

ISSN 2414-0171

ВОПРОСЫ
АТОМНОЙ
НАУКИ И
ТЕХНИКИ

Российский
федеральный
ядерный
центр –
ВНИИЭФ

СЕРИЯ

**Математическое моделирование
физических процессов**

2022

ВЫПУСК 3

Главный редактор
Шагалиев Р. М.

Заместители главного редактора:
Алексеев А. В., Тишкин В. Ф.

Ответственный секретарь:
Соколовская Е. В.

Члены редколлегии:

Бартенев Ю. Г., Бетелин В. Б., Бочков А. И., Вронский М. А.,
Дрёмов В. В., Залялов Н. Н., Кибзун А. И., Козелков А. С.,
Козманов М. Ю., Куркин А. А., Мартынов А. П., Петров И. Б.,
Прилуцкий М. Х., Смирнов Н. Н., Соколов С. С., Старостин Н. В.,
Степаненко С. А., Храмченков М. Г., Четверушкин Б. Н.,
Шестаков А. А., Янилкин Ю. В.

Адрес редакции и издателя: 607188,
г. Саров Нижегородской обл., пр. Мира, 37
тел.: (83130)28406, *e-mail*: sokol@vniief.ru.
Адрес сайта журнала: <http://vant.vniief.ru/>

ВОПРОСЫ АТОМНОЙ НАУКИ И ТЕХНИКИ

СЕРИЯ

Математическое моделирование физических процессов

НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

ВЫПУСК 3

Издается с 1978 г.

Саров — 2022

СОДЕРЖАНИЕ

<i>Янцкин Ю. В., Стаценко В. П., Чернышёв Ю. Д., Беляев С. П., Колобянин В. Ю., Кондрашенко А. В., Топорова О. О., Турутмина И. Ю.</i> Численное моделирование трехмерного течения газа в газовой центрифуге. Часть 1. Описание методики	3
<i>Янцкин Ю. В., Стаценко В. П., Чернышёв Ю. Д., Беляев С. П., Колобянин В. Ю., Кондрашенко А. В., Топорова О. О., Турутмина И. Ю.</i> Численное моделирование трехмерного течения газа в газовой центрифуге. Часть 2. Результаты расчетов модельной центрифуги "Игуассу"	20
<i>Хмельницкая А. Д., Карлыханов Н. Г.</i> Квазиспектральный метод решения кинетического уравнения переноса излучения в двумерной осесимметричной постановке	37
<i>Нуждин А. А.</i> Иерархический алгоритм распараллеливания вычислений при решении трехмерного уравнения переноса нейтронов на гибридных супер-ЭВМ	48
<i>Ерофеев А. М., Ветчинников М. В.</i> Оптимизация алгоритмов в прикладном методическом тесте MD для эффективного использования GPU	63
<i>Борисенко О. Н., Кузьменко М. В., Черенкова М. В., Гиниятуллина А. Г., Чухманов Н. В., Смолкина Д. Н., Тимаева Т. Е., Блажнова К. А.</i> Улучшение качества граней объемных ячеек при генерации неструктурированных сеток в пакете программ "Логос"	73
<i>Титов М. А.</i> Реализация в программном комплексе "Виртуальный 3D-принтер" возможности применения векторных шрифтов для аддитивного производства	86
Сведения об авторах	97

C O N T E N T S

<i>Yanilkin Yu. V., Statsenko V. P., Chernyshev Yu. D., Belyaev S. P., Kolobyanin V. Yu., Kondrashenko A. V., Toporova O. O., Turutina I. Yu.</i> Numerical simulation of 3D gas flow in gas centrifuga. Part 1. Description of the method	3
<i>Yanilkin Yu. V., Statsenko V. P., Chernyshev Yu. D., Belyaev S. P., Kolobyanin V. Yu., Kondrashenko A. V., Toporova O. O., Turutina I. Yu.</i> Numerical simulation of 3D gas flow in gas centrifuga. Part 2. Computation results of model "Iguassu" centrifuga	20
<i>Khmelnitskaya A. D., Karlykhanov N. G.</i> Quasi-spectral method to solve the kinetic equation of radiation transfer in 2D axially symmetric setup	37
<i>Nuzhdin A. A.</i> Hierarchical parallelization algorithm when computing 3D neutron transfer equation on hybrid super-computers	48
<i>Erofeev A. M., Vetchinnikov M. V.</i> Optimization of algorithms in applied technical MD test for efficient GPU implementation	63
<i>Borisenko O. N., Kuzmenko M. V., Cherenkova M. V., Giniyatullina A. G., Chukhmanov N. V., Smolkina D. N., Timaeva T. E., Blazhnova K. A.</i> Quality improvement of the facets of 3D cells when generating unstructured meshes in "Logos" software package	73
<i>Titov M. A.</i> Possible implementation of vector fonts for additive production realized in "Virtual 3D-printer" software complex	86
Information about authors	98

Ответственный за выпуск Е. В. Соколовская

Редактор Е. Н. Старченко

Корректоры Т. А. Меркушева
Е. А. Окатьева
А. В. Федоренко

Дата выхода в свет 25.09.2022

Усл. печ. л. ~ 12

Тираж 1000 экз.

Зак. тип. 1341-2022

Формат 60×84/8

Уч.-изд. л. ~ 15

7 статей

Учредитель: ФГУП "РФЯЦ-ВНИИЭФ"

Свидетельство о регистрации ПИ № ФС77-29789 от 04 октября 2007 г.
выдано Роскомнадзором

Оригинал-макет подготовлен
в Математическом отделении ФГУП "РФЯЦ-ВНИИЭФ"
Отпечатано в ИПЦ ФГУП "РФЯЦ-ВНИИЭФ"
607188, г. Саров Нижегородской обл., ул. Силкина, 23

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНОГО ТЕЧЕНИЯ ГАЗА В ГАЗОВОЙ ЦЕНТРИФУГЕ. ЧАСТЬ 1. ОПИСАНИЕ МЕТОДИКИ

Ю. В. Янилкин, В. П. Стаценко, Ю. Д. Чернышёв, С. П. Беляев,
В. Ю. Колобянин, А. В. Кондрашенко, О. О. Топорова, И. Ю. Турутина
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Статья представляет собой первую часть работы, посвященной трехмерному численному моделированию разреженного газа в реальной газовой центрифуге с учетом физической вязкости и теплопроводности. Приведено описание численной методики, реализация которой проведена на основе методики ЭГАК на цилиндрической сетке в специальной системе координат на параллельной ЭВМ. Часть процессов аппроксимируется в неподвижной декартовой системе координат, а часть — в инерциальной цилиндрической системе. Методика предназначена для сквозного моделирования всего процесса работы центрифуги без применения приближенных методов "сшивки" результатов расчетов в разных областях центрифуги. С этой целью используется приближение, основанное на том, что уравнения газовой динамики, теплопроводности и диффузии применимы всюду. Используется подход, основанный на решении трехмерных уравнений Навье—Стокса во всей центрифуге, в том числе в области двумерного течения в разделительной камере. Для улучшения эффективности программы используется декомпозиция по процессам, в которой на область двумерного течения приходится всего один процесс. Это позволяет экономить вычислительные ресурсы на порядки. Результаты тестирования методики приводятся во второй части работы.

Ключевые слова: газовая центрифуга, разделительная камера, отборная и отвальная камеры, методика ЭГАК, вязкая жидкость, теплообмен, численная методика.

Введение

Газовая центрифуга (ГЦ) представляет собой цилиндрический ротор, вращающийся вокруг оси цилиндра (рис. 1). Она состоит из нескольких элементов: разделительной камеры (РК), в которой течение является двумерным (2D) с осевой симметрией, а также отборной и отвальной камер. Поскольку в этой части работы разница между отборной и отвальной камерами не принципиальна, для обозначения их обеих будем использовать аббревиатуру ОК. ОК содержат диафрагмы и газоотборники фракций урана, нарушающие осевую симметрию, поэтому течение здесь является трехмерным (3D).

