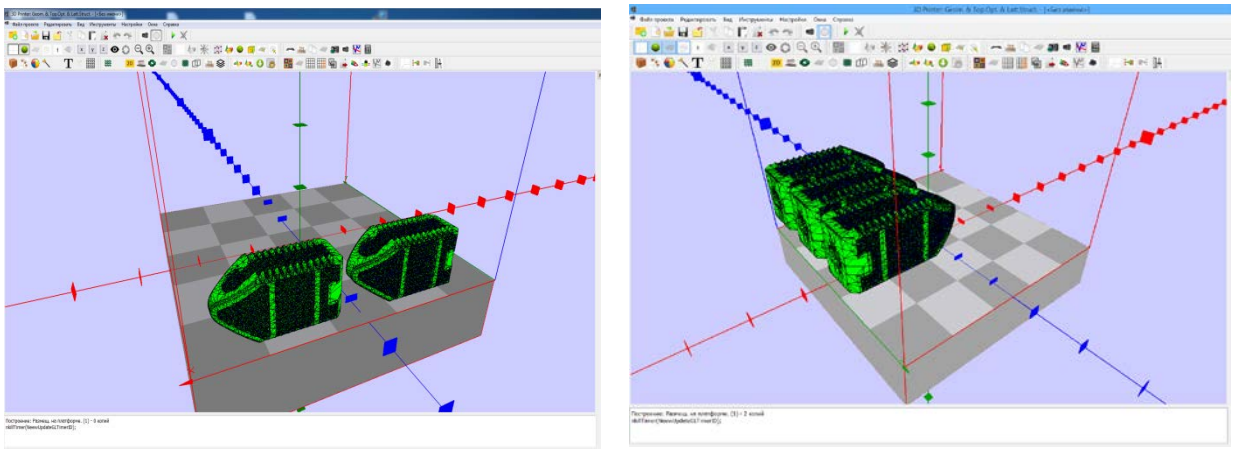


Рис. 1. Вид платформы с размещенными на ней деталями

Операция «Компоновка деталей на платформе» позволяет размещать детали, как в ручном режиме, так и в автоматическом. В случае автоматической компоновки операция размещает детали с учетом заданных в диалоге параметров: расстояние от края платформы, расстояние между копиями детали, высота над платформой. Примеры ориентации деталей на платформе в автоматическом режиме с заданием числа размещаемых копий оси X и по оси Y показаны на рис. 2. Так же возможно расположение детали на платформе в ручном режиме путем задания смещения по трем координатным осям X, Y, Z, с указанием углов поворотов изделия вокруг этих осей. Пример размещения нескольких деталей на платформе в ручном режиме показан на рис. 3.



а б

Рис. 2. Расположение изделий на платформе в автоматическом режиме (а – по оси X, б – по оси Y)

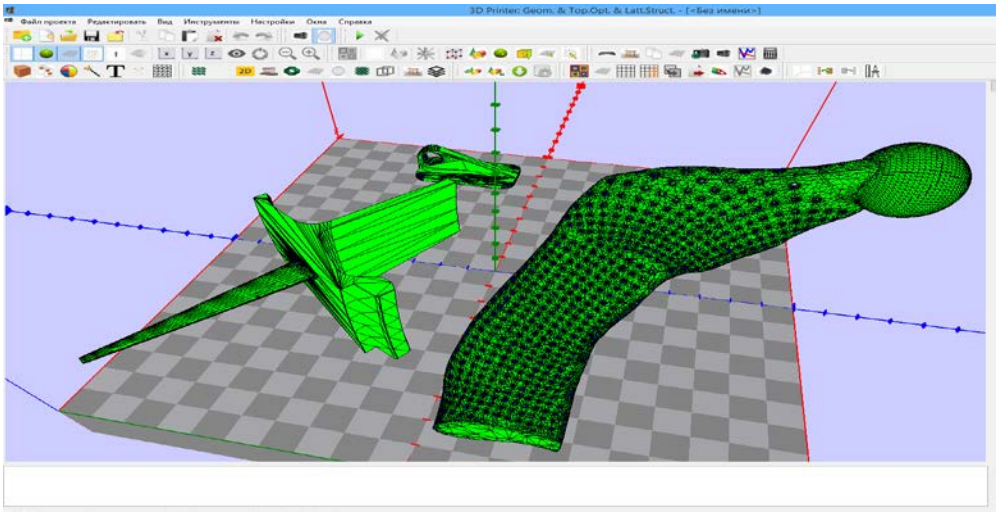


Рис. 3. Расположение изделий на платформе в ручном режиме

Задание области(ей) действия лазера(ов)

Печать деталей осуществляется с использованием одного или нескольких лазеров. Использование нескольких лазеров в аддитивных машинах необходимо, для ускорения построения деталей в рабочей камере за счет разделения области рабочей камеры между лазерами. Для включения использования лазера и задания области действия лазера используется операция задания области(ей) действия лазера(ов). Реализованная операция позволяет выбрать способ работы лазера: все лазеры работают по всей рабочей камере или каждый в своей области. Для разделения рабочей камеры

на области используются габаритные коробки, которые определяют область действия выбранного лазера. Габаритная коробка формируется посредством задания минимальной и максимальной точки коробки, в случае необходимости поворота вокруг оси Z задается угол поворота fi. Пример работы данной операции представлен на рис. 4.

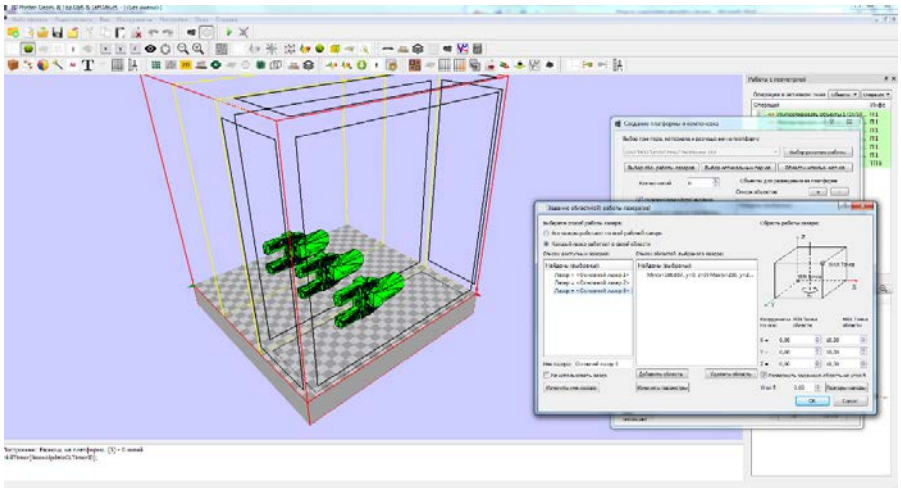


Рис. 4. Платформа с размещенными на ней деталями и выделенной областью действия лазера 3

Распределение порошкового материала по рабочей камере

Современные 3D-принтеры позволяют выпекать изделия, используя несколько материалов для печати. Для распределения порошкового материала по рабочей камере используется операция задания области распределения материала. Данная операция позволяет включать использование выбранного материала из списка доступных материалов и указывать область распределения материала. Список с доступными материалами зависит от модели принтера. Реализованная операция позволяет задавать распределение порошкового материала послойно или произвольно. Послойное задание порошкового материала подразумевает, что на одном слое не может быть больше одного материала.

Для распределения порошка по рабочей камере принтера используются габаритные коробки, которые определяют область нахождения материала в рабочей камере. Габаритная коробка формируется посредством задания минимальной и максимальной точки коробки. Пример работы данной операции представлен на рис. 5.

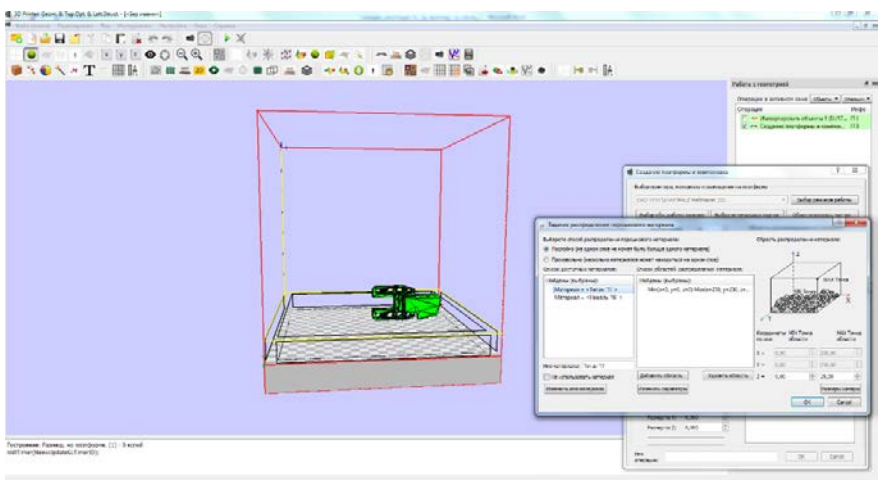


Рис. 5. Платформа с размещенной на ней деталью и выделенной областью материала 1

На рис. 5 черной рамкой выделена область материала 1. В качестве материала 1 используется «Ti», в качестве материала 2 – никель «Ni».

Выбор взаимосвязанных параметров предпечатной подготовки деталей

Рад параметров предпечатной подготовки деталей, режимов работы лазеров и качественных характеристик получаемых деталей взаимосвязаны между собой. Для точного определения их значений и последующего задания при предпечатной подготовке в общем случае необходимо проводить многочисленные серии расчетов физических процессов на макроуровне. Такие расчеты занимают много времени и во многом повторяют друг друга при изготовлении геометрически разных деталей. В рамках проекта специалистами Удмуртского Государственного Университета (Кривилев М. Д., Гордеев Г. А. и др.) разрабатывается модуль «Геометрия зоны сплавления», выходной информацией которого служат расчетные зависимости параметров друг от друга, которые получены в результате обработки экспериментальных данных и данных математического моделирования формирования ванны расплава. Эти зависимости позволяют априори выбрать важные взаимосвязанные параметры, на этапе предпечатной подготовки деталей, так чтобы качество деталей после изготовления оказалось допустимым. Важными выбираемыми параметрами на этом этапе являются: толщина порошкового слоя; расстояние между треками (дорожками сплавления), удельная мощность лазерного излучения.

Генерация поддержек

Печать деталей по технологии селективного лазерного сплавления требует наличие вспомогательных структур («поддержек») для:

– поддержания выступающих элементов геометрии;

– упрощения отделения готовых деталей от платформы;

Изготовление выступающих элементов геометрии, не опирающихся на другие элементы конструкции, требует использования поддержек для уменьшения или устранения возникающих деформаций в этих элементах геометрии. Примерами таких элементов можно считать горизонтальные и наклонные поверхности, арки.

Правильная расстановка поддержек позволяет сократить количество используемого материала и уменьшить время изготовления деталей.

Реализованная операция «Генерация поддержек» позволяет создавать поддержки балочного типа (призмы) для деталей, размещенных на платформе. Пользователь программы имеет возможность задать следующие настройки: предельный угол отклонения от вертикали граней деталей, под которые строятся поддержки, диаметр поддержек, расстояние между поддержками. Пример создания поддержек для нескольких разных деталей показан на рис. 6.

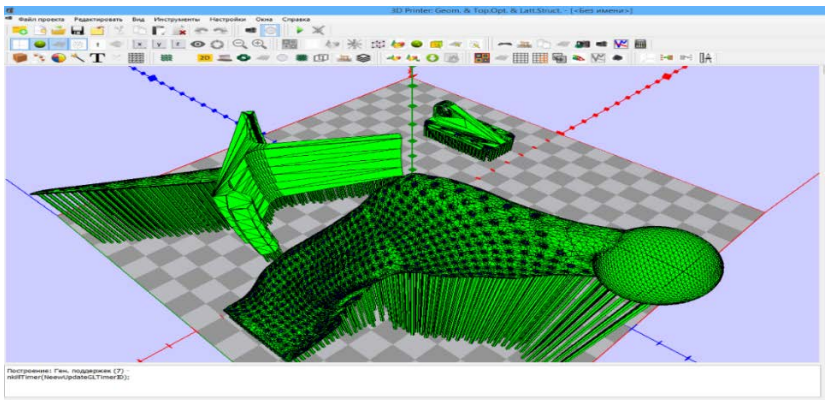


Рис.6. Пример создания поддержек для нескольких деталей

Нарезание деталей на слои «слайсинг»

Так как 3D-принтеры создают изделие по слоям, то программное обеспечение для подготовки работы принтера должно формировать слои изделия для печати на основе исходной объемной модели изделия, обычно представленной в формате STL. STL файл представляет собой список координат вершин треугольных граней (фасетов), образующих поверхность [3].

Существует множество алгоритмов для нарезания модели изделия на слои, но все они построены по принципу поиска пересечений элементов поверхности (фасетов) моделей деталей с секущими плоскостями. Основными шагами их работы являются: создание секущих плоскостей, определение векторов, образуемых пересечением секущей плоскости с поверхностью деталей (состоящей из фасетов), и генерация замкнутых контуров [3].

Реализованная операция позволяет нарезать детали, размещенные на платформе, на слои. Пользователь программы имеет возможность задавать следующие настройки: толщину слоя, количество слоев, высоту первого слоя. Пример нарезания деталей на слои представлен на рис. 8. Исходные детали, размещенные на платформе, показаны на рис. 7.

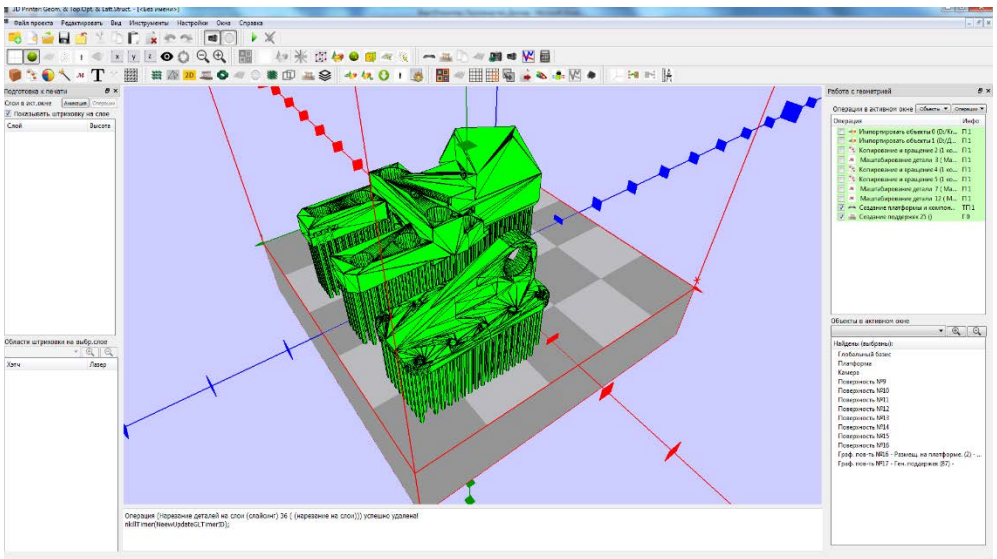


Рис. 7. Вид платформы с размещенными на ней деталями и построенными поддержками

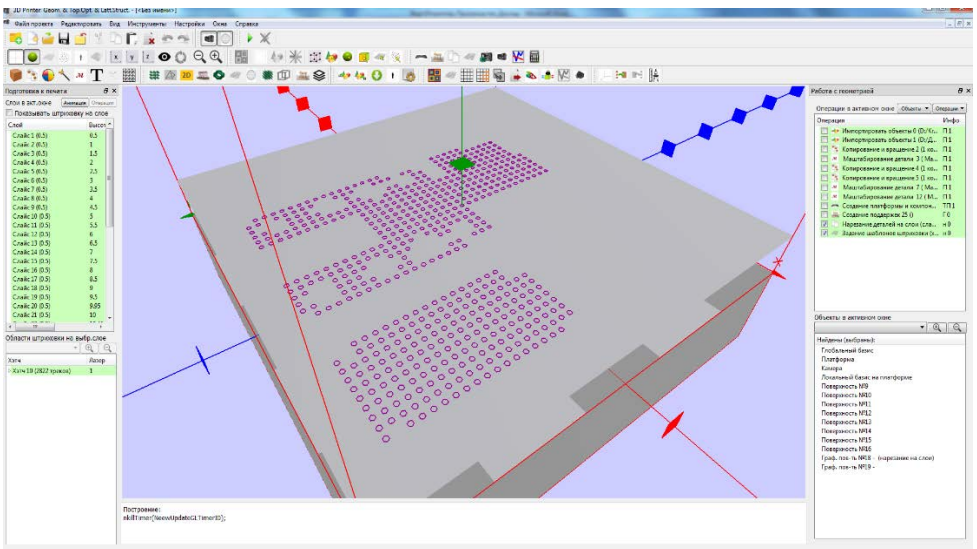


Рис. 8. Результат выполнения операции нарезания деталей на слои («слайсинг»)

Заполнение слоев штриховкой «хэтчинг»

После этапа нарезания объемной модели на слои, необходимо выделить зоны для отдельной обработки, задать режимы построения в этих зонах и заполнить полученные слои штриховкой. Реализованная операция заполнение слоев штриховкой «хэтчинг» позволяет выделить зоны, задать режимы и заполнить полученные слои штриховкой. Для выделения доступно 4 зоны: Down-Skin (Низ детали), Up-Skin (Вверх детали), Support (поддержки), NearPlatform (Около-платформенные слои). Пользователь программы имеет возможность задать следующие настройки: режим работы в активированной зоне (мощность и скорость лазера), тип штриховки, шаблон штриховки, шаг штриховки.

Основными шагами этапа заполнения слоя штриховкой являются: создание семейства прямых в соответствие с входными данными (шаг штриховки, тип штриховки); вычисление отрезков (треков движения лазера), полученных пересечением найденных контуров с семейством прямых; упорядочивание отрезков (треков) расположенных внутри детали по заданным критериям и сохранение их в специальных структурах, данных для последующего использования. Результат выполнения операции «хэтчинг» в программном комплексе «Виртуальный 3D-принтер» представлен на рис. 9.

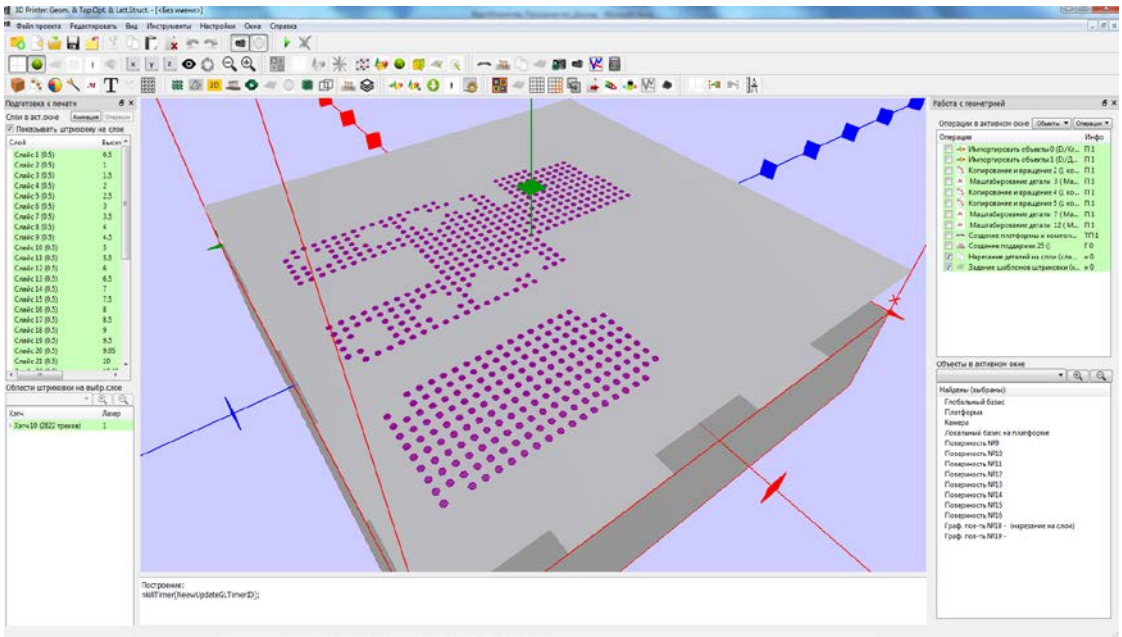


Рис. 9. Результат выполнения операции заполнения слоев штриховкой («хэтчинг»)

Генерация управляющего кода

Генерация управляющего кода реализована отдельной операцией, которая по результатам предыдущих операций формирует управляющий код в соответствии с заданными характеристиками конкретного 3D-принтера. Модели 3D-принтеров разных производителей используют различные реализации управляющего кода.

В программном комплексе «Виртуальный 3D-принтер» на данный момент реализована возможность генерации управляющего кода для принтеров линейки «MeltMaster», в частности «MeltMaster – 250». Сгенерированные командные файлы для аддитивной машины имеют текстовый формат и состоят из команд собственного языка управления для этого устройства. Основные команды: выбор режимов работы лазеров; управление перемещением луча лазера; опускание платформы и насыпка слоя порошка (начало и конец слоя). Пример управляющего кода представлен ниже.

```
-- Слой 1
F_In
Laser.Power = 76
Laser.MarkSpeed = 650
Laser.Active = 1
Image.Line(-119.5,-93.5,-118.5,-93.5)
Image.Line(-119.5,-93.5,-119.5,-92.5)
Image.Line(-119.5,-92.5,-118.5,-93.5)
Image.Line(-119.5,-92.5,-118.5,-92.5)
Image.Polyline3D(false,-106.8,-33.91,0,-106.8,-33.93, ….. , 33.65,0,-106.74,-33.76,0,-106.8,-33.91,0)
...
Image.Line(-119.5,-93.5,-118.5,-93.5)
Image.Line(-119.5,-93.5,-119.5,-92.51)
F_Out
-- Слой 2
F_In
...
```

Здесь: F_In – тег указывающий начало слоя; F_Out – тег указывающий конец слоя; Laser.Power – мощность лазера; Laser.MarkSpeed – скорость лазера; Laser.Active – номер лазера; Image.Line – трек, по которому бежит лазер; Image.Polyline3D–последовательность треков, по которым бежит лазер.

Заключение

В рамках проекта «Разработка комплекса программного обеспечения для моделирования физических процессов, протекающих при селективном лазерном сплавлении с целью прогнозирования структуры, свойств материалов, а также получения изделий с заданными свойствами, и проведения топологической оптимизации изделий» в ФГУП «РФЯЦ-ВНИИЭФ» с 2019 ведется разработка модуля подготовки детали к печати, который является частью программного комплекса «Виртуальный 3D-принтер». В докладе приведено описание программной реализации этапов подготовки деталей к печати. Этапы подготовки к печати тестовых деталей продемонстрированы на рисунках.

Литература

1. Гибсон Я., Розен Д., Стакер Б. Технологии аддитивного производства – М.: Техносфера, 2016. С. 656.

2. Камоничкин Д. Как правильно расположить деталь на платформе 3D - принтера? // Режим доступа: https://st3d.com/blog/positioning/

3. Гущин И. А., Авдеев А. Р., Швец А. А., Дроботов А. В. Алгоритм деления объемной модели на слои для 3D-печати. // Режим доступа: https://cyberleninka.ru/article/n/algoritm-deleniya-obemnoy-modeli-na-sloi-dlya-3b-pechati

STAGES OF PREPARING A COMPONENT FOR PRINTING IN SOFTWARE PACKAGE "VIRTUAL 3D PRINTER"

E. A. Sinitsyn, O. I. Britova, V. V. Popov, A. N. Bakhaev, L. V. Gamov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The "Virtual 3D printer" software package has been being developed at FSUE "RFNC-VNIIEF" since 2019 under the project "The development of software to simulate physical processes in selective laser melting in order to predict the structure and properties of materials, manufacture products with the desired properties and perform topological optimization for products". One of the program modules in this software is a module used to prepare components for printing on additive equipment.

The report describes the software implementation of stages of preparing components for 3D printing using the selective laser melting (SLM), or selective laser sintering (SLS) technology. The main stages described in the report are, as follows:

- arrangement of components on a platform;
- generation of supports;
- slicing;
- hatching;
- generation of control code.

Key words: additive technologies, selective laser sintering, selective laser melting, "virtual 3D printer", arrangement of components on a platform, generation of supports, slicing, hatching, generation of control code.

УДК 004.94:528.952
DOI 10.53403/9785951505071_2022_426

ЯЗЫК ПРОГРАММИРОВАНИЯ АГЕНТОВ OPAL++ ВИЗУАЛИЗАЦИОННО-ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ «ОПТИМУС»

Д. С. Собанин, И. А. Крючков, А. В. Огородников, О. В. Коваленко

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Разработка программных агентов и эффективное задание их логики поведения являются ключевыми и наиболее сложными элементами при разработке имитационных мультиагентных моделей.

Наращивание сложности и подробности моделей зачастую требует разработки большого количества программного кода в сжатые сроки проекта. Реализация высокоуровневых визуальных и скриптовых средств программирования логики поведения агентов в мультиагентных системах может обеспечить решение этой задачи.

В работе представлены базовые подходы и средства предметно-ориентированного языка программирования агентов Opal++ визуализационно-интеграционной платформы «ОптИМУС». Показаны перспективы развития языка программирования Opal++ с целью упрощения восприятия и сопровождения разрабатываемых моделей агентов, а также интеграции технологий машинного обучения и искусственного интеллекта в имитацию интеллектуального поведения агента.

Ключевые слова: мультиагентное имитационное моделирование, ВИП «ОПТИМУС», агент, решатель, макроязык, Opal++.

Для обеспечения эффективной и качественной разработки специализированных программных комплексов поддержки принятия решений, имитационного моделирования и оптимизации во ФГУП «РФЯЦ-ВНИИЭФ» разработана визуализационно-интеграционная платформа «ОптИМУС» [1, 2]. В ВИП «ОптИМУС», в рамках мультиагентного подхода к моделированию, анализируемая система представляется в виде совокупности взаимодействующих по формализованным правилам программных агентов.

При разработке имитационных мультиагентных моделей наиболее трудоемким процессом на практике является разработка агентов и задание их логики функционирования.

Наращивание сложности и подробности моделей зачастую требует разработки большого количества программного кода в сжатые сроки проекта. При этом для выполнения работы необходимо привлекать к процессу программирования специалистов в исследуемой предметной области, которые могут не обладать требуемым уровнем подготовки в прикладном программировании и мультиагентном моделировании.

Для решения данной проблемы, экспертов предметной области необходимо обеспечить специальными средствами, позволяющими разрабатывать сложные и подробные модели без глубокого погружения в низкоуровневое программирование.

Бурное развитие технологий искусственного интеллекта во всем мире [3] и необходимость их внедрения в различные моделируемые системы также накладывает дополнительные требования к программным средствам [4]. С одной стороны, мультиагентное моделирование является одной из технологий искусственного интеллекта, с другой, в работе мультиагентных систем, при повышении детальности и сложности модели, требуется внедрение других технологий ИИ. В частности,

нейросетевых технологий, нечеткой логики, динамических навигационных сетей, метаэвристической оптимизации, онтологического представления знаний и других.

Таким образом, при создании языка программирования агентов, ставились следующие цели:

− средства программирования агентов должны обеспечивать возможность разработки без погружения в низкоуровневое программирование (это необходимо для специалистов предметной области);

− органичная интеграция технологий искусственного интеллекта в разрабатываемые мультиагентные системы.

На первой стадии реализации ВИП «ОптИМУС» в ее составе были реализованы два крупных программных пакета: мультиагентного моделирования Оптимус.Решатель и препостпроцессинга мультиагентных задач Оптимус.Препост с графическим пользовательским интерфейсом. На этой стадии, разработка агентов выполнялась с помощью языка программирования общего назначения C++. Разработчику агента требовалось, с использованием API решателя и API визуализационных средств платформы реализовать все компоненты агента. Как следствие, средства программирования агентов очень сильно связаны с обоими программными пакетами.

При проработке архитектуры модуля языка программирования логики агентов Opal++ выполнена его интеграция в Оптимус.Решатель и Оптимус.Препост (см. рис. 1).

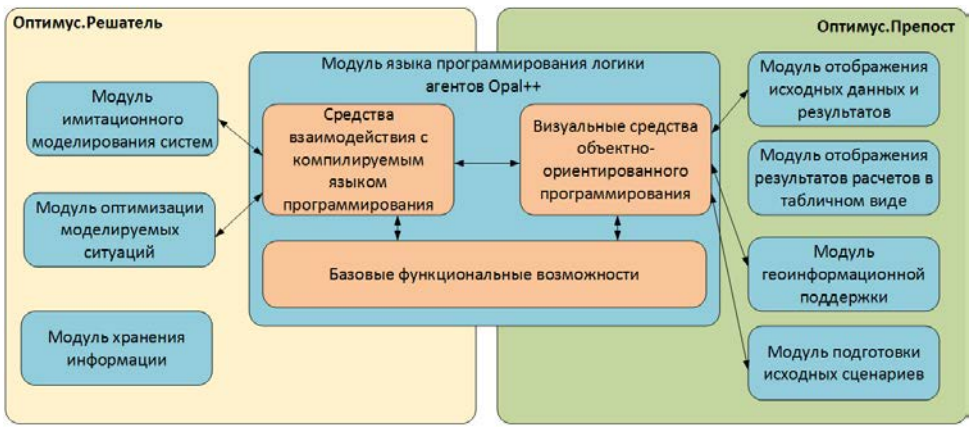


Рис. 1. Интеграция языка Opal++ в архитектуру ВИП «ОптИМУС»

Программный агент представляет собой сложную структуру (см. рис. 2), которая содержит множество параметров, поведения, задачи и т. д. Разработка агента с «чистого листа» на языке C++ – это трудоемкий и сложный процесс, требующий от разработчика множество разноплановых навыков. В частности для этого необходимы, помимо знаний в предметной области, знания особенностей работы API мультиагентного решателя, языка программирования C++, API препостпроцессора интеграционной платформы. Для того чтобы упростить (меньше специализированных знаний) и ускорить (радикальное сокращение рутинных операций) создание новых и развитие существующих агентов, в язык программирования логики агентов Opal++ заложены следующие возможности:

− поддержка функциональных возможностей мультиагентного решателя;

− поддержка функциональных возможностей визуализационно-интеграционной платформы;

− возможность конструирования новых типов агентов;

− средства разработки логики поведения агентов.

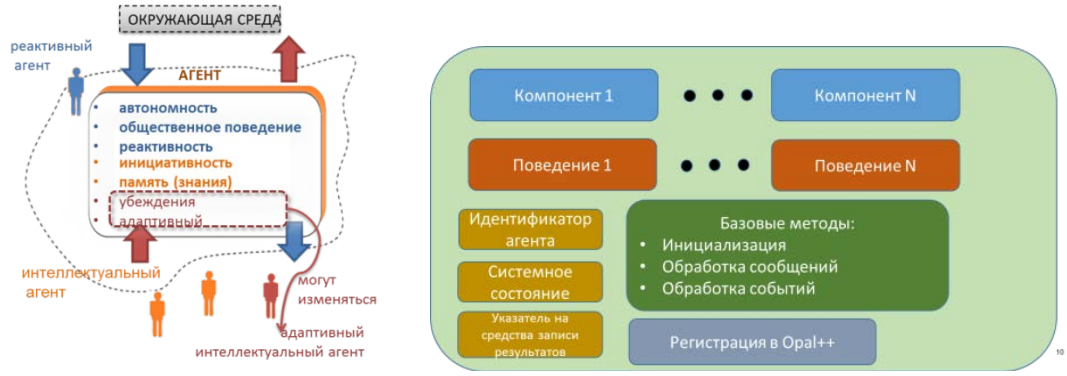
Конструктор агентов должен формировать нового агента с требуемыми предметному специалисту качествами. Новый агент основывается на классе существующего (или базового универсального агента) и содержит:

− базовый класс агента (от которого происходит наследование);

− атрибуты;

− поведение;

− дочерние объекты.



а б

Рис. 2. Программный агент: а – логическое представление; б – представление в формате мультиагентного решателя

Разработка логики агента может вестись разными специалистами. Базовый элементарный уровень может разрабатываться на низкоуровневом языке C++, а для поведенческой логики требуются сложные высокоуровневые конструкции. При создании сложной логики могут быть эффективны средства скриптового и визуального программирования. При реализации интеллектуальных агентов в перспективе потребуется встраивание в состав агента контроллеров на основе различных технологий ИИ (нейросетевой, нечеткологический, программа действий на предметно-ориентированном языке, онтологический и другие).

В состав Opal++ входят средства для программирования на различных уровнях абстракции:

– низкоуровневое задание логики и вычисление пользовательских выражений;

– объектно-ориентированное программирование;

– предметно-ориентированное программирование на основе специализированных языковых средств;

– высокоуровневое программирование с помощью различных графических диаграмм.

Каждый более низкий уровень абстракции позволяет реализовывать средства для ведения разработки на более высоком уровне. В результате каждый уровень агента может быть проработан максимально детально и при этом на уровне логики принятия решений не будет мелких подробностей, которые перегружают восприятие.

Как уже было сказано, средства разработки Opal++ могут использоваться для создания агента с «чистого листа», для разработки высокоуровневой логики поведения и для реализации мелких частных методов. При этом все эти элементы могут быть разработаны на C++. Таким образом, потребовалась возможность осуществлять разнонаправленные вызовы программного кода:

– из визуального кода на Opal++ выполнять вызовы скриптового кода на Opal++;

– из скриптового кода на Opal++ выполнять вызовы визуального кода на Opal++;

– из кода (визуального и скриптового) на Opal++ выполнять вызовы программных компонент, разработанных на C++;

– из кода на C++ осуществлять вызовы программных компонент, разработанных на Opal++ (визуально и/или на скриптах).

Для реализации этих возможностей все видимые для языка сущности (например, классы, методы, свойства, типы сообщений, типы событий и др.) должны быть зарегистрированы в единой интерпретационной системе Opal++. При разработке кода в Opal++ это происходит автоматически, а для сущностей C++ требуется прямая регистрация. Например, регистрация метода, осуществляется C++ кодом:

OPAL_BIND_METHOD (METHOD("step2", "param"), &MyAgent::step2);

А, например, вызов метода *step2* осуществляется:

obj->opal()->call("step2", 10)

при этом неважно на каком языке реализован метод *step2*, на C++ или на Opal++.

В настоящее время выполнение программного на Opal++ реализовано через интерпретацию для возможности запуска в окружениях без поддержки средств разработки. Это накладывает естественные ограничения в производительности разрабатываемого кода. Для решения этой проблемы

в перспективе планируется реализация генерации C++ кода для идентичной замены отработанных ресурсоемких методов.

Для обеспечения возможности функционирования агентов, задаваемых на Opal++, в мультиагентной среде Оптимус.Решатель, реализован доступ к программным интерфейсам (API) ядра мультиагентного решателя через регистрацию в общем интерпретационном механизме Opal++. Это позволяет пользователям в программном коде на текстовом скриптовом языке и в визуальных программах использовать базовые классы модуля мультиагентного моделирования, необходимые для программирования агентов.

Агентная система, с точки зрения мультиагентного решателя, состоит из следующих логических компонентов, каждый из которых представляет собой набор логических возможностей:

- агентная платформа (Agent Platform, AP) – обеспечивает инфраструктуру, в которой агенты могут создаваться и функционировать. Агентная платформа функционирует на каждом вычислительном процессе;
- агент – программный объект (существует в единственном экземпляре на каком-либо вычислительном процессе), реализующий автономную функциональность. Агенты обмениваются информацией с помощью систем обмена сообщениями и событиями;
- сервис – программный объект (существует на каждом вычислительном процессе), реализующий функциональность, необходимую для использования агентам. Сервисы получают информацию непосредственно от агентов или от системы обмена событиями;
- справочник вспомогательных функций и сервисов – компонент, предоставляющий для агентов доступ к сервисам;
- система управления агентами – осуществляет управление доступом к агентам, поддерживает актуальное состояние контейнера агентов;
- система обмена сообщениями – средство обмена информацией (передачи сообщений) между агентами (в том числе и на разных вычислительных процессах);
- система обмена событиями – средство публикации и доставки событий (порождаемых агентами) в мультиагентной системе для агентов и сервисов (в том числе и на разных вычислительных процессах);
- планировщик – специальное средство, которое обеспечивает общую синхронизацию вычислительных процессов, продвижение модельного времени, активацию агентов и т. д.

Для доступа к этим средствам, основные компоненты зарегистрированы в интерпретационных средствах Opal++ и доступны, как глобальные переменные в любой программе:

- AMS – система управления агентами (Agent Management System);
- SCHEDULER – планировщик (Scheduler);
- DF – справочник вспомогательных сервисов (Directory Facilitator);
- MTS – сервис обмена сообщениями (Message Transport Service);
- ETS – сервис обмена событиями (Event Transport Service).

Также зарегистрированы и доступны для расширения через наследование необходимые для разработки функционала агентов и средств их взаимодействия классы библиотеки базовых сущностей модуля мультиагентного моделирования:

- BaseAgent – класс базового агента, обеспечивающий базовые средства функционирования агента в мультиагентной среде решателя;
- Component – базовый элемент для расширения функциональности агентов;
- Behaviour – базовый объект, который реализует вызовы поведенческой логики агентов;
- BaseService – класс базового сервиса, обеспечивающего базовые средства для предоставления различных предметных сервисов агентам в мультиагентной среде;
- BaseEvent – класс базового события, с помощью которого агенты могу обмениваться информацией;
- BaseMessage – класс базового сообщения, с помощью которого агенты могу обмениваться информацией.

Для ведения разработки в едином программном пространстве средства Opal++ были интегрированы в программный пакет Оптимус.Препост (см. рис. 3).

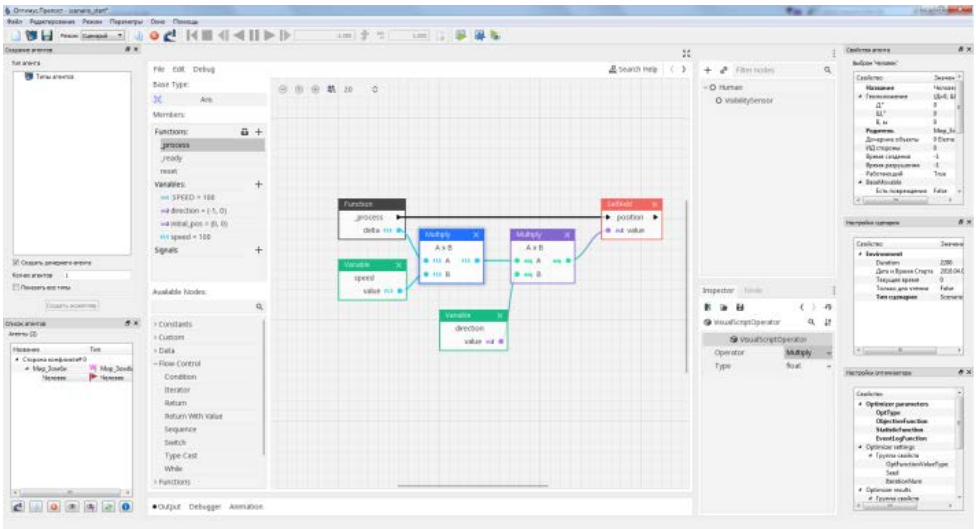


Рис. 3. Интеграция средств OPAL++ в ГПИ ВИП «ОптИМУС»

Основными визуальными интерфейсами Opal++ являются:
- редактор текстовых скриптов;
- визуальный редактор;
- конструктор агентов;
- вспомогательные элементы управления.

Редактор текстовых скриптов обеспечивает написание программ на специализированном скриптовом языке, динамически типизированном с опциональной статической типизацией (см. рис. 4). Синтаксис приближен к Python (используются отступы для выделения структурных блоков). Текстовый редактор поддерживает подсветку синтаксиса и предметно-ориентированные возможности дополнения кода.

```
# A file is a class!
# Inheritance
extends UniversalOpalAgent

# variables
var a = 5
var s = "Hello Optimus"
var arr = [1, 2, 3]
var dict = {"key": "value", 2:3}

# Constants
const ANSWER = 42
const THE_NAME = "Charly"

# Enums
enum {UNIT_NEUTRAL, UNIT_ENEMY, UNIT_ALLY}
enum Named {THING_1, THING_2, ANOTHER_THING = -1}
```

```
# Built-in vector types
var v2 = Vector2(1, 2)
var v3 = Vector3(1, 2, 3)

# function
func some_function(param1, param2):
    var local_var = 5
    if param1 < local_var:
        print(param1)
    elif param2 > 5:
        print(param2)
    else:
        print("Fail!")
    for i in range(20):
        print(i)

    while param2 != 0:
        param2 -= 1
```

Рис. 4. Пример кода на текстовом скриптовом языке

Визуальный редактор обеспечивает редактирование визуального кода и высокоуровневых поведенческих схем с различным графовым синтаксисом и семантикой. Диаграмма с визуальным программным кодом (см. рис. 5) состоит из функциональных блоков, имеющих входные и выходные порты, а также настраиваемые атрибуты. Порты соединены связями различных видов: одни определяют порядок выполнения функциональных блоков и задают поток управления, другие – передачу данных (порядок выполнения определяется зависимостями по данным). Реализован большой набор готовых функциональных блоков, в том числе для управления потоком выполнения, математических операций, вставки фрагментов текстовых скриптовых программ, создания и уничтожения объектов,

взаимодействия объектов, доступ к методам и свойствам объектов. Реализована возможность определения новых функциональных блоков различными способами.

Одной из больших проблем визуального программирования является борьба со сложностью (программы становятся слишком большими для восприятия в виде схем). Данная проблема решается путем иерархической декомпозиции (выделение части кода в отдельный функциональный блок).

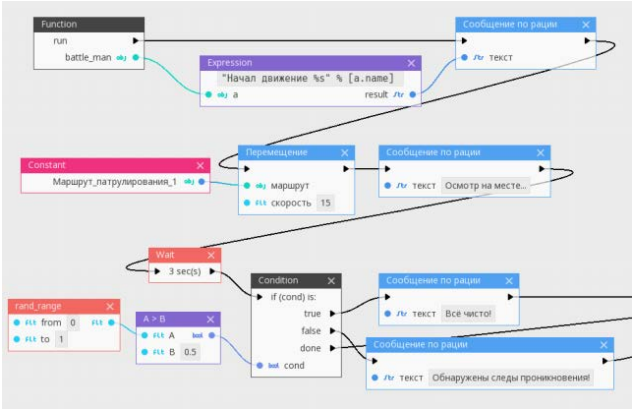


Рис. 5. Пример визуального кода

Процесс редактирования визуального кода включает в себя добавление новых блоков, соединение блоков связями, редактирование свойств блоков. Для быстрого добавления функциональных блоков из предметных библиотек присутствует поиск по нечеткому совпадению названия сущности и за счет перетаскивания визуальных сущностей (агентов, свойств, типов) из других окон ГПИ Оптимус.Препост.

Реализация функциональной и поведенческой логики агентов не может обойтись без взаимодействия агентов и составляющих их объектов между собой. Opal++ поддерживает весь спектр способов взаимодействия, доступный в рамках мультиагентного решателя:

− синхронные (прямой вызов метода, сигнал-слот) и асинхронные (сообщения, события, в том числе с временной задержкой через *SCHEDULER*);

− между агентами и внутри агента между подобъектами, совместно обеспечивающими его функционирование;

− адресно (адресат явно указывается) и широковещательно (адресат сам подписывается).

Для высокоуровневого описания поведения агента в Opal++ задаются способности агента (см. рис. 6) в виде доступных для вызова действий. Действия могут быть мгновенными, протяженными во времени с событием на завершение или бесконечной длительности. Некоторые заданные действия могут комбинироваться и запускаться на выполнение одновременно. Действие может требовать входные параметры, иметь результат успешности выполнения и выходные периметры, иметь предусловия доступности и ограничения на комбинирование с другими действиями, быть связанным с механизмами представления знаний интеллектуальных агентов.

Реализация способностей может осуществляться на любом из заявленных уровней абстракции. Например, элементарные способности могут быть реализованы в низкоуровневом C++ коде, более сложные в виде текстовых скриптов, а высокоуровневые – средствами визуального программирования.

При внедрении в ВИП «ОптИМУС» новых технологий ИИ, возможно их задействование при реализации различных способностей агентов.

Комбинирование этих подходов и технологий обеспечивает иерархическую декомпозицию и комбинирование различных способов описания логики простых и, в перспективе, интеллектуальных агентов, в рамках единого подхода на базе общего набора переиспользуемых сущностей.

Одной из задач создания собственного языка программирования агентов Opal++ в составе ВИП «ОптИМУС» является возможность создавать специальные предметно-ориентированные языки (ПОЯ) (в англ. литературе – Domain-specific language – DSL) (рис. 7).

Такие языки, с одной стороны, специализированы под конкретные задачи и области знаний (домены), а с другой – сохраняют полноту языка, не ограничивая программиста заданными рамками.