Численное моделирование течений в ГЦ представляет собой достаточно сложную задачу в силу нескольких причин. Во-первых, в этих задачах из-за вращения с огромной скоростью плотность меняется на много порядков до состояния разреженного газа. Во-вторых, необходимо моделировать несколько одновременно протекающих процессов, таких как газовая динамика (для краткости *газодинамика*), теплопроводность, диффузия компонентов газа, турбулентное перемешивание, поток питания. В-третьих, в задаче, как уже было сказано, имеются четко выделенные области 2D течений (в РК) и 3D течений (в ОК).

Как отмечается в [1], при общепринятом подходе пространство ротора разбивается на три характерные области (см. рис. 1): 1) область $r_1 \lesssim r \leq R$, где применима газодинамика; 2) область сильно разреженного газа вблизи оси вращения (*вакуумное ядро*); 3) переходная область между ними.

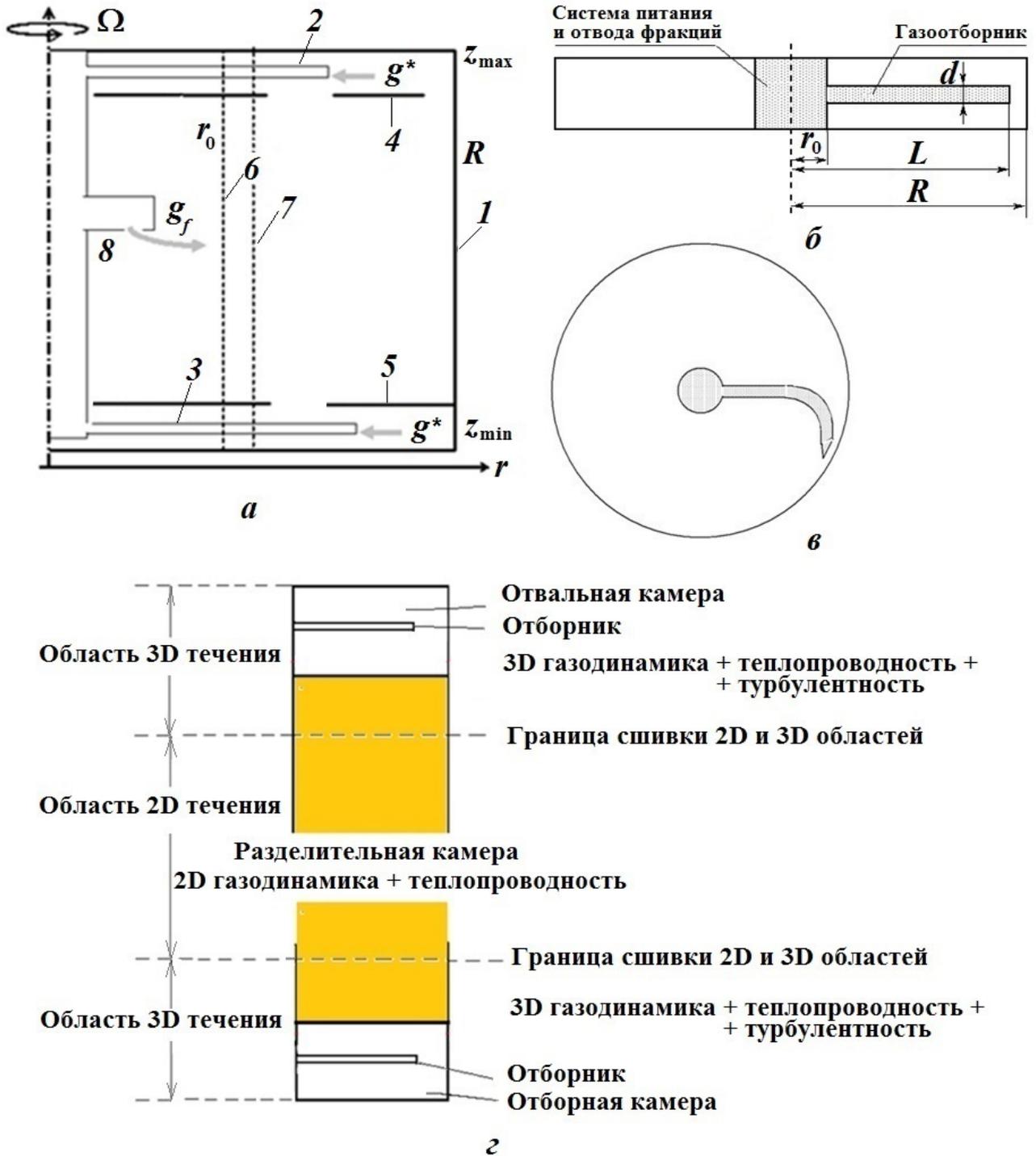


Рис. 1. Геометрия центрифуги: *а* — общая схема ГЦ; 1 — боковая стенка ротора; 2 — отборник тяжелой фракции; 3 — отборник легкой фракции; 4 — отвальная диафрагма; 5 — отборная диафрагма; *б* — граница области разреженного газа; 7 — граница области решения методом МК; 8 — источник питания; *б* — вертикальная проекция отборника; *в* — горизонтальная, симметричная по высоте, проекция отборника в отборной/отвальной камере; *г* — схема разбиения задачи на подобласти

В первом приближении процессами переноса в разреженных областях 2 и 3 пренебрегают. При этом распределение потока питания на внутренней границе плотной области остается неизвестным и

заменяется каким-либо модельным распределением. Несмотря на простоту, такая модель учитывает важные для центробежного разделения эффекты.

Последующие приближения связаны с учетом процессов переноса в разреженных областях 2 и 3. Учет разреженных областей может дать лишь малые поправки к первому приближению ($\sim 1\%$), хотя он может быть важен для обоснования условий на границе *газ—вакуум* и уточнения параметров текущих через разреженную зону потоков, в том числе потока питания.

В настоящее время известно несколько 2D методов моделирования течений в РК [1, 2]. Что касается 3D течений в ОК, то здесь для проведения расчетов используют приближенные граничные условия на границе с РК [3]. 3D расчет [4] был выполнен лишь для части ОК и части отборника с использованием приближенных граничных условий на границе счетной области.

Авторы настоящей работы поставили перед собой задачу — создать методику сквозного моделирования всего процесса работы центрифуги. С этой целью используется указанное выше первое приближение с формальным предположением, что уравнения газодинамики, теплопроводности и диффузии применимы всюду.

В качестве базовой была выбрана методика ЭГАК [5], в которой имеются достаточно широкие возможности проведения 2D и 3D расчетов газодинамических течений с сопутствующими процессами. Однако их недостаточно для проведения расчетов рассматриваемой задачи, поэтому дополнительно был модифицирован, разработан и реализован ряд программ. Основные из них имеют следующее назначение:

- переход в специальную декартово-цилиндрическую систему координат, когда часть программ работает с декартовой системой координат, а другая часть — с цилиндрической;*
- расчет движения вязкого газа;
- сглаживание скоростей;
- расчет диффузии и теплопроводности с учетом граничных условий, а также наличия слоя между двумя камерами и отборником;
- связь 2D и 3D областей;
- распараллеливание по процессам с учетом циклических граничных условий.

В первой части работы описывается сама методика решения указанной выше задачи. Результаты ее тестирования будут приведены во второй части.

1. Постановка задачи и основные уравнения

Геометрия задачи. На схеме ГЦ (см. рис. 1, *a*) счетная газодинамическая область расположена между боковой стенкой ротора 1 и границей области разреженного газа *b*. В ней происходят следующие процессы: циркуляционное течение в рабочей камере, состоящей из РК и ОК; обтекание отборников и отбор ими газа; течения через отверстия диафрагм, жестко соединенных с ротором; ввод потока питания через границу *b* с разреженным газом. Существует мнение, что в рамках газодинамики корректно смоделировать поток питания не удастся. Поэтому в некоторых работах (например, [6]) его распределение получают из решения по методу Монте-Карло (МК) для точечного потока питания g_f , и тогда граничное условие задается на границе γ внутри газодинамической области. Платой за это является необходимость согласования решений МК в разреженной области и газодинамики в зоне Кнудсена.

В настоящей работе всюду используется газодинамическое приближение. Предлагаемый подход притягателен тем, что в нем нет проблем, связанных с разработкой соответствующего метода МК и его "сшивкой" с газодинамикой.

Граничные условия. Граничные условия для системы уравнений Навье—Стокса следующие: на границах ГЦ задается газодинамическая скорость $\mathbf{u} = (U_r, U_\varphi, u_z)$. При этом радиальная и

*Это связано с тем, что к началу работы некоторые программы уже были ориентированы на декартовы координаты и верифицированы на широком классе задач, поэтому создание новых было признано нецелесообразным.

осевая скорости U_r и u_z равны нулю на всей границе ротора ГЦ — на поверхностях $z = z_{\min}$, $z = z_{\max}$ (верхний и нижний торцы ГЦ) и на границе $r = R$ (внешняя цилиндрическая граница ГЦ). Окружная скорость U_φ на всех границах ротора (при $r = R$, на поверхностях $z = z_{\min}$ и $z = z_{\max}$) определяется угловой скоростью Ω вращения ГЦ: $U_\varphi = \Omega r$. Кроме того, для U_φ имеются периодические по φ граничные условия: $U_\varphi(r, 0, z) = U_\varphi(r, 2\pi, z)$. На поверхности газоотборников, внутренней цилиндрической поверхности $r = r_0$ и расположенном на ней источнике питания все компоненты скорости U_r , U_φ , u_z равны нулю. На входе отборников задается условие свободного протекания. На источнике питания задается поток питания.

Для задач с теплопроводностью заданы распределения температуры на всех границах ГЦ: при $r = r_0$ и $r = R$, на поверхностях $z = z_{\min}$ и $z = z_{\max}$.

Основные уравнения. Рассматриваемые задачи относятся к случаю, когда компоненты газовой смеси имеют относительно близкие молярные массы; далее предполагается, что имеются лишь $^{235}\text{UF}_6$ и $^{238}\text{UF}_6$. Следуя [1], полагаем, что в этом случае распределение концентраций не влияет на движение газа как целого. Это, так называемое *изотопное*, приближение позволяет независимо находить решение задачи о газодинамическом течении в ГЦ и на его основе решать диффузионную задачу.

Система уравнений, которая описывает движение газовой смеси из нескольких компонентов, состоит из уравнений непрерывности для каждого компонента, уравнений Навье—Стокса и сохранения энергии для газа в целом. Приведем эти уравнения в декартовой системе координат в векторных обозначениях [1].

Уравнение неразрывности для j -го компонента смеси имеет следующий вид:

$$\frac{\partial \rho_j}{\partial t} + \nabla \cdot (\rho_j \mathbf{u}) = -\nabla \cdot (\rho_j \mathbf{w}_j). \quad (1)$$

Здесь \mathbf{w}_j — диффузионная скорость j -го компонента; $\rho_j \mathbf{w}_j$ — диффузионный поток. В силу сохранения импульса $\sum \rho_j \mathbf{w}_j = 0$. Суммирование уравнений (1) по j дает уравнение неразрывности, которое используется при решении задачи о течении в ГЦ:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0.$$

Уравнение (1) для $j = 1$ (легкий компонент) используется при решении диффузионной задачи (см. далее).

Уравнение движения (Навье—Стокса) имеет вид

$$\rho \left[\frac{\partial}{\partial t} + (\mathbf{u} \cdot \nabla) \right] \mathbf{u} = -(\nabla \cdot \sigma). \quad (2)$$

Приведем также формулы для двух компонент ускорения в цилиндрических координатах (в инерциальной системе отсчета), они понадобятся далее:

$$\begin{aligned} \dot{U}_r &= \frac{\partial U_r}{\partial t} + U_r \frac{\partial U_r}{\partial r} + \frac{U_\varphi}{r} \frac{\partial U_r}{\partial \varphi} + U_z \frac{\partial U_r}{\partial z} - \frac{U_\varphi^2}{r}; \\ \dot{U}_\varphi &= \frac{\partial U_\varphi}{\partial t} + U_r \frac{\partial U_\varphi}{\partial r} + \frac{U_\varphi}{r} \frac{\partial U_\varphi}{\partial \varphi} + U_z \frac{\partial U_\varphi}{\partial z} + \frac{U_r U_\varphi}{r}. \end{aligned} \quad (3)$$

Уравнение сохранения энергии:

$$\rho \left[\frac{\partial}{\partial t} + (\mathbf{u} \cdot \nabla) \right] e = -(\nabla \cdot \mathbf{q}) - (\sigma \cdot \nabla) \mathbf{u}. \quad (4)$$

Здесь e — удельная внутренняя энергия; \mathbf{q} — вектор теплового потока, $\mathbf{q} = -\lambda \nabla T$, где λ — коэффициент теплопроводности.

В уравнениях (2) и (4) $\sigma = \sigma(P_{kl})$ — тензор напряжений, выражение для компонент которого имеет вид

$$P_{kl} = p\delta_{kl} - \eta \left[\frac{\partial U_k}{\partial x_l} + \frac{\partial U_l}{\partial x_k} - \frac{2}{3}\delta_{kl}(\nabla \cdot \mathbf{u}) \right] - \zeta\delta_{kl}(\nabla \cdot \mathbf{u}),$$

где полагаем $\zeta = 0$.

Записывая уравнения движения и сохранения энергии в компонентах, получаем

$$\begin{aligned} \rho \left[\frac{\partial}{\partial t} + \left(U_k \frac{\partial}{\partial x_k} \right) \right] U_l &= -\frac{\partial p}{\partial x_l} + \frac{\partial}{\partial x_k} \left\{ \eta \left[\frac{\partial U_k}{\partial x_l} + \frac{\partial U_l}{\partial x_k} - \frac{2}{3}\delta_{kl}(\nabla \cdot \mathbf{u}) \right] \right\}; \\ \rho \left[\frac{\partial}{\partial t} + (\mathbf{u} \cdot \nabla) \right] l &= -\alpha(\nabla \cdot \mathbf{q}) - p(\nabla \cdot \mathbf{u}) + \eta \left[\frac{1}{2} \left(\frac{\partial U_k}{\partial x_l} + \frac{\partial U_l}{\partial x_k} \right)^2 - \frac{2}{3}(\nabla \cdot \mathbf{u})^2 \right], \end{aligned}$$

где α — коэффициент распределения теплового потока на вязкую и тепловую части.

Решение диффузионной задачи в ГЦ. Скорость j -го компонента газа равна сумме массовой скорости \mathbf{u} газа как целого и диффузионной скорости \mathbf{w}_j , которая для бинарной смеси удовлетворяет соотношению

$$\rho_1 \mathbf{w}_1 = -\rho_2 \mathbf{w}_2 = -\rho \frac{M_1 M_2}{(\rho/n)^2} D (\mathbf{d}_1 + k_T \nabla \ln T), \quad \mathbf{d}_j = \nabla \left(\frac{n_j}{n} \right) + \left(\frac{n_j}{n} - \frac{\rho_j}{\rho} \right) \nabla \ln p,$$

где M_1, M_2 — молярные массы компонентов; D — коэффициент концентрационной диффузии; n, n_j — молекулярная концентрация среды и j -го компонента соответственно.

Итак, уравнение (1) можно записать как уравнение для концентрации (молярной доли) легкого компонента $N = n_1/n$

$$\rho \left(\frac{\partial N}{\partial t} + \mathbf{u} \nabla N \right) = -\nabla (\rho_1 \mathbf{w}_1).$$

Отметим, что $\frac{\rho_1}{\rho} = \frac{M_1 N}{M_1 N + M_2 (1 - N)}$.

2. Сшивка областей 2D и 3D течений и счетная сетка

Покажем, как осуществляется сшивка областей РК центрифуги с 2D характером течения и ОК — с 3D течением (далее 2D и 3D областей). В начале работы предполагалось, что достаточно создать программы расчета указанных выше процессов — и можно будет решать задачи в сквозной постановке на 3D сетке, включающей в себя все элементы РК и ОК центрифуги. Однако оказалось, что подобный подход требует настолько больших ресурсов ЭВМ, что проведение сквозного моделирования становится затруднительным. Поэтому предлагается подход, который, хотя и содержит в себе сквозное моделирование, однако требует в десятки раз меньше ресурсов ЭВМ (экономия на тестовой задаче 5 из второй части работы составляет ~ 16 раз по количеству ячеек и, соответственно, по времени).