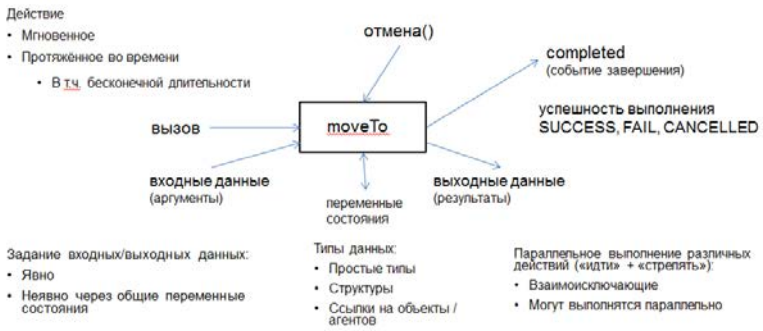


Рис. 6. Пример высокоуровневой способности агента

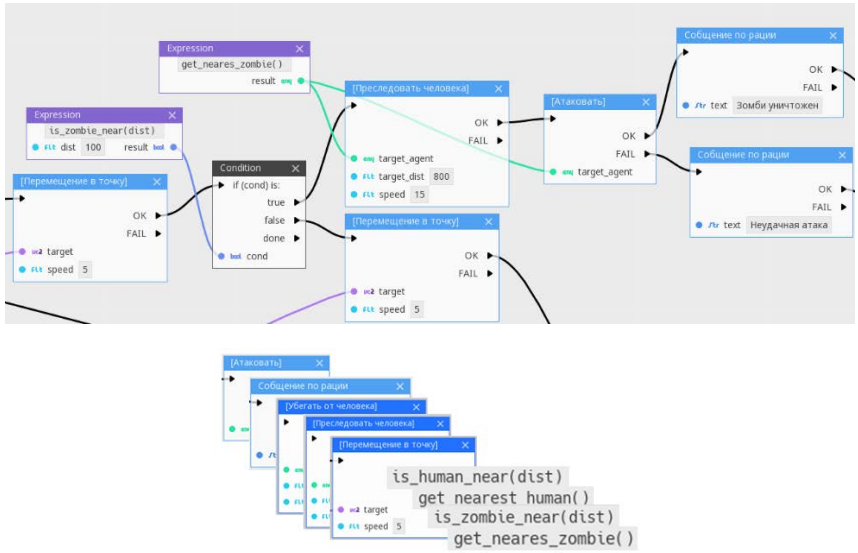


Рис. 7. Пример использования предметно-ориентированного языка (ПОЯ / DSL) для описания логики поведения агента

Использование ПОЯ в сочетании с визуальным программированием позволяет очень эффективно привлекать к процессу программирования агентов специалистов в исследуемой предметной области, а также существенно упрощает разработку, понимание и сопровождение разрабатываемых моделей агентов. Пример использования ПОЯ для описания логики поведения агента приведен на рис. 7.

Специализированный в сторону искусственного интеллекта ПОЯ может позволять внедрять блоки для интеграции технологий машинного обучения, онтологического подхода к представлению знаний, вероятностного анализа для облегчения процесса создания программной модели ИИ. Само создание и использование ПОЯ также относится к технологиям ИИ.

Для ИИ более важным является то, что пространство программ на ПОЯ имеет значительно меньшую размерность по сравнению с аналогичной программой, написанной на языке программирования более низкого уровня, что позволяет в пространстве таких структур проводить эффективный поиск оптимальных программ методами оптимизации. Для этих целей целесообразно использовать имеющиеся в составе ВИП «ОптИМУС» инструменты метаэвристической оптимизации.

Таким образом, в составе ВИП «ОптИМУС» реализован язык программирования агентов Opal++. Выполнена его интеграция с платформой на уровне пользовательского интерфейса и мультиагентного решателя. Изложены реализованные подходы и технологии программирования агентов,

показаны перспективные направления. В частности, приведены пути развития Opal++ с целью интеграции с технологиями искусственного интеллекта. Особенный интерес представляет реализация и использование:

- средств по созданию предметно-ориентированных языков;
- средств модификации в процессе функционирования поведенческой логики;
- механизмов принятия решений на основе онтологического подхода;
- средств постановки оптимизационных задач;
- процессного подхода при разработке моделей;
- средств обучения агентов (в том числе с подкреплением);
- средств нечеткологического вывода.

Литература

1. Свидетельство о государственной регистрации программы для ЭВМ №2019617157 «Визуализационно-интеграционная платформа для оптимизационного имитационного моделирования и управления системами (ВИП «ОптИМУС»)» / Коваленко О. В., Крючков И. А., Ежов Д. В., Огородников А. В., Ерошкина И. В., Собанин Д. С., Хочкин Н. И., Варгина Е. Ф., Тихомиров Ю. В., Рыжих А. В., Васильева Е. А., Кондратьев А. Б.

2. Коваленко О. В., Крючков И. А., Огородников А. В., Ежов Д. В., Собанин Д. С. Возможности визуализационно-интеграционной платформы ОптИМУС для имитационного моделирования вооружений, военной и специальной техники // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2019. Вып. 4

3. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М., 2006.

4. Rafael H. Bordini, Mehdi Dastani, Jurgen Dix, Amal El Fallah Seghrouchni. Multi-Agent Programming. Languages, Tools and Applications – Springer Science+Business Media, LLC 2009

THE *OPAL*++ AGENT PROGRAMMING LANGUAGE OF THE VISUALIZATION AND INTEGRATION PLATFORM “OPTIMUS”

D. S. Sobanin, I. A. Kryuchkov, A. V. Ogorodnikov, O. V. Kovalenko

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The development of agents and effective programming of their behavior logics are the key and most complex stages in the development of multiagent simulation models. The paper presents the basic approach and means of the domain-specific Opal++ agent programming language in the “OptIMUS” visualization and integration platform used to resolve the given problems.

Key words: visualization and integration platform “OptIMUS”, multiagent model, agent behavior logics, Opal++ agent programming language.

УДК 539.3
DOI 10.53403/9785951505071_2022_434

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ГАМИЛЬТОНОВОЙ ДИНАМИКИ В ЧИСЛЕННЫХ РАСЧЕТАХ ЗАДАЧ МЕХАНИКИ СПЛОШНОЙ СРЕДЫ

В. Н. Софронов, М. В. Ветчинников, М. А Демина

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе приводится построение дискретной модели численного метода решения задач механики сплошной среды на основе дискретной гамильтоновой динамики. Для численного решения гамильтоновых уравнений используются симплектические разностные схемы. Приведены примеры моделирования бездиссипативных процессов.

Ключевые слова: дискретная гамильтонова динамика, фазовый объем, симплектические разностные схемы, динамические задачи упругости, комплекс программ.

Введение

Гамильтоновы методы, основанные на приведении уравнений к гамильтоновой форме, используются в различных разделах теоретической физики. В задачах механики сплошной среды(МСС) почти всегда присутствуют диссипативные процессы, разрушающие многие свойства гамильтоновых систем (например, сохранение фазового объема), и проблема полной интегрируемости становится почти неразрешимой. Действительно, примеры полной интегрируемости динамических систем малочисленны и является редким исключением из общих правил [1]. В данной статье построение бессеточного численного метода для МСС основано на использовании методов дискретной гамильтоновой динамики [2].

В бессеточных методах (подвижные клеточные автоматы (МСА) [3], перидинамика (PD) [4]–[5], кластерная динамика (КД) [6]) сплошная среда заменяется системой взаимодействующих мезо-частиц. Для методов этого класса не всегда удается сконструировать микроскопические взаимодействия так, чтобы получить заданные макроскопические свойства. Уравнения движения мезо-частиц обычно записаны в интегро-дифференциальной форме и не всегда устанавливается их связь с уравнениями гамильтоновой динамики. Последние обладают множеством интересных свойств. Гамильтоновы системы, кроме стандартных интегралов движения, имеют N дополнительных интегральных инвариантов (N – число частиц). Одним из таких инвариантов является фазовый объем. Естественно, что разностные методы должны наследовать все свойства дифференциальных уравнений. Можно надеяться, что в этом случае будет лучше сохранена структура решения в фазовом пространстве.

Построение дискретной модели, представленной в статье, начинается с задания упругого потенциала (упругой энергии деформирования) системы частиц. Для изотропной среды эта энергия зависит от двух модулей упругости. Таким образом, мы получаем гамильтониан системы с разделяющимися переменными (координатами частиц и их импульсами). При построении дискретной модели используется процедура вычисления тензора дисторсии для системы частиц [7]. В исходном виде модель не предполагает наличия диссипативных процессов и может использоваться только для решения ограниченного круга задач (акустики, сейсмики, динамики гипо-упругих сред). Учет диссипации возможен в рамках диссипативной динамики частиц (dissipative particle dynamics, DPD) [8].

При моделировании упругопластических течений система уравнений дискретной модели должна быть дополнена соотношениями для вычисления тензора пластической дисторсии.

Программная реализация дискретной модели в рамках пакета MoDyS [10] сводится к замене процедуры вычисления сил. Основные счетные модули, программы пре- и постобработки остаются без изменения. Использование дискретного метода (как и любого разностного метода) в практически важных расчетах невозможно без распараллеливания вычислительных процессов.

Несомненным преимуществом реализации нового метода на базе комплекса программ MoDyS является возможность проведения расчетов на новых супер-ЭВМ с использованием интерфейсов MPI, OpenMP и технологии CUDA с высокой (более 80 %) эффективностью распараллеливания.

1. Построение дискретной модели

Классический метод МСС достиг больших успехов в решении прикладных задач. К сожалению, аппарат континуальной механики плохо приспособлен для расчета эффектов, связанных с большими деформациями и нарушением сплошности материала (дробление, пыление, диспергирование). Кроме того, для континуального подхода достаточно острой является проблема построения замыкающих соотношений. В последнее время предпринимаются попытки решить эту проблему с помощью многоуровневого моделирования. Тем не менее, следует признать – в настоящее время нет законченной теории, позволяющей с единых позиций отслеживать эволюцию дефектов на разных уровнях (с сильно различающимися пространственными и временными масштабами).

Промежуточное положение между микроскопическим методом МД и макроскопическим методом механики сплошной среды занимают методы физической мезомеханики (метод клеточных автоматов, метод кластерной динамики (КД), перидинамический метод), основанные на микроскопической динамике имитирующей среды. Трудности такого подхода состоят в том, что требуется конструирование микроскопических взаимодействий, приводящих к заданным макроскопическим свойствам. Метод КД, предложенный Пискуновым В. Н. и Давыдовым И. А. [6] основан на замене уравнений механики сплошной среды на уравнения движения выделенных структурных элементов (кластеров). Динамика взаимодействующих частиц (кластеров) описывается гамильтоновой системой с потенциалом, зависящим только от расстояния между частицами и не зависящим от их взаимного расположения. Выяснилось, что при таком подходе не удается получить упругую среду в наиболее общем представлении, то есть среду с двухпараметрическим произволом.

Рассмотренная в этом разделе дискретная модель является макроскопической и не претендует на многоуровневое моделирование. Уравнения этой модели, как и уравнения МД, могут быть записаны в гамильтоновой форме с отличной от МД (и КД) процедурой вычисления сил.

Предполагается, что каждая частица (например, с номером i) является макрообъектом и наделена параметрами (массой – m_i, плотностью – ρ_i, начальными координатами $\bar{r}_i^0\left(x_i^0, y_i^0, z_i^0\right)$ и компонентами скорости $\bar{u}_i\left(u_i^x, u_i^y, u_i^z\right)$). Кроме того, для каждой частицы заданы какие-либо два модуля упругости (например, K_i – модуль объемного сжатия, G_i – модуль сдвига). Движение деформируемой среды характеризуется дискретным вектором смещений $\bar{w}_i\left(w_i^x, w_i^y, w_i^z\right)$

Пусть N_i – список соседей частицы с номером i. В этот список входят частицы, координаты которых удовлетворяют условию: $\underbrace{\left|\bar{r}_i^0 - \bar{r}_k^0\right|}_{k \in N_i} < \delta_r$ (δ_r - горизонт взаимодействия).

Тензор дисторсии $\left(C^i\right) = \left(\partial\bar{w}/\partial\bar{r}\right)^i$ для частицы i на дискретном множестве частиц N_i может быть определен по методу наименьших квадратов. В этом случае он примет следующий вид [7]:

$$\left(C^i\right)^T = \left(A^i\right)^{-1}\left(B^i\right) \tag{1}$$

Таким образом, для вычисления тензора дисторсии требуется определить элементы двух матриц (симметричной (A^i) и несимметричной (B^i)):

$$\left(A^i\right)=\sum_{k\in N_i}\left(\bar{r}_i^0-\bar{r}_k^0\right)\otimes\left(\bar{r}_i^0-\bar{r}_k^0\right),$$

$$A_{11}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(x_i^0-x_k^0\right),\ A_{12}^i=A_{21}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(y_i^0-y_k^0\right),\ A_{13}^i=A_{31}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(z_i^0-z_k^0\right),$$

$$A_{23}^i=A_{32}^i=\sum_{k\in N_i}\left(y_i^0-y_k^0\right)\left(z_i^0-z_k^0\right),\ A_{22}^i=\sum_{k\in N_i}\left(y_i^0-y_k^0\right)\left(y_i^0-y_k^0\right),\ A_{33}^i=\sum_{k\in N_i}\left(z_i^0-z_k^0\right)\left(z_i^0-z_k^0\right)$$

$$\left(B^i\right)=\sum_{k\in N_i}\left(\bar{r}_i^0-\bar{r}_k^0\right)\otimes\left(\overline{w}_i-\overline{w}_k\right)$$

$$B_{11}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(w_i^x-w_k^x\right),\ B_{12}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(w_i^y-w_k^y\right),\ B_{13}^i=\sum_{k\in N_i}\left(x_i^0-x_k^0\right)\left(w_i^z-w_k^z\right),$$

$$B_{21}^i=\sum_{k\in N_i}\left(y_i^0-y_k^0\right)\left(w_i^x-w_k^x\right),\ B_{22}^i=\sum_{k\in N_i}\left(y_i^0-y_k^0\right)\left(w_i^y-w_k^y\right),\ B_{23}^i=\sum_{k\in N_i}\left(y_i^0-y_k^0\right)\left(w_i^z-w_k^z\right),$$

$$B_{31}^i=\sum_{k\in N_i}\left(z_i^0-z_k^0\right)\left(w_i^x-w_k^x\right),\ B_{32}^i=\sum_{k\in N_i}\left(z_i^0-z_k^0\right)\left(w_i^y-w_k^y\right),\ B_{33}^i=\sum_{k\in N_i}\left(z_i^0-z_k^0\right)\left(w_i^z-w_k^z\right)$$

В (1) $\left(A^i\right)^{-1}$ – матрица обратная к $\left(A^i\right)$

$$\left(A^i\right)^{-1}=\frac{1}{\left|A^i\right|}\begin{pmatrix}A_{22}^iA_{33}^i-A_{23}^iA_{23}^i & A_{13}^iA_{23}^i-A_{12}^iA_{33}^i & A_{12}^iA_{23}^i-A_{13}^iA_{22}^i\\ A_{13}^iA_{23}^i-A_{12}^iA_{33}^i & A_{11}^iA_{33}^i-A_{13}^iA_{13}^i & A_{13}^iA_{12}^i-A_{11}^iA_{23}^i\\ A_{12}^iA_{23}^i-A_{13}^iA_{22}^i & A_{13}^iA_{12}^i-A_{11}^iA_{23}^i & A_{11}^iA_{22}^i-A_{12}^iA_{12}^i\end{pmatrix},$$

$$\left|A^i\right|=A_{11}^iA_{22}^iA_{33}^i+2A_{12}^iA_{13}^iA_{23}^i-A_{11}^iA_{23}^iA_{23}^i-A_{22}^iA_{13}^iA_{13}^i-A_{33}^iA_{12}^iA_{12}^i$$

В условиях однородной деформации вектор смещения имеют вид:

$$\overline{w}=\left(\breve{C}\right)\overline{r},\quad\left(\breve{C}\right)=\left(\partial\overline{w}/\partial\overline{r}\right)\tag{2}$$

Здесь матрица дисторсии $\left(\breve{C}\right)$ не зависит от координат. Покажем, что элементы тензора дисторсии для поля смещений (2), вычисленные с помощью процедуры численного дифференцирования (1), совпадают с элементами матрицы $\left(\breve{C}\right)$.

$$\left(B^i\right)=\sum_{k\in N_i}\left(\bar{r}_i^0-\bar{r}_k^0\right)\otimes\left(\overline{w}_i-\overline{w}_k\right)==\sum_{k\in N_i}\left(\bar{r}_i^0-\bar{r}_k^0\right)\otimes\left(\breve{C}\right)\left(\bar{r}_i^0-\bar{r}_k^0\right)=$$

$$=\left(\sum_{k\in N_i}\left(\bar{r}_i^0-\bar{r}_k^0\right)\otimes\left(\bar{r}_i^0-\bar{r}_k^0\right)\right)\left(\breve{C}\right)^T=\left(A^i\right)\left(\breve{C}\right)^T$$

Таким образом, $\left(C^i\right)^T=\left(A^i\right)^{-1}\left(B^i\right)=\left(\breve{C}\right)^T$. Тензор деформаций $\left(E^i\right)$ является симметричной частью тензора дисторсии $\left(C^i\right)$

$$\left(E^i\right)=0.5\left(\left(C^i\right)^T+\left(C^i\right)\right)$$

Как обычно, тензор деформаций может быть представлен в виде суммы шарового тензора (E_V^i) и девиатора (E_S^i).

$$\left(E^i\right)=\left(E_V^i\right)+\left(E_S^i\right),\quad \left(E_V^i\right)=\frac{1}{3}\left(\delta_{kl}E_{mm}^i\right),\quad E_{mm}^i=\frac{\rho_i^0}{\rho_i}-1$$

Энергия, приходящаяся на одну частицу i, с точностью до членов второго порядка имеет вид [10]:

$$U_i=\frac{\lambda_i}{2}\sum_m E_{mm}^i E_{mm}^i+G_i\sum_{k,l}E_{kl}^i E_{kl}^i=\frac{K_i}{2}Sp^2\left(E_V^i\right)+G_i\left(E_S^i\right):\left(E_S^i\right) \tag{3}$$

$\lambda_i=K_i-\frac{2}{3}G_i$, G_i- коэффициенты Ламэ. Тензор напряжений Пиолы – Кирхгофа $\left(\sigma^i\right)$ определяется дифференцированием потенциала U_i (3)

$$\left(\sigma^i\right)=\frac{\partial U_i}{\partial E_{kl}^i}=K_i\delta_{kl}\sum_m E_{mm}^i+2G_i\left(E_{kl}^i-\frac{1}{3}\delta_{kl}\sum_m E_{mm}^i\right)$$

$$\sigma_{11}^i=\left(K_i-\frac{2G_i}{3}\right)\left(E_{11}^i+E_{22}^i+E_{33}^i\right)+2G_iE_{11}^i,\quad \sigma_{12}^i=\sigma_{21}^i=2G_iE_{12}^i,$$

$$\sigma_{22}^i=\left(K_i-\frac{2G_i}{3}\right)\left(E_{11}^i+E_{22}^i+E_{33}^i\right)+2G_iE_{22}^i,\quad \sigma_{13}^i=\sigma_{31}^i=2G_iE_{13}^i,$$

$$\sigma_{33}^i=\left(K_i-\frac{2G_i}{3}\right)\left(E_{11}^i+E_{22}^i+E_{33}^i\right)+2G_iE_{33}^i,\quad \sigma_{23}^i=\sigma_{32}^i=2G_iE_{23}^i$$

Тензор напряжений Пиолы – Кирхгофа $\left(\sigma^i\right)$ отнесен к системе координат начального (до деформации) состояния. Переход к тензору напряжений

Коши $\left(\Sigma^i\right)$, отнесенного к актуальному (текущему) состоянию осуществляется по формулам [11] – [13]: $\left(\Sigma^i\right)=\frac{\rho_i}{\rho_i^0}\left(\sigma^i\right)\left(I+\left(C^i\right)\right)$.

С учетом (3) гамильтониан системы представим в виде:

$$H=\sum_i\left(\frac{\left|\overline{P_i}\right|^2}{2m_i}+V_i^0U_i\right)=\sum_i\left(\frac{\left|\overline{P_i}\right|^2}{2m_i}+\frac{m_i}{\rho_i^0}U_i\right) \tag{4}$$

Динамические переменные $\left(\overline{r_i},\overline{p_i}\right)$ изменяются во времени согласно уравнениям Гамильтона:

$$\frac{d\overline{r_i}}{dt}=\frac{\partial H}{\partial\overline{p_i}}=\overline{u}_i,\quad \frac{d\overline{p_i}}{dt}=m_i\frac{d\overline{u_i}}{dt}=-\frac{\partial H}{\partial\overline{r_i}}=\overline{F}_i \tag{5}$$

Сила, действующая на частицу, определяется дифференцированием гамильтониана

$$\overline{F_i}=-\frac{\partial H}{\partial\overline{r_i}}=-V_i^0\frac{\partial U_i}{\partial E^i}\frac{\partial E^i}{\partial\overline{r_i}}=-\sum_{k\in N_i}V_i^0\left(\sigma^i\right)\left(A^i\right)^{-1}\left(\overline{\overline{r_i^0}}-\overline{\overline{r_k^0}}\right)=\sum_{k\in N_i}\overline{F_{ik}} \tag{6}$$

В молекулярной (и кластерной) динамике тензор напряжений $\left(S^i\right)$ при заданной силе взаимодействия частиц i, k вычисляется по формуле:

$$\left(S^i\right)V_i=\left(\sum_{k\in N_i}\overline{F_{ik}}\otimes\left(\overline{r}_i-\overline{r}_k\right)\right) \tag{7}$$

Покажем, что так определенный тензор напряжений $\left(S^i\right)$ будет совпадать с тензором Коши. С учетом (6) имеем

$$\left(S^i\right)V_i = \sum_{k\in N_i} V_i^0\left(\sigma^i\right)\left(A^i\right)^{-1}\left(\bar{r}_i^0 - \bar{r}_k^0\right)\otimes\left(\bar{r}_i - \bar{r}_k\right) = V_i^0\left(\sigma^i\right)\left(A^i\right)^{-1}\left(A^i + B^i\right) = V_i^0\left(\sigma^i\right)\left(I + C^i\right) =$$
$$= V_i^0\left(\Sigma^i\right)\rho_i^0/\rho_i = m_i\left(\Sigma^i\right)/\rho_i = \left(\Sigma^i\right)V_i$$

Классическими следствиями уравнений (5) являются законы сохранения энергии, импульса и момента импульса. Гамильтоновы системы, кроме этих стандартных интегралов движения, имеют дополнительные интегральные инварианты. Одним из таких инвариантов является фазовый объем.

Фазовый поток $G: R^{6N}\left(\bar{r}_0, \bar{p}_0\right) \to R^{6N}\left(\bar{r}, \bar{p}\right)$, $\bar{r} = \left(\bar{r}_1, \bar{r}_2, ..., \bar{r}_N\right)$, $\bar{p} = \left(\bar{p}_1, \bar{p}_2, ..., \bar{p}_N\right)$ гамильтоновой системы преобразует ее начальное состояние $\left(\bar{r}_0, \bar{p}_0\right)$ в текущее $\left(\bar{r}(t), \bar{p}(t)\right)$, сохраняя симплектическую структуру на $R^{6N}(\bar{r}, \bar{p})$, заданную 2–формой [14]

$$\omega^2 = \sum_i d\bar{p}_i \Lambda d\bar{r}_i, \quad d\bar{p}_i \Lambda d\bar{r}_i = dp_{ix}\Lambda dx_i + dp_{iy}\Lambda dy_i + dp_{iz}\Lambda dz_i.$$

Следствием сохранения симплектической структуры является инвариантность интегралов $\int\left(\omega^2\right)^L = const$, $\left(\omega^2\right)^L = \omega^2\Lambda\omega^2...\Lambda\omega^2$, $1 \le L \le N$ по соответствующим $2L$ – многообразиям. Конструкция данных инвариантов состоит в последовательном применении операции внешнего умножения 2–форм. Последним в ряду перечисленных инвариантов стоит фазовый объем.

Пусть начальные данные принадлежат некоторой *6N*-мерной области фазового пространства $\left(\bar{r}_0, \bar{p}_0\right) \in \Omega_0$. В соответствии с теоремой Лиувилля [14] фазовый поток *G* сохранит фазовый объем $\int d\bar{r}_0 d\bar{p}_0 = \int d\bar{r} d\bar{p}$.

Критерием симплектичности фазового потока с матрицей Якоби $D = \partial\left(\bar{r}, \bar{p}\right)/\partial\left(\bar{r}_0, \bar{p}_0\right)$ является выполнение матричного соотношения [14]

$$(D)^T JD = J, \; J = \begin{pmatrix} 0 & -E_{3N} \\ E_{3N} & 0 \end{pmatrix}, \tag{8}$$

где E_{3N} – единичная матрица порядка 3*N*.

2. Симплектические разностные схемы численного решения натуральных гамильтоновых систем

Общеизвестны требования, предъявляемые к разностным схемам механики сплошной среды. Эти требования заключаются в выполнении законов сохранения на дискретном уровне [15]. Так как гамильтоновы системы, кроме стандартных интегралов движения, имеют дополнительные интегральные инварианты, то и разностные методы решения таких систем должны наследовать все свойства соответствующих дифференциальных уравнений [16].

В соответствии с (8) одношаговый разностный метод называется симплектическим, если оператор перехода $D^{n+1} = \partial\left(\bar{r}^{n+1}, \bar{p}^{n+1}\right)\Big/\partial\left(\bar{r}^{n}, \bar{p}^{n}\right)$ осуществляет симплектическое преобразование динамических переменных на каждом временном шаге [16]–[17]

$$\left(D^{n+1}\right)^T J D^{n+1} = J \tag{9}$$

Для систем с гамильтонианами общего вида не существует явных симплектических схем [18]. Такие схемы могут быть построены только для натуральных гамильтоновых систем (систем с распадающимися гамильтонианами):

$$H\left(\overline{r},\overline{p}\right) = K\left(\overline{p}\right) + U\left(\overline{r}\right)$$

Построенная нами дискретная модель принадлежит именно к этому классу натуральных гамильтоновых систем. Для численного решения уравнений (4), (5) будем использовать разностные схемы с выполнением условия симплектичности (9). Можно надеяться, что подобные схемы будут лучше воспроизводить структуру решения в фазовом пространстве.

Для m – стадийных разностных схем Рунге–Кутты–Нистрема (RKN):

$$\overline{r}^{n+1} = \overline{r}^n + \Delta t\,\overline{r}\overline{u}^n + \Delta t^2 \sum_{l=1}^{l=m} \beta_l \overline{F}\left(q^l\right), \quad \overline{u}^{n+1} = \overline{u}^n + \Delta t \sum_{l=1}^{l=m} \gamma_l \overline{F}\left(\overline{q}^l\right),$$

$$\overline{q}^l = \overline{r}^n + \Delta t \alpha_l\, \overline{u}^n + \Delta t^2 \sum_{k=1}^{m} a_{lk} \overline{F}\left(q^k\right).$$

условие симплектичности (9) будет выполнено при следующих ограничениях на коэффициенты разностной схемы [16] – [17]:

$$\beta_l = \gamma_l\left(1-\alpha_l\right), \quad l = 1,2,....m,$$

$$\gamma_l a_{lk} - \gamma_k a_{kl} + \gamma_l\gamma_k\left(\alpha_k - \alpha_l\right) = 0, \quad l,k = 1,2,....m$$

Для явных схем матрица a_{lk} является нижне-треугольной. В этом случае связь коэффициентов этой матрицы с параметрами α_l, γ_l определяется однозначно:

$$a_{lk} = \begin{cases} 0, & 1 \le l \le k \le m \\ \gamma_k\left(\alpha_l - \alpha_k\right), & 1 \le k < l \le m \end{cases}.$$

Таким образом, *m*–стадийные симплектические схемы RKN полностью определяется заданием двух векторов $(\alpha_l),\ (\gamma_l), l = 1,2,...,m$ размерности *m*.

Следующим шагом в процессе построения схемы является выполнение условий порядка на коэффициенты схемы. Эти условия получаются приравниванием соответствующих коэффициентов в степенных разложениях точного и численного решений, что и обеспечивает нужный порядок аппроксимации.

В классе одностадийных схем *m=1* существует единственная симплектическая схема второго порядка – схема Verlet [19] с параметрами $\alpha_1 = 0.5, \gamma_1 = 1$. Данная схема получила наибольшее распространение в МД моделировании. Порядок аппроксимации выше второго может быть получен только в том случае, если число стадий увеличивается до трех ($m \ge 3$) [20]. В этом классе существует семь схем четвертого порядка [20 – 21] три из которых – с действительными коэффициентами.

Схема SIA4 [22]:

$$\alpha_1 = \frac{2^{1/3}}{6} + \frac{2^{2/3}}{12} + \frac{1}{3}, \qquad \alpha_2 = \frac{1}{2}, \qquad \alpha_3 = \frac{2}{3} - \frac{2^{1/3}}{6} - \frac{2^{2/3}}{12},$$

$$\gamma_1 = \frac{2^{1/3}}{3} + \frac{2^{2/3}}{6} + \frac{2}{3}, \qquad \gamma_2 = -\frac{1}{3} - \frac{2\cdot 2^{1/3}}{3} - \frac{2^{2/3}}{3}, \quad \gamma_3 = \frac{2^{1/3}}{3} + \frac{2^{2/3}}{6} + \frac{2}{3}$$

Два варианта Схемы RKN4 [20] в исходном и сопряженном (в алгебраическом смысле – с изменением знаков при радикалах) виде:

$$\alpha_1 = \frac{3 \mp \sqrt{3}}{6}, \quad \alpha_2 = \frac{3 \pm \sqrt{3}}{6}, \quad \alpha_3 = \frac{3 \mp \sqrt{3}}{6},$$

$$\gamma_1 = \frac{3 \pm 2\sqrt{3}}{12}, \quad \gamma_2 = \frac{1}{2}, \qquad \gamma_3 = \frac{3 \mp 2\sqrt{3}}{12}$$

Большая часть последующих тестовых расчетов проведена с использованием схемы Verlet.

3. Программная реализация дискретной модели в рамках пакета программ MoDyS

Представленная нами дискретная модель отличается от метода МД процедурой вычисления сил и была реализована в рамках пакета программ молекулярной динамики MoDyS [9]. В пакете программ MoDyS используется сеточный подход и на каждом шаге происходит обновление связных списков, используемых для быстрого поиска взаимодействующих друг с другом частиц. Для дискретной модели отпадает необходимость в обновлении связных списков каждый шаг и достаточно один раз после задания начальной геометрии построить список Верле, который остается неизменным на всем протяжении расчета.

Все программы пре- и постобработки пакета MoDyS остаются практически без каких-либо особых изменения. Для параллельных вычислений необходимы небольшие модификации программ, выполняющих обмены данными между MPI-процессами.

4. Численные расчеты тестовых задач

Расчеты тестовых задач из раздела 4 проведены с использованием стандартной разностной схемы «Крест» [24] и метода, основанного на дискретной гамильтоновой механике.

Дискретная модель является нелокальной теорией, содержащей масштаб длины – горизонт взаимодействия (δ_r). Ответ на вопрос – что происходит с уравнениями дискретной модели при уменьшении масштаба длины, переходят ли эти уравнения в классические уравнения динамической теории упругости? – получен в [5]. Было показано, что если деформации, определяющие модель, и какие-либо неоднородности достаточно гладкие, то тензор напряжений дискретной модели в этом предельном случае $\delta_r \to 0$ переходит в тензор Пиолы – Кирхгофа, а уравнения гамильтоновой механики превращаются в уравнения теории упругости. Соответственно, предельные численные решения этих уравнений (при измельчении сетки для МСС и, соответственно, при увеличении числа частиц для дискретной модели) также будут совпадать. Однако, при конечных размерах сетки и при конечном числе частиц результаты расчетов могут различаться, так как для уравнений механики сплошных сред и уравнений гамильтоновой механики используются совершенно разные численные методы. «Хорошим» качеством симплектических разностных схем является отсутствие диссипации – все характеристические числа матрицы перехода лежат на единичной окружности комплексной плоскости [14].

Для оценки точности численных методов будем использовать достаточно простые задачи, имеющие эталонное решение. Рассмотрим прохождение упругих волн по цепочке частиц и соответственно прохождение волн по тонкому стержню.

Малым параметром в этих расчетах является размер счетной ячейки $\sim L/N$, а также объем, приходящий на одну частицу, и соответствующий этому объему размер частицы $\sim L/N$. Здесь N – число ячеек либо число частиц вдоль одного из направлений.

Во всех расчетах выбрана следующая система единиц [г, см, 10^{-5}сек].

4.1. Прохождение гармонических волн вдоль тонкого стержня

Рассмотрим прохождение упругих волн по цепочке частиц и соответственно прохождение волн по тонкому стержню. Геометрия задачи:

$$0 < x < 2L, \quad -h < y < h, \quad -h < z < h, \quad L = 1см, \quad h = L/N, \quad N = 100.$$

Свойства материала: модуль Юнга $E = 200\text{ГПа}$, коэффициент Пуассона $\nu = 0$, плотность $\rho = 7.85$ г/см^3. Скорости звука:

продольная $C_l = \sqrt{\frac{1}{\rho}\left(K + \frac{4}{3}G\right)} = \sqrt{\frac{E}{\rho}} = 5.0475446\frac{\text{км}}{\text{сек}}$,

поперечная $C_t = \sqrt{\frac{G}{\rho}} = \sqrt{\frac{E}{2\rho}} = 3.569153\frac{\text{км}}{\text{сек}}$

Начальные данные: $w(x, t=0) = \varepsilon L \sin\frac{\pi x}{2L}, \quad \varepsilon = 0.001$.

Граничные условия: $u(x=0, t) = 0$, $u(x=2L, t) = 0$. Эталонное решение имеет вид:

$$w(x,t) = \varepsilon L \sin\frac{\pi x}{2L}\cos\sqrt{\frac{E}{\rho}}\frac{\pi t}{2L}, u(x,t) = \frac{\partial w(x,t)}{\partial t}, \sigma(x,t) = E\frac{\partial w(x,t)}{\partial x}$$

В этой задаче представляет интерес исследование амплитудной и фазовой ошибок разностных схем на достаточно большом временном интервале. Конец счета $t_k = 50 \cdot 10^{-5}$ сек соответствует $\frac{t_k}{4L}\sqrt{\frac{E}{\rho}} \approx 60$ волновым периодам.

Зависимость нормального напряжения от времени для всех вариантов расчета в сечении $x = L$ представлена на рис. 1, 2.

Для метода, основанного на дискретной гамильтоновой механике, амплитуда звуковых возмущений сохраняется на значительных пространственных (либо временных) интервалах. Тем не менее, существует фазовая ошибка (возможно из-за схемы Verlet). В расчетах с использованием разностных схем МСС звуковые возмущения (гладкие и негладкие) быстро затухают и это обстоятельство не позволяет проводить расчеты на больших пространственных (либо временных) интервалах.

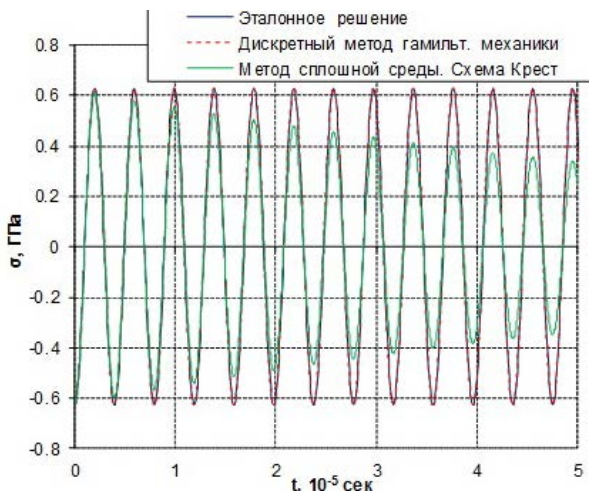


Рис. 1. Зависимость нормального напряжения от времени в сечении $x = L$ для гладких начальных данных (начальный временной интервал). Расчет в рамках механики сплошной среды с числом ячеек $N = 200$ и дискретного метода гамильтоновой механики с числом частиц $N = 200$

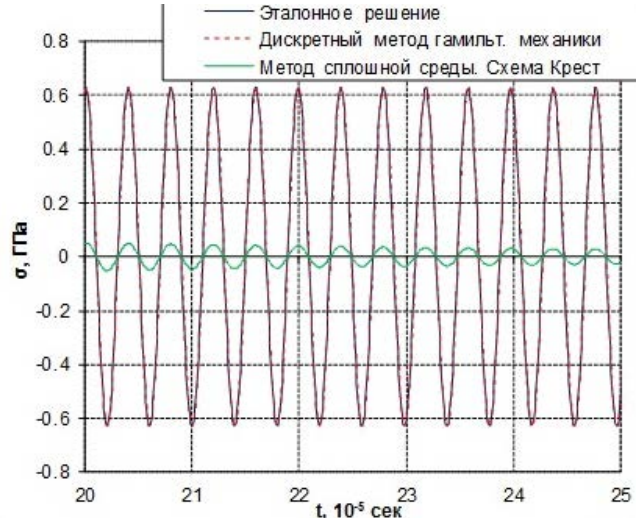


Рис. 2. Зависимость нормального напряжения от времени в сечении $x = L$ для гладких начальных данных (промежуточный временной интервал). Расчет в рамках механики сплошной среды с числом ячеек $N = 200$ и дискретного метода гамильтоновой механики с числом частиц $N = 200$

4.2. Взаимодействие гармонической волны с контактной границей

Важным качеством разностных схем является сохранение точности расчетов вблизи разрывов в решении или в значениях параметров веществ [23]. Рассмотрим задачу о взаимодействии гармонической волны с контактной границей, разделяющей счетную область $0 < x < 2L$ на две половины. Геометрия задачи:

$$0 < x < 2L, \quad -h < y < h, \quad -h < z < h, \quad L = 1см, \quad h = L/N, \quad N = 100 .$$

Выбраны вещества (Fe и Be) с сильно различающимися значениями акустической жесткости $(\rho c)_1/(\rho c)_2 \approx 3$ (см. табл. 1).

Таблица 1

Параметры веществ в задаче о взаимодействии гармонической волны с контактной границей

Fe	0< x >L	ρ_1 = 7.85 г/см^3	ν_1 =0	E_1 = 200 ГПа	$c_1=\sqrt{E_1/\rho_1}$=5.0475 км/сек
Be	L< x <2L	ρ_2 = 1.85 г/см^3	ν_2 =0	E_2=106.13 ГПа	$c_2=\sqrt{E_2/\rho_2}$=7.5741 км/сек

Начальные условия:

$$w(x,t=0) = \varepsilon \begin{vmatrix} \sin\dfrac{8\pi x}{L}, & 0 < x < L \\ 0, & L < x < 2L \end{vmatrix}, \quad \varepsilon = 0.001, \quad \rho = \begin{vmatrix} \rho_1, & 0 < x < L \\ \rho_2, & L < x < 2L \end{vmatrix},$$

$$u(x,t=0) = 0 .$$

Граничные условия:

$$u(x=0,t) = 0 , \quad u(x=2L,t) = 0 .$$

Эталонное решение получено суперпозицией падающей, отраженной и преломленной волн [23].

Профиль нормального напряжения от нормированный массы в момент времени $t = 0.1\cdot 10^{-5}$сек представлен на рис. 3. Расчеты, проведенные в рамках гамильтоновой механики, с хорошей точностью

воспроизводят амплитуды и длины отраженной и преломленной волн. Вместе с тем, в этих расчетах отсутствует счетная диссипация, что приводит к появлению слабой немонотонности вблизи разрывов.

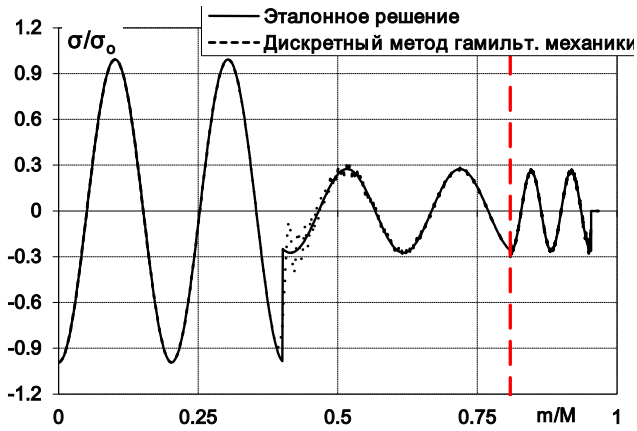


Рис. 3. Профиль нормального напряжения в задаче о взаимодействии гармонической волны с контактной границей Fe-Be. Расчет в рамках гамильтоновой механики с числом частиц $N = 2000$. Пунктирной линей обозначено положение контактной границы

4.3. Сферически – симметричная расходящаяся упругая волна

В однородной упругой среде с параметрами Ламэ λ, μ на сфере радиуса r_0 задан источник $w\big|_{r=r_0} = f(t)$. Рассматривается распространение расходящейся упругой волны.

Счетная область ($r_0 < |x| < r_1, r_0 < |y| < r_1, r_0 < |z| < r_1$), $r_0 = 1$см, $r_1 = 40$см. Граничные условия ставятся на гранях внутреннего куба: ($|x| = r_0, |y| = r_0, |z| = r_0$):

$$w_x(x,y,z,t) = \frac{x}{r} w(r,t),\ w_y(x,y,z,t) = \frac{y}{r} w(r,t),\ w_z(x,y,z,t) = \frac{z}{r} w(r,t).$$

Параметры источника: $T = 1 \cdot 10^{-5}$ сек, $w_0 = 0.01$ см.

Свойства материала: $\rho_0 = 7.85 \text{г}/\text{см}^3$, $C_0 = 5$км/сек, $\nu = 0.33$,

$$\mu = \frac{3K(1-2\nu)}{2(1+\nu)} = \frac{3\rho_0 C_0^2 (1-2\nu)}{2(1+\nu)} = 75.253759 \text{ ГПа}$$

$$\lambda = K - \frac{2\mu}{3} = \frac{3K\nu}{1+\nu} = \frac{3\rho_0 C_0^2 \nu}{1+\nu} = 146.080827 \text{ ГПа}$$

$$C_l = \sqrt{\frac{\lambda + 2\mu}{\rho_0}} = \sqrt{\frac{3K(1-\nu)}{\rho_0(1+\nu)}} = C_0 \sqrt{\frac{3(1-\nu)}{1+\nu}} = 6.146703 \text{ км/с}$$

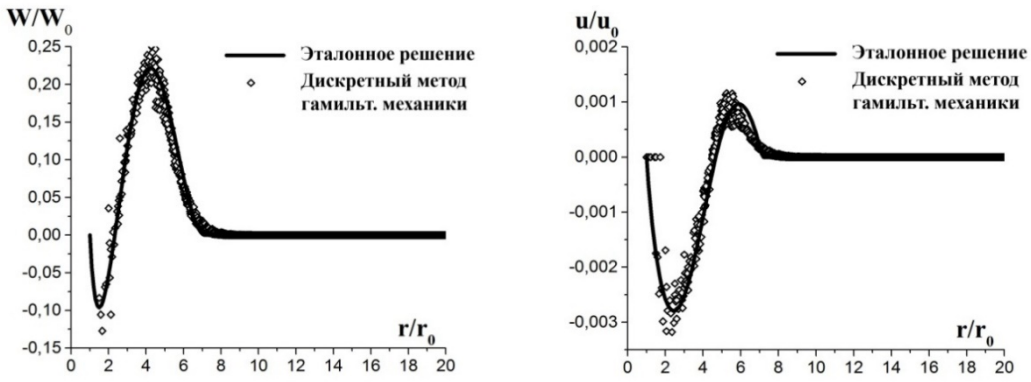
Конец счета $t = 3 \cdot 10^{-5}$с

Профили радиальных компонент смещения и скорости приведены на рис. 4–6. В силу сферической симметрии все параметры течения, зависят от одной пространственной координаты $r = \sqrt{x^2 + y^2 + z^2}$, поэтому на приведенных графиках нанесены значения сеточных функций для всех частиц.

Растровые картины в естественном масштабе и шлирен-визуализация (в нелинейной шкале) изображены на рис. 7–8. Оттенки цветов в шлирен-визуализации соответствуют величине:

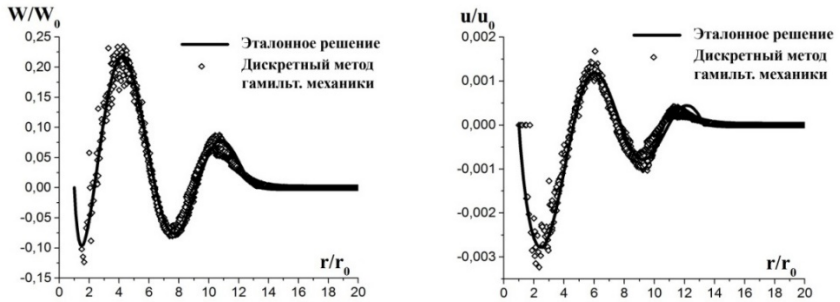
$$S = 0.8\exp\left(-5\left(\frac{\left|\frac{\partial W}{\partial r}+2\frac{W}{r}\right|}{\left|\frac{\partial W}{\partial r}+2\frac{W}{r}\right|_{\max}}\right)\right)$$

Отметим хорошее сохранение сферической симметрии в представленных расчетах.