Как уже отмечалось, рассматриваемая счетная область моделируемой системы (см. рис. 1, 2) состоит из двух подобластей, течения в которых принципиально отличаются. Ярко выраженный 3D характер течения в области ОК с небольшой прилегающей к ней частью РК и возможность считать течение как 2D почти во всей РК (см. [1, 3, 7]) позволяют в первой области использовать 3D программы, а во второй — 2D либо даже одномерное (1D) аналитическое решение для вращающегося газа. Влияние одной области на другую можно учесть путем передачи соответствующих решений между областями. При подобном подходе в указанных работах [1, 3, 7] границей стыковки является граница между ОК и РК (диафрагма), расположенная в области 3D течения. Такой выбор границы сшивки требует применения упрощающих предположений в обеих областях, погрешность которых трудно проконтролировать.

В данной работе используется другой подход. Во-первых, граница между 3D и 2D областями задается в РК на некотором удалении от границы отборника, так, чтобы на этой границе течение было 2D. Таким образом, шивка решений будет осуществляться в области 2D течения. В этом случае никаких упрощений задачи не требуется ни в одной из областей. Во-вторых, в обеих областях используются 3D программы счета, что делает счет однородным, т. е. по одним и тем же программам, и, соответственно, с одной и той же точностью. Как это осуществляется, проиллюстрировано на рис. 2, а. На этом рисунке область LMN включает в себя ОК с прилегающей к ней частью РК и представляет собой область, в которой решаются уравнения Навье—Стокса, температуропроводности, турбулентного перемешивания с учетом наличия отборника и отверстий на границе РК и ОК. Область $ABCDEF$ представляет собой вырезанную из оставшейся части РК пространственную область, в которой решаются уравнения Навье—Стокса, температуропроводности, диффузии изотопов урана с включением потока питания. Связь между этими двумя областями при численном моделировании осуществляется обычным образом, как связь между соседними процессами, реализованная в методике ЭГАК [5], т. е. с передачей газодинамических величин из двух соседних приграничных ячеек в обе стороны. При этом вторая (2D) область рассчитывается одним MPI-процессом, с которым соседствует один из процессов, рассчитывающих первую (3D) область, и данные из 2D области разносятся по всем процессам 3D области. Такой подход позволяет осуществлять полностью сквозной расчет течения, но при этом в десятки раз экономить вычислительные ресурсы.

В области 3D течения расчет производится следующим образом. Для центрифуги осью вращения является ось OZ . На рис. 2, б показан поперечный разрез цилиндра с цилиндрической сеткой для четырехпроцессного случая (номера процессов обозначены цифрами); $\Gamma 2$, $\Gamma 3$ — внешние границы счетной области. Циклическое замыкание геометрии осуществляется в плоскостях OXY по совпадающим границам $\Gamma 4$ и $\Gamma 5$ процессов 3 и 4. Для этого на границе $\Gamma 4$ процесса 3 для приграничных ячеек соседями объявляются приграничные ячейки процесса 4 со стороны $\Gamma 5$ и наоборот. При этом передача данных между процессами 3 и 4 выполняется обычным образом, например, как для процессов 1 и 2.

Отметим, что в данном случае с одним из указанных процессов в 3D области соседствует процесс с согласованной по углу сеткой из 2D области. Передача данных между ними осуществляется так же, как между процессами 1 и 2.

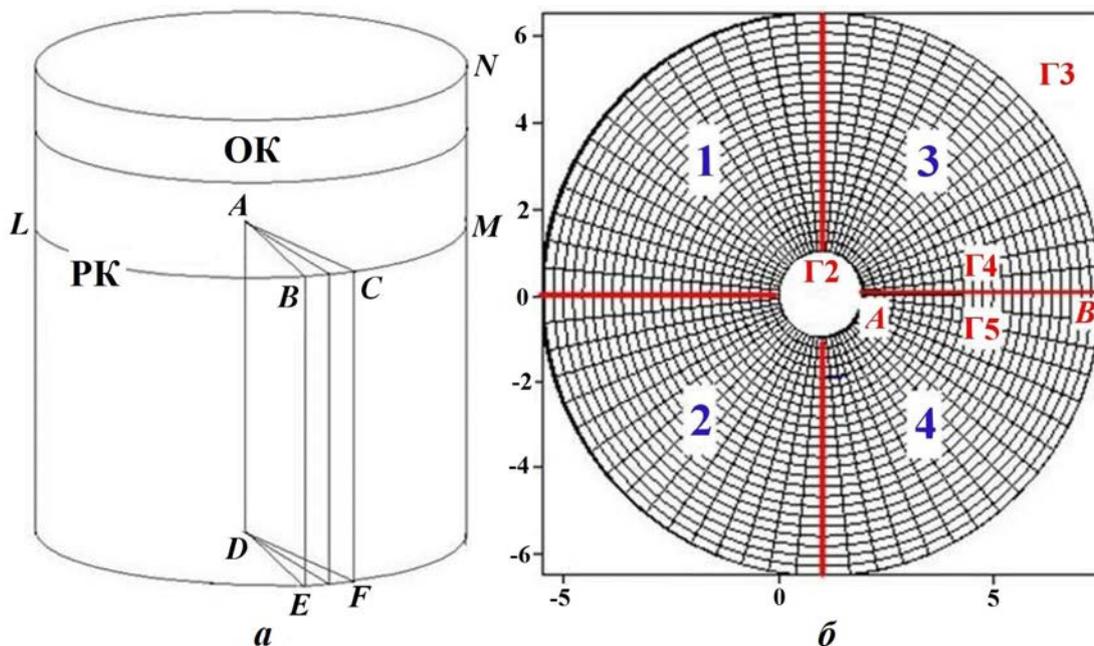


Рис. 2. Счетная сетка и схема разбиения задачи: а — 3D и 2D области; б — цилиндрическая счетная сетка и циклические граничные условия

3. Аппроксимация уравнений

В расчетах используется неподвижная цилиндрическая сетка (см. рис. 2, б), однако аппроксимация производится в разных программах по-разному: либо в декартовых, либо в цилиндрических координатах. Расчет полной задачи осуществляется с использованием метода расщепления в несколько этапов (рис. 3). Значения величин после каждого этапа являются исходными данными для следующего. Отметим, что процессы газодинамики и теплопроводности имеют различные временные масштабы, поэтому этап 5 выполняется после нескольких шагов этапов 1–4 (см. разд. 4).

Аппроксимация уравнений газовой динамики. Аппроксимация уравнений газовой динамики производится в декартовых координатах.

Рассмотрим для простоты течение идеального газа без учета вязкости. В этом случае первый этап вычислений (см. рис. 3) отсутствует. В случае наличия вязкости описанные ниже схемы легко обобщаются.

Дифференциальные уравнения многокомпонентной газодинамики в эйлеровых координатах имеют следующий вид:

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla(\rho \mathbf{u} \mathbf{u}) = -\nabla p; \quad (5)$$

$$\frac{\partial}{\partial t} \rho_\xi + \nabla(\rho_\xi \mathbf{u}_\xi) = 0; \quad (6)$$

$$\frac{\partial}{\partial t}(\beta_\xi \rho_\xi e_\xi) + \nabla(\beta_\xi \rho_\xi e_\xi \mathbf{u}_\xi) = -\beta_\xi p_\xi \nabla \mathbf{u}_\xi. \quad (7)$$

Здесь массовая скорость $\mathbf{u}(u_x, u_y, u_z)$ определена в узлах счетной сетки, скалярные величины ($\rho_\xi, \rho, e_\xi, p_\xi, p, \beta_\xi = V_\xi/V$) — в центрах ячеек; ξ — номер компонента, причем индекс ξ у скорости относится не к ней самой, а к ее дивергенции. В рассматриваемой области может быть несколько веществ с различными уравнениями состояний. Система (5)–(7) замыкается уравнениями состояний компонентов среды $p_\xi = p_\xi(\rho_\xi, e_\xi)$.

Аппроксимация системы уравнений (5)–(7) осуществляется с применением метода расщепления в два этапа. На первом (лагранжевом) этапе решаются уравнения без конвективных членов, т. е. уравнения газодинамики в лагранжевых переменных. Кроме того, на первом этапе после аппроксимации системы (5)–(7) производится учет кориолисова ускорения. На втором (эйлеровом) этапе сетка возвращается в начальное состояние и осуществляется пересчет величин на эту сетку, т. е.

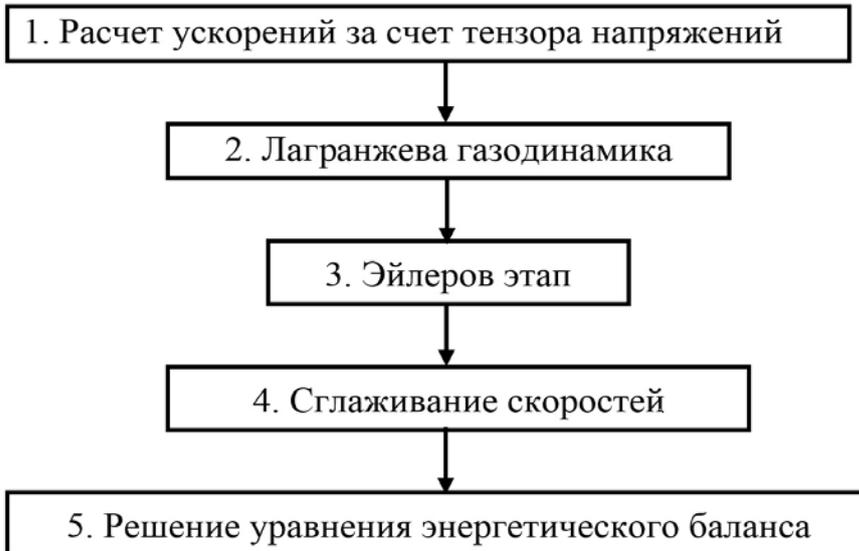


Рис. 3. Последовательность вычислений по этапам

аппроксимация отброшенных на первом этапе членов уравнений. При этом в качестве начальных данных используются значения величин, полученные на первом этапе вычислений. Данный этап, в свою очередь, разбивается на три подэтапа, а именно, аппроксимация уравнений адвекции выполняется с использованием расщепления по направлениям. Кроме того, после второго этапа возможно применение процедуры сглаживания скоростей.

Лагранжев этап. Исходная система уравнений, решаемая на данном этапе:

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= -\frac{\nabla p}{\rho}; \\ \frac{d\rho_\xi}{dt} &= -\rho_\xi (\nabla \cdot \mathbf{u})_\xi; \\ \frac{de_\xi}{dt} &= -\frac{p_\xi}{\rho_\xi} (\nabla \cdot \mathbf{u})_\xi; \\ \frac{d\mathbf{r}}{dt} &= \mathbf{u}; \\ p_\xi &= p_\xi(\rho_\xi, e_\xi). \end{aligned} \quad (8)$$

Здесь необходимо иметь в виду, что при программной реализации в этой системе вместо давления используется сумма давления и искусственной (счетной) вязкости.

Последовательность шагов аппроксимации этих уравнений схематично показана на рис. 4. Верхний индекс t_2 в обозначениях величин здесь и далее указывает на их значения после лагранжева этапа.

Аппроксимация уравнения движения в общем случае. На первом шаге при аппроксимации уравнения движения (первого уравнения системы (8)) сначала ищутся ускорения в центрах граней ячеек, направленные по нормали к грани. Формулы для ускорения на грани имеют вид [5]

$$g_x = \frac{G_1 - G_2}{m_1 + m_2} S_{yz}; \quad g_y = \frac{G_1 - G_2}{m_1 + m_2} S_{xz}; \quad g_z = \frac{G_1 - G_2}{m_1 + m_2} S_{xy}, \quad (9)$$

где S_{yz} , S_{zx} , S_{xy} — проекции площади грани на координатные плоскости; G — сумма давления и искусственной вязкости; m — четверти масс ячеек; индексы величин означают номера ячеек по обе стороны от рассматриваемой грани.

Ускорение в узле находится интерполяцией ускорений на гранях, прилегающих к узлу. Отметим, что при таком способе аппроксимации для используемой цилиндрической сетки фактически получаются ускорения в радиальном и угловом направлениях. Интерполяционные формулы получения ускорения в узле приводятся в [5].



Рис. 4. Последовательность вычислений на лагранжевом этапе газодинамики

Аппроксимация уравнения движения для рассматриваемого течения. Для рассматриваемой задачи и соответствующей цилиндрической сетки вычисление ускорения на гранях, расположенных перпендикулярно к оси цилиндра и в радиальном направлении, выполняется вполне корректно по стандартным формулам и дает приемлемый по точности результат. Однако при вычислении ускорения на гранях, перпендикулярных к радиальному направлению, эти формулы дают большую погрешность, связанную прежде всего с экспоненциальным распределением плотности и давления в радиальном направлении. Для уменьшения этой погрешности обычно уравнения Навье—Стокса переписываются так, чтобы в них выделить твердотельное вращение и использовать в качестве основных переменных логарифмы плотности и давления (см., например, [1]). В настоящей работе предлагается оставить в качестве основных переменных сами плотность и давление и модифицировать приведенную ниже разностную схему.

Рассмотрим сначала твердотельное равномерное вращение ротора (рис. 5). В этом случае ускорение в точке 0 полностью определяется указанными на рисунке ускорениями в радиальном направлении; ускорения в угловом направлении и по оси OZ равны нулю. Радиальное ускорение в узле будет равно полученному ускорению g на гранях ячейки.

При твердотельном вращении центростремительное ускорение в точке 0 на рис. 5 имеет вид (см. первое уравнение из (3))

$$g_{ц} = -\frac{U_{\varphi}^2}{r}.$$

Это ускорение должно определяться градиентом давления в радиальном направлении. Для корректного счета для ускорения g (см. рис. 5), полученного из разностного решения, должно иметь место равенство

$$g = g_{ц}. \quad (10)$$

Однако ускорение, рассчитанное по формулам (9), в общем случае не удовлетворяет условию (10). Таким образом, величина

$$\Delta = g - g_{ц}$$

представляет собой абсолютную погрешность метода определения соответствующего ускорения по формулам (9). Если в процессе счета к полученному по формулам (9) ускорению g добавлять величину Δ , то рассматриваемое твердотельное вращение останется без изменений.

Введем в рассмотрение отношение ускорений

$$\lambda = \frac{g_{ц}}{g}.$$

Если теперь в процессе счета ускорение, полученное по формулам (9), умножить на λ , то такое ускорение и будет точным решением для твердотельного вращения.

В общем случае (не твердотельного вращения) необходимо найти твердотельную компоненту скорости. Для этого выполняется разложение скорости на тангенциальную и радиальную компоненты (третье направление остается без изменения), т. е. $\mathbf{u}(u_x, u_y, u_z) \rightarrow \mathbf{u}(U_r, U_{\varphi}, u_z)$.

Предлагаемая модификация разностной схемы состоит в том, что далее для радиальной компоненты ускорения будет предполагаться неизменность значения λ на каждом шаге по времени и в каждом слое ячеек по радиусу. Это позволяет по ускорению, определенному по формулам (9), найти скорректированное его значение по формуле

$$\tilde{g} = \lambda g.$$

Коэффициент λ может быть определен на начальный момент времени по заданным полям скоростей, плотности и давления, которые полностью соответствуют параметрам течения для твердотельного

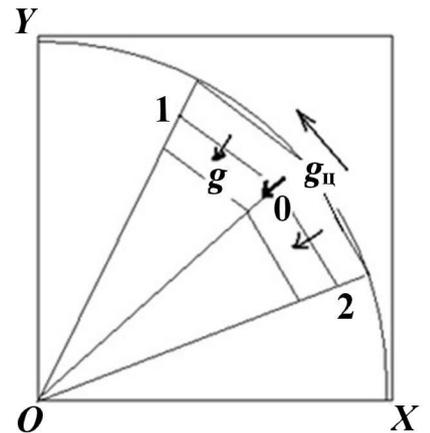


Рис. 5. Фрагмент счетной сетки для определения ускорения

вращения. В этом случае их необходимо запоминать в каждом слое ячеек по радиусу. Можно определять λ и в каждый момент времени по начальным данным задачи. С точки зрения экономии вычислительных ресурсов предпочтительней первый подход.