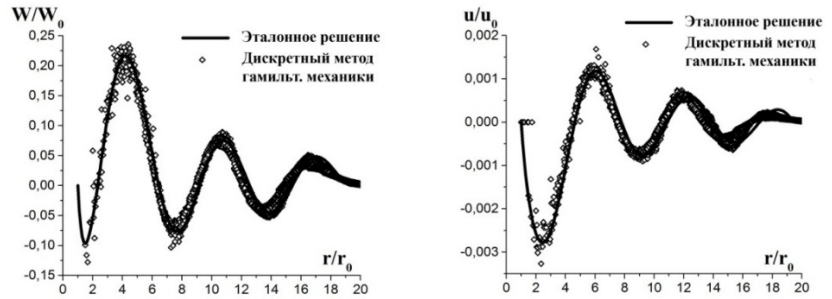
a b

Рис. 4. Профили смещения (a) и скорости (b) в момент времени $t = 1\cdot 10^{-5}$ сек . $u_0=w_0\omega$



a b

Рис. 5. Профили смещения (a) и скорости (b) в момент времени $t = 2\cdot 10^{-5}$ сек . $u_0=w_0\omega$



a b

Рис. 6. Профили смещения (a) и скорости (b) в момент времени $t = 3\cdot 10^{-5}$ сек . $u_0=w_0\omega$

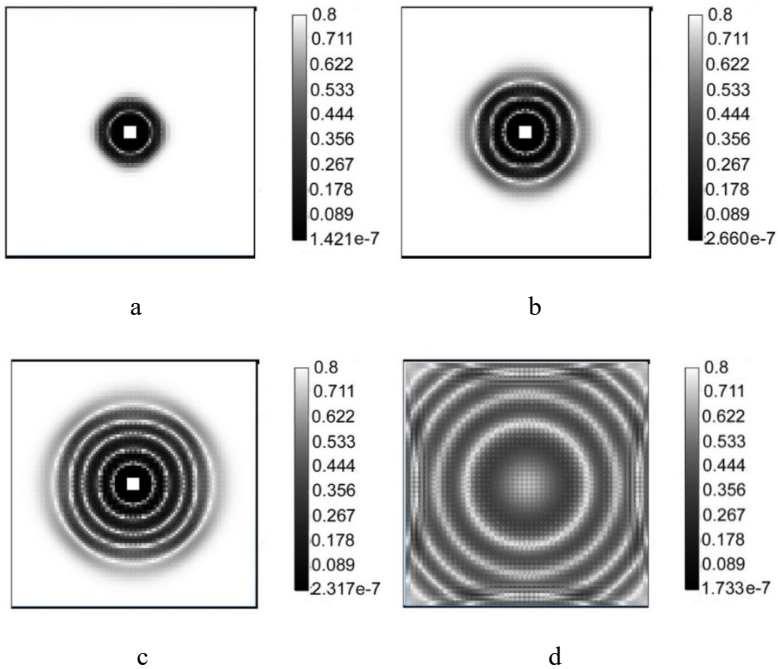


Рис. 7. Растровая картина модуля скорости (шлирен – визуализация в плоскости z = 0) в задаче со сферически-симметричной расходящейся упругой волной. Расчет в рамках гамильтоновой механики с числом частиц $N = 1030274$ на времена: a – $t = 1 \cdot 10^{-5}$ c, b – $t = 2 \cdot 10^{-5}$ c, c – $t = 3 \cdot 10^{-5}$ c, d – $t = 7 \cdot 10^{-5}$ c

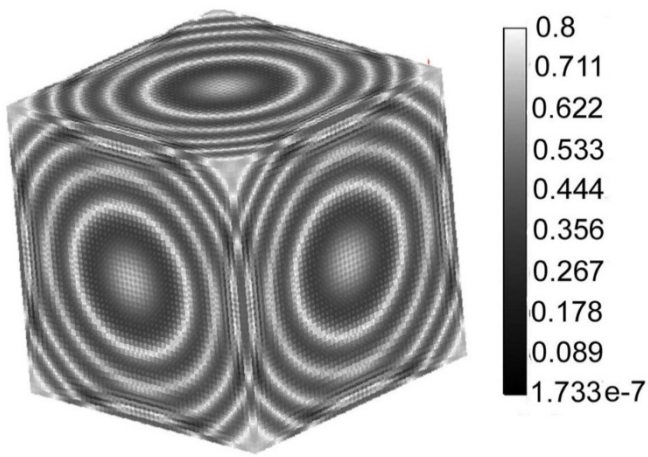


Рис. 8. Растровая картина модуля скорости (шлирен – визуализация на внешней поверхности куба) в задаче со сферически-симметричной расходящейся упругой волной. Расчет в рамках гамильтоновой механики на момент времени $t = 7 \cdot 10^{-5}$ c

4.4 Изгибные волны на круглой пластине с эффектом «шепчущей» галереи

В теории изгиба полу-бесконечных пластин хорошо изучено распространение вдоль прямолинейных границ локализованных волн Релея. Обобщение этих результатов на пластины с криволинейными границами позволило исследовать другой тип поверхностных – волн с эффектом «шепчущей галереи» [25]. Такие волны исследованы в акустике. Эффект шепчущей галереи возникает при движении звуковых волн вдоль изогнутых границ.

Рассматриваются изгибные колебания тонкой круглой пластины в рамках теории Кирхгофа:

$$\frac{3(1-\nu^2)\rho}{Eh^2}\frac{d^2w}{dt^2} = -\Delta\Delta w$$

Решение представлено в виде гармонической изгибной волны, распространяющейся вдоль границы пластины

$$w(r,\theta,t) = w_r(r)\exp(i(n\theta - \omega t)) \qquad \omega = \frac{k^2n^2}{R^2}\sqrt{\frac{Eh^2}{3(1-\nu^2)\rho}} \qquad T = \frac{2\pi}{\omega}$$

Дисперсионное уравнение связывает частотный параметр k с номером гармоники n. Эффект шепчущей галереи возникает для изгибных волн с малым волновым числом n, так как такие волны имеют возможность многократно отражаться от границы пластины. Мы ограничились рассмотрением изгибных волн со значениями чисел n = 3,5,7. Расчеты проведены в трехмерной постановке. Начальные данные соответствуют эталонному решению. Мы выбрали вторую форму третьей и седьмой гармоник и первую форму пятой гармоники.

Начальные данные

$$\rho = 7.85 \quad E = 200 \quad \nu = 0.333 \quad R = 1 \qquad h = R/N_x \qquad w = 0.1h$$

$$0 < r < R, -h < z < h, 0 < \theta < 2\pi$$

$$\frac{w_x(r,\theta,z,t=0)}{w_0} = -z\frac{\partial w_z}{\partial x} = -\frac{zx}{r}\frac{dw_r(r)}{dr}\sin(n\theta) + \frac{zy}{r}w_r(r)n\cos(n\theta)$$

$$\frac{w_y(r,\theta,z,t=0)}{w_0} = -z\frac{\partial w_z}{\partial x} = -\frac{zy}{r}\frac{dw_r(r)}{dr}\sin(n\theta) - \frac{zx}{r}w_r(r)n\cos(n\theta)$$

$$\frac{w_z(r,\theta,z,t=0)}{w_0} = w_r(r)\sin(n\theta)$$

$$\frac{u_x(r,\theta,z,t=0)}{w_0\omega} = -z\frac{\partial u_z}{\partial x} = -\frac{zx}{r}\frac{dw_r(r)}{dr}\cos(n\theta) - \frac{zy}{r}w_r(r)n\sin(n\theta)$$

$$\frac{u_y(r,\theta,z,t=0)}{w_0\omega} = -z\frac{\partial u_z}{\partial x} = -\frac{zy}{r}\frac{dw_r(r)}{dr}\cos(n\theta) + \frac{zx}{r}w_r(r)n\sin(n\theta)$$

$$\frac{u_z(r,\theta,z,t=0)}{w_0\omega} = w_r(r)\cos(n\theta)$$

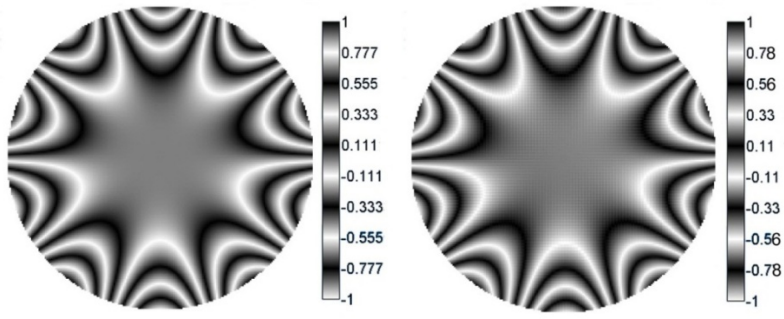
$$\theta = \begin{cases} ar\cos(x/r) & y > 0 \\ \pi + ar\cos(x/r) & y < 0 \end{cases}$$

Конец счета $t = 1.25T$

Эталонное решение

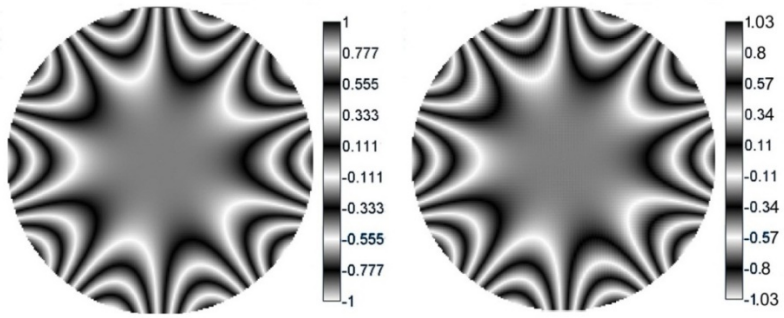
$$\frac{w_z(r,\theta,z,t)}{w_0} = w_r(r)\sin(n\theta + \omega t), \qquad \frac{u_z}{w_0\omega}(r,\theta,z,t) = w_r(r)\cos(n\theta + \omega t)$$

Растровые картины вертикальных смещений в линейной шкале (фигуры Хладни) изображены на рис. 9–14. Фигуры Хладни вращаются вокруг оси z с постоянной угловой скоростью ω. Результаты расчетов соответствуют эталонному решению.



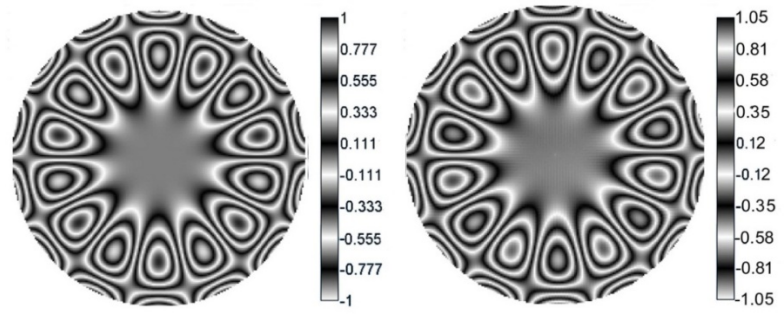
a b

Рис. 9. Растровая картина вертикальных смещений (w_z/w_0) для гармонической волны (n = 5) в моменты времени t = 0 (a) и t = T (b). Расчет в рамках гамильтоновой механики



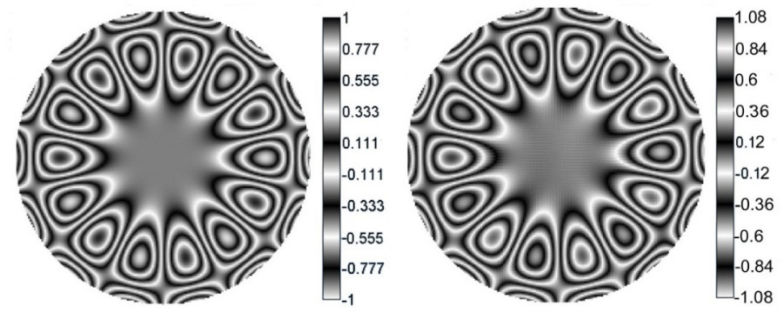
a b

Рис. 10. Растровая картина вертикальных смещений (w_z/w_0) для гармонической волны (n = 5) в момент времени t = 5T/4. (a) – эталонное решение, (b) – расчет в рамках гамильтоновой механики



a b

Рис. 11. Растровая картина вертикальных смещений (w_z/w_0) для гармонической волны (n = 7) в моменты времени t = 0 (a) и t = T (b). Расчет в рамках гамильтоновой механики



a b

Рис. 12. Растровая картина вертикальных смещений (w_z/w_0) для гармонической волны (n = 7) в момент времени t = 5T/4. (a) – эталонное решение, (b) – расчет в рамках гамильтоновой механики

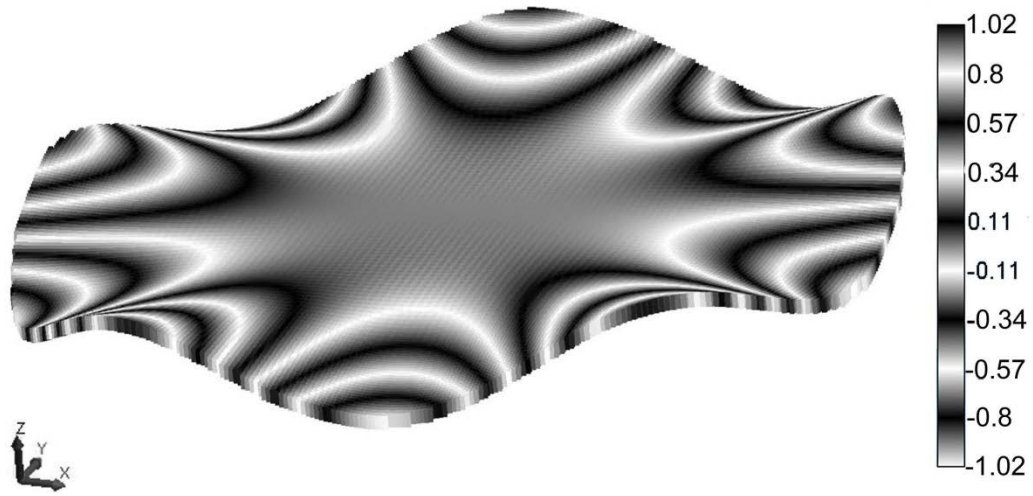


Рис. 13. Форма пластины для гармонической волны *(n*=5) на время *t*=*T*. Вертикальные размеры увеличены в 80 раз. Расчет в рамках гамильтоновой механики

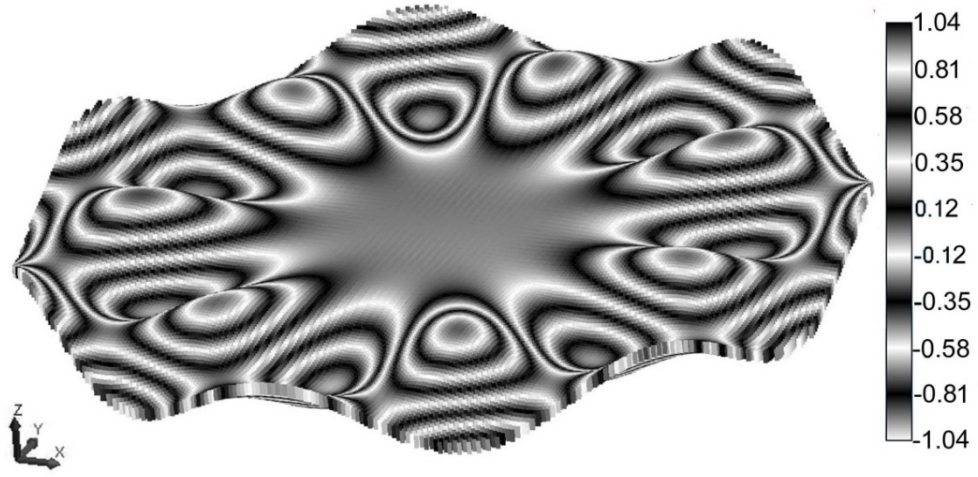


Рис. 14. Форма пластины для гармонической волны *(n* = 7) на время *t* = *T*. Вертикальные размеры увеличены в 80 раз. Расчет в рамках гамильтоновой механики

Заключение

В данном докладе приводится построение и реализация в рамках комплекса молекулярной динамики бессеточного численного метода решения задач механики сплошной среды на основе дискретной гамильтоновой динамики. В исходном виде метод не предполагает наличия диссипативных процессов и может использоваться только для решения ограниченного круга задач (акустики, сейсмики, динамики гипо-упругих сред). Учет диссипации возможен в рамках диссипативной динамики частиц (DPD) [26]. При моделировании упруго-пластических течений система уравнений дискретной модели должна быть дополнена соотношениями для вычисления тензора пластической дисторсии.

Несомненными достоинствами представленного нами метода являются:

1) отсутствие неопределенностей, возникающих при решении задач, содержащих разрывы и нарушение сплошности;

2) отсутствие проблем, связанных с генерацией высококачественных расчетных сеток, так как метод является бессеточным;

3) отсутствие этапов, связанных с корректировкой и перестроением расчетных сеток в результате сильных деформаций и изменения геометрии;

4) наличие независимых арифметических вычислений, позволяющих рассчитывать на высокую эффективность использования GPU.

Литература

1. Фадеев Л. Д. Новая жизнь полной интегрируемости // Успехи физических наук. 2013. Т.183, №5. С.487–495.

2. Sofronov V. N., Shemarulin V. E., Vetchinnikov M. V., High-Order Symplectic Difference Schemes for Numerical Solution of Natural Hamiltonian System. Applications to Molecular Dynamics and Continuum Mechanics Problems// Horizons in World Physics. New York: Nova Science. 2019. V. 301. P.1–79.

3. Псахье С. Г., Остермайер Г. П., Дмитриев А. И., Шилько Е. В., Смолин А. Ю., Коростелев С. Ю. Метод подвижных клеточных автоматов как новое направление дискретной вычислительной механики. I.Теоретическое описание // Физ. Мезомех. 2000. Т.3, №2. С. 5–13.

4. Silling S. A., Epton M., Weckner O., Xu J., Ascory E. Peridynamic States and Constitutive Modeling // J. Elastisity. 2007. V.88. P.151–184.

5. Silling S. A., Lechoucq R. B. Convergence of Peridynamics to Classical Elasticity Theory // SAND 2007–7242.

6. Пискунов В. Н., Давыдов И. А. Метод кластерной динамики и его применение в задачах импульсного нагружения твердых тел // Деформация и разрушение материалов. 2008. № 1.

7. Надарейшвили А. И., Петушков В. А. Моделирование нелинейной динамики гамильтоновых систем в биомеханике с использованием образов компьютерной томографии // Математическое моделирование. 2014. Т.26, №1. С. 109–121.

8. Франкель Д., Смит Б. Принципы компьютерного моделирования молекулярных систем. М.: Научный мир, 2013.

9. Анисимов А. Н., Грушин С. А., Воронин Б. Л., Копкин С. В., Ерофеев А. М., Демин Д. А., Демина М. А., Здорова М. В., Ветчинников М. В., Еричева Н. С., Коваленко Н. О., Крючков И. А., Кечин А. Г., Дегтярев В. А., Урм В. Я. Комплекс программ молекулярно-динамического моделирования(MoDyS). Свидетельство о государственной регистрации программы для ЭВМ № 2010614974. 2010.

10. Ландау Л. Д., Лифшиц Е. М. Теория упругости. Теоретическая физика. Т. VII. М.: Наука, 1987.

11. Куликовский А. Г., Свешникова Е. И. Нелинейные волны в упругих средах. М.: Московский лицей, 1998.

12. Блэнд Д. Р. Нелинейная динамическая теория упругости. М.: Мир, 1972.

13. Седов Л. И. Механика сплошной среды. М.: Наука, 1994.

14. Арнольд В. И. Математические методы классической механики. М.: Наука, 1979.

15. Самарский А. А., Попов Ю. А. Разностные методы решения задач газовой динамики. М.: Наука, 1992.

16. Сурис Ю. Б. О сохранении симплектической структуры при численном решении гамильтоновых систем. // Численное решение обыкновенных дифференциальных уравнений. М.: ИПМ. АН ССС, 1988. С. 148–160.

17. Sanz-Serna J.M. Runge-Kutta schemes for Hamiltonian system // BIT. 1988. V. 22, № 4.

18. Еленин Г. Г., Шляхов П. И. Геометрическая структура пространства параметров трехстадийных симплектических методов Рунге-Кутты // Математическое моделирование. 2011. Т. 23, № 5. С. 16–34.

19. Verlet L. Computer “experiments” on classical fluids. Thermodynamical properties of Lennard – Jones molecules // Phys. Rev. 1967. Vol. 159. № 1. P. 98-103.

20. Софронов В. Н., Шемарулин В. Е. Классификация явных трехстадийных симплектических разностных схем численного решения натуральных гамильтоновых систем. Сравнительное исследование схем высокого порядка на задачах молекулярной динамики // Журнал вычисл. мат. и мат. физ. 2016. Т. 56, № 45. С. 551–571.

21. Shemarulin V. E., Sofronov V. N. Three – stage symplectic difference schemes for molecular dynamics. Description, classification, analytical and numerical investigation // Proceedings of the 10th Seminar on New Models and Hydro-codes for Shock Wave Processes on Condensed Matter. Pardubice. Czech Republic. July 27 – August 1. 2014. P. 125–137.

22. Candy J., Rozmus W. A symplectic integration algorithm for separable Hamiltonian function // Journal of Computational Physics. 1991. Vol. 92. P. 230–256.

23. Бондаренко Ю. А., Софронов В. Н. Взаимодействие звука с границей разных веществ в разностных схемах одномерной лагранжевой газовой динамики. Зависимость точности счета от способов согласования сетки // ВАНТ. Сер. Математическое моделирование физических процессов. 2015. Вып. 2. С. 12–27.

24. Бондаренко Ю. А. Консервативное расщепление уравнения энергии в разностных схемах типа «Крест» для лагранжевой газодинамики // Журнал вычислительной математики и математической физики. 1997. Т.37, №8. С. 1020–1023.

25. Украинский Д. В. О типе изгибной краевой волны на круглой пластине // Известия РАН. Сер. Механика твердого тела. 2018. Т. 5. С. 29–38.

26. Остермайер Г. П., Попов В. Л. Многочастичные неравновесные потенциалы взаимодействия в методе мезочастиц // Физ. мезомех. 1999. Т.6, № 2.

USE OF HAMILTONIAN DYNAMICS METHODS IN NUMERICAL SIMULATION OF CONTINUUM MECHANICS PROBLEMS

V. N. Sofronov, M. V. Vetchinnikov, M. A. Dyemina

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes the construction of a discrete model of the numerical scheme used to solve the continuum mechanics problems, which is based on the discrete Hamiltonian dynamics. Symplectic difference schemes are used to solve Hamilton's equations numerically. Examples of simulating dissipationless processes are presented.

Key words: discrete Hamiltonian dynamics, phase volume, symplectic difference schemes, dynamic strength problems, software package.

УДК 004.272.44:004.382.2+535.2
DOI 10.53403/9785951505071_2022_451

ФОТОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА. ОПТИЧЕСКИЕ ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

С. А. Степаненко

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Предложены структуры интерференционных логических элементов, образующих полный функциональный базис. Выполняются требования идентичности значений интенсивности, соответствующих логическим константам «0» и «1», вырабатываемых различными элементами в заданные фиксированные интервалы времени.

Масштабирование значений интенсивности логически констант достигается увеличением количества уровней волноводов и сопровождается пропорциональным увеличением длительности выполнения операций.

На примере цифровой фотонной вычислительной машины показаны целесообразность и перспективность применения предложенных элементов.

Ключевые слова: фотонная машина, оптичейкие логический элемент, волновод.

Введение

Возможность и целесообразность реализации фотонной вычислительной машины [1] определяются, в частности, параметрами оптических логических элементов (ОЛЭ).

Активные ОЛЭ из известных материалов, задействующие нелинейные эффекты, уступают по производительности и энергоэффективности современным электронным элементам [2].

Пассивные ОЛЭ [3–7] имеют малые длительности выполнения операции, однако характеризуются большими потерями энергии и трудностями сохранения параметров сигнала.

Предлагаемые ниже интерференционные ОЛЭ задействуют различие значений интенсивности света, возникающее в результате отсутствия или наличия интерференции световых волн и их взаимодействия с волноводами. Показано применение этих ОЛЭ для создания фотонной вычислительной машины и других, конкурентных с электронными аналогами цифровых устройств.

1. Идентификация логических констант

ОЛЭ выполняют логические операции над световыми импульсами, образованными электромагнитными волнами. В результате операций получаются логические константы – логический нуль («0») и логическая единица («1»). Они идентифицируются интенсивностью I электромагнитной волны [8], распространяемой в оптическом волноводе [9].

Электромагнитная волна – линейно поляризованная монохроматическая [8], далее – бегущая волна, представляется в форме светового импульса длительностью $\tau = m\lambda/\upsilon$, где m – целое число, λ – длина волны [8], υ – скорость света в волноводе. Величина $d = m\lambda$ называется размером импульса.

Значение интенсивности выходного сигнала ОЛЭ вырабатывается либо при отсутствии входных импульсов – в ОЛЭ не задействуется никаких эффектов, либо в случае одного входного импульса –

задействуются эффекты, порожденные бегущей волной, либо в случае двух входных встречных импульсов – задействуются эффекты, порожденные стоячей волной [8].

Чтобы выполнять логические операции необходимо перераспределять интенсивность в волноводах. Для этого применяются щели. Щелью [8] называется часть поверхности волновода, через которую излучение передается в другой волновод или в окружающую среду. Для отвода энергии из щели используется однонаправленный типа (1×2) ответвитель [9] с заданным коэффициентом ответвления $0 < q < 1$, означающим долю интенсивности, отводимую через щель в ответвительный волновод.

Остаток энергии в волноводе, имеющем m-щелей, после прохождения бегущей волны пропорционален величине $I_m = I_0(1-q)^m$, где I_0 – интенсивность волны на входе в волновод.

Согласно закону сохранения, количество энергии, выделенной из волновода в результате взаимодействия бегущей волны с m щелями, пропорционально величине $\overset{\Delta}{I}_m = I_0 - I_0(1-q)^m$.

В результате столкновения когерентных импульсов длительностью τ образуется стоячая волна с $2m$ пучностями на расстоянии $\lambda/2$ одна от другой и $2m$ узлами, также удаленными на $\lambda/2$ один от другого [8]. Если щели волновода расположены над пучностями, то суммарная интенсивность на обоих выходах волновода составит $I_\Sigma = 4I_0(1-q)^{2m}$; при этом, суммарная интенсивность, выделяемая в $2m$ щелей волновода, равна $\overset{\Delta}{I}_{2m} = 2I_0 - 4I_0(1-q)^{2m}$. Если щели волновода расположены над узлами, то $I_\Sigma = 2I_0(1-q)^m$; при этом, $\overset{\Delta}{I}_{2m} = 2I_0 - 2I_0(1-q)^m$.

Элементы, в волноводах которых щели расположены над пучностями либо над узлами стоячей волны, называем ОЛЭ первого типа и соответственно ОЛЭ второго типа.

2. Функционально полный базис

Применим рассмотренные эффекты для создания функционально полного базиса ОЛЭ из элементов «И», «исключающее ИЛИ» и «НЕ», реализующих функции &, $\oplus$ и $^-$ [11].

В рассматриваемых далее ОЛЭ используется волноводная структура, показанная на рис. 1.

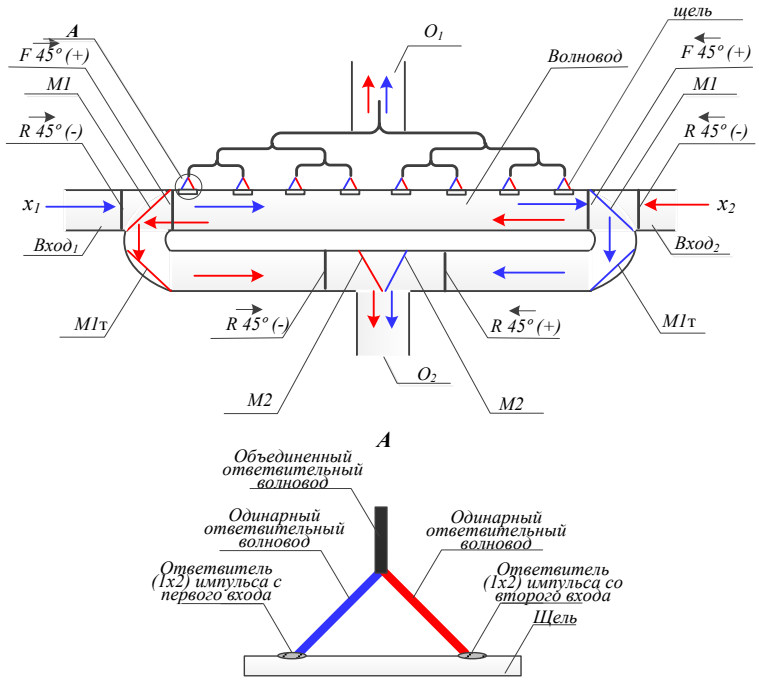


Рис. 1. Волноводная структура. Элементы конструкции

Назначение ее элементов охарактеризуем согласно [12].

Поступающий в волновод на вход 1 импульс с поляризацией 0º изменяет после левостороннего вращателя $R\overrightarrow{45}(-)$ плоскость поляризации на 45º по правилу левого винта и через поляризационное зеркало $M1_1$ поступает в волновод. После фарадеевского вращателя $F\overrightarrow{45}(+)$ он восстанавливает плоскость поляризации и взаимодействует со щелями и (при наличии) со встречным импульсом, поступившим после аналогичных операций со входа 2. Оба импульса когерентны, имеют одинаковую поляризацию. В результате их столкновения образуются пучности и узлы. Над пучностями (либо над узлами) в той же плоскости расположены щели и двунаправленные ответвительные волноводы. Устройство такого волновода показано на вставке к рис. 1.

После взаимодействия со щелями оставшаяся часть импульса со входа 1 поступает на фарадеевский вращатель $F\overleftarrow{45}(-)$, меняет плоскость поляризации и отражается поляризационными зеркалами $M1_1$, $M1_2$ на вращатель $R\overleftarrow{45}(+)$, восстанавливающий исходную поляризацию. Далее отраженный зеркалом М2 импульс поступает на выход O_1, где объединяется (при наличии) с импульсом со входа 2, над которым выполнялись такие же операции.

На выход O_2 поступает энергия из ответвительных волноводов, получаемая взаимодействием импульсов со щелями волновода.

2.1. Элементы первого типа

ЭЛЕМЕНТ «И»

Пусть в одном из направлений, обозначенных на рис. 1 символами $\boldsymbol{x_1}$ и $\boldsymbol{x_2}$, распространяется импульс размером $\boldsymbol{m\lambda}$ (это означает $\boldsymbol{x_2 = 1,\ x_1 = 0}$ или $\boldsymbol{x_2 = 0,\ x_1 = 1}$). На выходе $\boldsymbol{O_1}$ элемента имеем $\boldsymbol{I_{\&} = I_0(1 - q_1)^{2m}}$; это соответствует $\boldsymbol{f_{\&}(x_2, x_1) = 0}$.

Если импульсы распространяются в обоих направлениях (это означает $x_2 = 1,\ x_1 = 1$), то образуется стоячая волна. На выходе O_1 имеем $I_{\&} = 4I_0(1 - q_1)^{2m}$; это соответствует $f_{\&}(x_2, x_1) = 1$.

ЭЛЕМЕНТ «ИСКЛЮЧАЮЩЕЕ ИЛИ»

Выход элемента обозначен $\boldsymbol{O_2}$, он объединяет волноводы от 2m щелей над пучностями стоячей волны. В случае одного импульса на $\boldsymbol{O_2}$ имеем $\boldsymbol{I_{\oplus} = I_0 - I_0(1 - q_2)^{2m}}$, что соответствует $\boldsymbol{f_{\oplus}(x_2, x_1) = 1}$. Для двух встречных импульсов на $\boldsymbol{O_2}$ имеем, $\boldsymbol{I_{\oplus} = 2I_0 - 4I_0(1 - q_2)^{2m}}$, что соответствует $\boldsymbol{f_{\oplus}(x_2, x_1) = 0}$.

ЭЛЕМЕНТ «НЕ»

Реализация функции $^{-}$ («НЕ») достигается подачей на один из входов элемента $\oplus$ единичного сигнала – логической единицы.

Значения функций $f_{\&}(x_2, x_1) = x_2 \& x_1$, $f_{\oplus}(x_2, x_1) = x_2 \oplus x_1$ и соответствующие значения $I_{\&}$, $I_{\oplus}$ представлены в табл. 1.

Таблица 1

Значения $f_{\&}(x_2, x_1)$, $f_{\oplus}(x_2, x_1)$ и $I_{\&}$, $I_{\oplus}$. Элементы первого типа

x_2	x_1	$f_{\&}$	$I_{\&}$	$f_{\oplus}$	$I_{\oplus}$
0	0	0	0	0	0
0	1	0	$I_0(1 - q_1)^{2m}$	1	$I_0 - I_0(1 - q_2)^{2m}$
1	0	0	$I_0(1 - q_1)^{2m}$	1	$I_0 - I_0(1 - q_2)^{2m}$
1	1	1	$4I_0(1 - q_1)^{2m}$	0	$2I_0 - 4I_0(1 - q_2)^{2m}$

ИДЕНТИЧНОСТЬ ЗНАЧЕНИЙ ИНТЕНСИВНОСТИ

Для корректного функционирования устройств из ОЛЭ необходимо, чтобы одинаковым логическим значениям, получаемым через одинаковые временные интервалы, на выходах ОЛЭ, реализующих &, $\oplus$ и $\bar{}$, соответствовали одинаковые интенсивности $\boldsymbol{I}_{\&}$, $\boldsymbol{I}_{\oplus}$. Необходимо выполнение:

$$\begin{cases} 4I_0(1-q_1)^{2m} = I_0 - I_0(1-q_2)^{2m}, \\ I_0(1-q_1)^{2m} = 2I_0 - 4I_0(1-q_2)^{2m}. \end{cases}$$

Решением этой системы уравнений является:

$$(1-q_1)^{2m} = 0{,}133 \text{ и } (1-q_2)^{2m} = 0{,}466.$$

Значения m задаются требуемой длительностью операции.

Получен функционально полный базис интерференционных ОЛЭ первого типа.

2.2. Элементы второго типа

Структура ОЛЭ второго типа показана на рис. 2, выходы обозначены $\boldsymbol{O_1}$ и $\boldsymbol{O_2}$. Перед каждым входным вращателем устроены $(\boldsymbol{2m} - \frac{\boldsymbol{k_1}}{\boldsymbol{2}})$ щелей. Между вращателями устроены $\boldsymbol{k_1 < 2m}$ щелей, расположенные над узлами. Остальные конструкционные элементы совпадают с указанными на рис. 1.

ЭЛЕМЕНТ «И»

Одиночный импульс, означающий $x_2 = 1, x_1 = 0$ или $x_2 = 0, x_1 = 1$, взаимодействует с $2m$ щелями. На выходе O_1 имеем $I_{\&} = I_0(1-q_1)^{2m}$.

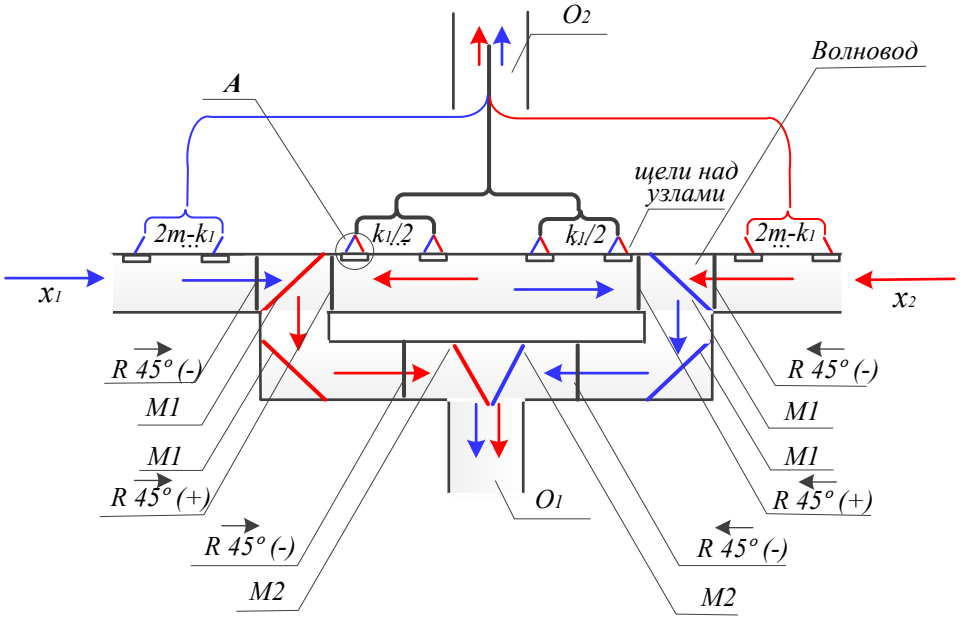


Рис. 2. Структура ОЛЭ второго типа

Если импульсы распространяются в обоих направлениях (т. е. $x_2 = 1, x_1 = 1$), то перед столкновением импульсов и образованием стоячей волны интенсивность в каждом направлении равна $I_0(1-q_1)^{2m-k_1/2}$. После столкновения интенсивность в щели не выделяется, так как щели расположены над узлами. На выходе O_1 имеем $I_{\&} = 2I_0(1-q_1)^{2m-k_1/2}$.

ЭЛЕМЕНТ «ИСКЛЮЧАЮЩЕЕ ИЛИ»

Выход элемента обозначен O_2, он объединяет ответвительные волноводы. В случае одного импульса имеем $I_{\oplus} = I_0 - I_0(1-q_2)^{2m}$, что соответствует $f_{\oplus}(x_2, x_1) = 1$. Для двух встречных импульсов имеем, $I_{\oplus} = 2I_0 - 2I_0(1-q_2)^{2m}$, что соответствует $f_{\oplus}(x_2, x_1) = 0$.

ЭЛЕМЕНТ «НЕ»

Реализация функции $\bar{}$ («НЕ») достигается подачей на один из входов элемента $\oplus$ постоянного единичного сигнала – логической единицы.

Значения функций $f_{\&}(x_2, x_1) = x_2 \& x_1$, $f_{\oplus}(x_2, x_1) = x_2 \oplus x_1$ и соответствующие значения $I_{\&}$, $I_{\oplus}$ представлены в табл. 2.

Таблица 2

Значения $f_{\&}(x_2, x_1)$, $f_{\oplus}(x_2, x_1)$ и $I_{\&}$, $I_{\oplus}$. Элементы второго типа

x_2	x_1	$f_{\&}$	$I_{\&}$	$f_{\oplus}$	$I_{\oplus}$
0	0	0	0	0	0
0	1	0	$I_0(1-q_1)^{2m}$	1	$I_0 - I_0(1-q_2)^{2m}$
1	0	0	$I_0(1-q_1)^{2m}$	1	$I_0 - I_0(1-q_2)^{2m}$
1	1	1	$2I_0(1-q_1)^{2m-k_1/2}$	0	$2I_0 - 2I_0(1-q_2)^{2m-k_2/2}$

ИДЕНТИЧНОСТЬ ЗНАЧЕНИЙ ИНТЕНСИВНОСТИ

Потребуем равенства значений $\boldsymbol{I}_{\&}$ и $\boldsymbol{I}_{\oplus}$, соответствующих одинаковым логическим константам:

$$\begin{cases} 2I_0(1-q_1)^{2m-\frac{k_1}{2}} = I_0 - I_0(1-q_2)^{2m}; \\ I_0(1-q_1)^{2m} = 2I_0 - 2I_0(1-q_2)^{2m-\frac{k_2}{2}}. \end{cases}$$

Решения этой системы уравнений получаются численными методами. В частности, решением являются $2m = 6$, $k_1 = k_2 = 4$, $q_1 \approx 0{,}15$ и $q_2 = 0{,}1$. Значения m определяются требуемой длительностью операции.

Получен функционально полный базис интерференционных ОЛЭ второго типа.

2.3. Y-волновые элементы

Импульсы, имеющие различные λ_i, не взаимодействуют.

Структуры ОЛЭ, одновременно и независимо выполняющие операции над данными, представленными на различных λ, получаются соответствующим размещением щелей и узлов. Параметры ОЛЭ первого типа и ОЛЭ второго типа, обрабатывающих длины λ_i, $i = 1,2,\dots,y$ вычисляются соответственно из:

$$\begin{cases} 4I_0(1-q_1)^{my+m} = I_0 - I_0(1-q_2)^{2my}; \\ I_0(1-q_1)^{2my} = 2I_0 - 4I_0(1-q_2)^{my+m}; \end{cases}$$

и

$$\begin{cases} 2I_0(1-q_1)^{2m-\frac{k_1}{2}y} = I_0 - I_0(1-q_2)^{2m}; \\ I_0(1-q_1)^{2m} = 2I_0 - 2I_0(1-q_2)^{2m-\frac{k_2}{2}y}. \end{cases}$$

3. Масштабирование интенсивности

Оценим возможности увеличения различия значений интенсивности, соответствующих логическим «0» и логическим «1».

3.1. J-уровневые структуры элементов

Волновод, в который входные импульсы поступают извне элемента, называем волноводом первого уровня. Волновод, в который входные импульсы поступают из волновода k-го уровня, где $k = 1,2, \ldots$, называем волноводом $(k+1)$ уровня.

Введем, как показано на рис. 3, в ОЛЭ первого типа еще один волновод – волновод второго уровня. Эту структуру назовем двухуровневым ОЛЭ первого типа.

Назначения и обозначения конструкционных элементов, совпадают с указанными в 2.1.

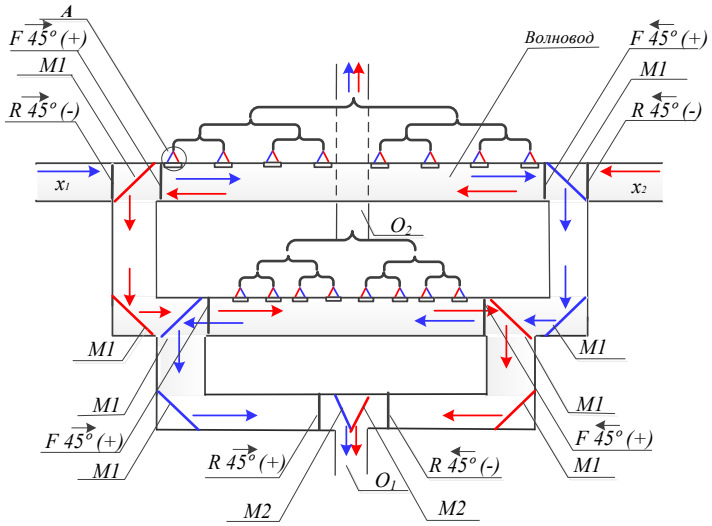


Рис. 3. Структура двухуровневого ОЛЭ первого типа

В волноводе второго уровня ОЛЭ, реализующего &, устроены $2m$ щелей в местах пучностей, образуемых импульсами, поступающими из волновода первого уровня. Выход элемента обозначен O_1.

В случае $x_1 = 1, x_2 = 0$ или $x_1 = 0, x_2 = 1$ имеем одиночный импульс, интенсивность которого после волновода первого уровня равна $I_{2m} = I_0(1-q_1)^{2m}$. В результате взаимодействия с $2m$ щелями волновода второго уровня его интенсивность составит $I_{\&} = I_{4m} = I_{2m}(1-q_1)^{2m} = I_0(1-q_1)^{4m}$. Общее энерговыделение в щели волноводов первого и второго уровней пропорционально $\overset{\Delta}{I}_{4m} = I_0 - I_0(1-q_1)^{4m}$.

В случае $x_1 = 1, x_2 = 1$ имеем на входах волновода второго уровня два встречных импульса; интенсивность каждого $I_{2m} = 2I_0(1-q_1)^{2m}$. В результате их столкновения суммарное значение интенсивности на выходе O_1, соответствующее логической единице равно $I_{\&} = 8I_0(1-q_1)^{4m}$. Общее энерговыделение в щели пропорционально $\overset{\Delta}{I}_{4m} = 2I_0 - 8I_0(1-q_1)^{4m}$.

Структура двухуровневого ОЛЭ, реализующего $\oplus$, представлена на рис. 3. Выход обозначен O_2. Значения $I_{\oplus}$ получаются аналогично вышеизложенному.

Элемент «НЕ» получается из ОЛЭ, реализующего $\oplus$, подачей на один из входов сигнала логической «1». Длительности выполнения операций «$\bar{}$» и $\oplus$ совпадают.

Значения функций $f_{\&}(x_2, x_1) = x_2 \& x_1$, $f_{\oplus}(x_2, x_1) = x_2 \oplus x_1$ и соответствующие им значения $I_{\&}$, $I_{\oplus}$, получаемые на выходах двухуровневых ОЛЭ первого типа, представлены в табл. 3.

Таблица 3

Значения $f_{\&}(x_2, x_1)$, $f_{\oplus}(x_2, x_1)$ и $I_{\&}$, $I_{\oplus}$. Двухуровневые ОЛЭ первого типа

x_2	x_1	$f_{\&}$	$I_{\&}$	$f_{\oplus}$	$I_{\oplus}$
0	0	0	0	0	0
0	1	0	$I_0(1-q_1)^{4m}$	1	$I_0(1-q_2)^{2m} - I_0(1-q_2)^{4m}$
1	0	0	$I_0(1-q_1)^{4m}$	1	$I_0(1-q_2)^{2m} - I_0(1-q_2)^{4m}$
1	1	1	$8I_0(1-q_1)^{4m}$	0	$2I_0(1-q_2)^{2m} - 8I_0(1-q_2)^{4m}$

Одинаковым логическим константам должны соответствовать одинаковые интенсивности на выходах ОЛЭ. Необходимо:

$$\begin{cases} I_0(1-q_1)^{4m} = 2I_0(1-q_2)^{2m} - 8I_0(1-q_2)^{4m}, \\ 8I_0(1-q_1)^{4m} = I_0(1-q_2)^{2m} - I_0(1-q_2)^{4m}. \end{cases}$$

Находим: $(1-q_1)^{2m} \approx 0{,}15$ и $(1-q_2)^{2m} = 0{,}238$.

Здесь различие интенсивностей логической «1» и логического «0» в 2 раза больше, чем в одноуровневом ОЛЭ.

Приведенные соотношения обобщаются на J уровней волноводов, где $J > 1$, целое. Получаем:

$$\begin{cases} 2 \cdot 2^J I_0(1-q_1)^{2mJ} = I_0(1-q_2)^{2m(J-1)} - I_0(1-q_2)^{2mJ}, \\ I_0(1-q_1)^{2mJ} = 2I_0(1-q_2)^{2m(J-1)} - 2 \cdot 2^J I_0(1-q_2)^{2mJ}. \end{cases}$$

При однократном срабатывании, соответствующие логическому «0» и логической «1», значения интенсивности в J-уровневых ОЛЭ отличаются в $2 \cdot 2^J$ раз. Длительность выполнения операции составляет не менее τJ, где τ – длительность выполнения операции одноуровневым элементом.

ДИНАМИКА ЗНАЧЕНИЙ ИНТЕНСИВНОСТИ

В результате последовательных операций значения интенсивностей, соответствующие различным логическим константам, могут сближаться. Это обусловлено различием значений выходной интенсивности при подаче на оба входа «1» и подачей на один вход «1», а на другой – «0». Оценим границы значений интенсивности.

Пусть с выхода i-го элемента импульс поступает на вход $(i+1)$ элемента, где $i = 1,2, \ldots, (h-1)$. Обозначим значения интенсивностей «1» и «0» после h операций на выходе h-го элемента символами $I_1^{\widetilde{(h)}}$ и $I_0^{\widetilde{(h)}}$. Начальное входное значение, соответствующее логической «1», обозначим $I_1^{\widetilde{(0)}}$. Для простоты изложения используем J уровневый элемент «И» первого типа.

В общем виде имеем:

$$I_1^{\widetilde{(h)}} = (2 \cdot 2^J (1-q_1)^{2mJ})^h \cdot I_1^{\widetilde{(0)}},$$

$$I_0^{\widetilde{(h)}} = (1-q_1)^{2mJ} \left(\sqrt{\tilde{I}_0^{(h-1)}} + \sqrt{\tilde{I}_1^{(h-1)}} \right)^2.$$

Подставляя значения интенсивностей, получаем:

$$2^{J-1} < \frac{I_1^{\widetilde{(h)}}}{I_0^{\widetilde{(h)}}} < 2 \cdot 2^J.$$

Это справедливо и для ОЛЭ «исключающее ИЛИ» и «НЕ».