При описанном выше стандартном методе определения ускорения в узлах счетной сетки не учитывается член $\frac{U_r U_\varphi}{r^{t_2}}$ (см. второе уравнение из (3)). Он учитывается в конце лагранжева этапа при вычислении скорости следующим образом:

$$U_\varphi^{t_2} = \tilde{U}_\varphi^{t_2} - \tau \frac{U_r^{t_2} U_\varphi^n}{r^{t_2}}, \quad (11)$$

где $\tilde{U}_\varphi^{t_2}$ — компонента скорости, полученная по стандартным формулам; индекс n обозначает предыдущий временной шаг.

Легко показать, что уравнение (11) гарантирует сохранение момента импульса $mU_\varphi r$. Умножим обе части уравнения (11) на r^{t_2} (новый радиус узла), используя представление

$$r^{t_2} = r^n + \tau U_r^{t_2} \equiv r^n + \Delta r.$$

Получим

$$r^{t_2} U_\varphi^{t_2} = r^{t_2} \left(U_\varphi^n - \frac{U_r^{t_2} U_\varphi^n}{r^{t_2}} \right) = (r^n + \Delta r) U_\varphi^n - r^{t_2} \tau \frac{U_r^{t_2} U_\varphi^n}{r^{t_2}} = r^n U_\varphi^n + \Delta r U_\varphi^n - \Delta r U_\varphi^n = r^n U_\varphi^n.$$

После определения скоростей в радиальном и тангенциальном направлениях компоненты скорости в декартовой системе координат вычисляются по обычным формулам. Все остальные формулы лагранжева этапа, касающиеся вычислений плотности, объемных концентраций и энергий компонентов, остаются без изменений [5] и ниже не описываются.

Эйлеров этап. На эйлеровом этапе производится построение счетной сетки, которая совпадает с сеткой на начало временного шага, и осуществляется пересчет на нее газодинамических величин, т. е. производится аппроксимация конвективных членов уравнений

$$\begin{aligned} \frac{\partial \rho_\xi}{\partial t} + \nabla (\rho_\xi \mathbf{u}) &= 0; \\ \frac{\partial (\rho_\xi \mathbf{u})}{\partial t} + \nabla (\rho_\xi \mathbf{u} \mathbf{u}) &= 0; \\ \frac{\partial e_\xi}{\partial t} + \nabla (e_\xi \mathbf{u}) &= 0. \end{aligned}$$

При этом в качестве начальных данных используются значения величин, полученные на лагранжевом этапе вычислений.

Прежде всего необходимо определить потоки объемов, масс и энергий всех компонентов газа через грани ячейки. При этом используется важное предположение, что поток может быть только между ячейками, имеющими общую грань; потоки в соседние по диагонали ячейки полагаются равными нулю.

В предлагаемой методике на эйлеровом этапе используется схема с расщеплением по направлениям, т. е. весь этап разбит на три последовательных подэтапа. Сначала пересчитываются величины за счет потоков только через две противоположные грани ячеек, перпендикулярные к направлению I вдоль оси OX . При этом потоки через остальные четыре грани (направления J и K вдоль осей OY и OZ соответственно) полагаются равными нулю. Затем выполняется пересчет величин за счет потоков через две противоположные грани вдоль направления J . При этом в качестве входных данных для этого подэтапа используются значения величин, полученные после пересчета вдоль направления I эйлерова этапа. Аналогично происходит пересчет величин и вдоль направления K . Порядок вычислений вдоль одного направления приведен на рис. 6. Здесь верхний индекс $t_{2+N/3}$ означает соответствующий подэтап, на котором происходят вычисления вдоль направления N ($N = 1, 2, 3$);

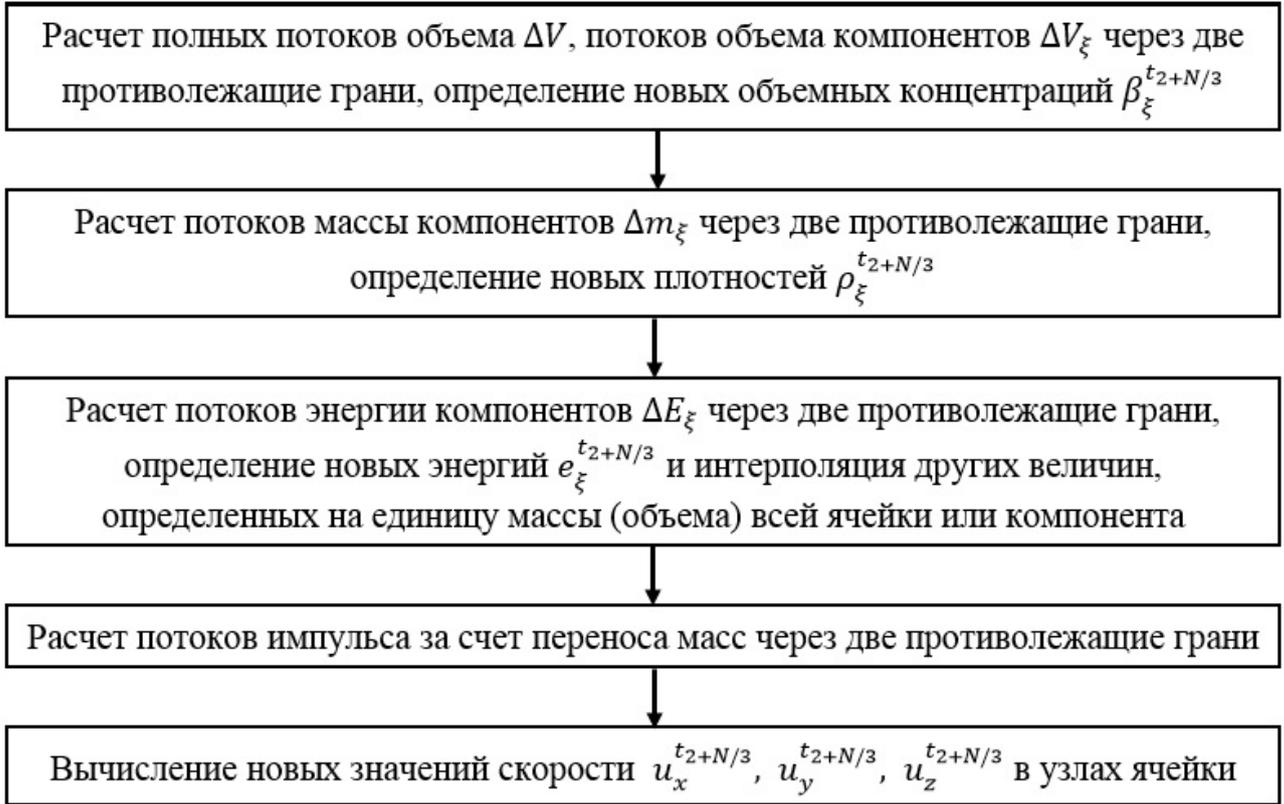


Рис. 6. Схема вычислений величин в одном направлении

t_2 означает, что значение величины берется после лагранжева этапа. Для обеспечения равноправности граней направления I , J и K на каждом шаге по времени меняются местами следующим образом: $I \rightarrow J \rightarrow K, J \rightarrow K \rightarrow I, K \rightarrow I \rightarrow J$.

Далее все уравнения будут описаны только для одного направления, для других направлений все вычисления проводятся аналогичным образом. Временной индекс не будет указываться, если его отсутствие не будет вызывать разночтений; при этом будет предполагаться, что значение данной величины берется после лагранжева этапа или после предыдущего направления.