Для J-уровневых ОЛЭ второго типа имеем:

$$\begin{cases} 2 \cdot I_0(1-q_1)^{(2m-\frac{k_1}{2})J} = I_0(1-q_2)^{2m(J-1)} - I_0(1-q_2)^{2mJ}, \\ I_0(1-q_1)^{2mJ} = 2I_0(1-q_2)^{(2m-\frac{k_2}{2})(J-1)} - 2 \cdot I_0(1-q_2)^{(2m-\frac{k_2}{2})J}. \end{cases}$$

Динамические значения интенсивностей удовлетворяют условию: $\frac{1}{2(1-q_1)^{\frac{k_1}{2}J}} < \frac{I_1^{\widetilde{(h)}}}{I_0^{\widetilde{(h)}}} < \frac{2}{(1-q_1)^{\frac{k_1}{2}J}}$.

4. Пример реализации и применения

Для реализации ОЛЭ могут использоваться оптоволоконные [10], либо планарно-интегральные [9] волноводы, фотонные кристаллы (ФК) [12], графеновые структуры [5]. Для локализации интенсивности в пучностях и узлах необходимо, чтобы размер щели составлял $\sim 0{,}1\lambda$.

Для $\lambda = 1{,}55$ мкм этому условию удовлетворяет ФК из кремния [4]. Оценим параметры ОЛЭ из этого ФК.

Значения $m = 4$ и $J = 2$ задаем из требований к быстродействию и различию интенсивностей. Из условий идентичности вычисляем $q_1 = 0{,}22$ и $q_2 = 0{,}15$.

Щели реализуются линейными дефектами в ФК [4]. Из графика относительных значений интенсивности [4] находим, что $q_1 = 0{,}22$ достигается при $r_d/p = 0{,}38$, где r_d – радиус стержня дефекта; $p = 0{,}418$ мкм – постоянная решетки кремния. Получаем $r_d \approx 160$ нм. Аналогично находим, что $q_2 = 0{,}15$ достигается при $r_d/p = 0{,}35$. Получаем $r_d \approx 120$ нм. Изготовление таких дефектов освоено [13].

Чтобы импульсы в ответвительных волноводах не взаимодействовали между собой, расстояние между центрами щелей выберем $8p = 3{,}3$ мкм [4]. Размещение 8 щелей требует 26 мкм. Фарадеевские вращатели в ФК требуют 1,5 мкм, другие конструкционные элементы около 1,0 мкм [13]. Длину волновода принимаем L=75 мкм. Полагая скорость света в волноводе $\upsilon = 2 \cdot 10^8$ м/с, находим $\tau = L/\upsilon = 375 \cdot 10^{-15}$ с длительность выполнения операции.

Периодичность регенерации. В процессе выполнения логических операций интенсивность уменьшается. Критичным является уровень интенсивности, соответствующий логической «1», определяющий наличие сигнала. Логический «0» может быть как угодно близок к физическому нулю.

Количество энергии в ОЛЭ из кремния при $\lambda = 1{,}55$ мкм и длительности выполнения операции $\tau = 375 \cdot 10^{-15}$ с составит $E = IS\tau \approx 1{,}23 \cdot 10^{-11}$ Дж, где $I = \frac{\varepsilon_{разр}}{\tau} = 0{,}67 \cdot 10^{13}$ Вт/см^2 - пороговое значение интенсивности, превышение которого разрушает волновод, $\varepsilon_{разр} = 2{,}5$Дж/см^2 - лучевая прочность кремния при длительности импульса $\tau = 300 - 400 \cdot 10^{-15}$ с [14], $S \approx 4{,}9 \cdot 10^{-12}$ м^2 – площадь светового пятна.

Энергия фотона при $\lambda = 1{,}5$ мкм составляет: $E_ф \approx 13{,}2 \cdot 10^{-20}$ Дж [8]. Количество фотонов в импульсе: $\frac{E}{E_ф} \approx 0{,}09 \cdot 10^9$ штук. Для надежной идентификации импульса при $T = 300$К достаточно 10^3 фотонов [12]. Мощность импульса $\frac{13{,}2 \cdot 10^{-20}\text{Дж} \cdot 10^3}{375 \cdot 10^{-15}} \approx 0{,}34$ мВт. Бюджет мощности 38 дБ.

Выполнение одной логической операции уменьшает мощность сигнала не более, чем на 5 дБ.

Без регенерации можно выполнить ≈ 7 шт. операций, в течение $2{,}5 \cdot 10^{-12}$ с. Для регенерации применяются нелинейные оптические ключи пикосекундного диапазона [15]. Их задействование не скажется на производительности и энергопотреблении устройства.

В процессе выполнения операций импульсы передаются только в волноводах, когерентность сохраняется. Энергия остается в ОЛЭ и задействуется для усиления и регенерации сигнала.

Общие потери энергии определяются потерями в волноводах, эта величина составляет 0,1 дБ/см, и эффективностью ввода излучения в волновод $\mu \geq 0{,}9$ [2].

Фотонная вычислительная машина, реализованная из предложенных ОЛЭ, при $\lambda = 1{,}5$ мкм, длине волновода L = 75 мкм и указанных значениях энергетических параметров имеет в 10^3–10^4 раз большую производительность, по сравнению с электронными аналогами при одинаковых энергозатратах [1].

Заключение

Предложены структуры интерференционных логических элементов, образующих полный функциональный базис. Выполняются требования идентичности значений интенсивности, соответствующих логическим константам «0» и «1», вырабатываемых различными элементами в заданные фиксированные интервалы времени.

Масштабирование значений интенсивности логически констант достигается увеличением количества уровней волноводов и сопровождается пропорциональным увеличением длительности выполнения операций.

На примере цифровой фотонной вычислительной машины показаны целесообразность и перспективность применения предложенных элементов.

Литература

1. Степаненко С. А. // Фотонная вычислительная машина. Принципы реализации. Оценки параметров. Доклады академии наук, 2017, т. 476, №4, С. 389–394.
2. Белов П. А., Беспалов В. Г., Васильев В. Н., и др. // Оптические процессоры: достижения и новые идеи [Электронный ресурс]. Режим доступа: http://pandia.ru/77/192/21190.php
3. Полуэктов А. О. // Оптический логический элемент. Патент РФ 20751061. Опубликован 10.03.1997г.
4. Сунь Сяо-Вэнь, Ян Сю-Лунь, Мэн Сян-Фэн, Чжу Цзи-Нань, Ван Юй-Жун, Инь Юн-Кай, Дун Го-Янь. // Разработка и анализ логических элементов «ИЛИ-НЕ», «И-НЕ» и «исключающее ИЛИ-НЕ» на основе эффекта интерференции. Квантовая электроника, 48, №2 (2018), С.178–183.
5. Xiaoting Wu, Jinping Tian, Rongcao Yang. // A type of all-optical logic gate base on graphene surface plasmon polaritons. Optics Communications 403 (2017), P. 185-192.
6. Maria Papaioannou, Eric Plum, João Valente, Edward T. F. Rogers, and Nikolay I. Zheludev // «All-optical multichannel logic based on coherent perfect absorption in a plasmonic metamaterial». APL PHOTONICS, 1, 090801 (2016); doi: 10.1063/1.4966269.
7. Hussein M. E., Tamer A. Ali, Nadia H. Rafab. // «New design of a complete set of Photonic Crystals logic gates» Optics Communications 411 (2018) P.175–181.
8. Яворский Б. М., Детлаф А. А., Лебедев А. К. // Справочник по физике для инженеров и студентов вузов М.: ООО «Издательство Оникс»: ООО «Издательство «Мир и Образование» 2008г.–С. 1056.
9. Никаноров Н. В., Шандаров С. М. // Волноводная фотоника. ИТМО. Санкт-Петербург, 2008 г.
10. Вербовецкий А. А. // Основы проектирования оптоэлектронных систем связи. М., Радио и связь, 2000г.
11. Поспелов Д. А. // Логические методы анализа и синтеза схем. М.: Энергия, 1974 г., С. 368 .
12. Салех Б., Тейх М. // Оптика и фотоника. Принципы и применения. т.1,2. Долгопрудный: «Интеллект», 2012г.
13. Котляр В. В. // Нанофотоника – манипулирование светом с помощью наноструктур. Компьютерная оптика, 2008, т. 32, №2, С. 119–135.
14. Gallais L., Douti D.-B., Commandre M., Bataviciute G. // Wavelength dependence of femtosecond laser-induced damage threshold of optical materials. Journal of Applied Physics 117, 223103 (2015), P.223103–6.
15. Гончарова О. В., Демин А. В. // Оптический переключающий элемент. Патент РФ 2096815, опубликован 20.11.1997г.

A PHOTONIC COMPUTER. OPTICAL LOGIC GATES

S. A. Stepanenko

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

Structures of interference-based logic gates constituting a complete functional unit are proposed. They meet the requirement that the intensity values corresponding to logical constants “0” and “1” generated by different gates in given fixed time intervals be identical.

The intensity values of the logical constants can be scaled up by increasing the number of levels of the waveguides with a proportional increase in the operation execution time.

A digital photonic computer is used as an example to demonstrate the feasibility and application potential of the proposed gates.

Key words: photonic machine, optical gate, waveguide.

УДК 533.6
DOI 10.53403/9785951505071_2022_461

ОСОБЕННОСТИ ПРИМЕНЕНИЯ МЕТОДА ГЕОМЕТРИЧЕСКОЙ МНОГОУРОВНЕВОЙ ИНИЦИАЛИЗАЦИИ ДЛЯ УСКОРЕНИЯ РЕШЕНИЯ ЗАДАЧ АЭРОДИНАМИКИ НА ПРОИЗВОЛЬНЫХ НЕСТРУКТУРИРОВАННЫХ СЕТКАХ

А. В. Стручков, А. С. Козелков, Р. Н. Жучков

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В работе представлен алгоритм ускорения решения задач аэродинамики на основе геометрической многоуровневой инициализации. Разработанный алгоритм помимо более быстрой сходимости, способствует устойчивому поведению решения на этапе формирования и установления структуры течения. Идея метода состоит в генерации последовательности грубых сеток на основе исходной с решением на каждой из них и интерполяцией полученного решения при переходе на более подробную сетку. После расчета на самой подробной сетке, полученное решение интерполируется на исходную сетку, являясь при этом начальной инициализацией на ней. Алгоритм разработан для применения на любых неструктурированных сетках с ячейками произвольной формы. При генерации последовательности грубых сеток в качестве критерия для объединения ячеек в новые контрольные объемы используется соотношение, рассчитываемое через площадь граней и объем ячейки. Процесс объединения ячеек основан на анализе взвешенного графа. Анализ влияния на устойчивость и скорость сходимости решения сделан на тестовых задачах обтекания.

Ключевые слова: аэродинамика, многосеточный решатель, взвешанный граф, критерий объединения, неструктурированная сетка, агломирация.

Введение

В настоящее время для численного решения задач аэродинамики наибольшее предпочтение отдается использованию неструктурированных расчетных сеток [1, 2]. В случае промышленно-ориентированных задач, например, расчета аэродинамических характеристик летательных аппаратов, для получения качественного решения, размерность расчетных моделей может достигать нескольких десятков миллионов ячеек [3]. Использование сеток большой размерности, естественно, приводит к существенному увеличению времени расчета. Для сокращения временных затрат разработаны специализированные численные методы и алгоритмы – широкое распространение получили алгебраический и геометрический многосеточные методы [2, 4]. Обзор по применению данных методов на практике для решения различных задач можно найти в многочисленных статьях и монографиях (см., [2, 4–6]).

Известно, что при решении задач аэро- гидродинамики значительная часть времени расчетного шага тратится на решение СЛАУ (системы линейных алгебраических уравнений), полученной в результате аппроксимации уравнений Навье – Стокса неявными методами, которые в настоящее время преобладают при решении промышлено-ориентированных задач на неструктурированных сетах [1]. Так, в [1–3] для ускорения решения СЛАУ при расчете течений жидкости и газа используется алгебраический многосеточный метод. В алгебраическом многосеточном методе построение иерархии вложенных сеток осуществляется на матричном уровне – для объединения в группы у каждой ячейки

рассматриваются ее соседи и выбираются ячейки, связь с которыми сильнее. Под связью ячеек подразумевается величина коэффициента для ячейки из матрицы, записанной в формате LDU [2], формируемой для решения СЛАУ. Для ускорения сходимости итерационного процесса решения СЛАУ учитываются свойства используемого метода, например, Гаусса – Зейделя.

Другим вариантом для ускорения решения СЛАУ является использование геометрического многосеточного метода, в рамках которого объединение ячеек происходит с учетом их геометрических параметров (площадь граней, объем ячейки). При использовании этого метода важным является сам способ построения последовательности вложенных сеток. В случае расчета на структурированных сетках последовательность вложенных сеток строится тривиальным образом по средствам удаления узлов подробной сетки в каждом координатном направлении. При реализации многосеточного метода на неструктурированных сетках для формирования контрольных объемов часто используется метод агломерации, во время работы которого для улучшения качества сеток на размер формируемых контрольных объемов накладываются дополнительные условия. Такой подход реализуется путем объединения ячеек на основе анализа взвешенного графа с применением критерия для объединения ячеек.

Как алгебраический, так и геометрический многосеточный метод используется на каждом счетном шаге итерационного процесса решения задачи. Однако ускорение решения можно добиться введением дополнительной процедуры не в счетный шаг итерационного процесса, а в начальный этап вычислительной процедуры, соответствующий инициализации расчетной области. Введение дополнительной процедуры начальной инициализации позволяет обеспечить формирование структуры потока вблизи обтекаемого объекта во время распространения по расчетной области начального возмущения. Наибольший эффект от применения этой процедуры достигается в случае моделирования течений с ударно-волновыми процессами, выражаемый в увеличении скорости сходимости решения и повышении его устойчивости.

Способом введения процедуры начальной инициализации является геометрическая многосеточная инициализация. Идея метода состоит в генерации на основе исходной базовой сетки последовательности грубых сеток, на которых решается рассматриваемая задача до полной сходимости численного решения. Решение задачи начинается с самой грубой сетки в последовательности. За счет того, что огрубленные сетки имеют меньшее число ячеек, в сравнении с исходной базовой, решение на них имеет более высокую скорость сходимости. Затем, полученное решение интерполируется на более подробную сетку, а в итоге – на исходную базовую сетку, и используется как начальная инициализация при расчете. Таким образом, за счет использования начальной инициализации, максимально близкой к конечному решению, ускоряется процесс сходимости этого решения как на каждой сетки в последовательности, так и в итоге – на базовой сетке.

Такой подход уже применяется для решения задач аэродинамики. Однако, в доступных публикациях описание технологии довольно общее и применимо на структурированных сетках. В случае неструктурированных сеток зачастую не уделяется должного внимания описанию алгоритма построения последовательности грубых сеток и методам решения на них, так как построенные сетки обычно характеризуются ячейками сложной формы, что вызывает трудности с дискретизацией основных уравнений на подобных сетках.

В настоящей работе рассматривается описание вычислительного алгоритма, включающего в себя шаг начальной инициализации на основе геометрического многосеточного метода и его применение при решении задач аэродинамики с использованием неструктурированной расчетной сетки. В предлагаемом алгоритме генерация последовательности грубых сеток рассматривается как задача оптимизации, решаемая с применением соответствующего критерия для объединения ячеек в контрольные объемы грубой сетки. В решении этой задачи используется подход на основе взвешенного графа. Представленный метод обеспечивает формирование сеток грубых уровней, решение на которых используется в качестве начальной инициализацией при расчете на исходной базовой сетке. Для решения рассматриваемой задачи на последовательности грубых сеток используется неявная разностная схема и алгебраический многосеточный решатель СЛАУ.

Основные уравнения и метод решения

Основные уравнения и итерационная процедура

Для описания нестационарного вязкого течения газа используется система уравнений Навье – Стокса, которая в консервативной форме, в декартовых координатах имеет вид:

$$\frac{\partial W}{\partial t}+\frac{\partial\left(F_e-F_{\mu}\right)}{\partial x}+\frac{\partial\left(G_e-G_{\mu}\right)}{\partial y}+\frac{\partial\left(E_e-E_{\mu}\right)}{\partial z}=H\,, \qquad (1)$$

где W – вектор консервативных переменных, $F_e(W), G_e(W), E(W_e)$ и $F_{\mu}(W,\nabla W), G_{\mu}(W,\nabla W), E_{\mu}(W,\nabla W)$ – вектора, соответственно, конвективных и диффузионных потоков через поверхности ортогональные осям координат, $H(W,\nabla W)$ – источник.

Для ламинарных течений вектора $W,F_e,G_e,E_e,F_{\mu},G_{\mu},E_{\mu}$ имеют вид:

$$W=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{pmatrix},\ F_e=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uH\end{pmatrix},\ G_e=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ \rho vH\end{pmatrix},\ E_e=\begin{pmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ \rho wH\end{pmatrix},\ H=\begin{pmatrix}0\\ 0\\ 0\\ 0\\ 0\end{pmatrix},$$

$$F_{\mu}=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ q_x+u\tau_{xx}+v\tau_{xy}+w\tau_{xz}\end{pmatrix},\ G_{\mu}=\begin{pmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ q_x+u\tau_{yx}+v\tau_{yy}+w\tau_{yz}\end{pmatrix},\ E_{\mu}=\begin{pmatrix}0\\ \tau_{zx}\\ \tau_{zy}\\ \tau_{zz}\\ q_x+u\tau_{zx}+v\tau_{zy}+w\tau_{zz}\end{pmatrix}.$$

Здесь используются общепринятые обозначения: ρ – плотность, $\mathrm{u,v,w}$ – компоненты вектора скорости, p — давление, $E=\varepsilon+\dfrac{u^2+v^2+w^2}{2}$ – удельная полная энергия, $H=\varepsilon+\dfrac{p}{\rho}+\dfrac{u^2+v^2+w^2}{2}$ – удельная полная энтальпия, ε – удельная внутренняя энергия. Система уравнений замыкается калорическим и термическим уравнениями состояния:

$$\rho=\frac{mp}{RT}\,,\ \varepsilon=C_VT\,.$$

Для описания турбулентных течений система уравнений (1) дополняется уравнениями моделями турбулентности.

Для дискретизации уравнений (1) используется метод конечных объемов, основанный на интегрировании исходных дифференциальных уравнений по контрольному объему, записываемых в дивергентном виде. Контрольные объемы (ячейки сетки) являются произвольными многогранниками, в общем случае, ограниченными произвольным числом граней и покрывающими расчетную область без зазоров и наложений. Общий вид ячейки произвольной неструктурированной сетки показан на рис. 1.

При дискретизации системы уравнений (1) диффузионные потоки через внутренние грани ячеек вычисляются в серединной точке грани как полусумма соответствующих узловых значений, умноженных на площадь грани, а для дискретизации невязких потоков используются схемы семейства TVD.

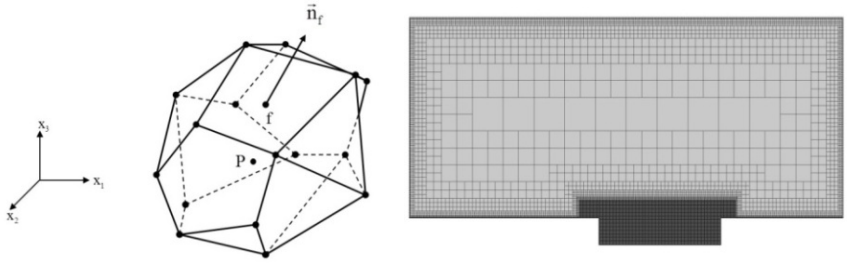


Рис. 1. Общий вид ячейки и базовой сетки

В результате дискретизации уравнений (1) формируется система разностных уравнений с разреженной матрицей большого размера, которая решается итерационным методом и является одной из самых затратных процедур с точки зрения вычислительных ресурсов [xxx]. Для максимально эффективного решения системы разностных уравнений используется алгебраический многосеточный решатель. Полный шаг численной итерационной процедуры представлен на рис. 2.

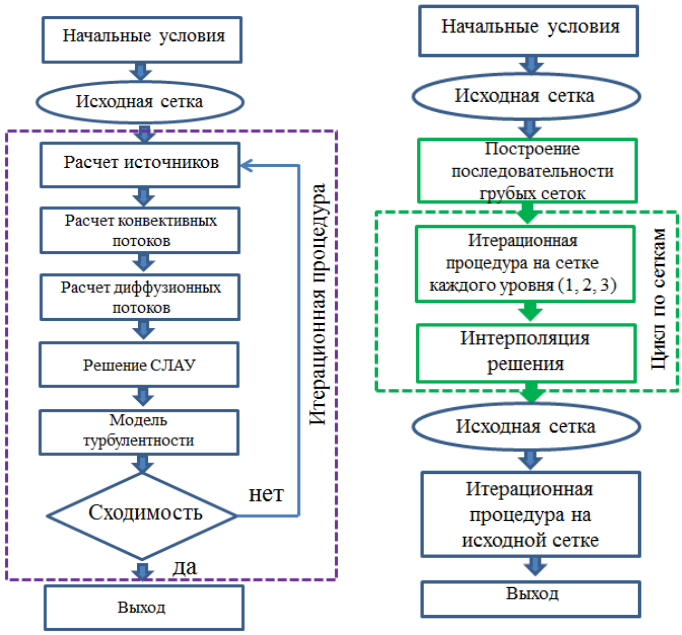


Рис. 2. «Главная» итерационная процедура (слева – без алгоритма начальной инициализации, справа – с алгоритмом начальной инициализации)

В соответствии с блок-схемой (рис. 2, слева), перед началом итерационной процедуры на исходной базовой сетке выполняется инициализация расчетной области однородным равномерным полем всех расчетных величин в соответствии с заданными начальными условиями. Далее выполняется расчет источников, конвективных и диффузионных потоков, а так же расчет уравнений модели турбулентности. Итерационный процесс продолжается до достижения сходимости численного решения.

Как уже отмечалось, алгоритм многосеточной инициализации предполагает построение последовательности грубых сеток на основе исходной базовой и последовательное решение (получение конечного сошедшегося численного решения) на каждой из них, начиная с самой грубой. При этом полученное решение используется в качестве начальной инициализации при переходе к более подробной сетке. Поэтому для реализации алгоритма многосеточной инициализации необходима модификация счетного алгоритма. Модификация заключается в добавлении двух основных шагов.

Первый шаг – это введение перед итерационной процедурой решения алгоритма построения последовательности грубых сеток на основе исходной базовой сетки. Количество грубых сеток предполагается равным 3 (алгоритм построения грубой сетки на основе исходной описан ниже) – треть-

его, второго и первого уровней. Сетка третьего уровня является самой грубой, состоит из контрольных объемов с максимальным числом ячеек исходной сетки. Самая подробная сетка в последовательности – сетка первого уровня.

Второй шаг – выполнение итерационной процедуры и получение конечного решения на каждой из построенных сеток, с интерполяцией решения при переходе от сетки к сетке (интерполяция осуществляется простым переносом сеточных массивов при переходе от грубой к более подробной сетке). Блок-схема численного метода с введением новых алгоритмов представлена на рис. 2, справа.

Основная проблема в случае однородной начальной инициализации для сверх- и гиперзвуковых течений – это устойчивость численного решения в момент формирование ударной волны. Зачастую при распространении возмущения по равномерному начальному полю ввиду возникающего сильного градиента физических величин происходит «развал» численного решения. Поэтому приходится вести счет с параметрами, обеспечивающими устойчивость численного решения, но приводящими к низкой скорости сходимости. В случае же многосеточной инициализации – на момент начала решения поле распределения физических величин максимально близко к конечному решению и особенности решения (например, ударная волна) имеет сформированную структуру. В процессе дальнейшего расчета у особенностей течения уточняются границы и интенсивность, что не связано с существенными трудностями сохранения устойчивости численного решения. Кроме того, такой вид начальной инициализации повышает скорость сходимости численного решения.

Алгоритм построения последовательности грубых сеток

Построение последовательности грубых сеток выполняется на основе взвешенного графа, соответствующего исходной сетки. Взвешенный граф строится по следующим правилам. Каждой ячейки сетки ставится в соответствие вершина графа. Две вершины графа соединены через ребро, если соответствующие ячейки сетки разделены общей гранью. Пример трехмерной сетки и соответствующий ей взвешенный граф представлен на рис. 3.

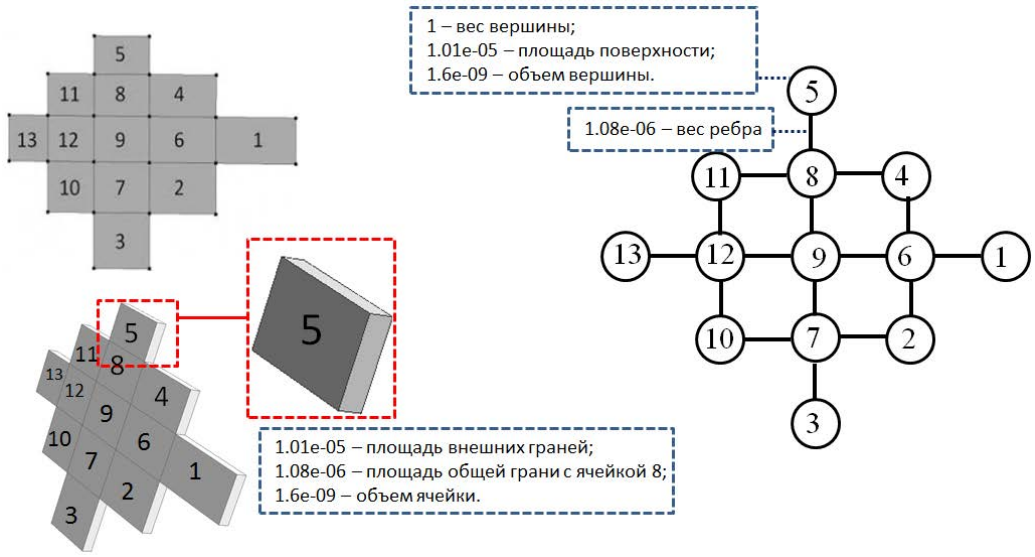


Рис. 3. Сетка и соответствующий ей взвешенный граф

Каждую вершину графа характеризуют три значения – вес вершины, площадь поверхности вершины и объем вершины. Вес вершины характеризует количество исходных ячеек сетки, образующих данный контрольный объем. Изначально вес каждой вершины графа исходной сетки равен одному. По мере огрубления сетки значение весов для вершин обновляются и становятся равными числу ячеек, образующих данный контрольный объем. Площадь поверхности вершины характеризует площадь поверхности, не используемой совместно с другими контрольными объемами, то есть являющейся внешней границей сетки. Следует отметить, что для всех внутренних контрольных объемов

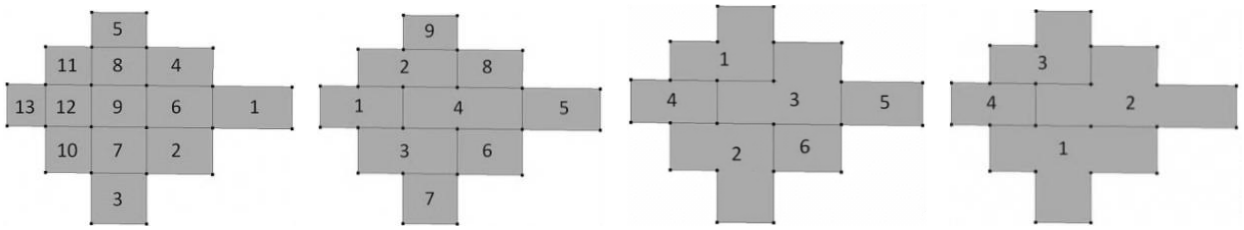
этот параметр равен нулю. Объем вершины графа соответствует объему контрольного объема. Кроме того, каждое ребро взвешенного графа имеет вес, характеризующий площадь поверхности, разделяющей два контрольных объема. Таким образом, взвешенный граф содержит всю информацию необходимую для формирования контрольного объема грубой сетки. Далее на основе алгоритма агломерации, с использованием взвешенного графа, выполняется построение последовательности грубых сеток, отличающихся друг от друга максимальным числом ячеек исходной сетки, используемых для объединения в новые контрольные объемы.

В алгоритме агломерации (agglomeration method) производится слияние контрольных объемов подробной сетки, имеющих общую грань, в контрольный объем грубой сетки (full coarsening). Для улучшения качества сеток на размер контрольных объемов грубой сетки накладываются дополнительные ограничения. Наиболее важным ограничением является размер и форма контрольного объема, полученного в результате слияния нескольких соседних контрольных объемов подробной сетки. Для преодоления проблем, связанных с дискретизацией основных уравнений (1) на грубых сетках, при формировании контрольного объема оценивают его *меру качества*. Мера качества – это критерий для оценки применимости контрольного объема при дискретизации основных уравнений. Для контрольного объема произвольной неструктурированной сетки меру качества $\boldsymbol{A}$, можно вычислить, используя геометрические параметры ячейки – площади сторон и объем. В численном виде данная мера имеет следующее выражение (2):

$$A = S^{3/2}/V, \tag{2}$$

где S – площадь сторон контрольного объема, V – объем.

Контрольный объем считается хорошо сформированным, если он является максимально компактным, т. е. отношение суммы площадей граней к объему является минимальным. Рассмотрим работу алгоритма на примере фрагмента трехмерной сетки, состоящего из 13 ячеек, показанного на рис. 4,а.



а б в г

Рис. 4. Фрагмент сетки, пример работы алгоритма

Для алгоритма объединения в новые контрольные объемы выполняется цикл по ячейкам сетки. Для каждой ячейки ее соседи рассматриваются как потенциальные варианты для объединения в один контрольный объем. Так для сетки на рис. 4,а, возможные новые контрольные объемы могут быть сформированы следующими парами ячеек: (6–1), (6–2), (6–4), (7–2), (7–3), (8–4), (8–5), (9–6), (9–7), (9–8), (10–7), (11–8), (12–9), (12–10), (12–11), (13–12). Так как каждая ячейка может быть включена только в один новый контрольный объем, то для выбора соседней ячейки, с которой она будет объединена, используется критерий вида (1) и выбирается та соседняя ячейка, при объединении с которой выражение (1) принимает минимальное значение. Таким образом, в результате первой итерации объединения новые контрольные объемы формируются следующими парами ячеек (рис. 4,б):

- новый контрольный объем 1 – формируется парой ячеек (13–12),
- новый контрольный объем 2 – формируется парой ячеек (11–8),
- новый контрольный объем 3 – формируется парой ячеек (10–7),
- новый контрольный объем 4 – формируется парой ячеек (9–6).

При этом ячейки 1, 2, 3, 4, 5 не вошли в новые контрольные объемы на первой итерации объединения. В конце операции объединения происходит переиндексация всех контрольных объемов сетки (рис. 4,б). Итерации объединения продолжаются до тех пор, пока не будет достигнуто максимальное количество ячеек в новом контрольном объеме, в данном случае оно равно четырем. На рис. 4,в и г

показаны новые контрольные объемы, полученные в результате второй и третьей итерации объединения. Как видно из рис. 4,г, максимальное число ячеек в новом контрольном объеме равно четырем и последующее объединение контрольных объемов не возможно.

Последовательность сеток различной разрешающей способности, построенная при помощи метода агломерации, для профиля крыла NACA0012 приводится на рис. 5. Для первого уровня объединение в новый контрольный объем выполняется по 4 ячейки, для второго уровня – по 8, для третьего уровня по 16. Исходная сетка наилучшей разрешающей способности содержит 157914 ячеек (фрагмент а). При переходе к грубым сеткам число ячеек уменьшается примерно в 4 раза (фрагмент б, 41907 ячеек), в 8 раз (фрагмент в, 21529 ячеек) и в 16 раз (фрагмент г, 11022 ячеек), а форма ячеек сетки становится более сложной.

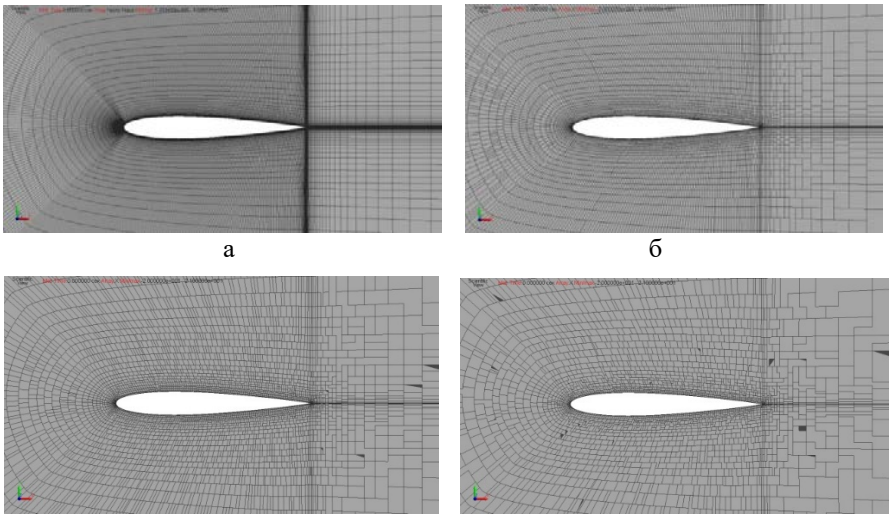


Рис. 5. Последовательность сеток, построенная при помощи метода агломерации:
а – исходная, б – 1го уровня, в – 2го уровня, г – 3го уровня

На рис. 6,г показано решение, полученное на сетке 3-го уровня. Далее это решение в качестве начальной инициализации переносилось на сетку 2-го уровня и полученное на ней решение представлено на рис. 6,в. Затем тоже самое выполняется для сетки 1-го уровня и исходной сетки. В итоге, решение на сетке 1-го уровня показано на рис. 6,б, а решение на исходной сетке – на рис. 6,а.

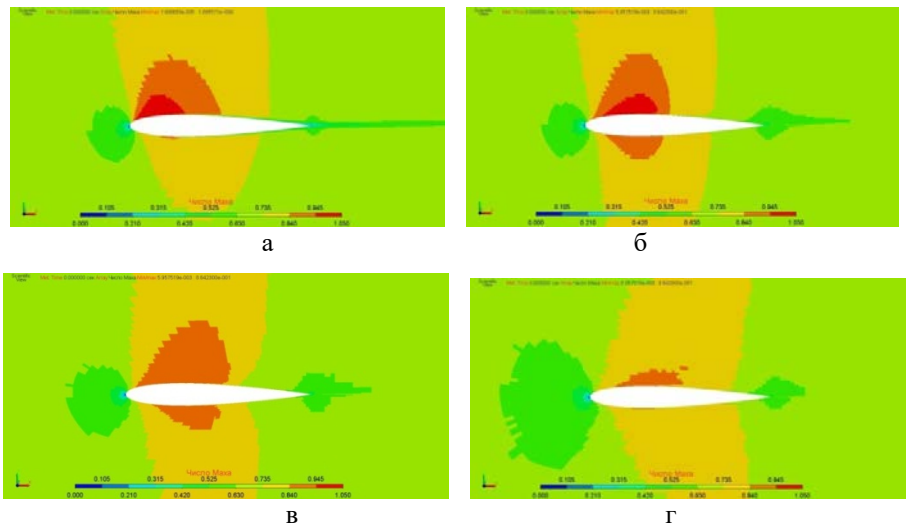


Рис. 6. Решение, поле распределение числа Маха, полученное на сетках: а – исходная,
б – 1-го уровня, в – 2-го уровня, г – 3-го уровня

Таким образом, применение шага геометрической многосеточной инициализации позволяет выбрать начальное приближение численного решения. При этом полученная в результате дискретизации на каждой из сеток система разностных уравнений решается с использованием алгебраического многосеточного решателя. Совместное применение геометрической многосеточной инициализации и алгебраического многосеточного метода обеспечивает более быстрый и устойчивый процесс сходимости численного решения, что показано на задаче далее.

Численные эксперименты

Описанная физико-математическая модель реализована в пакете программ ЛОГОС – отечественном программном продукте, предназначенном для решения трехмерных задач тепломассопереноса, аэродинамики и гидродинамики на параллельных ЭВМ. Пакет программ ЛОГОС ориентирован на решение задач вычислительной гидродинамики на произвольных неструктурированных сетках и успешно прошел верификацию на серии различных гидродинамических задач, включая расчеты турбулентных и геофизических течений.

Течение в канале с клином

Для исследования зависимости скорости сходимости численного решения от использования геометрической многосеточной инициализации рассматривается задача моделирования сверхзвукового течения в канале с клином. Данная задача имеет аналитическое решение и является известным тестом для верификации численных схем и алгоритмов.

По условию задачи во входном сечении канала (рис. 7) задается статическое давление 101325 Па, температура $T = 300°$ K и скорость потока 697.52 м/с, что соответствует числу Маха 2.

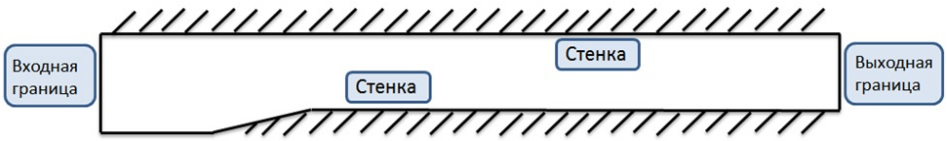


Рис. 7. Схема задачи

Исходная неструктурированная сетка состоит из 95000 ячеек, представлена на рис. 8.



Рис. 8. Расчетная сетка

При решении задачи использовался алгоритм геометрической многосеточной инициализации, в результате работы которого было получено следующее начальное приближение решения (рис. 9). Из рис. 9 видно, что в канале формируется ударно-волновая структура, распространяющаяся по его длине. Однако сеточного разрешения не хватило, чтобы сохранит интенсивность ударной волны, в результате чего волна в конце канала была «размыта».

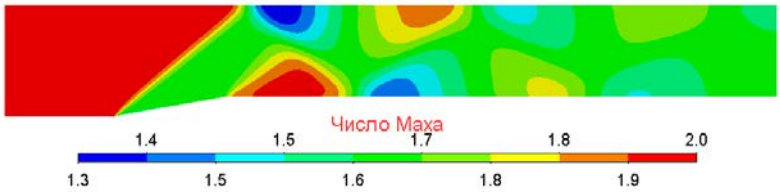


Рис. 9. Поле распределения числа Маха (решение, начальная инициализация)

Далее, полученное решение использовалось как начальная инициализация при расчете на исходной сетке. В результате было получено решение, показанное на рис. 10.

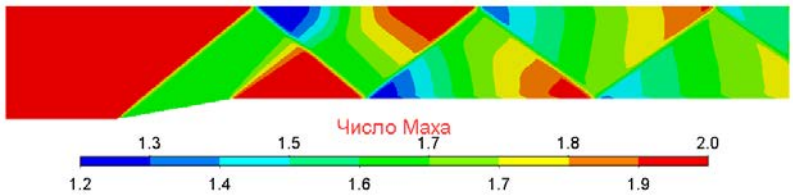


Рис. 10. Поле распределения числа Маха (конечное решение)

Можно отметить, что начальная инициализация и конечное решение имеют одинаковую структуру течения (система ударных волн, отразившихся от стенок канала) и отличаются лишь интенсивностью ударно-волновой структуры, что объясняется сеточным разрешением. График распределения по длине канала полного давления в сравнении с аналитическим решением представлен на рис. 11.

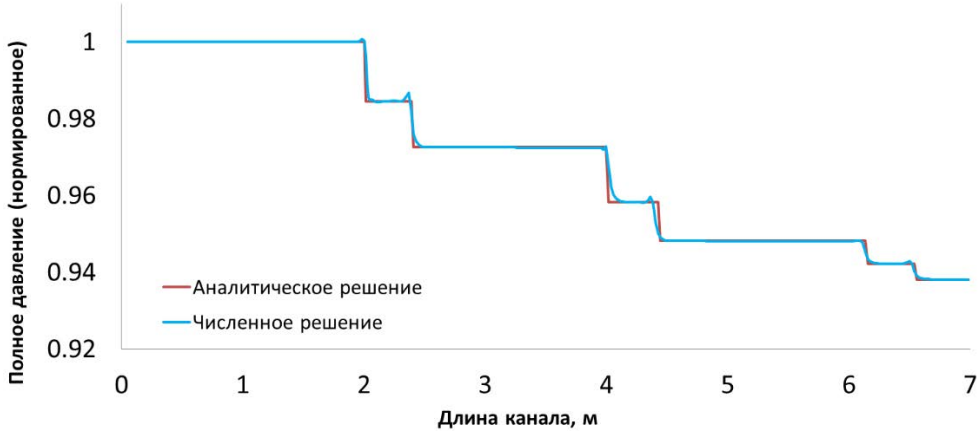


Рис. 11. График распределения полного давления

Оценим влияние начальной инициализации на скорость сходимости решения. Для этого сравним график установления величины расхода через выходную границу канала в случае использования алгоритма геометрической инициализации перед началом решения и без использования такового.

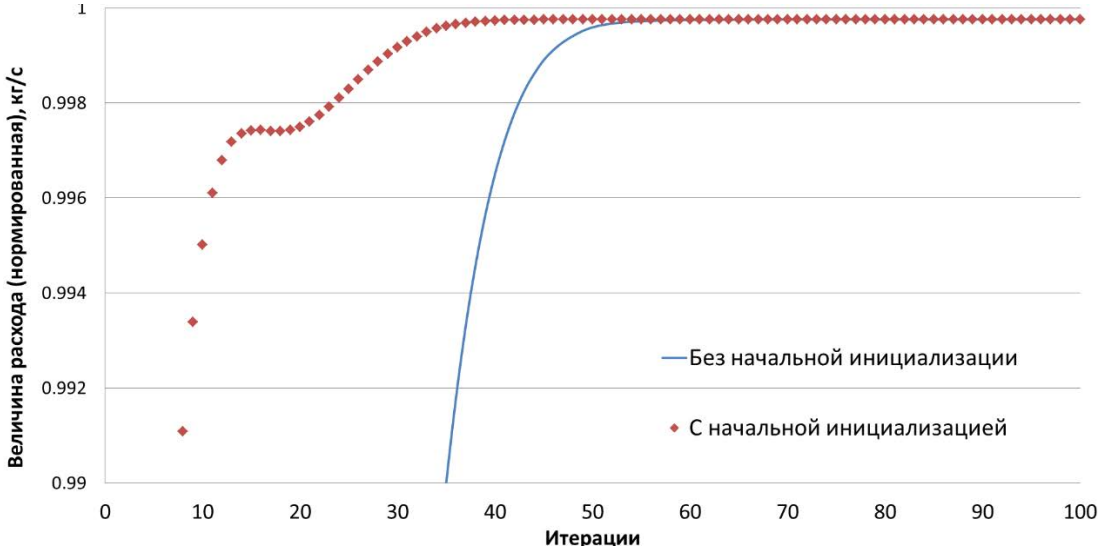


Рис. 12. График сходимости величины расхода

В случае использования начальной инициализации решение сошлось за 30 итераций, в то время как без нее – примерно, за 60 итераций. Как отмечалось выше, задача является простым тестом для верификации численных схем и алгоритмов и характеризуется быстрым процессом сходимости. Именно поэтому в данной задаче разница в скорости сходимости полностью нивелируется затраченным временем на выполнение алгоритма многосеточной инициализации.

По результатам можно сделать вывод, что начальная инициализация положительно влияет на скорость сходимости численного решения и при одинаковых параметрах решателя уменьшает число итераций для его получения.

Гиперзвуковое обтекание конуса

Для демонстрации полной возможности алгоритма многосеточной инициализации, в части ускорения сходимости численного решения и повышения его устойчивости, рассматривается задача обтекания конуса гиперзвуковым потоком вязкого сжимаемого газа.

Для постановки численного эксперимента на входной границе области обтекания (рис. 13) задается статическое давление 4.5 Па, температура $T = 216°$ K и скорость потока, равное числу Маха 7. Поверхность конуса считается твердой стенкой. Число ячеек в расчетной сетке составляет 557039.

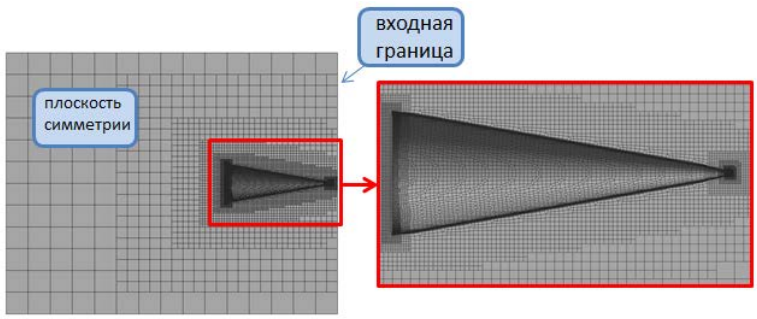


Рис. 13. Общий вид расчетной области

В результате применения алгоритма геометрической многосеточной инициализации и расчета на последовательности грубых сеток, было получено начальное приближение решения, показанное на рис. 14.

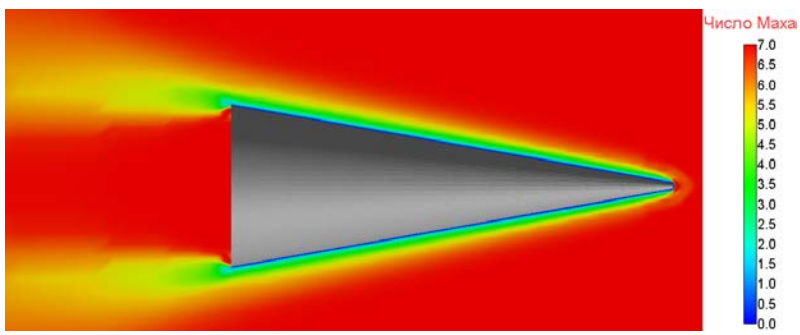


Рис. 14. Поле распределения числа Маха (решение, начальная инициализация)

Видно, что начальное возмущение, заданное на внешней границе, прошло расчетную область, и вблизи обтекаемого конуса сформировалась структура течения. Однако, как и в предыдущем примере, сеточного разрешения грубых сеток не хватило для более детального и качественного разрешения всех структур. Например, в донной области наблюдаются достаточно «размытые» отрывные зоны.

Далее, полученное решение использовалось как начальная инициализация при расчете на исходной сетке. В результате было получено следующее решение (рис. 15).

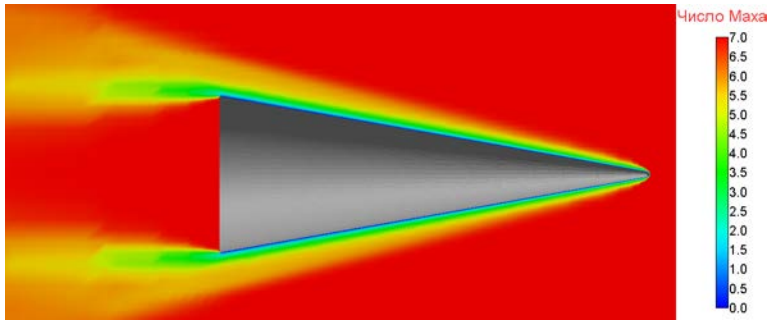


Рис. 15. Поле распределения числа Маха (конечное решение)

Можно отметить, что начальная инициализация и конечное решение имеют одинаковую структуру течения, отличаются лишь интенсивностью ударно-волновой структуры и их качеством разрешения. Оценим скорость сходимости решения в задаче с использованием начальной инициализации и без нее. Для этого сравним график сходимости величины силы лобового сопротивления.

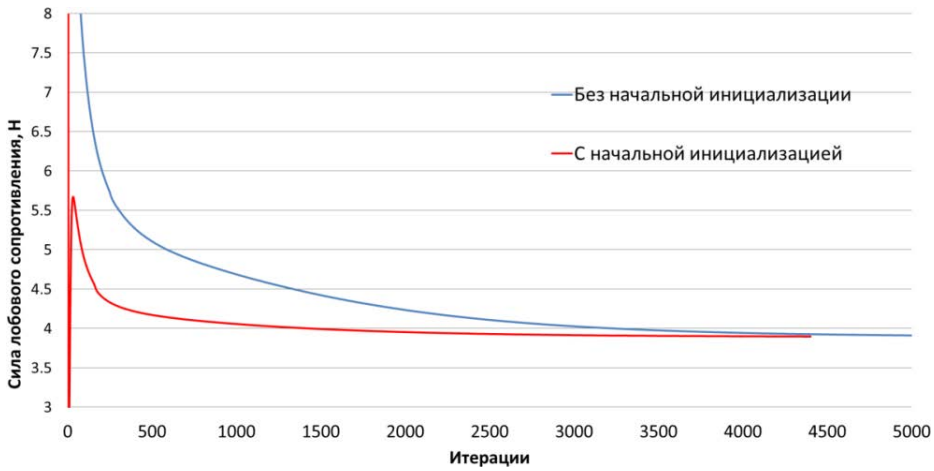


Рис. 16. График сходимости силы лобового сопротивления

Видно, что оба решения сходятся к одному значению, однако в случае использования начальной инициализации решение сошлось за 3000 итераций, в то время как без нее – примерно, за 4000 итераций. В табл. 1 представлена оценка времени при решении данной задачи в обоих вариантах.

Таблица 1

Способ решения	Время начальной инициализации, с	Число итераций	Общее время, с (время одной итерации 0.65 с)
Без начальной инициализации	–	4000	2600
С начальной инициализацией	45.55	3000	1995.55

Таким образом, с применением многосеточной начальной инициализации удалось сократить время решения задачи на ~23 %. Кроме того, применение начальной инициализации позволило вести счет с максимальным начальным значением числа Куранта, что говорит о повышении устойчивости численного решения. В то время как без использования алгоритма – значение числа Куранта приходится увеличивать постепенно в процессе счета задачи.

Следует отметить, что использование алгоритма многосеточной инициализации не «уводит» решение от истинного значения, о чем говорит график сходимости силы лобового сопротивления (рис. 16).

Заключение

В работе представлен алгоритм построения последовательности грубых сеток и используемая математическая модель. Формирование новых контрольных объемов происходит на основе взвешенного графа, содержащего необходимую информацию об элементах сетки. В качестве критерия для объединения ячеек в контрольные объемы рассматривается отношение площади к объему.

Описанный алгоритм был протестирован на примере решения задач сверх– и гиперзвукового течения.

Показано, что на рассматриваемой задачи применение многосеточной технологии может сократить время решения на 23 %.

Литература

1. Козелков А. С., Курулин В. В., Лашкин С. В., Шагалиев Р. М., Ялозо А. В. Исследование потенциала суперкомпьютеров для масштабируемого численного моделирования задач гидродинамики в индустриальных приложениях // Журнал вычислительной математики и математической физики, 2016, Т. 56, № 8, С. 1524–1535.

2. Волков К. Н., Дерюгин Ю. Н., Емельянов В. Н., Карпенко А. Г., Козелков А. С., Тетерина И. В. Методы ускорения газодинамических расчетов на неструктурированных сетках. – Москва: Физматлит, 2013, С. 536.

3. Козелков А. С., Корнев А. В., Стрелец Д. Ю., Танненберг И. А., Останко А. А., Расчетные исследования аэродинамических характеристик сверхзвукового самолета на крейсерских режимах полета // Полет. Общероссийский научно-технический журнал, 2017, № 6, С. 17–21.

4. Moinier P., Muller J–D., Giles M.B. Edge-based multigrid and preconditioning for hybrid grids // AIAA Journal, 2002, V. 40, №10, P. 1954–1960.

5. Mavriplis D. J. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers // Journal of Computational Physics. 2002. Vol. 175. № 1. P. 302–325.

6. Trottenberg U., Oosterlee C., Schuller A. Multigrid. London: Academic Press, 2001. P. 588.

SPECIAL ASPECTS OF USING THE GEOMETRIC MULTILEVEL INITIALIZATION METHOD TO SPEED UP AERODYNAMIC SIMULATIONS ON UNSTRUCTURED GRIDS

A. V. Struchkov, A. S. Kozelkov, R. N. Zhuchkov

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents a geometric multilevel initialization-based algorithm to speed up aerodynamic simulations [1]. This algorithm helps improve convergence rates and stability of numerical solution in the flow structure formation phase. The idea of the method is to generate a series of coarse grids [2–3] based on a parent grid to find solutions on each of the grids and to subsequently interpolate them to a finer grid. The algorithm can be used on any unstructured grids with an arbitrary cell shape. As a cell combination criterion to form new control volumes in the successive grid coarsening, the algorithm uses a relationship calculated from face areas and cell volumes. The cell combination process is based on the analysis of the weighted graph. Stability analysis of the solution was performed by simulations of an aerodynamic benchmark problem [5]. As a result, the algorithm shortened the runtime by 23 percent.

Key words: aerodynamics, multigrid solver, weighted graph, merging criterion, agglomeration.

УДК 517.9
DOI 10.53403/9785951505071_2022_473

АНАЛИЗ РЕЗУЛЬТАТОВ РАСПАРАЛЛЕЛИВАНИЯ ГРАДИЕНТНОГО АЛГОРИТМА РЕШЕНИЯ СИСТЕМ НЕЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

В. Г. Суфиянов, А. В. Новиков

Ижевский государственный технический университет им. М.Т. Калашникова, Ижевск

Целью данной работы – распараллеливание градиентного алгоритма решения СНАУ на GPU и анализ результатов решения СНАУ в зависимости от размерности задачи.

Ключевые слова: СНАУ, СЛАУ, распараллеливание градиентногго алгоритма, ArrayFire.

Многие практические задачи сводятся к решению систем нелинейных алгебраических уравнений (СНАУ). Например, задача о химически равновесном составе продуктов горения топлива [1] решается в процессе численного моделирования процессов горения топлива в ракетных двигателях. При этом необходимо многократно решать уравнения в каждой ячейке сетки и на каждом временном слое, что при большом числе ячеек ($>10^6$) является проблемой, так как требуется значительные вычислительные мощности.

Основным методом решения систем нелинейных уравнений является градиентный метод методом Ньютона, в котором на каждой шаге итерации решается система линейных алгебраических уравнений (СЛАУ). Метод Ньютона обладает вычислительной неустойчивостью при большом значении числа обусловленности матрицы Якоби. Сходимость алгоритма и время решения системы нелинейных уравнений зависит от выбора начального приближения [2]. Наряду с градиентными методами используются и другие методы, например, метод простой итерации, генетические алгоритмы [3] и другие, которые также обладают рядом недостатков.

Одним из способов повышения скорости решения систем СНАУ является распараллеливание вычисления на многопроцессорных ЭВМ. Современное параллельное программирование для высокопроизводительных систем подразумевает многопоточность для многоядерных процессоров и ускорителей, которые могут быть реализованы, например, с помощью технологий MPI, OpenMP, OpenCL для центральных процессоров (CPU) и CUDA, OpenCL, OpenACC для графических процессоров (GPU).

В данной работе использовалась универсальная библиотека ArrayFire с открытым исходным кодом на языке C++, которая упрощает процесс разработки программного обеспечения, ориентированного на параллельные и массивно-параллельные архитектуры, включая центральные процессоры, графические процессоры и другие устройства. В библиотеке ArrayFire реализованы возможности решения задач линейной алгебры на параллельных системах, например, арифметические действия с матрицами и векторами, транспонирование, нахождении обратных матриц, вычисление определителей, решение СЛАУ и др. Одной из важных особенностей библиотеки ArrayFire является возможность написания одного кода программы на языке C++ для различных платформ CUDA и OpenCL.

Проведено исследование времени решения СНАУ градиентным методом на тестовом примере в зависимости от размерности задачи. В результате проведенных вычислений показано, что распараллеливание, с использованием библиотеки ArrayFire, позволяет значительно сократить время решения СНАУ с использованием GPU. «Узким» местом при решение этих задач является процедура

передачи данных из оперативной памяти на вычислительные устройства. Время на передачу данных в некоторых случаях сопоставимо с решением самой задачи. Решением данной проблемы является сокращение передачи данных из оперативной памяти на вычислительные устройства за счет вычислений векторов и матриц непосредственно на GPU.

1. Разработка градиентного алгоритма решения СНАУ

Рассмотрим систему нелинейных уравнений в общем виде

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0, \end{cases} \quad (1)$$

где $x_1, x_2, \ldots, x_n$ – искомые переменные; n – размерность задачи.

Перепишем систему (1) в векторном виде

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix} = \mathbf{0}, \quad (2)$$

где $\mathbf{f}(\mathbf{x})$ – вектор-функция; $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ – вектор искомых переменных; $\mathbf{0} = (0, 0, \ldots, 0)^T$ – нулевой вектор.

При нахождении достаточно хорошего начального приближения к решению системы уравнений (1) эффективным методом решения задачи является метод Ньютона, который можно представить в виде

$$\mathbf{x}^{k+1} = \mathbf{F}(\mathbf{x}^k)$$

с начальным приближением

$$\mathbf{x}^0 = (x_1^0, x_2^0, \ldots, x_n^0)^T,$$

где вектор-функция $\mathbf{F}(\mathbf{x})$ имеет вид

$$\mathbf{F}(\mathbf{x}) = \mathbf{x} - \mathbf{J}^{-1}\mathbf{f}(\mathbf{x}); \quad (3)$$

$\mathbf{J} = \left[\dfrac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]$ – матрица первых производных (матрица Якоби) функций $\mathbf{f}(\mathbf{x})$ системы нелинейных уравнений (2) по элементам вектора $\mathbf{x}$.

Итерационный процесс представляется в виде:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]^{-1}_{\mathbf{x}=\mathbf{x}^t} \mathbf{f}(\mathbf{x}^t). \quad (4)$$

Согласно [4] система (4) представляется в виде системы линейных алгебраических уравнений (СЛАУ):

$$\left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^t} \Delta^{t+1} = -\mathbf{f}(\mathbf{x}^t), \quad (5)$$

где $\Delta^{t+1} = \mathbf{x}^{t+1} - \mathbf{x}^t$ – разность между значениями приближениями на текущей и предыдущей итерациях. Решение СЛАУ (5) находится методом Гаусса, LU– или QR-разложения, итераций Зейделя и др. Таким образом, после вычисления разности Δ^{t+1} текущее приближение определяется по формуле

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta^{t+1}.$$

Одно из следующих условий позволяет завершить итерационный процесс:

– число итераций $t > \mathrm{MaxIter}$, где $\mathrm{MaxIter}$ – максимальное число итераций;

– норма приращения $\left\|\Delta^{t+1}\right\| < \varepsilon_x$, где ε_x – минимальное значение приращения аргумента;

– норма вектор функции $\left\|\mathbf{f}\left(\mathbf{x}^{t+1}\right)\right\| < \varepsilon_f$, где ε_f – минимальное отклонение функции от нулевого вектора.

Ускорение решения систем уравнений на GPU требует представления системы уравнений в векторно-матричном виде. В общем случае решение системы нелинейных уравнений (2) достаточно сложно привести к векторно-матричному виду, т.к. требуется вычислять нелинейные функции и матрицу Якоби. В случае СНАУ представим функции в виде

$$f_i(x_1,x_2,\ldots,x_n) = \sum_{j=1}^{k} a_{ij} \cdot \varphi_j(x_1,x_2,\ldots,x_n) - b_i, \tag{6}$$

где $\varphi_j(x_1,x_2,\ldots,x_n) = x_1^{p_{j1}} \cdot x_2^{p_{j2}} \cdot \ldots \cdot x_n^{p_{jn}}$ – слагаемые функции, соответствующие произведению искомых переменных в степени p_{jl}, $l = 1,2,\ldots,n$; k – количество различных слагаемых φ_j в СНАУ; a_{ij}, b_i – константы СНАУ, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,k$.

Представим константы СНАУ в векторно-матричном виде:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{7}$$

Кроме того, представим степени слагаемых функций в виде матрицы

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kn} \end{pmatrix}. \tag{8}$$

Матрицу множителей функций φ_j представим в виде:

$$\mathbf{Y} = \mathbf{X}^{\mathbf{P}} = \begin{pmatrix} x_1^{p_{11}} & x_2^{p_{12}} & \ldots & x_n^{p_{1n}} \\ x_1^{p_{21}} & x_2^{p_{22}} & \ldots & x_n^{p_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{p_{k1}} & x_2^{p_{k2}} & \ldots & x_n^{p_{kn}} \end{pmatrix}, \tag{9}$$

тогда вектор $\mathbf{y}$ значений функций $y_j = \varphi_j(x_1,x_2,\ldots,x_n)$, примет вид:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix} = prod_2(\mathbf{Y}) = \begin{pmatrix} x_1^{p_{11}} \cdot x_2^{p_{12}} \cdot \ldots \cdot x_n^{p_{1n}} \\ x_1^{p_{21}} \cdot x_2^{p_{22}} \cdot \ldots \cdot x_n^{p_{2n}} \\ \vdots \\ x_1^{p_{k1}} \cdot x_2^{p_{k2}} \cdot \ldots \cdot x_n^{p_{kn}} \end{pmatrix}, \tag{10}$$

где $prod_2(\mathbf{Y})$ – оператор, соответствующий произведению элементов матрицы $\mathbf{Y}$ по строкам.

В результате получим векторно-матричное представление для вычисления вектор-функции $\mathbf{f}(\mathbf{x})$:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{y} - \mathbf{b}. \tag{11}$$

Аналогичным образом запишем матрицу Якоби:

$$\mathbf{J} = \left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right] = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \ldots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \ldots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \ldots & \frac{\partial f_n(\mathbf{x})}{\partial x_n} \end{pmatrix}, \tag{11}$$

где элементы матрицы Якоби можно представить в виде:

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} = \sum_{l=1}^{k} a_{il} \cdot \frac{\partial \varphi_l((\mathbf{x}))}{\partial x_j}. \tag{12}$$

Вычисление производных слагаемых можно представить в следующем виде

$$\frac{\partial \varphi_l(\mathbf{x})}{\partial x_j} = \begin{cases} 0, & p_{lj} = 0, \\ p_{lj} \cdot x_1^{p_{l1}} \cdot x_2^{p_{l2}} \ldots \cdot x_j^{p_{lj}-1} \cdot \ldots \cdot x_n^{p_{ln}}, & p_{lj} \neq 0. \end{cases} \tag{13}$$

Следующее представление производной (13) позволяет сократить затраты на ее вычисление

$$\frac{\partial \varphi_l(\mathbf{x})}{\partial x_j} = \begin{cases} 0, & p_{lj} = 0, \\ p_{lj} \cdot \dfrac{\varphi_l(\mathbf{x})}{x_j}, & p_{lj} \neq 0, \end{cases} \tag{14}$$

где $y_l = \varphi_l(\mathbf{x})$ – известное значение слагаемого, но, в этом случае, возникают сложности при вычислении функций при $x_j \approx 0$ и необходимо устанавливать дополнительные условия проверки точности вычислений.

В результате каждый столбец матрицы Якоби можно вычислить в векторно-матричном виде:

$$\mathbf{J}(:, j) = \mathbf{A} \cdot \begin{pmatrix} \dfrac{\partial \varphi_1(\mathbf{x})}{\partial x_j} \\ \dfrac{\partial \varphi_2(\mathbf{x})}{\partial x_j} \\ \vdots \\ \dfrac{\partial \varphi_k(\mathbf{x})}{\partial x_j} \end{pmatrix}. \tag{15}$$

где $\mathbf{J}(:, j)$ – соответствует j -ому столбцу матрицы Якоби.

Таким образом, получаем полностью векторно-матричный алгоритм решения СНАУ методом Ньютона, который реализуется программными средствами распараллеливания алгоритмов, например, для GPU – это технологии CUDA или OpenCL.

2. Анализ результатов решения СНАУ на GPU

Рассмотрим тестовый пример для решения СНАУ вида

$$\begin{cases} x_1^2 + x_2^2 + \ldots + x_n^2 - n = 0, \\ x_1 - x_2 = 0, \\ x_2 - x_1 = 0, \\ \ldots \\ x_{n-1} - x_n = 0. \end{cases} \tag{16}$$

где размерность задачи $n \geq 2$.

Система (16) имеет два точных решения:

$$\mathbf{x}_1^* = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}, \quad \mathbf{x}_2^* = \begin{pmatrix} +1 \\ +1 \\ \vdots \\ +1 \end{pmatrix}. \tag{17}$$

В случае $n = 2$ решение можно представить графически (рис. 1).

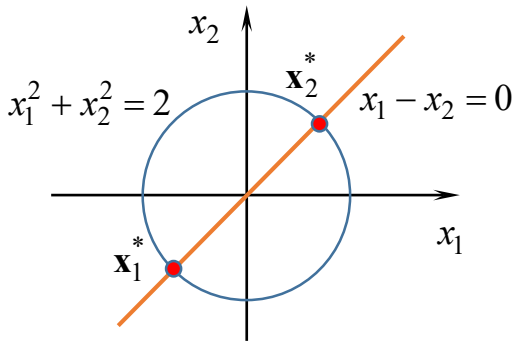


Рис. 1. Решение СНАУ для тестового примера при n=2

На рис. 1 представлено очевидные решения $\mathbf{x}_1^* = (-1, -1)$ и $\mathbf{x}_2^* = (+1, +1)$ системы двух уравнений, первое представляет собой окружность радиуса $\sqrt{2}$ с центром в начале координат и прямую, проходящую через начало координат.

Константы тестовой СНАУ в векторно-матричном представлении имеют вид

$$\mathbf{A} = \left(\begin{array}{cccccc|cccccc} 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right), \quad \mathbf{b} = \begin{pmatrix} -n \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{17}$$

Матрица степеней слагаемых функций записывается в виде

$$\mathbf{P} = \left(\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \hline 2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \end{array}\right). \tag{18}$$

В качестве начального приближения принималось значение:

$$\mathbf{x}^0 = \begin{pmatrix} 2 \\ 3 \\ \vdots \\ n+1 \end{pmatrix}. \tag{18}$$

Решение тестовой СНАУ осуществлялось в операционной системе Windows 10 64 bit. В качестве интегрированной среды разработки использовался MS Visual Studio 2019. Программа была написана на языке программирования C++ с использованием библиотеки ArrayFire v. 3.8.0 и технологии OpenCL 2.0.

Тестирование программы осуществлялось на 4 ядерном CPU Intel Core i7 2600 с частотой 3 400 МГц, объем оперативной памяти 16 Гб. В качестве вычислительного устройства использовался

графический процессор AMD Radeon VII с числом ядер – 3840 и базовой частота 1 400 МГц, объем графической память HBM2 составила 16 Gb.

В качестве условия выхода из итераций задавались два условия: максимальное число итераций задавалось равным $\mathrm{MaxIter} = 100$ и минимальное значение приращение аргумента $\varepsilon_x = 10^{-5}$.

Тестовая задача решалась при заданной размерности $n = 2, 10, 20, \ldots, 500$. Зависимость числа итераций от размерности решаемой тестовой СНАУ представлена на рис. 2.

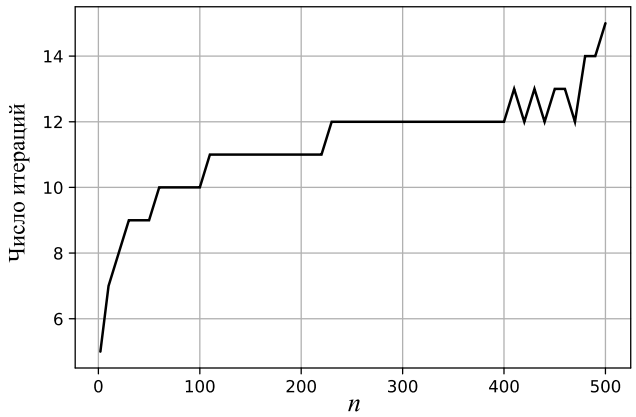


Рис. 2. Число итераций при решения тестовой СНАУ
в зависимости от размерности задачи n

Как видно из рис. 2, число итераций, требуемое для достижения заданной точности нелинейно растет с увеличением размерности тестовой задачи. Зависимость погрешности в логарифмических координатах от числа итераций для рассматриваемой тестовой задачи размерности $n = 500$ представлена на рис. 3.

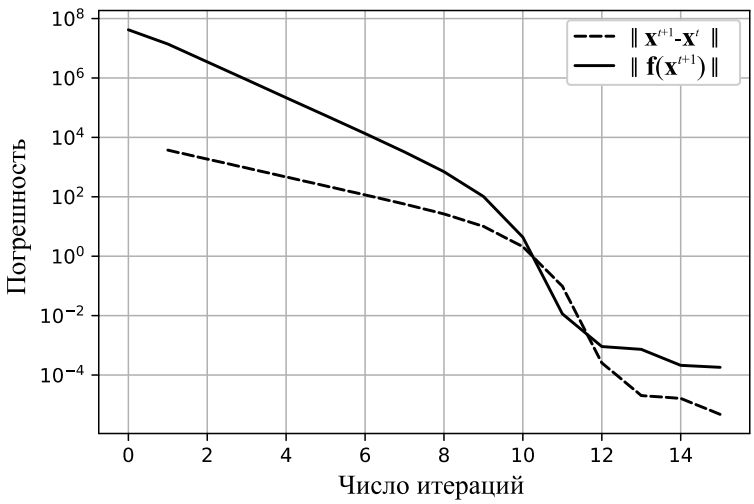


Рис. 3. Сходимость к точному решению от числа итераций
для тестовой СНАУ при $n = 500$

Как видно из рис. 3, значение приращения аргумента $\left\|\Delta^{t+1}\right\| = \left\|\mathbf{x}^{t+1} - \mathbf{x}^t\right\|$ и норма вектора функций $\left\|\mathbf{f}\left(\mathbf{x}^{t+1}\right)\right\|$ имеет одинаковый нелинейный характер убывания.

Зависимость времени решения тестовой СНАУ на CPU представлены на рис. 4 и на GPU – на рис. 5.

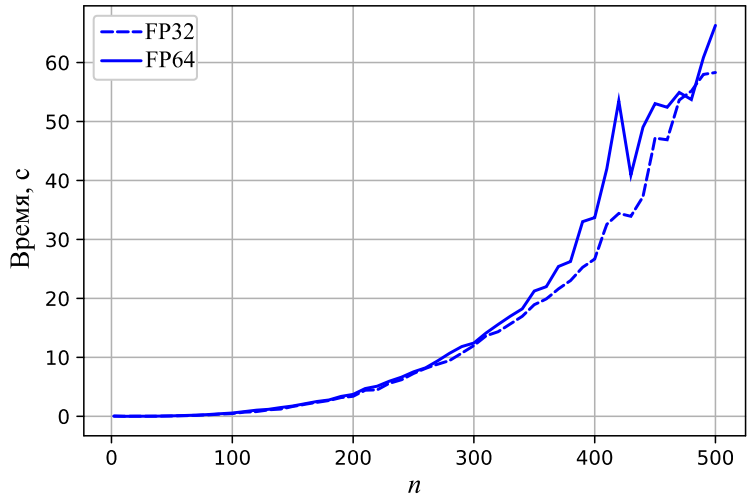


Рис. 4. Зависимость времени расчетов на CPU от размерности задачи с одинарной (FP32) и двойной точностью (FP64)

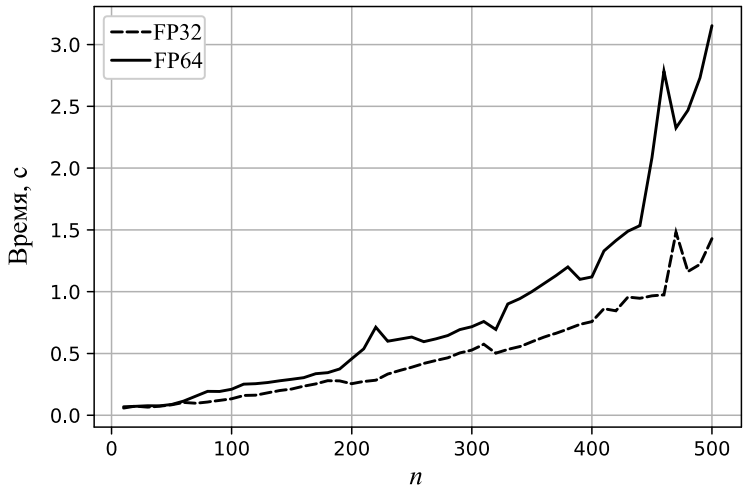


Рис. 5. Зависимость времени расчетов на GPU от размерности задачи с одинарной (FP32) и двойной точностью (FP64)

Из представленных выше рис. 4 и 5 видно, что время решения тестовой задачи растет нелинейно с увеличением размерности задачи. Сравнение рис. 4 и 5 показывает, значительное уменьшение времени решения тестовой СНАУ на GPU по сравнению с решением на CPU. Так, максимальное время расчета с одинарной точностью (FP32) на CPU составило 58,32 с, а на GPU – 1,43 с, т. е. время расчета уменьшилось в 40,8 раз. Максимальное время расчета с двойной точностью (FP64) на CPU составило 66,31 с, а на GPU – 3,15 с, т. е. время расчета уменьшилось в 21,0 раз.

Таким образом, можно сделать вывод, что значительный прирост производительности достигается за счет использования GPU с большим объемом памяти и большим числом вычислительных ядер. При этом для рассматриваемых вычислительных средств решение задачи с одинарной точностью (FP32) почти в 2 раза превосходит эффективность решения задачи с двойной точностью (FP64).

Анализ временных затрат на загрузку исходных данных на вычислительные устройства показал, что они весьма незначительные по сравнению со временем решения задачи.

Заключение

Таким образом, разработан обобщенный алгоритм решения системы СНАУ на графических процессорах с использованием библиотеки ArrayFire© и технологии параллельного программирования OpenCL. Исследование скорости сходимости алгоритма показало, что с увеличением размерности задачи увеличивается число итераций, а также наблюдается нелинейная сходимость к решению СНАУ.

Проведено исследование временных затрат на решение в зависимости от размерности задачи на тестовом примере для задач различной размерности. Проведенный анализ временных затрат показал, что распараллеливание алгоритма на GPU позволяет уменьшить время решения СНАУ с одинарной точностью (FP32) в 40 раз по сравнению со временем решения на CPU и в 21 раз – при решении с двойной точностью (FP64).

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-01-00072

Литература

1.Алемасов В. Е., Ваничев А. П., Гришин С. Д., Ильинский В. А., Дрегалин А. Ф., Тишин А. П., Илларионов Н. В. Термодинамические и теплофизические свойства продуктов сгорания; под ред. Глушко В. П. В 5 томах. Т. 1. М.: ВИНИТИ АН СССР, 1971. С. 266.

2.Суфиянов В. Г., Новиков А. В. Распараллеливание градиентного алгоритма решения систем нелинейных уравнений с использованием библиотеки ArrayFire // Сборник материалов XII Всероссийского совещания, посвященного 100-летию Калашникова М. Т. (Ижевск, 16-189 октября 2018). С. 80–87.

3.Попова Е. В. Решение систем нелинейных алгебраических уравнений гибридными генетическими алгоритмами // Интеллектуальные системы в производстве. 2011. № 2 (18). С. 33–40

4.Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. –М.: Мир, 1975. –С. 560 .

EVALUATION OF THE EFFECT OF PARALLELING THE GRADIENT SOLVER FOR SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS

V. G. Sufiyanov, A. V. Novikov

Kalashnikov Izhevsk State Technical University, Izhevsk

The objective of this work is to parallelize the gradient solver for GPU-assisted computations of systems of nonlinear algebraic equations and to evaluate the results of such computations as a function of problem dimensionality.

Key words: system of nonlinear algebraic equations, system of linear algebraic equations, parallelization of gradient solver, ArrayFire.

УДК 004.4
DOI 10.53403/9785951505071_2022_481

КЛИЕНТ-СЕРВЕРНАЯ СИСТЕМА КОНТРОЛЯ ЛИЦЕНЗИЙ ЛОГОС

Д. А. Тишкин, А. А. Ярулина, А. В. Дьяков, Д. О. Козлов, А. Г .Надуев, П. А. Тюхтина, А. В. Шемякин

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Пакет программ ЛОГОС – это программный продукт для комплексного математического моделирования с использованием ресурсов современных супер-ЭВМ. ЛОГОС – это наукоемкий комплексный цифровой продукт, состоящий из различных компонент. Использование типовых решений для контроля лицензии цифрового продукта ЛОГОС не представляется возможным. Поэтому было принято решение о разработке системы лицензирования, позволяющей контролировать соблюдение пользователями пакета программ ЛОГОС условий лицензии на его использование. К функциям сервера лицензии ЛОГОС относится:

– проверка возможности запуска программных модулей, подключенных к серверу лицензий;

– ведение статистики использования пакета программ ЛОГОС:

• типы, общее время исполнения и количество запусков программных модулей;

• получение информации о текущих запущенных программных модулях;

• информация об используемых и свободных ресурсах лицензии для запуска программных модулей.

– предоставление средств управления сервером лицензии:

• веб-консоль администратора;

• консольная утилита управления.

– проверка целостности лицензионных файлов с целью исключения возможности их несанкционированной модификации пользователем.

Чтобы обеспечить выполнение требований к серверу системы лицензирования, лицензия должна содержать следующую информацию:

– цель использования;

– количество вычислительных ядер;

– количество одновременно решающихся задач.

Для централизации управления лицензией на множество копий программного обеспечения ЛОГОС было принято решение разработать сервер лицензий. Клиент сервера лицензии должен встраиваться в каждый компонент программного пакета ЛОГОС. Так же сервер лицензии должен предоставлять механизм доступа к данным лицензии и управления.

Ключевые слова: сервер лицензий, ЛОГОС, клиент, управление, TCP.

Взаимодействие клиента и сервера лицензии ЛОГОС

Общая схема взаимодействия клиента с сервером лицензии ЛОГОС изображена на рис. 1.



Рис. 1. Схема взаимодействия клиентов с сервером лицензии ЛОГОС

Архитектура сервера лицензии ЛОГОС

Сервер лицензии состоит из трех частей: TCP сервера [1], HTTP сервера и программного интерфейса сервера лицензии. TCP сервер принимает подключения от клиентов и через программный интерфейс определяет возможность их работы в соответствии с лицензией. HTTP сервер, встроенный в сервер лицензии, предоставляет пользователям информацию о функционировании сервера лицензии, текущей лицензии и подключенных клиентах, позволяет управлять работой сервера лицензии. Клиентской частью для TCP сервера являются счетные модули, для HTTP сервера – браузеры пользователей. Архитектура сервера лицензий изображена на рис. 2.

Сервер лицензии выполняет следующие функции:

– проверка возможности запуска программных модулей, подключенных к серверу лицензий;

– ведение статистики использования пакета программ ЛОГОС:

• типы, общее время исполнения и количество запусков программных модулей;

• получение информации о текущих запущенных программных модулях;

• информация об используемых и свободных ресурсах лицензии для запуска программных модулей.



Рис. 2. Структура сервера лицензии

– предоставление средств управления сервером лицензии:

• веб-консоль администратора;

• консольная утилита управления.

– проверка целостности лицензионных файлов с целью исключения возможности их несанкционированной модификации пользователем.

Система выдачи и учета лицензий ЛОГОС

Для генерации входного файла лицензии сервера лицензий, была разработана система выдачи и учета лицензий ЛОГОС.

Система представляет из себя веб-приложение, запускаемое в веб-браузере пользователя. Система позволяет создавать и редактировать лицензионные файлы, хранить историю выдачи лицензионных файлов пользователям, а также она имеет разграничение прав на выдачу лицензий с помощью LDAP аутентификации.

Интерфейс пользователя системы выдачи и учета лицензий ЛОГОС показан на рис. 3.

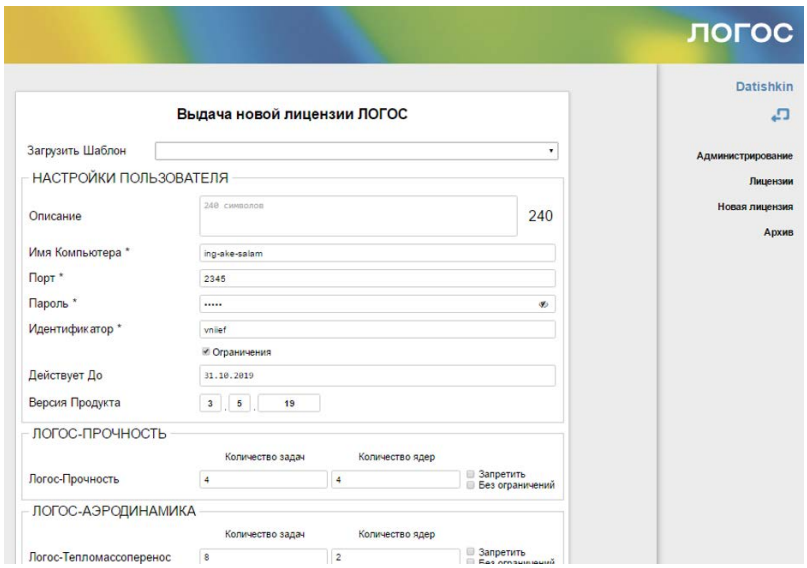


Рис. 3. Интерфейс пользователя системы выдачи и учета лицензий ЛОГОС

После создание лицензионного файла, можно просмотреть историю выдач лицензий пользователям, это окно изображено на рис. 4.

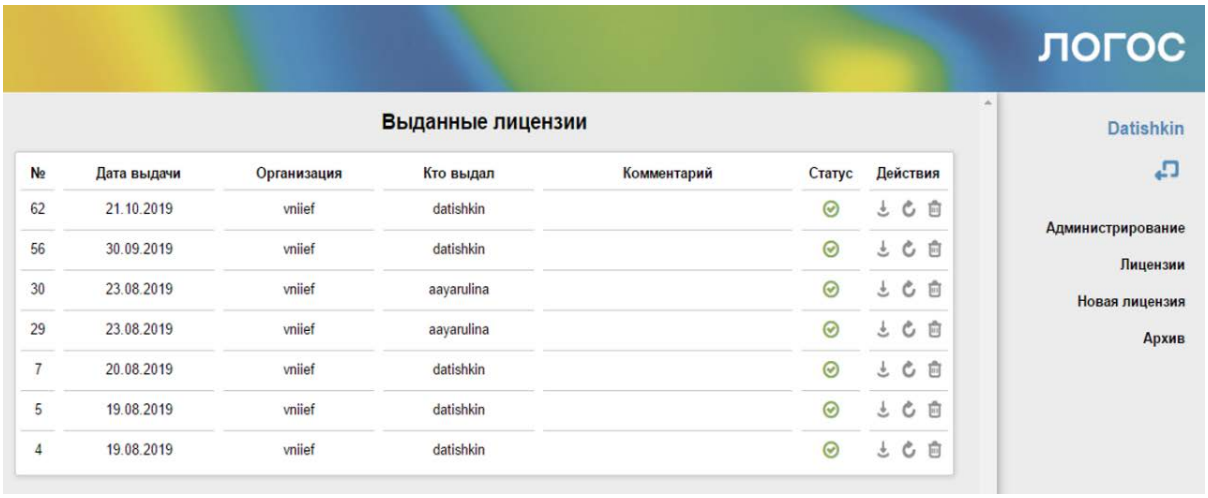


Рис. 4. История выдач лицензий с помощью системы выдачи и учета лицензий ЛОГОС

Так же с помощью кнопки скачать, расположенной с правой стороны интересующей строки с лицензией, можно сохранить файл лицензии на локальный компьютер, чтобы в дальнейшем использовать его с сервером лицензий.

Взаимодействие сервера лицензии с клиентами

На начальном этапе работы клиент выполняет процедуру инициализации для совместной работы с сервером лицензии, на сервере осуществляется проверка возможности запуска клиента, путем проверки лицензии. После инициализации клиент параллельно со своей основной работой отправляет уведомления серверу лицензии с определенным интервалом о продолжении работы. Сервер анализирует данные уведомления и отправляет клиенту сообщение о возможности продолжения работы. Когда клиент пытается завершить свою работу, он уведомляет об этом сервер. Сервер после данного уведомления освобождает ресурсы, занимаемые клиентом.

Сервер лицензии взаимодействует с клиентскими приложениями посредством сообщений, которые могут иметь следующие типы:

– "init" – инициализационное сообщение от клиента;

– "hb" – сообщение о подтверждении работы клиента;

– "resp" – ответ сервера в случае успешного выполнения запроса клиента;

– "fin" – ответ сервера в случае некорректного запроса (например, клиент запрашивает большее количество ресурсов, чем позволяет лицензия, или истекло время возможности подтверждения клиентом своей работы) или ответ на запрос клиента о завершении.

Обмен сообщениями производится посредством протокола TCP.

Программный интерфейс взаимодействия с сервером лицензии предоставляет следующие возможности: инициализация клиентов для работы с сервером лицензии, проверка продолжения работы клиента с текущей лицензией, завершение работы клиентов с сервером лицензии. Взаимодействие ЛОГОС-ПреПост и счетных модулей с сервером лицензии показано на рис. 5.

Клиентская часть сервера лицензии представляет собой динамически подключаемую библиотеку lib_lic_client.

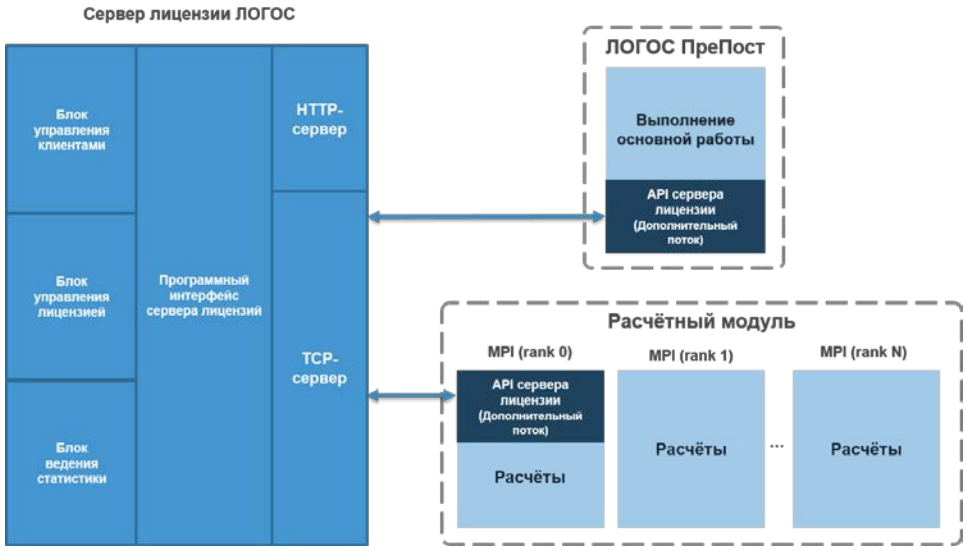


Рис. 5. Схема взаимодействия сервера лицензии с приложениями ЛОГОС-ПреПост и расчетными модулями

Для взаимодействия сервера лицензии с приложением ЛОГОС-ПреПост была реализована библиотека UPPLicClient, предоставляющая графический диалоговый интерфейс и использующая функции из клиентской библиотеки lib_lic_client для взаимодействия с сервером лицензий. Она осуществляет отправку уведомлений серверу лицензии о продолжении работы, с определенным интервалом; завершает работу приложения, в случае получения от сервера соответствующего уведомления; отправляет серверу лицензии уведомление о завершении приложения. С помощью предоставляемого графического интерфейса библиотеки UPPLicClient пользователь при превышении купленной лицензии получает диалоговое уведомление. В диалоговом уведомлении можно выбрать, перепроверить лицензию или закрыть приложение. При выборе пользователем перепроверки лицензии инициируется повторная проверка лицензии через сервер, если занятые ресурсы, купленной лицензии, освободились, то происходит запуск приложения, в противном случае снова появляется диалоговое уведомление о превышении лицензии.

Взаимодействие клиентской библиотеки сервера лицензий со счетными модулями осуществляется при помощи динамической библиотеки ccf_license_client, подключение которой к счетному модулю выполняет модуль сопряжения CCF_Bridge. При запуске расчета задачи, счетные модули, участвующие в расчете, в момент инициализации вызывают метод инициализации модуля сопряжения. Модуль сопряжения, в свою очередь, при инициализации, выполняет подключение библиотеки ccf_license_client. Далее из модуля сопряжения вызывается метод инициализации клиентской библиотеки сервера лицензии. Во время выполнения данного метода копии библиотеки ccf_license_client, каждая из которых принадлежит своему mpi-процессу модуля, обмениваются между собой именами счетных модулей. В результате данного обмена формируется словарь, который хранит список счетных модулей, участвующих в расчете и число mpi-процессов, на которых они запущены. Копия библиотеки ccf_license_client, запущенная на mpi-процессе, номер которого равен 0, передает данную информацию серверу лицензий, который, в свою очередь, вернет клиенту информацию о возможности запуска расчета, таймаут опроса клиентом лицензионного сервера и уникальный идентификатор задачи. Полученный ответ о возможности запуска расчета рассылается с процесса с нулевым номером остальным. Если запуск не возможен, будет выведена информация о причине остановки расчета, счет будет прекращен. В случае положительного ответа будет запущен дочерний поток, который с определенным интервалом будет отправлять уведомления серверу лицензии о выполнении счета. Сервер анализирует данные уведомления и отправляет клиенту сообщение о возможности продолжения работы. Если продолжение расчета невозможно произойдет его остановка, также будет выведена информация о причине остановки. Когда из mpi-процесса счетного модуля с рангом 0 будет вызван метод завершения, клиент остановит свою работу и уведомит об этом

сервер. Работоспособность перечисленной выше функциональности обеспечивается также для счетных модулей, собранных без использования mpi. В данном случае будет отсутствовать mpi-обмен внутри библиотеки ccf_license_client.

Веб-консоль управления сервером лицензии

Для управления сервером лицензии была реализована веб-консоль, состоящая из нескольких взаимосвязанных HTML – страниц, схема изображена на рис. 6.



Рис. 6. Схема веб-консоли сервера лицензии

При первом запуске сервера лицензии будет отображен диалог, предлагающий загрузить файл лицензии, изображенный на рис. 7.

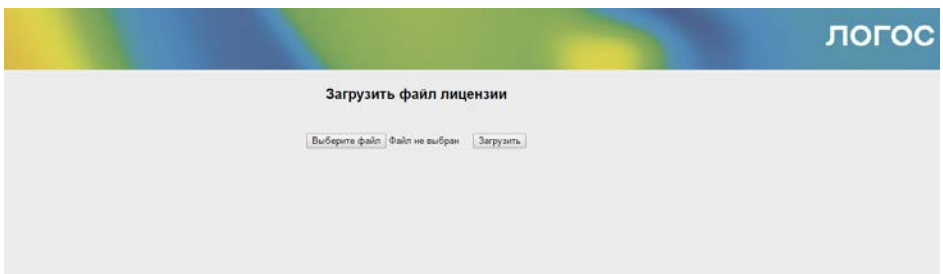


Рис. 7. Диалог загрузки файла лицензии

При успешной загрузке файла лицензии пользователь будет перенаправлен на страницу аутентификации, изображенную на рис. 8.

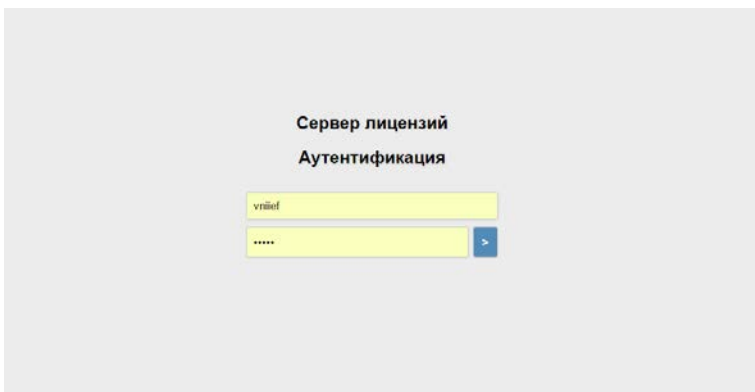


Рис. 8. Страница аутентификации

После успешной аутентификации, пользователь увидит главную страницу веб-консоли сервера лицензии ЛОГОС на которой отображена информация о текущей лицензии, это окно изображено на рис. 9.