Аппроксимация уравнений неразрывности и энергии. Рассмотрим уравнения, выражающие законы сохранения массы и энергии для случая однокомпонентной среды. В общем виде разностные уравнения, аппроксимирующие уравнения неразрывности на трех подэтапах ($N = 1, 2, 3$), имеют следующий вид:

$$\rho^{t_2+N/3} = \frac{m^{t_2+(N-1)/3} + \sum_{j=0}^1 \Delta m_j}{V^{t_2+N/3}},$$

где m — масса ячейки; Δm_j — потоки массы через две противоположащие грани ячейки. Аналогичный вид имеет и формула для энергии.

Рассмотрим расчет потоков между двумя соседними ячейками (рис. 7), где строчными буквами

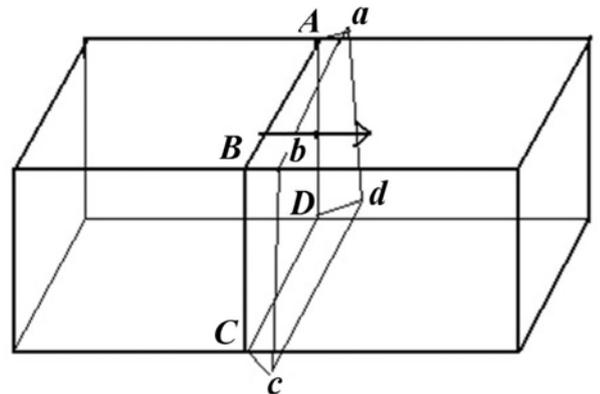


Рис. 7. Иллюстрация расчета потока объема на одном из подэтапов

обозначено положение узлов после предыдущего подэтапа, а прописными буквами — после данного подэтапа. В данной ситуации, когда скорость в целом направлена слева направо, донорной ячейкой является левая, и объем ΔV , который она отдает правой, акцепторной, ячейке, является объемом пространственной фигуры $abcdABCD$.

Формулу для определения потока массы можно записать в следующем общем виде:

$$\Delta m = \Delta V \rho^*,$$

где индекс * означает, что плотность относится к вытекающему объему ΔV .

Для рассматриваемого течения характерно наличие экспоненциального закона распределения плотности по радиусу. При использовании указанного метода вычисления потоков из ячеек в радиальном направлении имеет место заметная погрешность, связанная с приближенным определением значения плотности в потоке объема. Для рассматриваемого течения эта погрешность может быть минимизирована использованием экспоненциального распределения плотности в вытекающем из ячейки потоке. Так, в примере на рис. 7 значение плотности в объеме $abcdABCD$ вычисляется по формуле

$$\rho^* = \rho_d \exp(),$$

где ρ_d — плотность в донорной ячейке; показатель экспоненты тот же, что и в начальном распределении плотности.

Аппроксимация уравнений движения. При аппроксимации уравнений движения в исходной методике ЭГАК в декартовой системе координат используется согласованный с потоками массы расчет потоков количества движения [5]. Для рассматриваемой задачи применение только общих формул для вычисления скорости не обеспечивает строгого сохранения угловой симметрии течения для твердотельного вращения. Поэтому разностные уравнения движения на эйлеровом этапе также претерпевают изменения.

Сначала, как и на лагранжевом этапе, выполняется разложение скорости с компонентами в декартовой системе координат на тангенциальную и радиальную компоненты (третье направление остается без изменения), т. е. $\mathbf{u}(u_x, u_y, u_z) \rightarrow \mathbf{u}(U_r, U_\varphi, u_z)$.

В предлагаемой модификации разностной схемы далее будет предполагаться, что общие формулы остаются в силе только для скорости, представленной в локальной системе координат с радиальной и тангенциальной компонентами.

В стандартной газодинамике на эйлеровом этапе пересчет скоростей с новой сетки, полученной на лагранжевом этапе, на старую неподвижную сетку осуществляется с сохранением импульса. Однако для рассматриваемой задачи принципиально важно обеспечить сохранение момента импульса.

Процедура замены закона сохранения импульса на закон сохранения момента импульса на эйлеровом этапе схематично осуществляется следующим образом. Закон сохранения импульса в тангенциальном направлении для некоего узла имеет вид

$$U_\varphi^{t_3} = \frac{U_\varphi^{t_2} m^{t_2} + 8 \sum_{i=1}^8 I_i}{m^{t_3}}, \quad (12)$$

где m — масса в узле ячейки; I_i — потоки импульса в окружающих данный узел ячейках в тангенциальном направлении. Переход к сохранению момента импульса производится заменой потоков на величины $I_i r_i$, где r_i — расстояние от центра объема потока i до оси ГЦ. В этом случае формула (12) преобразуется к виду

$$U_\varphi^{t_3} = \frac{U_\varphi^{t_2} m^{t_2}}{m^{t_3}} + 8 \frac{\sum_{i=1}^8 I_i r_i^{t_3}}{m^{t_3} r_i^{t_3}}. \quad (13)$$

После этих вычислений осуществляется обратный переход к декартовой системе координат, что сводится к вычислению соответствующих компонент скорости, т. е.

$$\mathbf{u}(U_r, U_\varphi, u_z) \rightarrow \mathbf{u}(u_x, u_y, u_z).$$

Предложенная модификация разностной схемы решения эйлеровых уравнений газовой динамики легко обобщается на случай уравнений Навье—Стокса.

Сглаживание скоростей. Методические расчеты центрифуги показали, что устойчивый счет без сглаживания скоростей возможен до 3—5 оборотов. При дальнейшем счете без сглаживания в окрестности внутреннего цилиндра возникает несимметрия, которая приводит к неустойчивости и аварийным остановам счета. Для устранения этого недостатка используется сглаживание поля скорости, прежде всего угловое.

3D сглаживание скоростей производится по двум плоскостям, для чего 3D вектор проецируется на соответствующую плоскость. Полученные сглаженные добавки прибавляются к соответствующим компонентам 3D вектора скорости.

При сглаживании по кольцам $J = \text{const}$ ($0 < J < n_j$) сглаженные добавки получаются в плоскостях $I = \text{const}$ ($0 \leq I \leq n_i$). При сглаживании по радиусу $K = \text{const}$ это плоскости $K = \text{const}$ ($0 \leq K \leq n_k$). Здесь n_i, n_j, n_k — число узлов в соответствующих направлениях.

Угловое сглаживание скорости. Рассмотрим алгоритм углового сглаживания на примере линии сетки 5-0-1 на рис. 8. Здесь приняты следующие обозначения: φ — угол в узле 0 между ребрами 5-0 и 0-1; \mathbf{n} — вектор нормали к отрезку 1-5 в узле 0; $\mathbf{u}_0 = \mathbf{u}_0(U_{x0}, U_{y0})$ — вектор скорости в узле 0.

Добавка к скорости \mathbf{u}_0 от квадратичного и линейного углового сглаживания направлена по нормали \mathbf{n} к линии сетки и вычисляется по следующей формуле:

$$\Delta \mathbf{u}_0 = \begin{cases} A \frac{\tau \dot{\varphi} (|\dot{\varphi}| + d)}{\sin^2 \frac{\varphi}{2}} \mathbf{n}, & \text{если } \dot{\varphi} \sin \varphi < 0; \\ 0, & \text{если } \dot{\varphi} \sin \varphi \geq 0. \end{cases}$$

Здесь $d = Ba_0/s_{15}$ — линейная добавка, где $s_{15} = \sqrt{(x_1 - x_5)^2 + (y_1 - y_5)^2}$ — длина отрезка 1-5; a_0 — скорость звука в узле 0; $\dot{\varphi}$ — угловая скорость; A и B — коэффициенты, определяемые на основе расчетов.

Оператор квадратичного углового сглаживания имеет порядок малости τh^3 , а оператор линейного сглаживания — τh . Положительным свойством таких операторов является то, что при стремлении временного шага τ к нулю добавка к скорости также стремится к нулю.