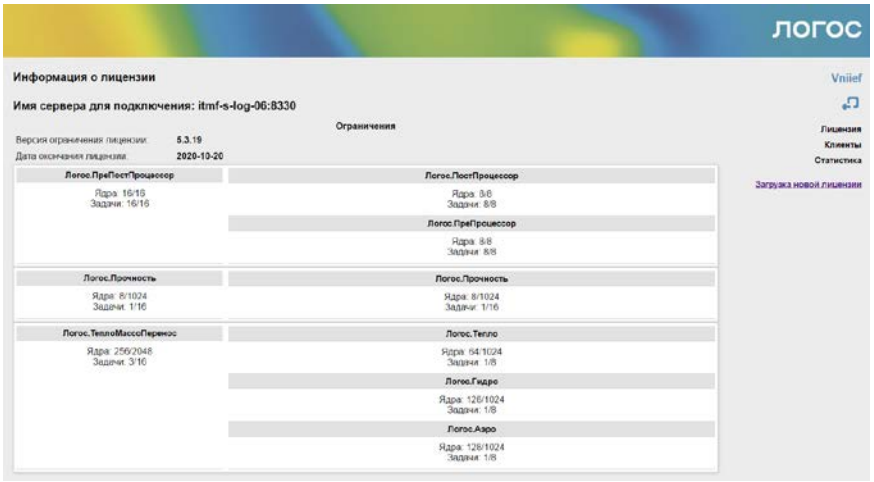
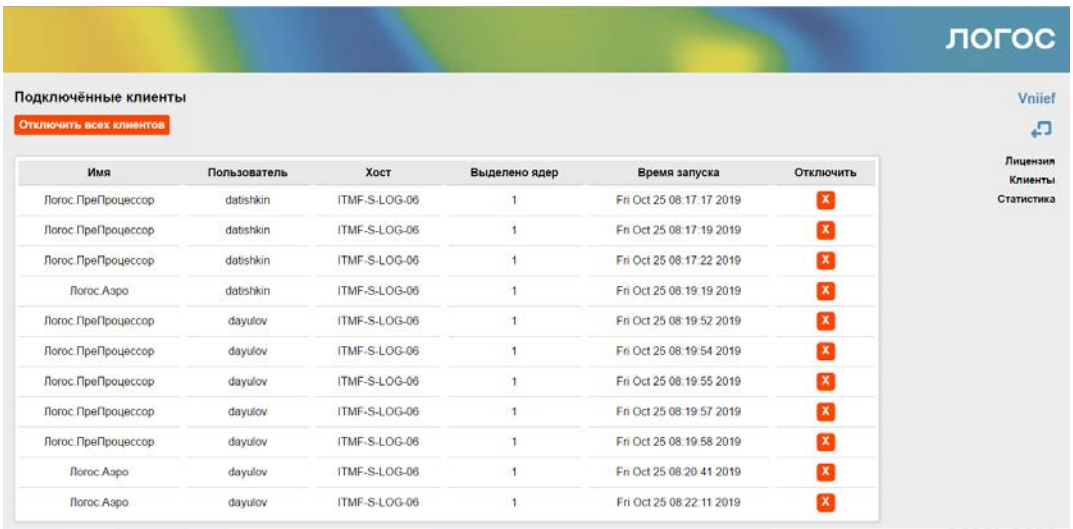


Рис. 9. Главная страница веб-консоли сервера лицензии

С правой стороны в веб-консоли располагается основное меню, в котором перечислены пункты с названиями страниц доступных пользователю.

На главной странице, она же страница «Лицензия», отображено общее количество доступных ресурсов лицензии, и количество использующихся в настоящее время ресурсов, а также есть ссылка на страницу загрузки нового файла лицензии.

На странице «Клиенты» отображаются, приложения, выполняющиеся в настоящий момент. Пример приведен на рис. 10.



Имя	Пользователь	Хост	Выделено ядер	Время запуска	Отключить
Логос.ПреПроцессор	datishkin	ITMF-S-LOG-06	1	Fri Oct 25 08:17:17 2019	X
Логос.ПреПроцессор	datishkin	ITMF-S-LOG-06	1	Fri Oct 25 08:17:19 2019	X
Логос.ПреПроцессор	datishkin	ITMF-S-LOG-06	1	Fri Oct 25 08:17:22 2019	X
Логос.Аэро	datishkin	ITMF-S-LOG-06	1	Fri Oct 25 08:19:19 2019	X
Логос.ПреПроцессор	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:19:52 2019	X
Логос.ПреПроцессор	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:19:54 2019	X
Логос.ПреПроцессор	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:19:55 2019	X
Логос.ПреПроцессор	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:19:57 2019	X
Логос.ПреПроцессор	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:19:58 2019	X
Логос.Аэро	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:20:41 2019	X
Логос.Аэро	dayulov	ITMF-S-LOG-06	1	Fri Oct 25 08:22:11 2019	X

Рис. 10. Страница «Клиенты», отображающая запущенные в настоящее время приложения

При переходе на страницу «Статистика» отобразится статистика выполнения всех приложений, использующих лицензию, как изображено на рис. 11. Верхняя таблица отображает агрегированную статистику, т. е. общее количество запусков приложения и ядро-часы (т. е. произведение количества используемых ядер на время выполнения). Нижняя таблица отображает название запущенного приложения, которое использовало лицензию, количество используемых ядер, время начала и завершения работы, имя пользователя от которого был произведен запуск и т. д. Информацию можно фильтровать и сортировать с помощью панели «Фильтры», расположенной справа под основным меню.

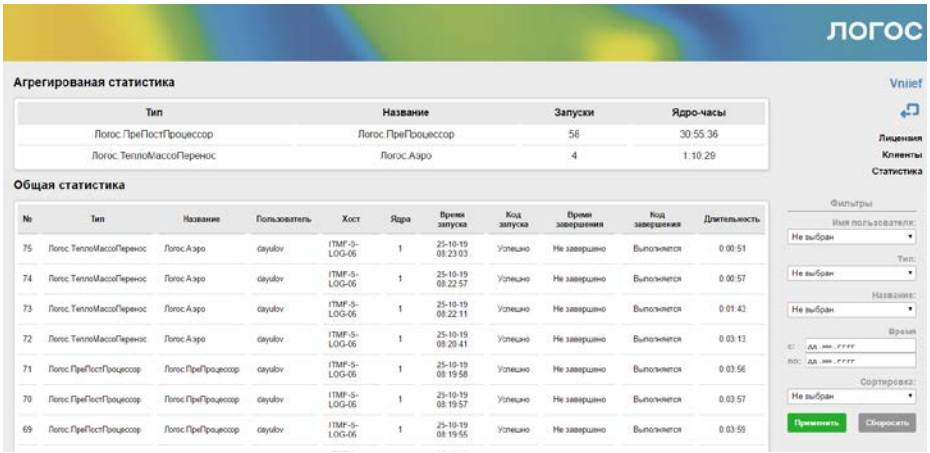


Рис. 11. Страница «Статистика», отображающая статистику использования всех приложений, которые используют лицензию

Заключение

Разработанная система лицензирования предоставляет возможность контроля использования лицензии программного пакета ЛОГОС. В результате работы:

– разработана клиент-серверная система контроля лицензий;
– разработана система выдачи и учета лицензий ЛОГОС;
– проведены предварительные испытания;
– по результатам испытаний функции сервера лицензии ЛОГОС реализованы в полном объеме;
– сервер лицензии будет входить в поставку дистрибутива ЛОГОС в этом году.

В дальнейшем планируется:

– расширение функциональности ограничения лицензии по времени использования;
– выгрузка базы данных статистики для ее анализа;
– увеличение количества собираемой для статистики информации.

Литература

1. Уолтон, Шон – Создание сетевых приложений в среде Linux. – М.: Издательский дом «Вильямс», 2001.

THE CLIENT-SERVER CONTROL SYSTEM FOR “LOGOS” LICENSES

D. A .Tishkin, A. A. Yarulina, A. V. Dyakov, D. O. Kozlov, A. V. Shemyakin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The LOGOS software package is a commercial software product for the integrated simulation using modern supercomputer resources. The LOGOS software package is a multi-purpose scientific digital product consisting of various components. The use of standard solutions to control the LOGOS licenses seems impossible. So, the decision is to develop a licensing system for monitoring how the LOGOS software users follow the terms and conditions of their licenses. The LOGOS license server functions include:

– checking a possibility of running program modules interconnected to the license server;
– producing statistical data on the usage of the LOGOS software package:
• the types, total runtimes, and total numbers of the program module runs;
• acquiring data on the current program modules in use;
• acquiring data on the resources in use and free resources of the license to run program modules.
– providing the license server management tools:
• the supervisor web-console;
• the command-line management tool.
– checking the license files for integrity to prevent unauthorized modifications by the user.

To ensure that the licensing system server requirements are met, each license should contain the following data:
– objectives;
– number of compute kernels;
– number of jobs, which are executed concurrently.

For the license control centralization, it was decided to develop a license server. The license server client must be inlined in each component of the LOGOS software package. Also, the license server must provide the data access arrangement and control mechanism.

Key words: license server, the LOGOS software package, client, control, TCP.

УДК 532.5
DOI 10.53403/9785951505071_2022_490

ПРИМЕНЕНИЕ НЕОТРАЖАЮЩИХ ГРАНИЧНЫХ УСЛОВИЙ ДЛЯ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ЗАДАЧ СО СВОБОДНОЙ ПОВЕРХНОСТЬЮ

Д. А. Уткин, Е. С. Тятюшкина, А. С. Козелков, В. В. Курулин

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В статье представлены способы реализации неотражающих граничных условий для задач со свободной поверхностью, моделируемых системой уравнений Навье – Стокса и методом VOF. Приводится описание двух подходов реализации неотражающих граничных условий – за счет вычисления интегральных выражений и за счет формирования демпфирующего слоя. Данные подходы сравниваются на примере решения задачи распространения солитона.

Ключевые слова: неотражающие граничные условия, метод VOF, уравнения Навье – Стокса, течение со свободной поверхностью.

Введение

Решение задач вычислительной гидродинамики зачастую требует использования неотражающих граничных условий, имитирующих бесконечное пространство за пределами расчетной области. Использование неотражающих граничных условий особенно актуально для задач со свободной поверхностью. Типичная задача – распространение волн цунами на расчетных областях с открытыми границами. Существуют некоторые методы и алгоритмы введения в математическую модель неотражающих граничных условий.

Распространенным является метод, основанный на использовании интегральных выражений, полученных, например, через характеристики [1, 2], либо через схемы высокого порядка [3–5], в дальнейшем применяемых для неотражающих границ. Метод используется для математических моделей распространения волн, основанных на уравнениях мелкой воды. Опыт применения данного подхода изложен, например, в [1, 2, 4]. Такая методика требует, чтобы расчетная область была разбита на структурированные счетные ячейки. В докладе приводится обобщение этого метода на трехмерные неструктурированные сетки. Так, в [2] авторы демонстрируют результат применения данного метода на задачи со свободной поверхностью, демонстрируя способность реализованной процедуры имитировать открытую границу. Однако, для демонстрации эффективности метода авторами не приведено информации об интенсивности отраженных волн по сравнению с набегающими.

Широко применяется подход, заключающийся в формировании зоны демпфирования волн в приграничной подобласти [1, 6–8]. Он удобен в реализации и заключается в добавлении дополнительного источника импульса, который действует в установленной приграничной области и функционирует как поглотитель волн. В литературе данный метод часто именуется «пористым слоем» (sponge layer) [9] и доступен в современных коммерческих пакетах для решения задач со свободной поверхностью по методике VOF.

В докладе представлена реализация данных двух методов и их сравнительное применение для задачи о распространении волн.

1. Численный метод

1.1. Математическая модель

Рассматривается математическая модель для решения задач распространения волн, основанная на системе уравнений Навье – Стокса совместно с методикой VOF [10]. Предполагается, что течение изотермическое, а поле скорости общее для всех фаз. Учитывая данные допущения, получим систему уравнений, состоящую из уравнения сохранения массы, уравнения сохранения импульса и уравнения переноса объемной доли, которая в декартовых координатах имеет вид:

$$\begin{cases} \dfrac{\partial u_i}{\partial x_i} = 0, \\ \dfrac{\partial u_i}{\partial t} + \rho \dfrac{\partial}{\partial x_j}\left(u_i u_j\right) = -\dfrac{\partial p}{\partial x_i} + \dfrac{\partial}{\partial x_j}\tau_{ij} + \rho g_i, \\ \dfrac{\partial \alpha_w}{\partial t} + \dfrac{\partial}{\partial x_i}\left(u_i \alpha_w\right) = 0, \end{cases} \tag{1}$$

где i, j – нижние индексы, указывающие на принадлежность векторных компонент к декартовым координатам, $i, j = \{x, y, z\}$, ρ – средняя плотность, вычисляемая как $\rho = \left(\rho_w \alpha_w + \rho_a \alpha_a\right)$, w – (water) нижний индекс, указывающий на принадлежность к фазе «вода», a – (air) нижний индекс, указывающий на принадлежность к фазе «воздух», α_w – объемная доля воды, u_i – компонента вектора скорости, $i = \{x, y, z\}$, t – время, p – давление, x_i – компонента вектора декартовых координат, $i = \{x, y, z\}$, τ_{ij} – тензор вязких напряжений, который, согласно гипотезе Буссинеска, принимает вид:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right), \tag{2}$$

где μ – динамическая вязкость, δ_{ij} – символ Кронекера, g_i – компонента вектора ускорения свободного падения.

Система уравнений решается путем численного интегрирования на конечно-объемной расчетной сетке. Для дискретизации уравнений используется оригинальный полностью неявный метод решения уравнений Навье – Стокса для расчета многофазных течений со свободной поверхностью [3].

Для дискретизации уравнений по времени используется трехслойная схема второго порядка точности [11]. Дискретизация конвективных членов в уравнении движения осуществляется по противопоточной схеме LUD [11], в уравнении переноса объемной доли фаз – по схеме HRIC[12], позволяющей предотвращать излишнюю численную диффузию раздела фаз. Учет силы тяжести производился с использованием алгоритма, основанного на поправке объемных сил [13].

Для данной математической модели на базе отечественного пакета программ ЛОГОС [14] были реализованы методы неотражающих граничных условий, которые подробно представлены и рассмотрены в разделах 1.2 и 1.3.

1.2. Метод открытой границы

Рассмотрим подход, реализованный в работе [2] для методики VOF на базе уравнений Навье – Стокса. Реализация неотражающих граничных условий, имитирующих открытые границы, основана на их выводе из уравнений теории мелкой воды. Ключевыми параметрами в контексте данного вывода являются параметры скорости течения и глубины воды. Для этого уравнения мелкой воды раскладывают на систему независимых уравнений с помощью инвариантов Римана, получив тем самым

гиперболические уравнения в частных производных. Далее, с помощью метода характеристик, получают аналитическое решение. Это аналитическое решение накладывается на выбранную границу, таким образом, позволяя определять значения скорости и глубины на ней, основываясь на данных о приходящей волне и на данных об уровне жидкости за пределами расчетной области. Техника реализации этих граничных условий для методики VOF требует вычисления интегралов, связанных с величиной объемной доли в областях при неотражающей границе.

$$h_1^n = \frac{\int_\Omega \alpha_1^n d\Omega}{\Omega} \qquad u_{1,i}^{n\,*} = \frac{\int_\Omega \left(\alpha_1^n u_{1,i}^n\right) d\Omega}{\Omega}, \tag{3}$$

где Ω – область интегрирования, являющаяся поверхностью неотражающей границы, α_1^n – объемная доля воды в слое ячеек, примыкающем к области Ω, $u_{1,i}^n$ – компонента скорости в слое ячеек, примыкающем к области Ω, h_1^n и $u_{1,i}^{n\,*}$ – значение глубины и скорости, вычисленные при области Ω.

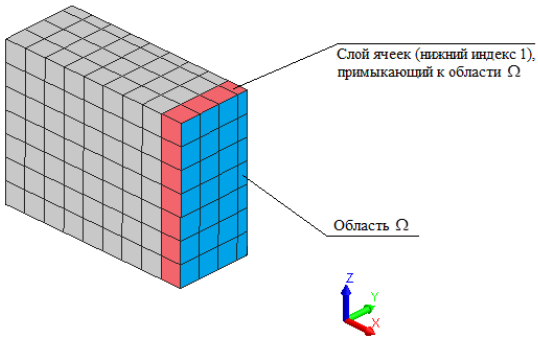


Рис. 1. Слой ячеек, примыкающий к области Ω

Величины, рассчитанные в (3), используются напрямую для вычисления значений объемной доли, а также скорости течения на рассматриваемой неотражающей границе:

$$h_b^n = \frac{1}{16|\vec{g}|}\left(2\sqrt{|\vec{g}|h_\infty} - u_1^n + 2\sqrt{|\vec{g}|h_1^n}\right)^2 \tag{4}$$

$$u_{b,i}^n = \frac{1}{2}\left(2\sqrt{|\vec{g}|h_\infty} + u_1^n - 2\sqrt{|\vec{g}|h_1^n}\right) \tag{5}$$

Величина объемной доли вычисляется по показаниям глубины воды h_b^n, рассчитанной на границе, относительно вертикальной оси z, которая, по условиям задачи, должна быть противоположно направлена вектору силы тяжести $\vec{g}$:

$$\alpha_b^n = \begin{cases} 0 & z \geq h_b^n \\ 1 & z < h_b^n \end{cases}$$

Основной вопрос, который связан с этим методом, заключается в том, как следует определить область для интегрирования в (3). Чтобы повысить точность вычисления этих интегралов, требуется разбивать область Ω неотражающей границы на несколько подобластей, по которым и следует вычислять интегралы. Данное обстоятельство накладывает ограничения на применяемую в численном моделировании расчетную сетку, а именно на слой ячеек, примыкающий к неотражающей границе. Далее следует рассмотреть особенности, связанные с этим алгоритмом – в качестве примера приводится граница, представленная на рис. 2:

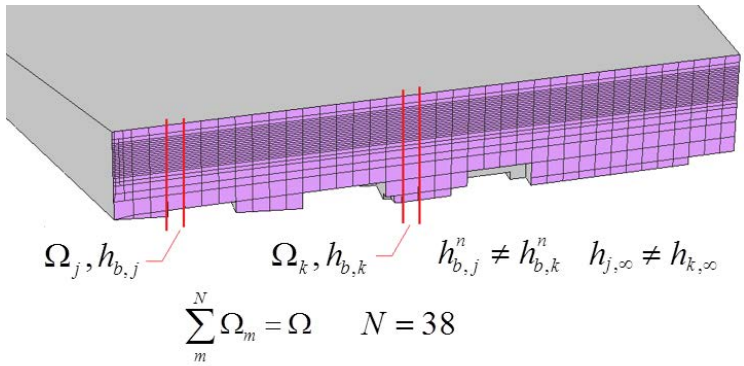


Рис. 2. Граница расчетной области

На рис. 2 представлен пример границы расчетной сетки, который рассматривается как неотражающая для рассмотренного метода. Данная расчетная область разбита на N подобластей для интегрирования Ω_m, причем подобласти не равны между собой, и глубина воды на границе $h^n_{b,m}$ вычисляется отдельно при каждой Ω_m. Помимо этого, требуется знать значение глубины воды за пределами расчетной области $h_{m,\infty}$, которая также изменяется при каждом отдельном участке Ω_m. Таким образом возрастает количество вычислительных операций для трехмерных задач, где неотражающая граница разбита на большое число подобластей Ω_m.

В таком виде данный метод был реализован, его применение для решения задач рассматривается в разделе 2.1.

1.3. Метод демпфирования волн

Метод демпфирования волн основан на использовании вблизи открытых границ демпфирующего приграничного слоя. В пределах данной зоны существует введенный источник импульса, задача которого заключается в том, чтобы уменьшить скорость приходящей в слой волны. Источник в общем случае может быть представлен формульным выражением:

$$I_i = -\max\left(\left(1-\frac{dL}{L}\right);0\right)\cdot k_{sour}\cdot u_i, \qquad (6)$$

где I_i – значение источника для уравнения сохранения импульса для компоненты скорости u_i, x_b – координата границы расчетной области, dL – кратчайшее расстояние до свободной границы, L – ширина зоны демпфирования, k_{sour} – параметр дампинга для источника. В данном методе свободными параметрами являются две величины: параметр k_{sour} и ширина демпфирующего слоя L. Данные параметры напрямую влияют на эффективность поглощения волн, и, следовательно, на амплитуду отраженных волн. Их оптимальные значения зависят от параметров приходящих волн: амплитуды волны, длины волны и глубины канала. Процедуре определения оптимального значения параметра U_{sour} и ширине демпфирующего слоя L посвящен раздел 2.2.

2. Численные эксперименты

2.1. Тестирование методики открытых границ

Для тестирования первого реализованного метода была выбрана задача распространения одиночной волны высотой H в водном бассейне. Волна с заданными параметрами распространяется от левой границы к правой неотражающей границе. Параметры волны: высота 0.32 м, длина канала 23 м, высота волны 0.064 м.

Начальный профиль волны задается следующей формулой:

$$\eta(x,0) = \mathrm{H}\,\mathrm{sech}^2\left(\gamma\left(x - X_s\right)\right),$$

где $\gamma = \sqrt{\frac{3H}{4d}}$, $L = X_s - X_0 = \frac{\arccos h\left(\sqrt{20}\right)}{\gamma}$.

Начальная скорость волны равна $u(x,0) = \sqrt{\frac{g}{d}}\eta(x,0)$.

Для численного моделирования использовалась расчетная сетка, состоящая из 200 тысяч ячеек (рис. 3). Для более точного моделирования распространения волны сетка имеет сгущение к поверхности раздела фаз.

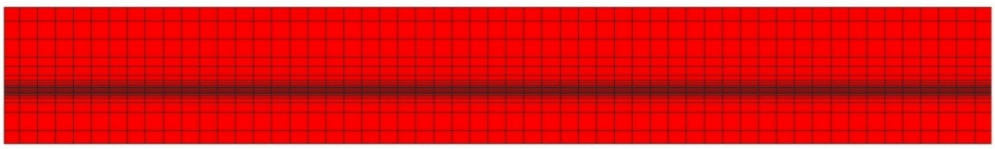


Рис. 3. Фрагмент расчетной сетки

В задаче участвует две фазы – вода и воздух. Плотность воды составляет 1000 кг/м^3, молекулярная вязкость равна 0.001 кг/(м·с).

Показания мареографа, размещенного на расстоянии одной ширины волны от неотражающей границы, представлены на рис. 4.

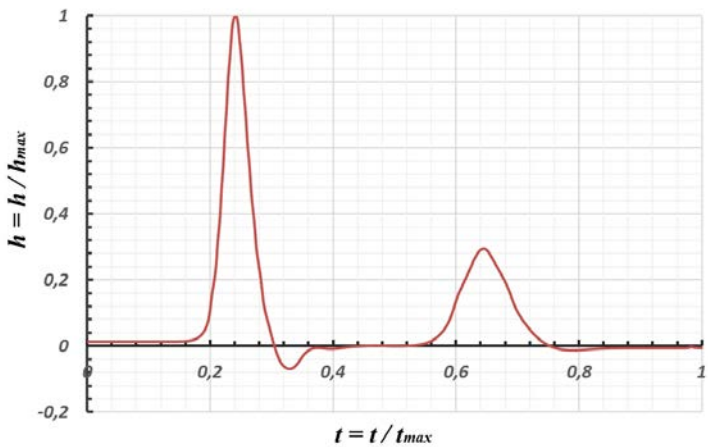


Рис. 4. Показания мареографа

На представленном графике приходящей волне соответствует участок $t < 0.5$. Отразившаяся волна наблюдается при $t > 0.5$. Видно, что амплитуда отраженной волны составляет 29 % от набегающей волны.

2.2. Определение оптимальных параметров для метода демпфирования волн

Для определения оптимальных параметров для метода демпфирования волн решалась задача из раздела 2.1.

Для определения оптимального значения параметра демпфирования k_{sour}, моделировалось распространение волны в каналах разной глубины. Рассматривались постановки, представленные в табл. 2.

Таблица 2

Постановки задачи

Постановка	Глубина канала, м	Длина канала, м	Высота волны ,м	Длина волны, м	L, м
1	0.32	23	0.064	4.24	4
2	3.2	230	0.64	40.24	40
3	32	2300	6.4	402.4	400
4	320	23000	64	4024	4000

На первом этапе моделировалось распространение волны в каналах разной глубины для выбора оптимального параметра демпфирования k_{sour}, при котором амплитуда отраженной волны наименьшая. Затем при выбранной постоянной глубине канала варьировалась ширина зоны демпфирования L для одной длины волны.

На первом этапе рассмотрим четыре варианта постановки задачи, представленные в табл. 2. Для выбора оптимального параметра демпфирования сравнивалась амплитуда отраженной волны с амплитудой приходящей волны в мареографе, который находится вблизи зоны демпфирования.

На рис. 5 представлены графики сравнения амплитуды отраженной волны при различных параметрах демпфирования для каналов различной глубины.

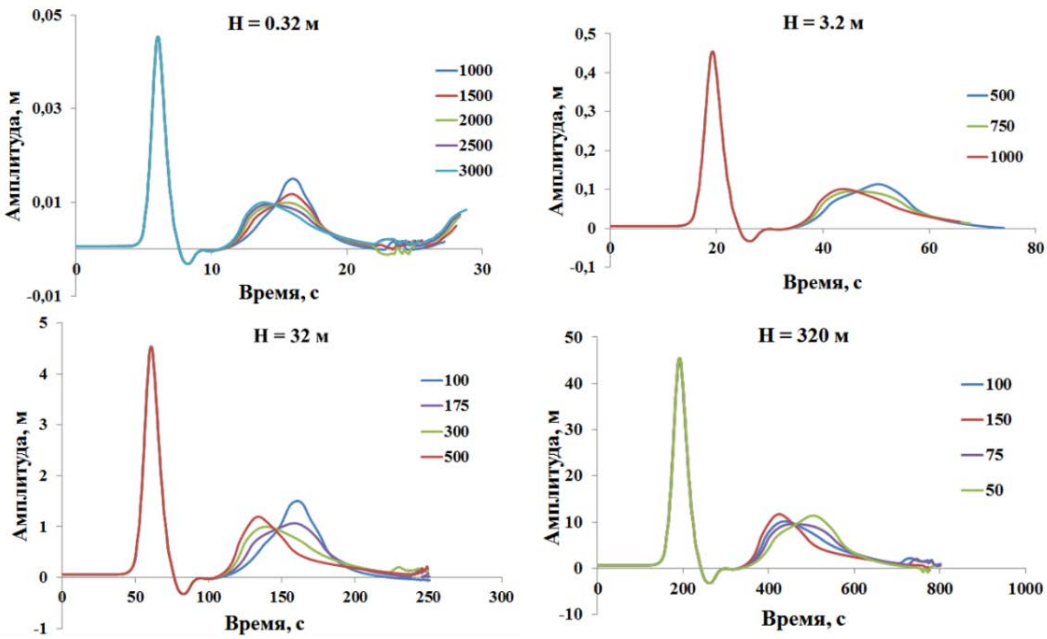


Рис. 5. Показания мареографа

По представленным графикам получены следующие результаты: оптимальный k_{sour} для глубины канала 0.32 м равен 2000 1/с, для глубины 3.2 м – 750 1/с, для глубины 32 м – 175 1/с, для глубины 320 м – 75 1/с.

Исходя из полученных данных, была подобрана зависимость параметра демпфирования от глубины канала (рис. 6):

$$k_{sour} = \frac{1.1 \cdot 10^3}{\sqrt{d}} \tag{7}$$

где d – глубина канала.

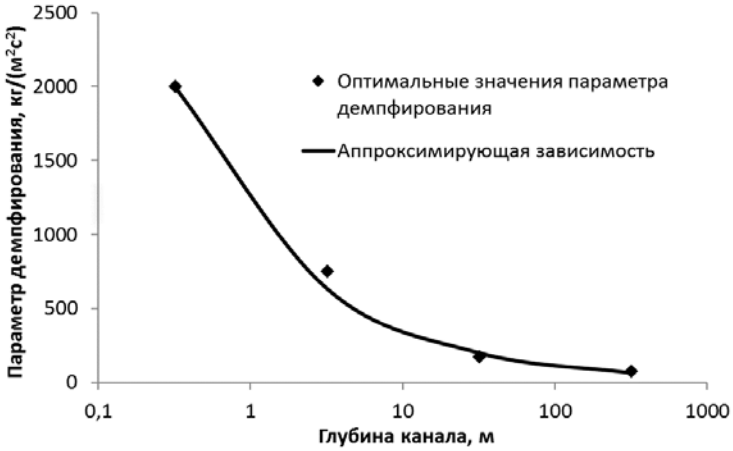


Рис. 6. Зависимость параметра демпфирования k_{sour} от глубины канала

Для определения оптимального значения ширины зоны демпфирования решалась задача с варьирование ширины зоны демпфирования L при неизменном параметре k_{sour}. Для этого был выбран канал глубиной 3.2 м, длиной 230 м, длина волны – 40.24 м, высота волны – 0.64 м, ширина зоны демпфирования L бралась равной одной, двум и трем длинам волн, параметр демпфирования $k_{sour} = 750\ 1/\text{с}$ – оптимальное значение для данной постановки, согласно проведенному исследованию и установленной зависимости (7).

На рис. 7 представлены показания мареографа, установленного в точке 100 м. Графики демонстрируют сравнение амплитуды приходящих в мареограф волн с амплитудой отраженных волн для трех вариантов постановок.

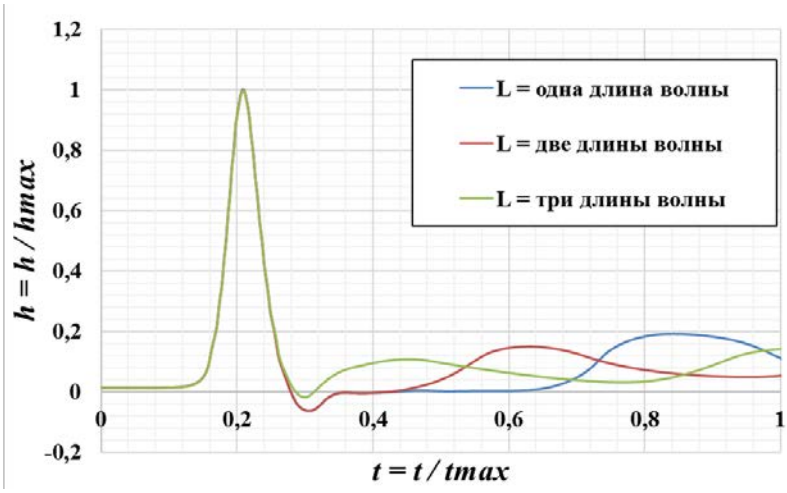


Рис. 7. Сравнение амплитуды приходящих волн с амплитудой отраженных волн

На рис. 7 показано, что при максимальной ширине слоя демпфирования, равному трем длинам волны, амплитуда отраженной волны наименьшая. Амплитуда отраженной волны составляет 18.9 % от амплитуды приходящей волны для первой постановки, 14.4 и 10 % – для второй и третьей соответственно.

По полученным результатам можно сделать вывод, что метод демпфирования волн является наиболее предпочтительным для решения реальных задач, поскольку данный метод возможно настроить для конкретной задачи за счет выбора оптимальных параметров демпфирования. В то время как метод открытой границы дал амплитуду отраженной волны в 29 % от амплитуды приходящей волны, метод зоны демпфирования дал результат в 10 % от амплитуды приходящей волны.

2.3. Использование неотражающих граничных условий для решения задачи распространения волн цунами

В качестве примера приводится численное моделирование схода подводного оползня в районе п-ова Камчатка. В результате схода оползня образуются волны, впоследствии натекающие на сам полуостров и на Командорские острова. На рис. 8 представлен вид сеточной модели для проведения численного расчета. Технология построения сеточных моделей для расчета волн цунами детально изложена в [15, 16].

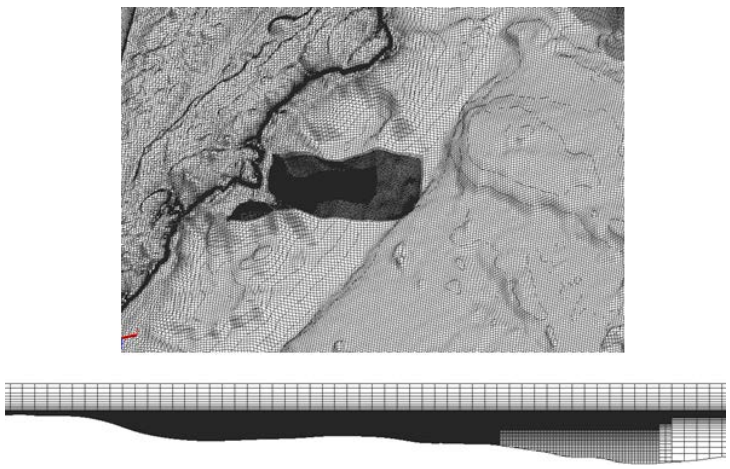


Рис. 8. Расчетная сетка в области источника цунами

В задаче применялась методика зоны демпфирования, где ширина зоны L задана равной двум длинам волн, а U_{sour} задан функцией вида (7). На рис. 9 представлена визуальная картина распространения волн в акватории. Визуализация картины течения зафиксирована по шкале минимальной и максимальной амплитуды волн.

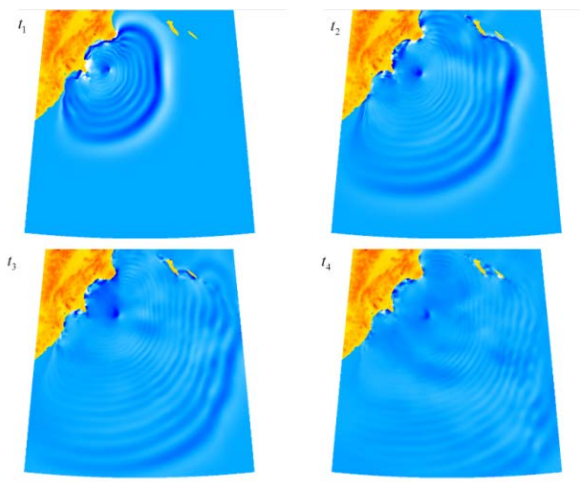


Рис. 9. Визуальная картина распространения волн в акватории п-ва Камчатка

На серии снимков, приведенных на рис. 9, можно наблюдать, как затухают волны, приближаясь к границам расчетной области, при этом отраженные волны, неизбежно образующиеся в процессе численного расчета, несущественно влияют на картину течения.

Заключение

В докладе рассмотрены методики реализации свободных границ для задач со свободной поверхностью. Описана реализация метода неотражающих граничных условий, основанного на методе характеристик – методе открытых границ. Показано, что метод не устраняет отраженные волны полностью и не может быть применен полноценно для решения задач на трехмерных неструктурированных сетках. Рассмотрена методика поглощения волн за счет формирования зоны демпфирования. На задаче распространения солитонной волны в канале были определены оптимальные параметры для метода зоны демпфирования. Было показано, что значение этих параметров определяется свойствами распространяемой волны и размерными масштабами решаемой задачи. Для демонстрации применения методики поглощения волн за счет зоны демпфирования решена задача распространения волн цунами в акватории полуострова Камчатка.

Литература

1. Kar S. K., Turco R. P. Formulation of a Lateral Sponge Layer for Limited-Area Shallow-Water Models and an Extension for the Vertically Stratified Case // Mon. Wea. Rev. 1994. Vol. 123. P. 1542–1559.

2. Fürst J., Musil J. Development of non-reflective boundary conditions for free-surface flows // Topical problems of fluid mechanics. Prague, 21-23 february, 2018. P. 97–104.

3. Подгорнова О. В. Построение дискретных прозрачных граничных условий для анизотропных и неоднородных сред: Диссертация на соискание ученой степени канд. физ.-мат. наук. Москва, 2008.

4. Givoli D., Neta B. High-order nonreflecting boundary conditions for the dispersive shallow water equations // Journal of Computational and Applied Mathematics. 2003. Vol. 158. P. 49–60.

5. Овчарова А. С. Метод расчета стационарных течений вязкой жидкости со свободной границей в переменных вихрь-функция тока // Прикладная механика и техническая физика. 1998. Т. 39, № 2. С. 59–66.

6. Pant C. S., Bhattacharya A. A viscous sponge layer formulation for robust large eddy simulation of thermal plumes // Computers and Fluids. 2016. Vol. 134. P. 177–189.

7. Wang Y., Yin Z., Liu Y., Yu N., Zou W. Numerical investigation on combined wave damping effect of pneumatic breakwater and submerged breakwater // International Journal of Naval Architecture and Ocean Engineering. 2019. Vol. 11. P. 314–328.

8. Choi J., Yoon S. B. Numerical simulations using momentum source wave-maker applied to RANS equation model // Coastal Engineering. 2009. Vol.56. P. 1043–1060.

9. Colombo A., Crivellini A. Assessment of a sponge layer non-reflecting boundary treatment for high-order CAA/CFD computations // Computers and Fluids. 2016. Vol. 140. P. 478–499.

10. Дородницын Л. В. Неотражающие граничные условия для систем уравнений газовой динамики // Журнал вычислительной математики и математической физики. 2002. Т. 42, № 11. С. 522–549.

11. Jasak H. Error Analysis and Estimation for the finite volume method with applications to fluid flows. Thesis submitted for the degree of doctor. – Department of Mechanical Engineering, Imperial College of Science, 1996.

12. Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. PhD thesis. – Imperial College, University of London, 1997.

13. Ефремов В. Р., Козелков А. С., Корнев А. В., Куркин А. А., Курулин В. В., Стрелец Д. Ю., Тарасова Н. В. Метод учета сил гравитации при моделировании течений со свободной поверхностью // Журнал вычислительной математики и математической физики. 2017. Т. 57, № 10.

14. Козелков А. С., Курулин В. В., Лашкин С. В., Шагалиев Р. М., Ялозо А. В. Исследование потенциала суперкомпьютеров для масштабируемого численного моделирования задач гидродинамики в индустриальных приложениях // Журнал вычислительной математики и математической физики. 2016. Т. 56, № 8. С. 1524–1535.

15. Козелков А. С. Методика численного моделирования цунами оползневого типа на основе уравнений Навье-Стокса // Вычислительная механика сплошных сред. 2016. Т. 9, №2. С. 218–236.

16. Kozelkov A. S., Kurkin A. A., Pelinovsky E. N., Tyatyushkina E. S., Kurulin V. V., Tarasova N. V., Landslide-type tsunami modelling based on the Navier-Stokes Equations // Science of tsunami Hazards, Journal of Tsunami Society International. 2016. Vol. 35, № 3. P. 106–144.
17. Уилкинсон У. Л. Неньютоновские жидкости. М.: Мир, 1964.

APPLICATION OF NON-REFLECTIVE BOUNDARY CONDITIONS IN NUMERICAL MODELING OF FREE-SURFACE FLOW PROBLEMS

D. A. Utkin, E. S. Tyatyushkina, A. S. Kozelkov, V. V. Kurulin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper describes implementations of non-reflective boundary conditions for free-surface problems simulated using the Navier-Stokes equations and the VOF method. Two implementation approaches for the non-reflective boundary conditions are described: calculation of integral expressions and creation of a damping layer. These approaches are compared based on the simulation of a soliton propagation problem.

Key words: Non-reflective boundary conditions, VOF method, Navier-Stokes equations, free-surface flow.

УДК 621.396.967
DOI 10.53403/9785951505071_2022_500

ИМИТАЦИОННАЯ МОДЕЛЬ РАДИОЛОКАЦИОННОЙ СТАНЦИИ ОБНАРУЖЕНИЯ И СЛЕЖЕНИЯ

И. И. Шанин, К. В. Задорожный

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

В докладе представлена имитационная компьютерная модель радиолокационной станции обнаружения и слежения (РЛС ОС), имитирующая основные процессы ее функционирования. В представленной модели воспроизводится цифровое представление физических сигналов, формируются их амплитудно-фазовые частотные характеристики, осуществляется обработка в полосовых фильтрах, выполняются другие процедуры по обработке и преобразованиям сигналов. Применяются алгоритмы быстрого преобразования Фурье. Приводится перечень изменяемых характеристик РЛС ОС. Представлено место модели РЛС ОС в моделирующей системе, ее взаимодействие с воздушными объектами и другими элементами фоновой обстановки.

Ключевые слова: компьютерная модель, радиолокационная станция, воздушный объект, поиск, обнаружение, сопровождение, помеха, сигнал, преобразование фурье, полосовой фильтр, угловая ячейка, стробирование, алгоритм сканирования, опорный сигнал.

Введение

В докладе представлена компьютерная модель радиолокационной станции обнаружения и слежения (РЛС ОС) импульсного излучения. Модель РЛС ОС предназначена для моделирования процессов поиска, обнаружения и сопровождения воздушных объектов, а также для оценки амплитуд отраженных сигналов и шумовых помех, оценки качества измерения отдельных характеристик, варьирования отражательными характеристиками воздушных объектов (ВО) и различными вариантами движения ВО [1–2]. Представляемая модель работы РЛС применима только для моделирования РЛС с неподвижными фазированными антенными решетками.

В модели РЛС ОС такие процессы как поиск, обнаружение, сопровождение, оценка амплитуд и шумовых помех в сигналах воспроизводится максимально подробно. Полностью воспроизводится физические сигналы во временно́й форме представления, формируются их амплитудно-фазовые частотные характеристики, представленные в комплексном виде. Применяются алгоритмы прямого и обратного быстрого преобразования Фурье. Выполняется полноценная обработка цифровых сигналов в полосовых фильтрах, осуществляются другие реальные процедуры обработки сигналов. Моделируется распространение и множественное переотражение амплитудно-фазовых частотных характеристик (АФЧХ) радиосигналов с учетом ЭПР, углов падения и отражения от множественных элементов групповых ВО. Моделирование и воспроизведение процессов, протекающих внутри РЛС, на таком уровне детализации занимает значительное вычислительное время. Это делает нерациональным применение представляемой и подобных ей моделей для оперативной оценки возможностей РЛС по обнаружению и сопровождению ВО. С учетом вышесказанного, представленная в докладе модель РЛС ОС используется для исследования возможностей РЛС по обнаружению и сопровождению групповых многоэлементных воздушно-космических объектов и для верификации более простых и более быстрых моделей РЛС. Верификация проводится на некотором ограниченном наборе

контрольных сценариев, предполагающих функционирование моделей РЛС с определенными характеристиками, работающих в заданном режиме работы, окруженных определенными воздушными объектами, двигающимися на фоне заданной фоновой обстановки.

Описание модели

Компьютерная модель РЛС ОС является частью комплекса компьютерных моделей в состав которого входят:

- компьютерная модель РЛС ОС;
- программный модуль прогноза траекторий движения ВО;
- компьютерная модель имитации движения ВО;
- модель Земли;
- модели потребителей РЛИ.

Комплекс компьютерных моделей моделирования работы РЛС ОС работает в составе системы компьютерного моделирования, которая включает графический интерфейс пользователя (ГИП) и моделирующую систему. Структурная схема компьютерной модели РЛС в составе комплекса компьютерных моделей и системы компьютерного моделирования представлена на рис. 1.

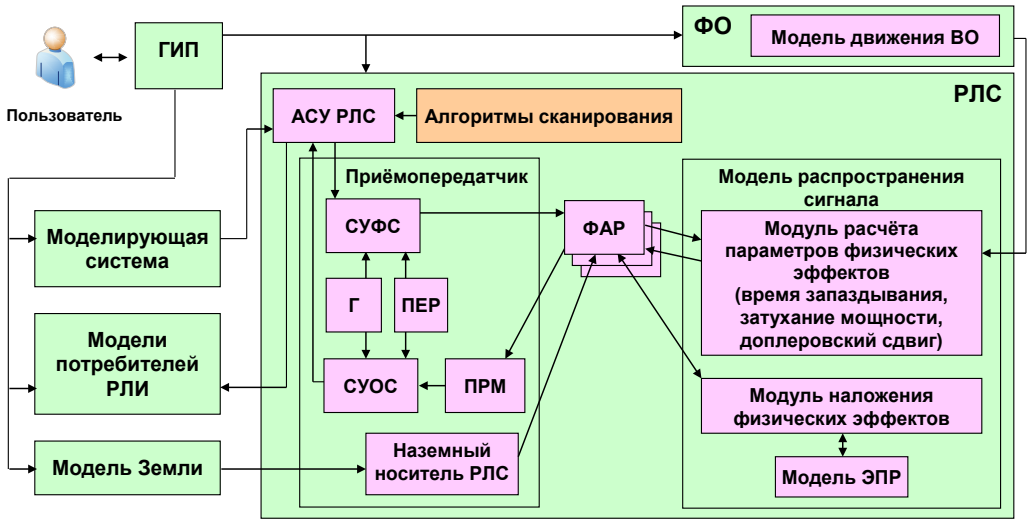


Рис. 1. Структурная схема модели РЛС

В модуле системы управления формированием сигнала (СУФС) на опорный сигнал гетеродина (Г) накладывается формируемый в передатчике (ПЕР) закон изменения амплитуды и формируется амплитудно-фазовая частотная характеристика (АФЧХ) излучаемого сигнала. В модуле фазированной антенной решетки (ФАР) для каждого облучаемого ВО, движение которого моделируются в блоке фоновой обстановки (ФО), с учетом текущего выставленного направления главного лепестка диаграммы направленности антенны (ДНА) формируется АФЧХ сигнала, излучаемого в направлении конкретного ВО – рис. 2.

АФЧХ отраженных сигналов и характеристики их распространения поступают в модуль принимающей ФАР. В модели распространения сигнала рассчитывается время запаздывания, затухание мощности, доплеровский сдвиг при отражении сигнала от каждого ВО. Принятые сигналы обрабатывается в модули наложения физических эффектов, в том числе учитывается эффективная площадь рассеивания (ЭПР) с учетом углов падения и отражения зондирующего сигнала. ЭПР каждого ВО задается в обобщенном табличном формате. Значения ЭПР для реализовавшихся во время моделирования углов падения и отражения интерполируются из таблично заданных отражательных характеристик ВО. Формируется АФЧХ общего принятого сигнала с учетом индивидуальных времен запаздывания каждой составляющей. Сформированная АФЧХ передается в Приемник – рис. 3.

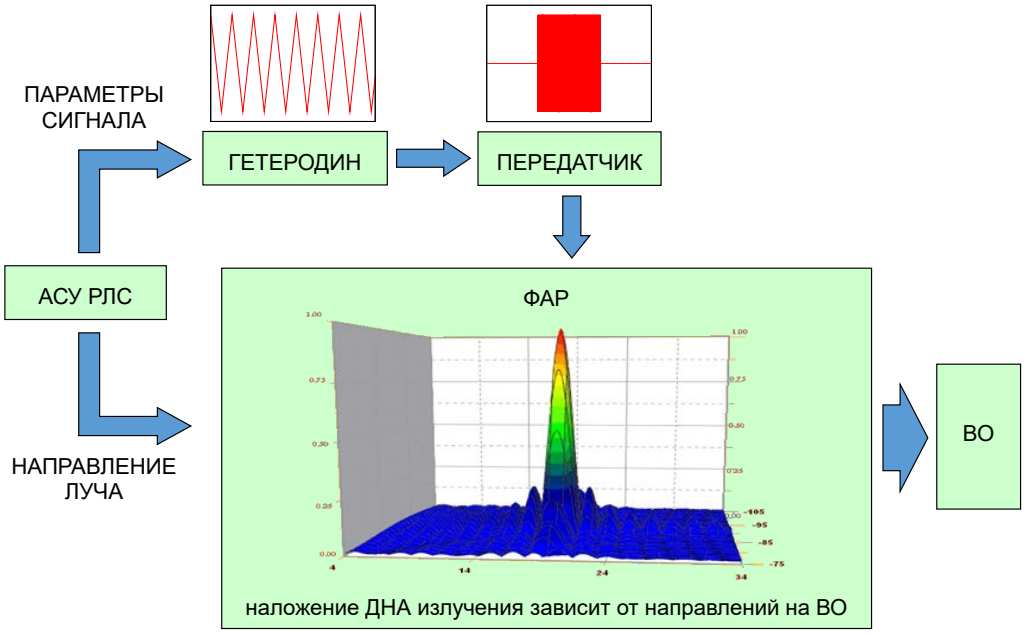


Рис. 2. Формирование АФЧХ индивидуального сигнала, излучаемого в направлении каждого ВО

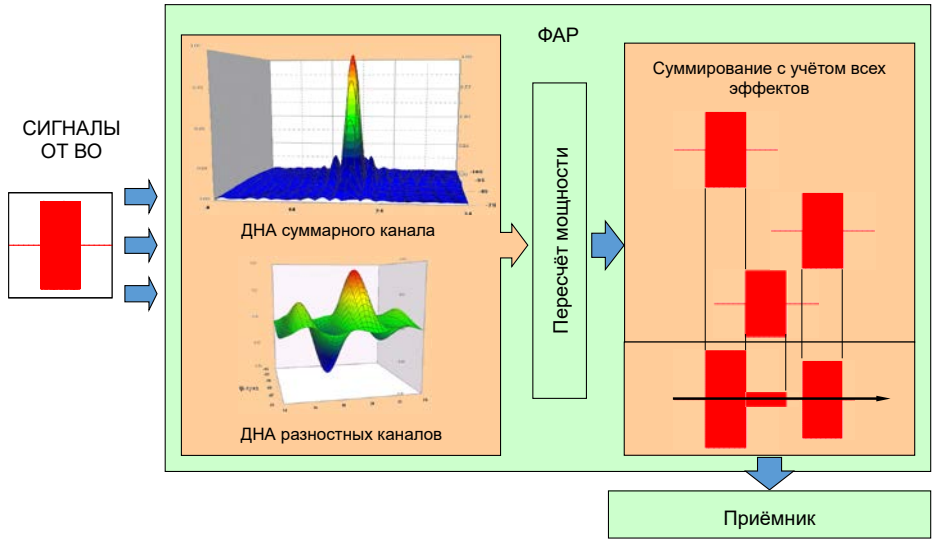


Рис. 3. Обработка АФЧХ отраженных сигналов в принимающей ФАР

Моделируется наложение внутренних тепловых шумов радиоэлектронных компонентов (РЭК) приемника, вычисляется АФЧХ сигнала на входе смесителя – рис. 4.

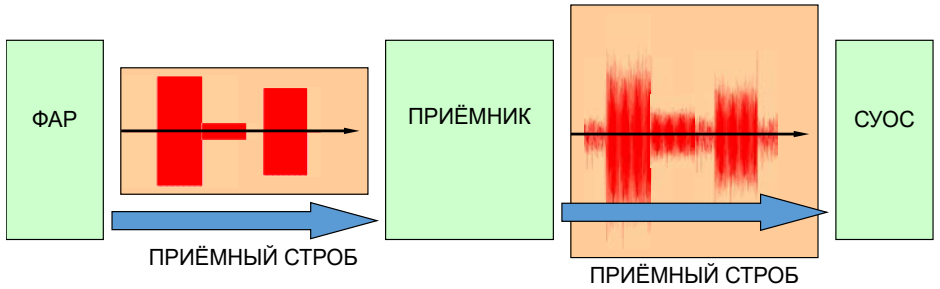


Рис. 4. Наложение внутренних тепловых шумов радиоэлектронных компонентов (РЭК) приемника

В модуле системы управления обработкой сигнала (СУОС) сигнал разбивается на стробы дальности, выделяются cos и sin составляющие сигнала на выходе смесителя, осуществляется проверка на наличие отраженного от ВО сигнала. Если сигнал присутствует, формируются специализированные разностные и суммарный сигналы для измерителей углов. По вычисленным сигналам и дискриминационным характеристикам в измерителях рассчитываются соответствующие принятому сигналу:

- время запаздывания отраженного сигнала;
- доплеровский сдвиг частоты;
- отклонение направления на ВО от главного лепестка диаграммы направленности антенны (ДНА);
- амплитуда сигнала;
- оценка среднеквадратичного значения амплитуды шума.

Далее в модели РЛС ОС имитируется работа измерителей угломерных и дальностного каналов, канала измерения радиальной скорости и ЭПР наблюдаемых воздушных объектов – рис. 5.

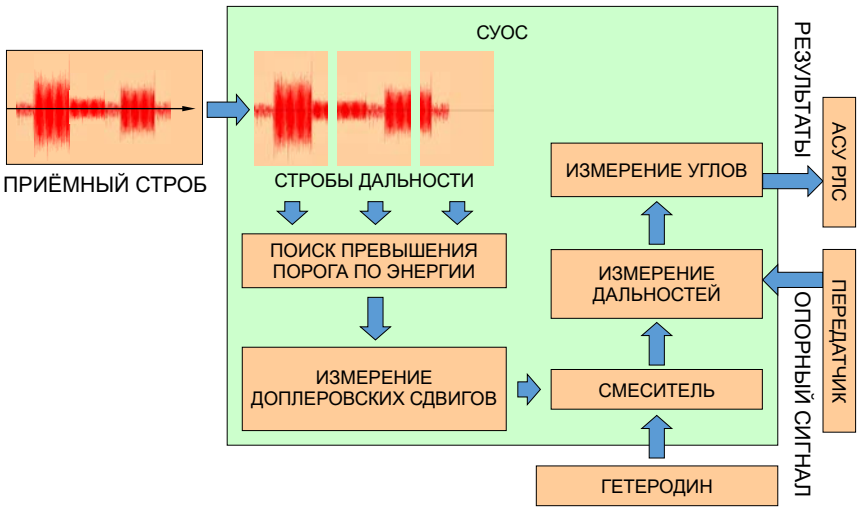


Рис. 5. Обработка приемного строба сигнала

В модели РЛС ОС имитируются процессы обнаружения, захвата и сопровождения ВО. Вначале работы выполняется поиск – в цикле последовательно зондируются все угловые ячейки сектора обзора, после чего выполняется цикл сопровождения.Если сопровождаемые ВО отсутствуют, то сканирование сектора обзора выполняется непрерывно. В момент обнаружения ВО последовательный просмотр сектора обзора начинает прерываться в соответствии с потребностями алгоритма обнаружения, захвата и сопровождения ВО. Его длительность зависит от количества сопровождаемых на текущий момент объектов – рис. 6.

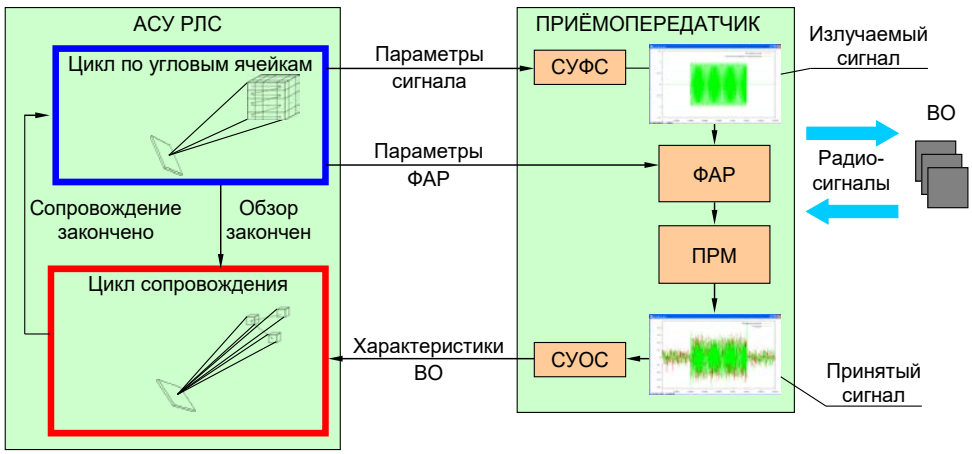


Рис. 6. Управление циклами обнаружения, захвата и сопровождения ВО

Для обзорных станций обнаружения в АСУ РЛС реализуется барьерный режим работы, который позволяет задавать границы сектора обзора по азимуту, углу места и дальности. Для квадратных секторов обзора (при a = b или a = b ± 1) применяется спиральный обход угловых ячеек – рис. 7.

Для прямоугольных (при a > b + 1 или a < b – 1) применяется построчный обход угловых ячеек – рис. 8.

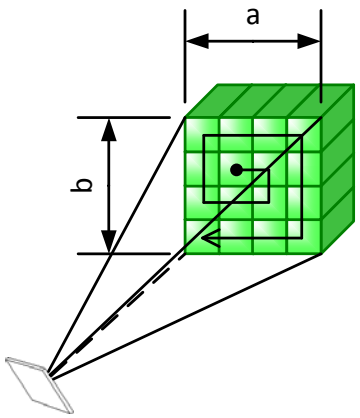


Рис. 7. Спиральный обход угловых ячеек

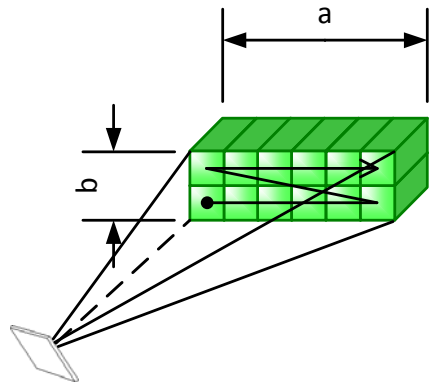


Рис. 8. Построчный обход угловых ячеек

Функции фильтров следящих систем и прогноз траекторий ВО осуществляется с помощью специально разработанного программного модуля прогноза траекторий движения ВО. Этот алгоритм сглаживает поступившие в него измерения полиномами. Коэффициенты полинома и соответствующая матрица ошибок вычисляются модифицированным методом наименьших квадратов. При этом выбираются оптимальные степень полинома и число сглаживаемых измерений, при которых оценка среднеквадратичного отклонения (с.к.о.) ошибок сглаживания (прогноза) минимальна. Вычисляются соответствующие ковариационные матрицы ошибок прогноза. Модель прогноза траекторий ВО позволяет в несколько раз уменьшить ошибки прогноза измеряемых величин по сравнению с традиционным методом наименьших квадратов при фиксированной модели движения (степени сглаживающего полинома) и полной выборке измерений.

Из измеренных характеристик наблюдаемых ВО формируются пакеты радиолокационной информации (РЛИ), которые передаются внешним моделям-потребителям РЛИ.

Принятые допущения и ограничения

Рассматриваемая модель работы РЛС применима только к РЛС с неподвижными фазированными антенными решетками. В настоящее время это самый современный, но не единственный класс РЛС. Поэтому это, несомненно является ограничением.

Верификация

Ниже приводится описание вычислительного эксперимента, в котором моделируется взаимодействие РЛС ОС с ВО. Характеристики, используемые при моделировании работы РЛС ОС, можно изменять, что обеспечивает возможность проведения исследований, касающихся функционирования как существующих, так и перспективных РЛС ОС в различных вариантах использования. Полный перечень изменяемых характеристик РЛС ОС и тестовых значений, используемых при получении представленных ниже результатов, представлен в табл. 1.

Таблица 1

Перечень изменяемых характеристик РЛС ОС

№	Название параметра	Размерность	Значение
1.	Несущая частота	Гц	$3{,}3 \cdot 10^{9}$
2.	Такт зондирования	с	$2 \cdot 10^{-3}$
3.	Импульсная мощность	Вт	$200 \cdot 10^{3}$
4.	Длительность импульса	с	$55 \cdot 10^{-6}$
5.	Пороговое значение сигнал/шум	б/р	100
6.	Мощность собственных шумов приемника	Вт	$7.59 \cdot 10^{-15}$
7.	Ширина ФАР	м	3
8.	Высота ФАР	м	3
9.	Сектор обзора: азимут мин./макс. угол места мин./макс. дальность мин. дальность макс.	 градус градус м м	 –60 / 60 1 / 80 $10 \cdot 10^{3}$ $200 \cdot 10^{3}$
10.	Количество неудачных попыток перед сбросом сопровождаемого объекта	б/р	3

Несущая частота, представленная в табл. 1, соответствует сантиметровому X-диапазону.

Коэффициенты усиления антенны (КУ) на передачу и на прием в данном случае равны и определяются выражением (1).

$$G_r = G_t = \frac{4\pi \cdot S_{а.эф.}}{\lambda^2} = \frac{4\pi \cdot S_{а.эф.}}{(c/f_0)^2}. \quad (1)$$

где $S_{а.эф.}$– эффективная площадь антенны в заданном направлении; λ – длина волны излучаемого сигнала.

В соответствии с табл. 1 в направлении нормали КУ антенны на передачу и на прием равны:

$$G_r = G_t = \frac{4\pi \cdot 9}{8{,}26 \cdot 10^{-3}} \approx 13.7 \cdot 10^{3}. \quad (2)$$

В соответствии с используемыми в вычислительных экспериментах характеристиками РЛС ОС ширина луча по половинной мощности в направлении нормали антенны составляет 1,54 градуса. Площадь луча в этом случае равна 2,37 квадратных градуса.

Длительность такта зондирования должна удовлетворять условию (3).

$$t_т \geq t_с + t_п \quad (3)$$

где $t_с$ – длительность сигнала; $t_п$ – длительность приемного строба.

Если длительность сигнала равна удвоенной длительности зондирующего импульса, то:

$$t_т \geq 110 \cdot 10^{-6} + \frac{2 \cdot 200 \cdot 10^{3}}{c} \approx 110 \cdot 10^{-6} + 1333 \cdot 10^{-6} = 1443 \cdot 10^{-6} \approx 1.4 \cdot 10^{-3}. \quad (4)$$

В соответствии с (4) такт зондирования длительностью $2 \cdot 10^{-3}$ (табл. 1) удовлетворяет условию (3). В этом случае скорость осмотра зоны поиска в близи нормали антенны составляет 1168 квадратных градуса в секунду или одну квадратную зоны поиска размером 35×35 градусов за одну секунду. Теоретически время просмотра сектора обзора РЛС ОС составляет:

$$t_{обз.т} = \frac{120 \cdot 79}{1182} = 8{,}12 \text{ с}. \quad (5)$$

С учетом того, что в имитационном вычислительном эксперименте с помощью РЛС ведется наблюдение за ВО с максимальными скоростями до 100 км/ч (28 м/с). Теоретически, в самом «плохом» случае ВО сможет войти в сектор обзора и приблизиться к РЛС на:

$$l_{max} = 28 \cdot 8 = 224 \text{ м}. \quad (6)$$

Таким образом, РЛС с приведенными в табл. 1 характеристиками обладает адекватными возможностями по поиску ВО и может быть успешно использована для обзора пространства.

Значения скоростных и сигнальных характеристик ВО заимствованы из [3 – 4] и представлены в табл. 2.

Таблица 2

Скоростные и сигнальные характеристики ВО

№	Воздушный объект	ЭПР, м^2	Скорость полета	
			км/ч	м/с
1.	Сизый голубь (Columba livia)	$10 \cdot 10^{-3}$	80	22,2
2.	Серый гусь (Anser anser)	$22 \cdot 10^{-3}$	80	22,2
3.	Белый аист (Ciconia ciconia)	$30 \cdot 10^{-3}$	60	16,6
4.	Обыкновенный скворец (Sturnus vulgaris)	$2{,}3 \cdot 10^{-3}$	50	13,9
5.	Кряква (Anas platyrhynchos)	$21 \cdot 10^{-3}$	90	25,0

Ниже приведены графики некоторых сигналов, формирующихся во время моделирования, и демонстрирующих результаты обработки сигналов в некоторых программных модулях модели РЛС ОС. На рис. 9 представлен увеличенный фрагмент сигнала идеальной прямоугольной формы, сформированного системой управления формированием сигнала.

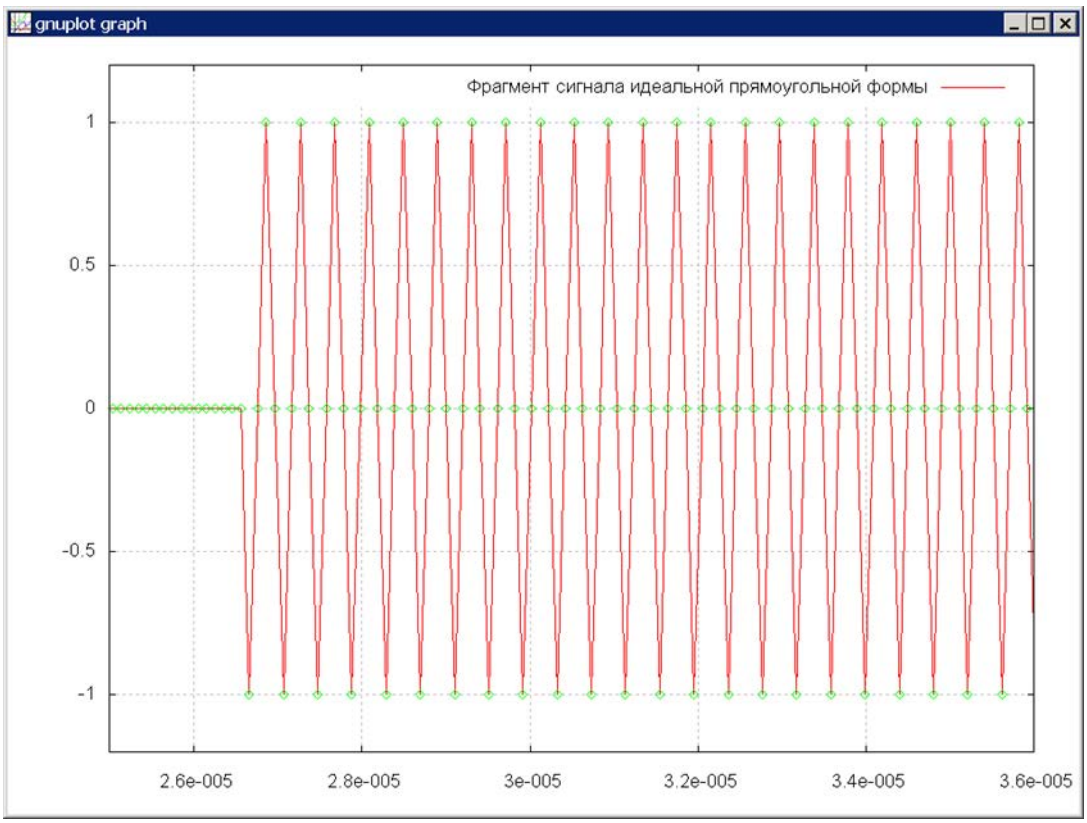


Рис. 9. Увеличенный фрагмент сигнала идеальной прямоугольной формы

На рис. 10 представлен строб дальности, в который попал отраженный от ВО сигнал. Отношение сигнал/шум в данном случае приблизительно равен пяти.

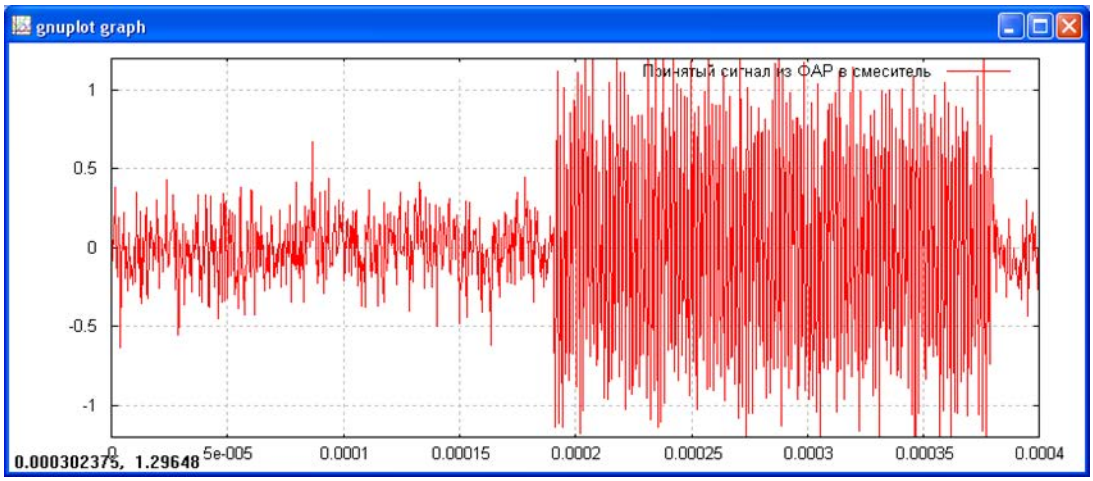


Рис. 10. Строб дальности с сигналом, отраженным от ВО

В алгоритме измерения дальности используется опорный импульс, который формируется в результате обработки немодулированного сигнала, изменяющегося по закону изменения амплитуды, реализованному в Передатчике, в ФНЧ измерителя дальности. В результате обработки в ФНЧ идеальный импульс длительностью t_{uu} «расползается», а его длительность увеличивается и становится равной t_u – рис. 11. Длительность зондирующего сигнала обозначена t_c.

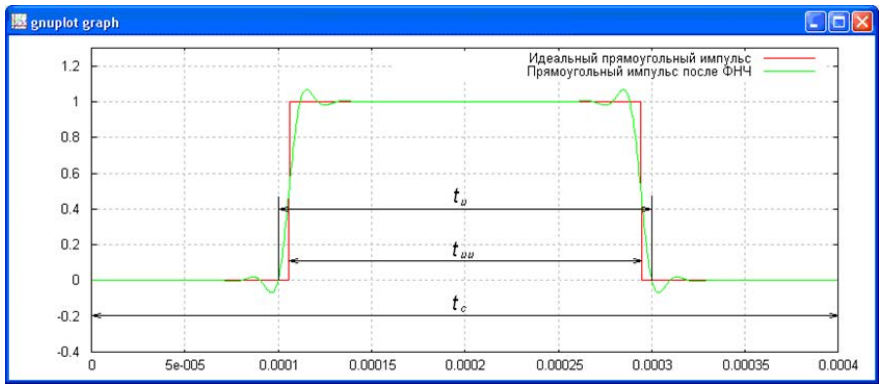


Рис. 11. Идеальный прямоугольный импульс до и после обработки в ФНЧ измерителя дальности

На рис. 12 представлен опорный строб и принятый сигнал с отношением сигнал/шум равным пяти в суммарном канале после квадратурной обработки.



Рис. 12. Опорный строб и принятый сигнал в суммарном канале после квадратурной обработки

На рис. 13 представлены косинусоидальная и синусоидальная составляющие сигнала с ненулевым доплеровским сдвигом относительно несущей частоты опорного гетеродина РЛС, вызванным отрицательной радиальной скоростью ВО относительно РЛС.

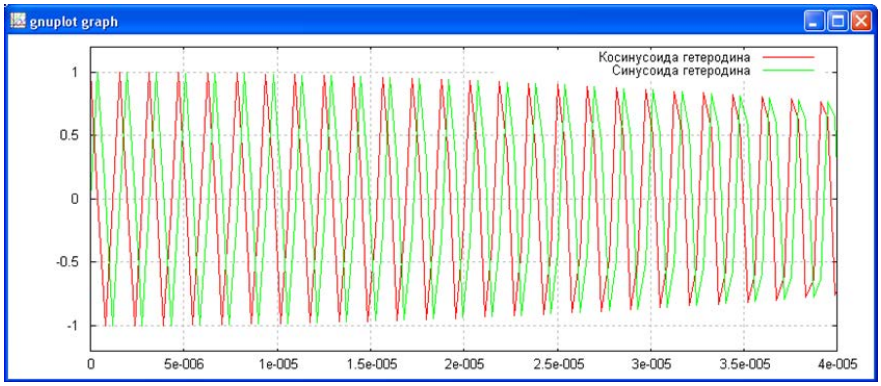


Рис. 13. Косинусоидальная и синусоидальная составляющие сигнала с доплеровским сдвигом, вызванным отрицательной радиальной скоростью ВО

Некоторые обобщенные показатели полученных результатов моделирования, теоретических расчетов и верификационные значения некоторых характеристик представлены в табл. 3.

Таблица 3

Обобщенные моделирования, теоретических расчетов и верификационные значения

№	Обобщенный показатель	Мод.	Теор.
1.	Размер зоны поиска в угловых ячейках по азимуту, б/р по углу места, б/р	 69 47	 78 52
2.	Время просмотра зоны поиска, с	6,486	8,12
3.	Дальность обнаружения ВО с ЭПР 0.030 м^2, м	$42 \cdot 10^3$	$50 \cdot 10^3$
4.	Дальность обнаружения ВО с ЭПР 0.018 м^2, м	$38 \cdot 10^3$	$44 \cdot 10^3$
5.	Дальность обнаружения ВО с ЭПР 0.010 м^2, м	$33 \cdot 10^3$	$38 \cdot 10^3$
6.	Дальность обнаружения ВО с ЭПР 0.001 м^2, м	$18 \cdot 10^3$	$21 \cdot 10^3$
7.	Максимальная ошибка измерения дальности[1], м	290	–
8.	Максимальная ошибка измерения азимута[1], градус	0,13	–
9.	Максимальная ошибка измерения угла места[1], градус	0,14	–
10.	Максимальная ошибка измерения радиальной скорости[1], м/с	149	–
11.	Максимальная ошибка измерения ЭПР[1], м^2	$1 \cdot 10^{-3}$	–

[1] Максимальные ошибки измерения приведены для ВО с ЭПР 0.030 м^2.

Анализ результатов

Фактический размер зоны поиска, выраженный в угловых ячейках, отличается от размера, полученного в результате теоретических расчетов. В действительности, с учетом увеличения ширины луча при его отклонении от нормали антенны сектор обзора, имеющий ширину 120 и высоту 79 градусов, разбивается 69 и 47 угловых ячеек соответственно. Таким образом, действительное время просмотра сектора составляет:

$$t_{\text{обз.д}} = \frac{69 \cdot 47}{500} = 6{,}486\text{с}. \quad (7)$$

Теоретические расчеты проводились в части дальностей обнаружения ВО с различными ЭПР. Для расчетов использовалось основное уравнение радиолокации, записанное в виде получения дальности действия РЛС с пассивным ответом (6).

$$D_{max} = \sqrt[4]{\frac{P_t \cdot G_t \cdot G_r \cdot \left({}^{c}/_{f_0}\right)^2 \cdot \sigma}{P_{r.noise} \cdot {}^{S}/_{N} \cdot (4\pi)^3}} =$$

$$= \sqrt[4]{\frac{200 \cdot 10^3 \cdot (13.7 \cdot 10^3)^2 \cdot \left(\frac{3 \cdot 10^8}{3.3 \cdot 10^9}\right)^2 \cdot \sigma}{7.59 \cdot 10^{-15} \cdot 100 \cdot (4\pi)^3}} =$$

$$= \sqrt[4]{2.05 \cdot 10^{20} \cdot \sigma} = 120 \cdot 10^3 \cdot \sqrt[4]{\sigma}, \quad (8)$$

где P_t – мощность передатчика; G_t – коэффициент усиления антенны на передачу; G_r – коэффициент усиления антенны на прием; ${}^{c}/_{f_0}$ – длина волны; σ – ЭПР ВО; $P_{r.noise}$ – мощность собственных шумов приемника; ${}^{S}/_{N}$ – пороговое отношение сигнал/шум.

Дальность обнаружения ВО с ЭПР 0.03 м^2 в приведенном вычислительном эксперименте составляет 42 км. В соответствии с теоретическим расчетом, дальность обнаружения должна равняться 50 км. ВО с ЭПР 0.018 м^2 обнаруживается на дальности 38 км, при теоретических 44 км. Такая ситуация наблюдается на всех, приведенных ВО – фактическая дальность обнаружения на 15 % меньше теоретической.

В результате пошагового исследования функционирования модели РЛС ОС на всех этапах, от формирования сигнала до получения измеренных характеристик ВО, проводимого с целью выявления причин несовпадения дальностей обнаружения ВО в приведенном вычислительном эксперименте и теоретическом расчете было выявлено, что это происходит по двум причинам.

Первая – в модели РЛС ОС учитывается эффект несовпадения оси главного лепестка ДНА с истинным направлением на ВО. В модели РЛС ОС формируется ДНА в соответствии с направлением, определяемым зондируемой угловой ячейкой, моделируется действительная реализация истинного направления на ВО и используется значение ДНА, которое соответствует истинному направлению на ВО в рамках выставленной ДНА.

Вторая причина – неполное попадание отраженного сигнала в строб дальности. В силу того, что устройство порогового уровня (УПУ) настроено на попадание в каждый рассматриваемый строб дальности более половины сигнала (в противном случае более половины сигнала всегда будет находится в следующем стробе дальности), часть его мощности, анализируемая при принятии решения, в общем случае всегда находится в диапазоне от 0.5 до 1.0 от его действительной принятой мощности. При этом, следует учитывать, что мощность принятого сигнала, также включает в себя мощность шума, оказавшегося на том же участке, что и принятый сигнал.

В основном уравнении радиолокации, используемом для теоретических расчетов, эти эффекты не учитываются, но могут быть учтены, если заранее известны значения соответствующих коэффициентов.

В первом случае значение коэффициента определяется случайной степенью попадания ВО в центры зондируемых угловых ячеек при обнаружении и точностью прогноза движения ВО при сопровождении. И то, и другое определяется множеством факторов, следовательно, может отличаться в различных ситуациях. В представленном эксперименте ВО двигался по одной и той же пролетающей траектории, при этом менялось значение его ЭПР. ВО с разными ЭПР обнаруживались на разных дальностях, т. е. в момент обнаружения они находились в разных угловых ячейках и/или в разных частях угловых ячеек сектора обзора.

Во втором случае значение коэффициента определяется степенью попадания отраженного от ВО сигнала в нарезанные по времени стробы дальности. Положение отраженного от ВО сигнала относительно границ стробов дальности определяется фактической удаленностью ВО от РЛС, ближней границей зондируемой угловой ячейки по дальности и длительностью стробов дальности. То есть, также, как и в первом случае, коэффициент носит случайный характер.

Максимальные ошибки измерения в табл. 3 приведены для ВО с ЭПР 0.030 м^2. Процентное отношение максимальной ошибки измерения дальности к дальности, эквивалентной длительности импульса, составило 1.75 %. Процентное отношение максимальной ошибки измерения угла места и азимута к ширине луча составило 8.44 %. Процентное отношение максимальной ошибки измерения радиальной скорости составило 414 % относительно истинной скорости движения ВО. Процентное отношение максимальной ошибки измерения ЭПР составило 3.3 % относительно истинного значения ЭПР ВО.

Достаточно высокие точности измерения дальности и углов объясняются, соответствующим высоким пороговым отношением сигнал/шум в УПУ, заданным в характеристиках модели. Полученная точность измерения радиальной скорости делает бессмысленным ее использование – очевидно, реализованная в модели РЛС ОС методика ее измерения требует тщательного пересмотра. Точность измерения ЭПР ВО вполне позволяет делать предположения о принадлежности ВО к определенным видам на основании сопоставления измеренных значений ЭПР с априорно известными эталонными значениями ЭПР видов ВО.

Теперь о верификации с публикуемыми в открытых источниках данными, полученными экспериментальным путем. Модель РЛС ОС обнаруживает ВО с ЭПР 0.018 м^2 на дальности 38 км. Как утверждается в открытых источниках, лучшие современные системы контрбатарейной борьбы способны обнаруживать снаряды гаубиц с ЭПР 0,001 м^2 на дистанциях около 30 км, а ракеты и артиллерийские мины с ЭПР от $10 \cdot 10^{-3}$ до $18 \cdot 10^{-3}$ м^2 – более, чем на 50 км. С учетом использовавшихся при моделировании характеристик, в том числе значений внутренних шумов приемника, а также отсутствия в модели РЛС ОС дополнительных систем и алгоритмов, применяемых в РЛС лучших современных системах контрбатарейной борьбы, можно считать, что полученные результаты моделирования, соответствуют данным, приведенным в открытых источниках.

К примеру, если уменьшить внутренние тепловые шумы приемника РЛС ОС до значения 5,7E – 15, а пороговое отношение сигнал/шум обнаружения ВО со 100 до 20, то ВО с ЭПР 0,001 м^2 обнаружится на дистанции 30 км, ВО с ЭПР 0,01 м^2 обнаружится на дистанции 51 км, а ВО с ЭПР 0,018 м^2 – на дистанции 53 км. При том, что в модели РЛС ОС отсутствуют какие-либо алгоритмы обработки сигналов, нацеленные повышение дальности обнаружения ВО.

Выводы

При сравнении дальностей обнаружения ВО, полученных в результате проведения имитационного вычислительного эксперимента, и дальностей обнаружения ВО, полученных теоретически с помощью основного уравнения радиолокации, не учитывающего рассогласование между направлением оси главного лепестка ДНА и истинным направлением на ВО, а также не учитывающего неполное попадание отраженного от ВО сигнала в анализируемые стробы дальности, оказалось, что в вычислительном имитационном эксперименте дальности обнаружения ВО на 15 % меньше, чем в теоретических расчетах [1–2]. С другой стороны, при использовании основного уравнения радиолокации, учитывающего неидеальность попадания ВО на ось главного лепестка ДНА, а также неполное попадание отраженных от ВО сигналов в анализируемые стробы дальности, результаты имитационного вычислительного эксперимента совпадают с теоретическими расчетами.

В целом это подтверждает правильность полученных результатов.

Модель РЛС ОС может быть использована для нахождения вероятностных характеристик, таких как математическое ожидание и среднеквадратичное отклонение коэффициентов, позволяющих учитывать рассогласование между направлением оси главного лепестка ДНА и истинным направлением на ВО, а также эффект неполного попадания отраженного от ВО сигнала в анализируемые стробы дальности для РЛС с конкретными характеристиками и определенных воздушных объектов на достаточном количестве реализаций. Полученные характеристики коэффициентов могут быть использованы для моделирования данных эффектов с помощью генератора псевдослучайных чисел, с использованием очень малого количества вычислительных операций и без снижения качества получаемых результатов.

Литература

1. Ширман Я. Д. Теоретические основы радиолокации. – Харьков: ВИРТА, 1984.
2. Бердышев В. П. Радиолокационные системы. – Красноярск: СФУ, 2012.
3. Эффективная площадь рассеяния [Электронный ресурс]: Википедия.
4. Полет птиц [Электронный ресурс]: Википедия.

A SIMULATION MODEL OF THE DETECTION AND TRACKING RADAR

I. I. Shanin, K. V. Zadorozhny

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

The paper presents a simulation computer model of the detection and tracking radar (DTR) that imitates its main operation processes. The model reproduces digital representation of physical signals, forms their amplitude and phase frequency parameters, performs processing in the bandpass filters as well as other signal transformation and processing procedures. Algorithms of fast Fourier transformation (FFT) are used. A list of changeable parameters of DTR is provided. The paper describes the place of the DTR model in the simulating system, its interaction with flying objects and other elements of background surroundings.

Key words: computer model, radar station, flying object, hunting, detection, tracking, spurious signal, Fourier transformation, bandpass filter, angular cell, strobing, scanning algorithm, reference signal

УДК: 004.94:519.854.3
DOI 10.53403/9785951505071_2022_512

АДАПТИВНАЯ ГЕНЕРАЦИЯ РЕГУЛЯРНЫХ РАСЧЕТНЫХ СЕТОК НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

И. А. Шумилов, А. Е. Тимофеев, А. Л. Потехин

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Для формирования расчетных сеток на поверхностях геометрических моделей существуют различные стратегии. Целью настоящего исследования является разработка алгоритма блочной генерации регулярных сеток с представлением граничных условий задания сетки в виде системы целочисленных линейных уравнений. Критерием задачи целочисленного программирования является минимизация числа изменений начального распределения узлов сетки на границе. Для решения составленной задачи используется алгоритм решения задач целочисленного линейного программирования, который на практике быстро находит допустимое решение, близкое к оптимальному и может быть использован для вычисления глобального оптимума. Разработанный алгоритм позволяет при необходимости изменять начальное число сеточных узлов на границе модели для применения блочной генерации регулярной сетки. Разработанный алгоритм позволил в рамках препостпроцессора пакета программ ЛОГОС получить возможность построения поверхностных сеток, обеспечивающих согласование сеточных фрагментов с разным уровнем детализации.

Ключевые слова: целочисленное линейное программирование; поверхностная сетка; расчетная сетка; блочная генерация.

Введение

Полностью автоматизированных генераторов поверхностных сеток, позволяющих построить регулярную поверхностную сетку на сложных поверхностях, не существует (рис. 1). Перспективным выглядит компромиссный подход: необходимо произвести дополнительное нарезание поверхностей на более простые, в которых возможно построение регулярных сеток. На простых поверхностях построение сетки достижимо путем их разделения на виртуальные регулярные блоки по заданной схеме с согласованной на границах блоков сеткой (рис. 2). Под регулярным блоком здесь мы будем понимать четырехугольник, у которого противоположные ребра имеют одинаковое число сеточных узлов. В развитых коммерческих продуктах под простыми поверхностями подразумеваются 3–6 угольные грани [1], а также фрагменты окружностей.

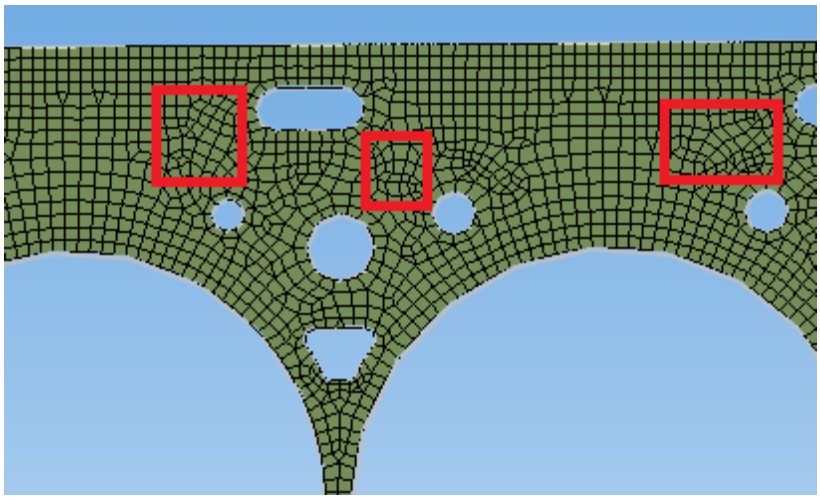


Рис. 1. Фрагмент сложной поверхности с сеткой, построенной фронтальным алгоритмом. Красным отмечены места схождения фронта генерации.

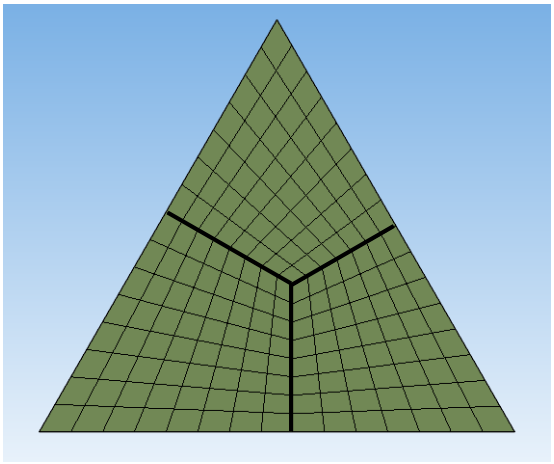


Рис. 2. Пример выделения трех виртуальных блоков на треугольной грани для построения регулярной сетки. Границы виртуальных блоков выделены черной линией.

Согласование виртуальных блоков

Очевидно, что разделение простых поверхностей на согласованные регулярные блоки возможно не всегда, а лишь при удовлетворении некоторых условий по числу узлов на их границах. В ходе построения поверхностных сеток на многогранном теле крайне желательно сохранение общего размера сеточных элементов. При этом число ячеек на границах граней может варьироваться. Для поддержания такой возможности ставится задача разбиения грани с минимальным изменением начального числа сеточных узлов контура грани.

Рассмотрим задачу выделения и согласования виртуальных блоков на грани на примере построения переходного сеточного слоя на четырехугольной грани. Под переходным слоем будем подразумевать сетку специального вида, обеспечивающую согласование сеток на соседних гранях.

Дана четырехугольная геометрическая грань вытянутой формы, на которой необходимо построить сетку, расширяющуюся с n узлов до m узлов вдоль широкой границы, где $n < m$. При этом на противоположных концах узкой границы число сеточных узлов одинаковое. Контур грани зада-

ется в виде набора ребер, представляющих собой геометрические кривые. Каждая кривая определяется однопараметрическим уравнением. Для каждого ребра заданы начальное число и координаты сеточных узлов. Также обозначим каждую вершину грани. Пример грани с исходными данными на ней изображен на рис. 3.

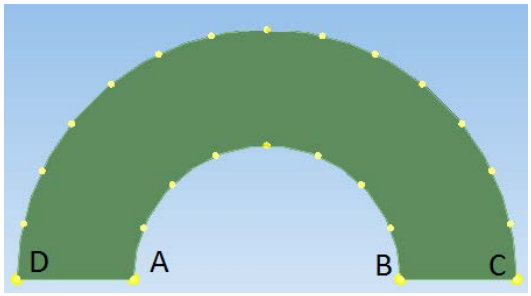


Рис. 3. Грань с заданным начальным положением
сеточных узлов на границе

На грани требуется построить поверхностную сетку, используя блоки трех видов, представленных на рис. 4. Внутренние области каждого блока представляют собой одну сеточную ячейку.

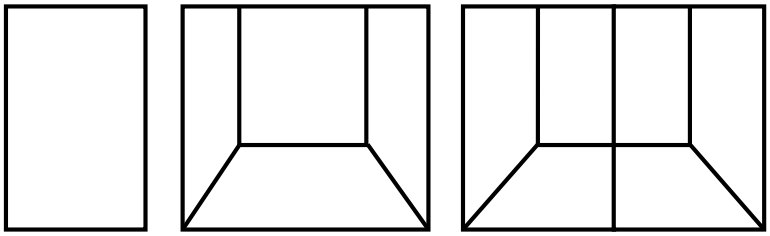


Рис. 4. Типы блоков, используемые для формирования переходного сеточного слоя

Пусть x_1, x_2, x_3 – количество блоков по каждому из трех типов, использованных для построения переходного сеточного слоя.

Для поиска набора блоков, наилучшим образом удовлетворяющего начальным ограничениям строится система уравнений:

$$\begin{cases} AB = \delta_1 + x_1 + x_2 + 2 * x_3 \\ CD = \delta_2 + x_1 + 3 * x_2 + 4 * x_3 \\ x_i \geq 0, i = \overline{1,3} \\ x_i \in \mathbb{Z}, i = \overline{1,3} \\ \delta_1 \in \mathbb{Z}, \delta_2 \in \mathbb{Z} \end{cases}, \quad (1)$$

где AB и CD – число сеточных ячеек на ребрах AB и CD соответственно (для простоты обозначения вершин выбраны таким образом, что $AB \leq CD$); δ_1 – число сеточных ячеек, добавленных к ребру AB; δ_2 – число сеточных ячеек, добавленных к ребру CD.

Для контроля за изменением числа сеточных узлов на ребрах AB и CD можно добавить дополнительные ограничения $\delta_1 = 0$ или $\delta_2 = 0$.

Критерием оптимизации является минимизация числа изменений на ребрах грани.

$$F(x) = |\delta_1| + |\delta_2| \to min \quad (2)$$

Для представления задачи (1) – (2) виде задачи целочисленного линейного программирования (ЦЛП) необходимо привести критерий к линейному виду. Для этого можно рассмотреть четыре задачи, в каждой из которых используется различная комбинация знаков функций модуля. Далее производится сравнение значений критерия по каждой задаче и выбирается наилучшее.

При решении задачи целочисленного линейного программирования получаем число каждого типа блоков, а также новое число ячеек на ребрах AB и CD, добавляя к существующим значениям δ_1 и δ_2 соответственно. После этого блоки равномерно, с учетом их ширины, распределяются по поверхности грани. Пример построенного переходного слоя изображен на рис. 5.

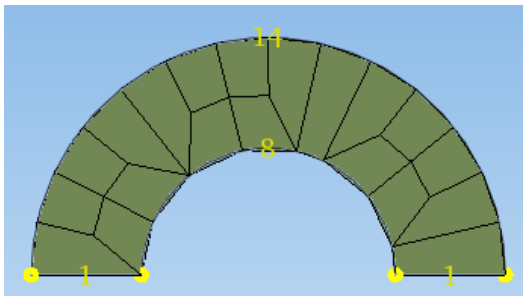


Рис. 5. Построенный переходный сеточный слой
на грани при помощи виртуальных блоков

Постановка задачи ЦЛП

Рассмотрим задачу целочисленного линейного программирования в общей постановке:

$$\begin{cases} Ax \le b \\ Dx = f \\ x \in \mathbb{Z}^n \end{cases}, \quad c^T x \to max \tag{3}$$

где c, b, f – векторы, а A и D – матрицы, все элементы которых являются целыми числами.

Ограничения на переменные, если присутствуют, включены в $Ax \le b$.

Обозначения a_i — это строки матрицы A.

Алгоритм полагается на два основных способа трансформации задачи ЦЛП:

1. Линейное преобразование переменных: мы производим замену переменных типа $x = Mx^1$, где x – старый набор переменных (система координат), x^1 – новый набор переменных, а M – унимодулярная матрица. Такое преобразование часто позволяет упрощать задачу и всегда сохраняет отношения один к одному между решениями старой и новой задачи.

2. Отсечение допустимого решения: если мы уже получили одно допустимое решение x^*, со значением критерия g, то мы можем ввести в задачу новое ограничение: $c^T x \ge g + 1$. Новая задача не содержит решения x^*, и все решения, которые она содержит, будут строго лучше x^* или любого другого решения, отсеченного новым условием. Таким образом, если глобальный оптимум новой задачи существует, то он является и глобальным оптимумом оригинальной задачи, если не существует, то x^* является глобальным оптимумом.

Алгоритм решения задачи ЦЛП

I. Если в задаче есть равенства, то матрицу равенств приведем к нормальной форме Смита (применив полученную матрицу замены переменных к неравенствам соответственно) [2, 3]. В получившейся задаче равенства содержат не более одной переменной и могут быть решены последовательно.

Возможны три случая:

1. Уравнения противоречивы (например, $x = 3$ и $x = 5$) или не имеют целочисленных решений (например, $2x = 3$ или $0x = 6$), следовательно модель не имеет решений.

2. Все переменные назначены. Если такое назначение удовлетворяет всем неравенствам, то получено единственное решение, иначе решений нет.

3. Часть переменных (равная количеству независимых уравнений) назначена, остальные переменные входят в систему неравенств, которая формулируется так:

$$\begin{cases} Ax \le b \\ x \in \mathbb{Z}^n \\ c^T x \to max \end{cases}, \tag{4}$$

где c, b – векторы, а A – матрица, все элементы которых являются целыми числами. В системе n переменных и m неравенств.

II. Проверим столбцы матрицы A на линейную зависимость. Если линейная зависимость найдена, проведем линейное преобразование, которое обратит зависимые столбцы в нули. Если у соответствующей переменной в критерии стоит ненулевой коэффициент, то, изменяя эту переменную, критерий задачи можно сделать неограниченно большим или малым при условии, что оставшаяся система совместна. Если коэффициент в критерии нулевой, то эта переменная не имеет никакого значения и может быть назначена нулем. После процедуры остается система линейных неравенств с гарантированно линейно независимыми переменными.

III. Проверим систему неравенств на неограниченность области допустимых значений. Для этого решим m систем нецелочисленных неравенств вида:

$$\begin{aligned} a_i x^j &\le -1,\ j = i,\ i \in [1..m] \\ a_i x^j &\le\ \ 0,\ j \ne i,\ i \in [1..m] \end{aligned} \tag{5}$$

для j от 1 до m симплекс-методом. Назначим $x^j = 0$, если -ая задача не имеет решений, и домножим x^j на общий знаменатель, если решение нашлось дробное – тогда эти вектора всегда существуют и являются целочисленными (умножение вектора x^j на любое число больше единицы не может нарушить условия задачи). Введем вектора $x' = \sum_{j=1}^{m} x^j$ и $b' = Ax'$.

Вектора x^j имеют смысл направления, в котором можно перемещать допустимое решение системы в n-мерном пространстве таким образом, что оно может бесконечно удаляться от гиперплоскости -ого ограничения, не нарушая остальные ограничения. Вектор x' – направление, по которому можно отдалить допустимое решение от максимального количества гиперплоскостей ограничений, не нарушив оставшиеся. Вектор b' – вектор относительных скоростей, с которыми решение будет отдаляться.

Возможны три случая:

1. Вектор b' – нулевой. В таком случае область допустимых решений ограничена.

2. Вектор b' состоит только из ненулевых отрицательных целых чисел. Найдем минимальное положительное целое число k, что $kb' \le b$. Тогда $kb' = kAx' = A(kx') \le b$ и kx' является допустимым решением исходной системы неравенств. Если при увеличении k критерий ухудшается, отсечем допустимое решение и решим получившуюся систему начиная с пункта III, в противном случае критерий не ограничен, и задача считается решенной.

3. Вектор b' имеет как нулевые, так и отрицательные значения. В таком случае составим систему неравенств, выбрав из исходной только те, которые соответствуют нулевым значениям b': $A^* x^* \le b^*$. Решим эту систему, начиная с пункта II. Если решений нет, то и исходная система несовместна. Если нашлось решение x^*, то найдем допустимое решение исходной системы в виде $x = x^* + kx'$, где k – минимальное положительное целое число. Если при увеличении k критерий ухудшается, отсечем допустимое решение и решим получившуюся систему начиная с пункта III, в противном случае критерий не ограничен, и задача считается решенной.

IV. Решим ограниченную систему линейных неравенств. В первую очередь воспользуемся алгоритмом линейной упаковки [4] для минимизации ограничивающего гиперпараллелепипеда области допустимых решений. Это уменьшит ветвление алгоритма и одновременно увеличит шанс успеха на каждом шаге.

Затем построим модифицированную версию задачи: оставим матрицу коэффициентов A и критерий c, но изменим вектор свободных членов следующим образом:

$$b_i^k = b_i - \sum_{j=1}^{n} \frac{|A_{i,j}|}{2},\ i \in [1..m] \tag{6}$$

Решим задачу $Aq \leq b^k,\ c^T q \rightarrow max,\ q \in \mathbb{R}$ симплекс-методом. Вектор q имеет смысл центра единичного гиперкуба, полностью принадлежащего области допустимых решений исходной системы неравенств. Единичный гиперкуб всегда содержит хотя бы одну целочисленную точку, и для вычисления достаточно округлить координаты центра гиперкуба [4].

Если удастся получить дробный вектор q, удовлетворяющий условиям задачи, то допустимое решение исходной целочисленной задачи найдено, его можно отсечь и продолжать решение с пункта IV**.** В противном случае выберем переменную с самым маленьким диапазоном в области допустимых значений, назначаем ей все возможные значения и рекурсивно решаем каждую получившуюся подзадачу (в которых будет на одну переменную меньше) с пункта IV**.**

Заключение

В статье рассмотрен блочный подход к генерации поверхностной расчетной сетки на примере создания переходного сеточного слоя. Приведено описание алгоритма решения задачи целочисленного линейного программирования, используемого для согласования числа сеточных узлов на границах грани и виртуальных блоков.

В качестве недостатка описанного подхода можно указать использование линейного критерия в задаче оптимизации. Возможный путь улучшения для задачи сеткогенерации – это переход к решению нелинейных оптимизационных задач, в частности включение функции модуля в критерий.

В тоже время, даже текущий вариант алгоритма позволил решить важную практическую задачу в рамках препостпроцессора пакета программ ЛОГОС. В целях экономии времени, затрачиваемого на математическое моделирование, инженеры стремятся выдержать примерно одинаковый размер сеточных элементов в разных частях конструкции, не допуская лишнего избыточного сгущения. Это приводит к появлению ситуаций с разным уровнем сеточной детализации на различных фрагментах геометрической модели, как показано на рис. 6. На основе описанного в работе алгоритма в рамках препостпроцессора реализован специальный режим сеткогенерации, решающий задачу согласования.

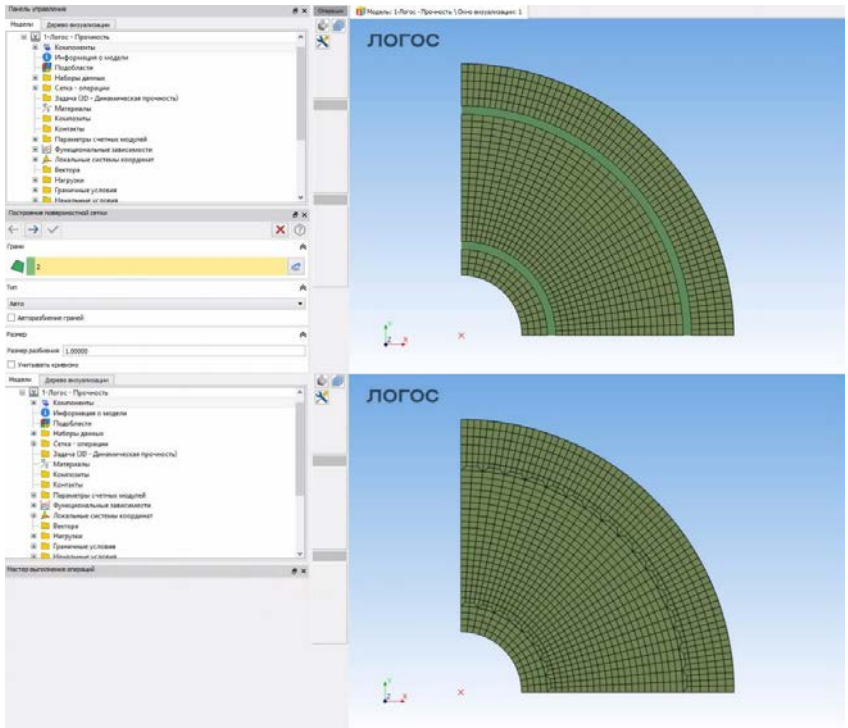


Рис. 6. Исходные данные для работы алгоритма (сверху) и результат применения (внизу)

Литература

1. Афраймович Л. Г., Прилуцкий М. Х., Потехин А. Л., Старостин Н. В., Тимофеев А. Е, Филимонов А. В. Распознавание геометрических сущностей с целью последующей генерации сеток для моделирования прочностных процессов // Супервычисления и математическое моделирование. Труды XVI Международной конференции» / Под ред. Шагалиева Р. М. – Саров: ФГУП «РФЯЦ-ВНИИЭФ». 2016. С. 50–51

2. Шумилов, И. А. Алгоритмы редукции систем целочисленных уравнений и неравенств: выпускная квалификационная работа бакалавра по напр. Прикладная информатика: 09.03.03: защищена 19.06.17. Н. Новгород, 2017. С. 56 .

3. Шумилов И. А. Алгоритмы редукции систем целочисленных уравнений и неравенств // XXIII Нижегородская сессия молодых ученых. Математические науки. Тезисы докладов. Нижний Новгород. 2018.

4. Шумилов, И. А. Алгоритмы решения задач целочисленного линейного программирования: магистерская диссертация по напр. Прикладная информатика: 09.04.03: защищена 10.06.19. Н. Новгород, 2019. С. 41 .

ADAPTIVE GENERATION OF REGULAR CALCULATION NETWORKS BASED ON THE SOLUTION OF TASKS OF AN INTEGER LINEAR PROGRAMMING

I. A. Shumilov, A. E. Timofeev, A. L. Potekhin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

For the formation of computational grids on the surfaces of geometric models, there are various strategies. The aim of this study is to develop an algorithm for block generation of regular grids with a representation of the boundary conditions for specifying a grid in the form of a system of integer linear equations. The criterion for the integer programming problem is to minimize the number of changes in the initial distribution of grid nodes at the boundary. To solve the problem, an algorithm is used to solve integer linear programming problems, which in practice quickly finds a feasible solution close to optimal and can be used to calculate the global optimum. The developed algorithm allows, if necessary, to change the initial number of grid nodes at the model boundary for applying block generation of a regular grid. The developed algorithm made it possible, within the framework of the prepost processor of the LOGOS software package, to obtain the possibility of constructing surface grids that ensure matching of mesh fragments with different levels of detail.

Key words: integer linear programming; surface grid; computational grid; block generation.

СВЕДЕНИЯ ОБ АВТОРАХ

Аблесимов Владимир Евгеньевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, кандидат физико-математических наук
Абросимов Николай Анатольевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, главный научный сотрудник, доктор физико-математических наук
Абузяров Мустафа Хасьянович – НИИМ ННГУ им. Н. И. Лобачевского, г. Нижний Новгород, Россия, ведущий научный сотрудник, кандидат физико-математических наук, e-mail: abouziar@mech.unn.ru
Алешкин Виталий Анатольевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: VAAleshkin@vniief.ru
Аликулов Елбек Абдукаимович – ВУНЦ «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина» Министерства обороны РФ, г. Воронеж, адъюнкт, e-mail: adunkt83@bk.ru
Афанасьев Владимир Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: VlAAfanasev@vniief.ru
Бабанов Алексей Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, кандидат физико-математических наук, e-mail: AVBabanov@vniief.ru
Баканов Егор Константинович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: EKBakanov@vniief.ru
Балуева Наталья Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: NNBalueva@vniief.ru
Барабанов Роман Анатольевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: RABarabanov@vniief.ru
Баранов Алексей Валерьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AlVBaranov@vniief.ru
Баранов Алексей Валерьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AlekVBaranov@vniief.ru
Бахаев Александр Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ANBakhaev@vniief.ru
Беженцева Мария Владимировна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: MVSkobeeva@vniief.ru
Бетелин Владимир Борисович – ФГУ ФНЦ НИИСИ РАН, г. Москва, Россия, научный руководитель, академик РАН
Бондаренко Сергей Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-теоретической лаборатории, e-mail: SVBondarenko@vniief.ru
Борисенко Ольга Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: ONBorisenko@vniief.ru
Борляев Владимир Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: VlVBorlyaev@vniief.ru
Бритова Ольга Игоревна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, математик, e-mail: OIBritova@vniief.ru
Быков Александр Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, кандидат физико-математических наук, e-mail: ANBykov@vniief.ru
Веселова Елена Александровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: EAVeselova@vniief.ru

Ветчинников Максим Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: MVVetchinnikov@vniief.ru
Вишняков Андрей Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AYVishnyakov@vniief.ru
Габзетдинова Лилия Ясировна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник
Галкин Валерий Алексеевич – Сургутский филиал ФГУ ФНЦ НИИСИ РАН, СурГУ, г. Сургут Тюменской области, Россия, директор, профессор, доктор физико-математических наук, e-mail: val-gal@yandex.ru
Галкин Максим Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: MVGalkin@vniief.ru
Гамов Лев Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: LVGamov@vniief.ru
Глазунов Виктор Алексеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель начальника научно-исследовательского отдела, начальник научно-исследовательской лаборатории, e-mail: VAGlazunov@vniief.ru
Голубкова Елена Федоровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий инженер-программист, e-mail: E. F.Golubkova@vniief.ru
Гордеев Алексей Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: AVGordeev@vniief.ru
Гордеев Дмитрий Геннадьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: DGGordeev@vniief.ru
Губайдулина Елена Алексеевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: EAEskova@vniief.ru
Гурин Алексей Михайлович – ИГиЛ СО РАН, г. Новосибирск, Россия, младший научный сотрудник, e-mail: guralisk@mail.ru
Данилов Артем Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: ASDanilov@vniief.ru
Девятых Дмитрий Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, кандидат физико-математических наук, e-mail: dvdevyatykh@vniief.ru
Демина Марина Александровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: MADemina@vniief.ru
Дерюгин Юрий Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, главный научный сотрудник, доктор физико-математических наук, e-mail: YNDeryugin@vniief.ru
Дордопуло Алексей Игоревич – ООО ««НИЦ супер-ЭВМ и нейрокомпьютеров», г. Таганрог Ростовской области, Россия, начальник отдела, кандидат технических наук, e-mail: dordopulo@superevm.ru
Доронченко Юрий Иванович – ООО ««НИЦ супер-ЭВМ и нейрокомпьютеров», г. Таганрог Ростовской области, Россия, технический директор, кандидат технических наук, e-mail: doronchenko@superevm.ru
Дубовик Алексей Олегович – СурГУ, г. Сургут Тюменской области, Россия, старший преподаватель кафедры прикладной математики, кандидат физико-математических наук, e-mail: alldubovik@gmail.com
Дьяков Александр Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: AVDyakov@vniief.ru
Дьянов Денис Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: DYDyanov@vniief.ru
Дюпин Владимир Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: VNDyupin@vniief.ru

Евстифеева Елена Олеговна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: EOEvstifeeva@vniief.ru
Егоров Александр Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник
Ежов Дмитрий Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: DVlEzhov@vniief.ru
Елесин Александр Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, кандидат физико-математических наук, e-mail: elesin@mech.unn.ru
Елизаров Георгий Сергеевич – ФГУП «Научно-исследовательский институт «Квант», г. Москва, Россия, директор, доктор технических наук, e-mail: elizarov@rdi-kvant.ru
Елизаров Сергей Георгиевич – МГУ имени М.В.Ломоносова, г. Москва, Россия, руководитель Дизайн-центра цифровой микроэлектроники, кандидат физико-математических наук, e-mail: elizarov@physics.msu.ru
Емельянов Андрей Борисович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ABEmelyanov@vniief.ru
Еременко Андрей Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: AYuEremenko@vniief.ru
Жильникова Наталия Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: NNZhilnikova@vniief.ru
Жирнов Владимир Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: VVZhirnov@vniief.ru
Журавлева Мария Вадимовна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник
Жучков Роман Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, кандидат технических наук, e-mail: RNZhuchkov@vniief.ru
Задорожный Константин Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: KVZadorozhnyy@vniief.ru
Занегин Игорь Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, e-mail: IVZanegin@vniief.ru
Зеленский Дмитрий Константинович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель начальника научно-исследовательского отдела – начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: DKZelenskiy@vniief.ru
Змеев Дмитрий Николаевич – ИППМ РАН, г. Москва, Россия, научный сотрудник, e-mail: zmejevdn@list.ru
Иванов Константин Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель начальника научно-исследовательского отдела – начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: KVIvanov@vniief.ru
Казанцев Александр Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AlVKazantsev@vniief.ru
Катыков Алексей Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ANKatykov@vniief.ru
Каякин Алексей Анатольевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, e-mail: AAKayakin@vniief.ru
Кидямкина Дина Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: DNKidyamkina@vniief.ru
Климов Аркадий Валентинович – ИППМ РАН, г. Москва, Россия, старший научный сотрудник, e-mail: arkvklimov@yandex.ru
Коваленко Олег Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородскойобласти, Россия, начальник научно-исследовательского отдела, кандидат физико-математических наук, e-mail: OVKovalenko@vniief.ru
Кожаев Дмитрий Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, e-mail: DAKozhaev@vniief.ru

Козелков Андрей Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, доктор физико-математических наук, e-mail: ASKozelkov@vniief.ru
Козлов Дмитрий Олегович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: DmOKozlov@vniief.ru
Конотопцев Валерий Николаевич – ФГУП «Научно-исследовательский институт «Квант», г. Москва, Россия, начальник отдела, e-mail: valko-19@rdi-kvant.ru
Константинов Александр Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, главный научный сотрудник, доктор физико-математических наук
Корнеев Виктор Владимирович – ФГУП «Научно-исследовательский институт «Квант», г. Москва, Россия, главный научный сотрудник, доктор технических наук, e-mail: korv@rdi-kvant.ru
Коротков Алексей Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: AVKorotkov@vniief.ru
Косарим Снежана Сергеевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: SSKosarim@vniief.ru
Кошелева Елена Викторовна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, инженер-исследователь 1 категории, e-mail: EVKosheleva@vniief.ru
Красноярова Дарина Александровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: DAKrasnoyarova@vniief.ru
Крючков Илья Алиевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: IAKryuchkov@vniief.ru
Куделькин Вадим Григорьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, кандидат физико-математических наук, e-mail: VGKudelkin@vniief.ru
Кузнецов Виктор Андреевич – ВУНЦ «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина» Министерства обороны РФ, г. Воронеж, доцент кафедры эксплуатации бортового авиационного радиоэлектронного оборудования, кандидат технических наук, e-mail: kuzzviktor@mail.ru
Кузнецов Владимир Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: vlykuznetsov@vniief.ru
Куликова Мария Вячеславовна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: MVKulikova@vniief.ru
Куркин Андрей Александрович – НГТУ им. Р.Е. Алексеева, г. Нижний Новгород, Россия, проректор по научной работе, доктор физико-математических наук, профессор, e-mail: aakurkin@gmail.com
Куркина Оксана Евгеньевна – НГТУ им. Р.Е. Алексеева, г. Нижний Новгород, Россия, ведущий научный сотрудник, кандидат физико-математических наук, доцент, e-mail: oksana.kurkina@mail.ru
Курулин Вадим Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской группы, кандидат физико-математических наук, e-mail: VVKurulin@vniief.ru
Лазарев Алексей Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: AALazarev@vniief.ru
Лазарев Владимир Валерьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: VlVLazarev@vniief.ru
Ларькина Ольга Сергеевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, математик, e-mail: OSLarkina@vniief.ru
Лашкин Сергей Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: SVLashkin@vniief.ru
Левин Илья Израилевич – ООО «НИЦ супер-ЭВМ и нейрокомпьютеров», г. Таганрог Ростовской области, Россия, директор, доктор технических наук, профессор, e-mail: levin@superevm.ru

Левина Марина Геннадиевна – ООО «НИЦ супер-ЭВМ и нейрокомпьютеров», г. Таганрог Ростовской области, Россия, начальник отдела, кандидат технических наук, доцент, e-mail: mgl-2004@yandex.ru
Левченко Николай Николаевич – ИППМ РАН, г. Москва, Россия, заведующий отделом, кандидат технических наук, e-mail: nick@ippm.ru
Лобовиков Павел Викторович – НГТУ им. Р.Е. Алексеева, г. Нижний Новгород, Россия, инженер, e-mail: plobovikov@gmail.com
Лобовикова Лидия Валерьевна – НГТУ им. Р.Е. Алексеева, г. Нижний Новгород, Россия, младший научный сотрудник, e-mail: talalushkina.lida@gmail.com
Ломайкин Александр Иванович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AILomaykin@vniief.ru
Ломтев Василий Валерьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: VVLomtev@vniief.ru
Машенькин Павел Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: PAMashenkin@vniief.ru
Медведкина Марина Валентиновна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: MVMedvedkina@vniief.ru
Меньших Николай Александрович – ФГАУ Военный инновационный технополис «ЭРА», Анапа, командир взвода (научного) – младший научный сотрудник, e-mail: nikolai19951225@gmail.com
Милешин Иван Геннадьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, математик, e-mail: igmileshin@vniief.ru
Митрофанов Евгений Игоревич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: EIMitrofanov@vniief.ru
Муратов Родион Владимирович – ФГУП «ВНИИА» им. Н.Л. Духова, г. Москва, Россия, научный сотрудник, e-mail: rodyon-mur@yandex.ru
Мустаева Надежда Анатольевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: NAMustaeva@vniief.ru
Надуев Александр Геннадиевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: AGNaduev@vniief.ru
Наумов Андрей Олегович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AONaumov@vniief.ru
Наумова Евгения Игоревна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: EINaumova@vniief.ru
Никитин Вячеслав Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: VyANikitin@vniief.ru
Ниткина Екатерина Сергеевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: ESNitkina@vniief.ru
Новиков Иван Геннадьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, кандидат физико-математических наук, e-mail: IGNovikov@vniief.ru
Новиков Олег Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: OVNovikov@vniief.ru
Огородников Андрей Валерьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: AVOgorodnikov@vniief.ru
Окунев Анатолий Семенович – ИППМ РАН, г.Москва, Россия, ведущий научный сотрудник, кандидат технических наук
Олесницкая Ксения Константиновна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: KKOlesnitskaya@vniief.ru
Панкратов Дмитрий Михайлович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: DMPankratov@vniief.ru
Панов Александр Иванович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AIPanov@vniief.ru

Пашарина Ольга Юрьевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник
Пищулин Игорь Анатольевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: IAPischulin@vniief.ru
Побуринная Нина Анатольевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: NAPoburinnaya@vniief.ru
Полищук Сергей Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: SNPolischuk@vniief.ru
Половникова Татьяна Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник
Попов Владимир Витальевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, кандидат физико-математических наук, e-mail: VVPopov@vniief.ru
Потапкина Людмила Федоровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: LFPotapkina@vniief.ru
Потапов Александр Алексеевич – ИРЭ им. В.А. Котельникова РАН, Москва, Россия, главный научный сотрудник, доктор физико-математических наук, профессор, e-mail: potapov@cplire.ru
Потехин Антон Львович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, кандидат физико-математических наук, e-mail: ALPotekhin@vniief.ru
Резвова Татьяна Владимировна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: TVRezvova@vniief.ru
Речкин Вадим Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель начальника расчетно-методического отдела, начальник расчетно-методической группы, кандидат технических наук, e-mail: VNRechkin@vniief.ru
Романов Владимир Игоревич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель начальника научно-исследовательского отделения – начальник расчетно-методического отдела, кандидат технических наук, e-mail: VIRomanov@vniief.ru
Русяк Иван Григорьевич – ИжГТУ им. М.Т. Калашникова, г. Ижевск, Россия, заведующий кафедрой, доктор технических наук, профессор, e-mail: primat@istu.ru
Рябов Александр Алексеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, заместитель главного конструктора РФЯЦ-ВНИИЭФ, начальник научно-исследовательского отделения, доктор физико-математических наук, e-mail: AARyabov@vniief.ru
Сайфуллин Айдар Ильдарович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AISayfullin@vniief.ru
Сайфуллина Рамиля Наиловна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: RNSayfullina@vniief.ru
Самодолов Алексей Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской группы
Санталов Алексей Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ASSantalov@vniief.ru
Сапронов Иван Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: ISSapronov@vniief.ru
Сапронова Ольга Вячеславовна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: OVSapronova@vniief.ru
Саразов Алексей Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AVSarazov@vniief.ru
Свидинский Артем Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AVSvidinskiy@vniief.ru
Сельченкова Надежда Ивановна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник
Серякова Ирина Валерьевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: IVSeryakova@vniief.ru

Симаков Виталий Юрьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: VYuSimakov@vniief.ru
Синицин Евгений Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: EASinitsin@vniief.ru
Собанин Данила Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: DSSobanin@vniief.ru
Соколов Сергей Сергеевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, доктор физико-математических наук, e-mail: SSSokolov@vniief.ru
Соловьев Александр Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской группы, e-mail: ANSolovev@vniief.ru
Софронов Василий Николаевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, e-mail: VNSofronov@vniief.ru
Степаненко Сергей Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, главный научный сотрудник, доктор физико-математических наук, e-mail: SAStepanenko@vniief.ru
Стручков Андрей Викторович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: AnVStruchkov@vniief.ru
Сурина Ольга Юрьевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: OYuSurina@vniief.ru
Суфиянов Вадим Гарайханович – ИжГТУ им. М. Т. Калашникова, г. Ижевск, Россия, профессор, доктор технических наук, e-mail: vsufiy@mail.ru
Тимофеев Алексей Евгеньевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области Россия, начальник научно-исследовательской лаборатории
Титов Михаил Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: MATitov@vniief.ru
Тишкин Дмитрий Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: DATishkin@vniief.ru
Трунин Иван Рюрикович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, ведущий научный сотрудник, e-mail: IRTrunin@vniief.ru
Тюндина Анна Александровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: AATyundina@vniief.ru
Тюхтина Полина Александровна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: PATyukhtina@vniief.ru
Тятюшкина Елена Сергеевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, кандидат физико-математических наук, e-mail: ESTyatyushkina@vniief.ru
Уткин Дмитрий Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник, e-mail: DAUtkin@vniief.ru
Учаев Александр Яковлевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, главный научный сотрудник
Филимонкин Евгений Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: EAFilimonkin@vniief.ru
Цалко Тарас Валентинович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, научный сотрудник, e-mail: TVTsalko@vniief.ru
Циберев Кирилл Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, кандидат физико-математических наук, e-mail: KVtsiberev@vniief.ru
Чайка Андрей Иванович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AIChayka@vniief.ru
Челаков Артем Александрович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AAChelakov@vniief.ru

Черевань Александр Дмитриевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ADCherevan@vniief.ru
Шанин Игорь Иннокентьевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательского отдела, кандидат технических наук
Шаров Олег Олегович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: OOSharov@vniief.ru
Шемякин Андрей Вячеславович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: AVShemyakin@vniief.ru
Шиберин Игорь Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории
Шувалова Екатерина Владимировна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, начальник научно-исследовательской лаборатории, e-mail: EVShuvalova@vniief.ru
Шумилина Ольга Николаевна – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: ONShumilina@vniief.ru
Шумилов Иван Андреевич – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, младший научный сотрудник
Шурыгин Александр Владимирович – ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской области, Россия, старший научный сотрудник, e-mail: AVShurygin@vniief.ru

INFORMATION ABOUT AUTHORS

Ablesimov Vladimir Evgenyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod region, Russia, leading scientist, Candidate of Science in Physics and Mathematics
Abrosimov Nikolay Anatolyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, chief scientist, Doctor of Science in Physics and Mathematics
Abuzyarov Mustafa Khasyanovich – NIIM NNSU, Nizhny Novgorod, Russia, leading scientist, Candidate of Science in Physics and Mathematics, e-mail: abouziar@mech.unn.ru
Afanasyev Vladimir Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: VlAAfanasev@vniief.ru
Aleshkin Vitaliy Anatolyevich – FSUE RFNC-VNIIEF, Sarov Nizhny Novgorod Region, Russia, scientist, e-mail: VAAleshkin@vniief.ru
Alikulov Elbek Abdukaimovich – Military Science and Training Center "Airforce Academy named after Professor N.E. Zhukovsky and Yu.A. Gagarin" of the Ministry of Defense
of the Russian Federation, Voronezh, adjunct, e-mail: adunkt83@bk.ru
Babanov Aleksey Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Physics and Mathematics, e-mail: AVBabanov@vniief.ru
Bakanov Egor Konstantinovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: EKBakanov@vniief.ru
Bakhaev Aleksandr Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ANBakhaev@vniief.ru
Balueva Natalya Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: NNBalueva@vniief.ru
Barabanov Roman Anatolyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: RABarabanov@vniief.ru
Baranov Aleksey Valeryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AlekVBaranov@vniief.ru
Baranov Aleksey Valeryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AlVBaranov@vniief.ru
Betelin Vladimir Borisovich – SRISA/NIISI RAS, Moscow, Russia, research supervisor, Academician of RAS
Bezhentseva Mariya Vladimirovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: MVSkobeeva@vniief.ru
Bondarenko Sergey Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of scientific-theoretical laboratory, e-mail: SVBondarenko@vniief.ru
Borisenko Olga Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: ONBorisenko@vniief.ru
Borlyaev Vladimir Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: VlVBorlyaev@vniief.ru
Britova Olga Igorevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, mathematician, e-mail: OIBritova@vniief.ru
Bykov Aleksandr Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Physics and Mathematics, e-mail: ANBykov@vniief.ru
Chayka Andrey Ivanovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AIChayka@vniief.ru
Chelakov Artem Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AAChelakov@vniief.ru
Cherevan Aleksandr Dmitrievich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ADCherevan@vniief.ru

Danilov Artem Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: ASDanilov@vniief.ru
Demina Marina Aleksandrovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: MADemina@vniief.ru
Deryugin Yury Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, chief scientist, Doctor of Science in Physics and Mathematics, e-mail: YNDeryugin@vniief.ru
Devyatykh Dmitriy Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, Candidate of Science in Physics and Mathematics, e-mail: dvdevyatykh@vniief.ru
Dordopulo Aleksey Igorevich – SRC SC&NC, Taganrog, Rostov region, Russia, head of department, Candidate of Science in Engineering, e-mail: dordopulo@superevm.ru
Doronchenko Yury Ivanovich – SRC SC&NC, Taganrog, Rostov region, Russia, engineering director, Candidate of Science in Engineering, e-mail: doronchenko@superevm.ru
Dubovik Aleksey Olegovich – SurSU, Surgut, Tyumen region, Russia, senior lecturer, Department of Applied Mathematics, Candidate of Science in Physics and Mathematics, e-mail: alldubovik@gmail.com
Dyakov Aleksandr Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: AVDyakov@vniief.ru
Dyanov Denis Yuryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: DYDyanov@vniief.ru
Dyupin Vladimir Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: VNDyupin@vniief.ru
Egorov Aleksandr Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist
Elesin Aleksandr Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, Candidate of Science in Physics and Mathematics, e-mail: elesin@mech.unn.ru
Elizarov Georgiy Sergeevich – FSUE Research Institue “Kvant”, Moscow, Russia, director, Doctor of Science in Engineering, e-mail: elizarov@rdi-kvant.ru
Elizarov Sergey Georgievich – Lomonosov Moscow State University, Moscow, Russia, head of design-center of digital microelectronics, Candidate of Science in Physics and Mathematics, e-mail: elizarov@physics.msu.ru
Emelyanov Andrey Borisovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ABEmelyanov@vniief.ru
Eremenko Andrey Yuryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: AYuEremenko@vniief.ru
Evstifeeva Elena Olegovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: EOEvstifeeva@vniief.ru
Ezhov Dmitriy Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: DVlEzhov@vniief.ru
Filimonkin Evgeniy Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: EAFilimonkin@vniief.ru
Gabzetdinova Liliya Yasirovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist
Galkin Maksim Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: MVGalkin@vniief.ru
Galkin Valeriy Alekseevich – Surgut Branch of SRISA/NIISI RAS, SurSU, Surgut, Tyumen region, Russia, director, Professor, Doctor of Science in Physics and Mathematics, e-mail: val-gal@yandex.ru
Gamov Lev Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: LVGamov@vniief.ru
Glazunov Viktor Alekseevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy head of research department, head of research laboratory, e-mail: VAGlazunov@vniief.ru
Golubkova Elena Fedorovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading software engineer, e-mail: E.F.Golubkova@vniief.ru

Gordeev Aleksey Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: AVGordeev@vniief.ru
Gordeev Dmitriy Gennadyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: DGGordeev@vniief.ru
Gubaydulina Elena Alekseevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: EAEskova@vniief.ru
Gurin Aleksey Mikhailovich – LIH SB RAS, Novosibirsk, Russia, junior scientist, e-mail: guralisk@mail.ru
Ivanov Konstantin Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy head of reasearch department – head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: KVIvanov@vniief.ru
Katykov Aleksey Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ANKatykov@vniief.ru
Kayakin Aleksey Anatolyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, e-mail: AAKayakin@vniief.ru
Kazantsev Aleksandr Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AlVKazantsev@vniief.ru
Kidyamkina Dina Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: DNKidyamkina@vniief.ru
Klimov Arkadiy Valentinovich –Institute for Design Problems in Microelectronics, Moscow, Russia, senior scientist, e-mail: arkvklimov@yandex.ru
Konotoptsev Valeriy Nikolaevich – FSUE Research Institute “Kvant”, Moscow, Russia, head of department, e-mail: valko-19@rdi-kvant.ru
Konstantinov Aleksandr Yurievich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, chief scientist, Doctor of Science in Physics and Mathematics
Korneev Viktor Vladimirovich – FSUE Research Institute “Kvant”, Moscow, Russia, chief scientist, Doctor of Science in Enginnering, e-mail: korv@rdi-kvant.ru
Korotkov Aleksey Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: AVKorotkov@vniief.ru
Kosarim Snezhana Sergeevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: SSKosarim@vniief.ru
Kosheleva Elena Viktorovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, first category research engineer, e-mail: EVKosheleva@vniief.ru
Kovalenko Oleg Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Physics and Mathematics, e-mail: OVKovalenko@vniief.ru
Kozelkov Andrey Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Doctor of Science in Physics and Mathematics, e-mail: ASKozelkov@vniief.ru
Kozhaev Dmitriy Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, e-mail: DAKozhaev@vniief.ru
Kozlov Dmitriy Olegovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: DmOKozlov@vniief.ru
Krasnoyarova Darina Aleksandrovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: DAKrasnoyarova@vniief.ru
Kryuchkov Ilya Alievich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: IAKryuchkov@vniief.ru
Kudelkin Vadim Grigoryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, Candidate of Science in Physics and Mathematics, e-mail: VGKudelkin@vniief.ru
Kulikova Mariya Vyacheslavovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: MVKulikova@vniief.ru

Kurkin Andrey Aleksandrovich – NNSTU n.a. R.E. Alekseev, Nizhny Novgorod, Russia, vice-president for scientific work, Doctor of Science in Physics and Mathematics, Professor, e-mail: aakurkin@gmail.com
Kurkina Oksana Evgenyevna – NNSTU n. a. R.E. Alekseev, Nizhny Novgorod, Russia, leading scientist, Candidate of Science in Physics and Mathematics, Assistant Professor, e-mail: oksana.kurkina@mail.ru
Kurulin Vadim Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research group, Candidate of Science in Physics and Mathematics, e-mail: VVKurulin@vniief.ru
Kuznetsov Viktor Adnreevich – Military Science and Training Center "Airforce Academy named after Professor N. E. Zhukovsky and Yu. A. Gagarin" of the Ministry of Defense of the Russian Federation, Voronezh, assistant professor, Department of Operation of On-board Aviation Radio Equipment, Candidate of Science in Engineering, e-mail kuzzviktor@mail.ru
Kuznetsov Vladimir Yuryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: vlykuznetsov@vniief.ru
Larkina Olga Sergeevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, mathematician, e-mail: OSLarkina@vniief.ru
Lashkin Sergey Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: SVLashkin@vniief.ru
Lazarev Aleksey Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: AALazarev@vniief.ru
Lazarev Vladimir Valeryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: VlVLazarev@vniief.ru
Levchenko Nikolay Nikolaevich –Institute for Design Problems in Microelectronics of RAS, Moscow, Russia, head of department, Candidate of Science in Engineering, e-mail: nick@ippm.ru
Levin Ilya Izrailevich – SRC SC&NC, Taganrog, Rostov region, Russia, director, Doctor of Science in Engineering, Professor, e-mail: levin@superevm.ru
Levina Marina Gennadievna – SRC SC&NC, Taganrog, Rostov region, Russia, head of department, Candidate of Science in Engineering, Assistant Professor, e-mail: mgl-2004@yandex.ru
Lobovikov Pavel Viktorovich – NNSTU n. a. R.E. Alekseev, Nizhny Novgorod, Russia, engineer, e-mail: plobovikov@gmail.com
Lobovikova Lidiya Valeryevna – NNSTU n. a. R.E. Alekseev, Nizhny Novgorod, Russia, junior scientist, e-mail: talalushkina.lida@gmail.com
Lomaykin Aleksandr Ivanovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AILomaykin@vniief.ru
Lomtev Vasiliy Valeryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: VVLomtev@vniief.ru
Mashenkin Pavel Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: PAMashenkin@vniief.ru
Medvedkina Marina Valentinovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: MVMedvedkina@vniief.ru
Menshikh Nikolay Aleksandrovich – Federal State Autonomous Institution "Military Innovative Technopolis "ERA", Anapa, (scientific) platoon commander– junior scientist, e-mail: nikolai19951225@gmail.com
Mileshin Ivan Gennadyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, mathematician, e-mail: igmileshin@vniief.ru
Mitrofanov Evgeniy Irorevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: EIMitrofanov@vniief.ru
Muratov Rodion Vladimirovich – FSUE VNIIA, Moscow, Russia, scientist, e-mail: rodyon-mur@yandex.ru
Mustaeva Nadezhda Anatolyevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: NAMustaeva@vniief.ru
Naduev Aleksandr Gennadyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: AGNaduev@vniief.ru
Naumov Andrey Olegovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AONaumov@vniief.ru

Naumova Evgeniya Igorevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: EINaumova@vniief.ru
Nikitin Vyacheslav Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: VyANikitin@vniief.ru
Nitkina Ekaterina Sergeevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: ESNitkina@vniief.ru
Novikov Ivan Gennadiyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: IGNovikov@vniief.ru
Novikov Oleg Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: OVNovikov@vniief.ru
Ogorodnikov Andrey Valeryevich – FSUE RFNC-VNIIEF, Sarov Nizhny Novgorod Region, Russia, Head of Research Laboratory, e-mail: AVOgorodnikov@vniief.ru
Okunev Anatoliy Semenovich –Institute for Design Problems in Microelectronics of RAS, Moscow, Russia, leading scientist, Candidate of Science in Engineering
Olesnitskaya Kseniya Konstantinovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: KKOlesnitskaya@vniief.ru
Pankratov Dmitriy Mikhaylovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: DMPankratov@vniief.ru
Panov Aleksandr Ivanovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AIPanov@vniief.ru
Pasharina Olga Yuryevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist
Pishchulin Igor Anatolyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: IAPischulin@vniief.ru
Poburinnaya Nina Anatolyevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: NAPoburinnaya@vniief.ru
Polishchuk Sergey Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: SNPolischuk@vniief.ru
Polovnikova Tatyana Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist
Popov Vladimir Vitalyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, Candidate of Science in Physics and Mathematics, e-mail: VVPopov@vniief.ru
Potapkina Ludmila Fedorovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: LFPotapkina@vniief.ru
Potapov Aleksandr Alekseevich – Kotelnikov Institute of Radioengineering and Electronics (IRE) of RAS, Moscow, Russia, chief scientist, Doctor of Science in Physics and Mathematics, Professor, e-mail: potapov@cplire.ru
Potekhin Anton Lvovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Physics and Mathematics, e-mail: ALPotekhin@vniief.ru
Rechkin Vadim Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy head of computational and methodological department – head of computational and methodological group, Candidate of Science in Engineering, e-mail: VNRechkin@vniief.ru
Rezvova Tatyana Vladimirovna – FSUE RFNC-VNIIEF, Sarov Nizhny Novgorod Region, Russia, senior scientist, e-mail: TVRezvova@vniief.ru
Romanov Vladimir Igorevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy head of research department – head of computational and methodological department, Candidate of Science in Engineering, e-mail: VIRomanov@vniief.ru
Rusyak Ivan Grigoryevich – Kalashnikov Izhevsk State Technical University, Izhevsk, Russia, head of university department, Doctor of Science in Engineering, Professor, e-mail: primat@istu.ru
Ryabov Aleksandr Alekseevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy chief designer – head of research department, Doctor of Science in Physics and Mathematics, e-mail: AARyabov@vniief.ru

Samodolov Aleksey Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research group
Santalov Aleksey Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ASSantalov@vniief.ru
Sapronov Ivan Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: ISSapronov@vniief.ru
Sapronova Olga Vyacheslavovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: OVSapronova@vniief.ru
Sarazov Aleksey – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AVSarazov@vniief.ru
Sayfullin Aydar Ildarovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AISayfullin@vniief.ru
Sayfullina Ramilya Nailovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: RNSayfullina@vniief.ru
Selchenkova Nadezhda Ivanovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist
Seryakova Irina Valeryevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: IVSeryakova@vniief.ru
Shanin Igor Innokentyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Engineering
Sharov Oleg Olegovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: OOSharov@vniief.ru
Shemyakin Andrey Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: AVShemyakin@vniief.ru
Shiberin Igor Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory
Shumilina Olga Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: ONShumilina@vniief.ru
Shumilov Ivan Andreevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist
Shurygin Aleksandr Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AVShurygin@vniief.ru
Shuvalova Ekaterina Vladimirovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: EVShuvalova@vniief.ru
Simakov Vitaliy Yuryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: VYuSimakov@vniief.ru
Sinitsin Evgeniy Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: EASinitsin@vniief.ru
Sobanin Danila Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: DSSobanin@vniief.ru
Sofronov Vasiliy Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, e-mail: VNSofronov@vniief.ru
Sokolov Sergey Sergeevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Doctor of Science in Physics and Mathematics, e-mail: SSSokolov@vniief.ru
Solovyev Aleksandr Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research group, e-mail: ANSolovev@vniief.ru
Stepanenko Sergey Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, chief scientist, Doctor of Science in Physics and Mathematics, e-mail: SAStepanenko@vniief.ru
Struchkov AndreyViktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: AnVStruchkov@vniief.ru
Sufiyanov Vadim Garaykhanovich – Kalashnikov Izhevsk State Technical University, Izhevsk, Russia, professor, Doctor of Science in Engineering, e-mail: vsufiy@mail.ru

Surina Olga Yuryevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: OYuSurina@vniief.ru
Svidinskiy Artem Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AVSvidinskiy@vniief.ru
Timofeev Aleksey Evgenyevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region Russia, head of research laboratory
Tishkin Dmitriy Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: DATishkin@vniief.ru
Titov Mikhail Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: MATitov@vniief.ru
Trunin Ivan Ryurikovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, e-mail: IRTrunin@vniief.ru
Tsalko Taras Valentinovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: TVTsalko@vniief.ru
Tsiberev Kirill Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, Candidate of Science in Physics and Mathematics, e-mail: KVtsiberev@vniief.ru
Tyatyushkina Elena Sergeevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, Candidate of Science in Physics and Mathematics, e-mail: ESTyatyushkina@vniief.ru
Tyukhtina Polina Aleksandrovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: PATyukhtina@vniief.ru
Tyundina Anna Aleksandrovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: AATyundina@vniief.ru
Uchaev Aleksandr Yakovlevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, chief scientist
Utkin Dmitriy Aleksandrovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist, e-mail: DAUtkin@vniief.ru
Veselova Elena Aleksandrovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: EAVeselova@vniief.ru
Vetchinnikov Maksim Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research laboratory, e-mail: MVVetchinnikov@vniief.ru
Vishnyakov Andrey Yuryevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: AYVishnyakov@vniief.ru
Zadorozhny Konstantin Viktorovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: KVZadorozhnyy@vniief.ru
Zanegin Igor Vladimirovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, head of research department, e-mail: IVZanegin@vniief.ru
Zelenskiy Dmitriy Konstantinovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, deputy head of reasearch department – head of research laboratory, Candidate of Science in Physics and Mathematics, e-mail: DKZelenskiy@vniief.ru
Zhilnikova Nataliya Nikolaevna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, senior scientist, e-mail: NNZhilnikova@vniief.ru
Zhirnov Vladimir Vyacheslavovich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, scientist, e-mail: VVZhirnov@vniief.ru
Zhuchkov Roman Nikolaevich – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, leading scientist, Candidate of Science in Engineering, e-mail: RNZhuchkov@vniief.ru
Zhuravleva Mariya Vadimovna – FSUE RFNC-VNIIEF, Sarov, Nizhny Novgorod Region, Russia, junior scientist
Zmeev Dmitry Nikolaevich –Institute for Design Problems in Microelectronics of RAS, Moscow, Russia, scientist, e-mail: zmejevdn@list.ru

Научное издание

СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

SUPERCOMPUTER SIMULATIONS AND ARTIFICIAL INTELLIGENCE

Международная конференция
XXII Харитоновские тематические научные чтения
24–27 мая 2021 года

International conference
XXII Khariton scientific talks
24–27 May 2021

СБОРНИК НАУЧНЫХ ТРУДОВ

PROCEEDINGS

Под редакцией
доктора физ.-мат. наук Р. М. Шагалиева

Печатается с оригинальных текстов авторов

Компьютерная подготовка оригинала-макета
Н. В. Мишкина, Ю. В. Пичурова

Подписано в печать 15.03.2022 Формат 60×84/8
Печать офсетная Уч.-изд. л. ~55,7 Усл. печ. л. ~67,2
Тираж 180 экз. Заказ 242-2022

Отпечатано в ИПЦ ФГУП «РФЯЦ-ВНИИЭФ»
607188, г. Саров Нижегородской обл., ул. Силкина, 23