Рассмотрим в разностях угловое сглаживание вдоль линии сетки 5-0-1. В этом случае при определении добавки к скорости $\Delta \mathbf{u}_0$ в узле 0 используются скорости в узлах 5, 0, 1 (при сглаживании вдоль линии сетки 3-0-7 рассматриваются узлы 3, 0, 7).

Угловая скорость $\dot{\varphi}_{01}$ ребра 0-1 вокруг узла 0 имеет вид

$$\dot{\varphi}_{01} = \frac{(x_1 - x_0)(u_{y1} - u_{y0}) - (y_1 - y_0)(u_{x1} - u_{x0})}{s_{01}^2},$$

где s_{01} — длина ребра 0-1.

Аналогично вычисляется угловая скорость ребра 0-5. Тогда угловая скорость узла 0 будет равна $\dot{\varphi} = \dot{\varphi}_{01} + \dot{\varphi}_{05}$.

Сглаживание скорости производится в случае, если угол φ уменьшается и выполняется условие $\dot{\varphi} \sin \varphi < 0$, где $\sin \varphi = \frac{(x_5 - x_0)(y_1 - y_0) - (x_1 - x_0)(y_5 - y_0)}{s_{01}s_{05}}$. Добавки к скорости ребер 0-1 и 0-5 вычисляются по формуле (13). Разностная добавка к скорости $\Delta \mathbf{u}_0$ берется равной полусумме этих добавок.

Далее добавки нормируются таким образом, чтобы отношение модулей добавки и скорости не превышало заданного значения. Нормировка производится в том случае, если $|\Delta \mathbf{u}_0|/|\mathbf{u}_0| > \text{const} \sim 0,1$.

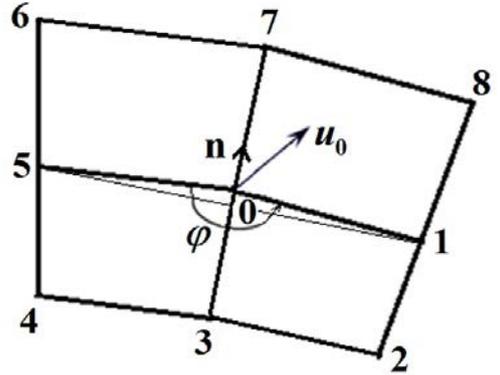


Рис. 8. Шаблон для углового сглаживания скорости

При использовании данного алгоритма сглаживания скоростей возникает дисбаланс кинетической энергии, который переводится во внутреннюю энергию ячеек, примыкающих к узлу 0.

Метод решения уравнения энергетического баланса. В данном подразделе рассматривается полная (с учетом вязкости и теплопроводности) модель теплопереноса в газе и теплообмена между газом и стенками ГЦ. Уравнение энергетического баланса для газа в цилиндрических координатах имеет следующий вид:

$$\begin{aligned} \frac{\partial E}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (rEU_r) + \frac{1}{r} \frac{\partial}{\partial \varphi} (EU_\varphi) + \frac{\partial}{\partial z} (Eu_z) = \\ = -P \left[\frac{1}{r} \frac{\partial}{\partial r} (rU_r) + \frac{1}{r} \frac{\partial U_\varphi}{\partial \varphi} + \frac{\partial u_z}{\partial z} \right] - \frac{1}{r} \frac{\partial r q_r}{\partial r} - \frac{1}{r} \frac{\partial q_\varphi}{\partial \varphi} - \frac{\partial q_z}{\partial z}. \end{aligned}$$

Здесь P — полное давление.

Отщепленное уравнение теплопроводности. На данном этапе отдельно рассчитывается теплопроводность, т. е. решается отщепленное уравнение

$$\frac{\partial E}{\partial t} = -\frac{1}{r} \frac{\partial r q_r}{\partial r} - \frac{1}{r} \frac{\partial q_\varphi}{\partial \varphi} - \frac{\partial q_z}{\partial z} + (1 - \alpha) Q_T, \quad (14)$$

где через Q_T обозначены члены, описывающие вязкий теплообмен. Здесь $\alpha = 0,5$ — весовой множитель: при расчете газодинамики члены, описывающие вязкий теплообмен, входят в уравнение энергии с весом α , а при расчете теплопроводности — с весом $(1 - \alpha)$. Значения Q_T во всех пространственных ячейках рассчитываются на этапе расчета газодинамики, а при расчете уравнения (14) "замораживаются".

Для численного решения этого уравнения из-за большого разброса (несколько порядков) в теплофизических параметрах рабочего газа и материала газоотборника удобнее перейти к уравнению температуропроводности [8]

$$\frac{\partial T}{\partial t} = -\text{div}(\chi \text{grad}(T)) + (1 - \alpha) q_T, \quad (15)$$

где χ — коэффициент температуропроводности,

$$\chi = \frac{\lambda \gamma}{\rho c_p}; \quad (16)$$

$q_T = Q_T \gamma / \rho c_p$; c_p — удельная теплоемкость газа. При этом считаем, что внутренняя энергия E и температура T газа связаны соотношением

$$E = \frac{\rho c_p T}{\gamma}; \quad (17)$$

для гексафторида урана

$$c_p = \frac{R\gamma}{(\gamma - 1)M},$$

где $R = 8,314 \cdot 10^7$ эрг/(К · моль) — универсальная газовая постоянная; $\gamma \approx 1,066$; $M = 352$ г/моль — молярная масса гексафторида урана ($^{238}\text{UF}_6$).

Для численного решения уравнения (15) используется явная схема расщепления по направлениям с семиточечным по пространству шаблоном [9].

Для уравнения температуропроводности (15) могут задаваться следующие граничные условия:

- 1) температура на границах ОК, т. е. на верхнем/нижнем торце центрифуги, диафрагме, боковой цилиндрической поверхности и цилиндрической поверхности (обращенной внутрь ОК), питающей магистрали (см. рис. 1);

- 2) условие *черного излучения* на торце центрифуги, диафрагме и боковой поверхности ОК, т. е. соотношение

$$\lambda \frac{\partial T}{\partial n} = \psi \sigma T^4,$$

где T — температура граничной поверхности ОК (торец ГЦ + диафрагма + боковая поверхность ГЦ); $\frac{\partial T}{\partial n}$ — внешняя нормальная к граничной поверхности производная температуры; $\sigma = 5,67 \cdot 10^{-8}$ — постоянная Стефана–Больцмана; ψ — степень черноты материала, из которого изготовлены ГЦ и ОК (для алюминия $\psi \approx 0,8 \div 0,93$ [9]).

Аппроксимация уравнения температуропроводности. Общая схема вычислений с учетом температуропроводности следующая. По программе газодинамики с учетом физической вязкости вычисляются плотность ρ^{n+1} и удельная энергия e^{n+1} на момент времени t^{n+1} . Затем по формуле связи энергии и температуры (17) вычисляется температура T^n :

$$T^n = \frac{e^{n+1} \gamma}{c_p}, \quad e^{n+1} = E^{n+1} \rho^{n+1}. \quad (18)$$

С использованием найденных значений T^n (18) решается разностный аналог уравнения (15). Уравнение (15) аппроксимируется по явной *потокковой* схеме расщепления по направлениям счетной сетки.

Далее по новым температурам T^{n+1} корректируется удельная энергия:

$$e_{\text{new}}^{n+1} = \frac{T^{n+1} c_p}{\gamma}.$$

Проведенные методические расчеты показали, что предложенная явная схема устойчива для данного класса задач при счете с газодинамическим временным шагом $\tau \cong h/a$, где h — размер ячейки; a — скорость звука.

4. Организация расчета полной задачи

Для определения последовательности расчетов процессов теплопроводности и газодинамики при осуществлении расчета полной задачи сопоставим характерные времена этих процессов.

Оценим характерные времена процессов теплопереноса в ОК ГЦ. Расстояние x_Φ , на которое продвигается фронт тепловой волны (или, другими словами, линейный размер области, в которой сосредоточено основное количество тепла) от мгновенного плоского источника к моменту времени t можно оценить как [9]

$$x_\Phi \sim \sqrt{\chi t}.$$

Согласно [1]

$$\lambda = 607 \cdot [1 + 0,0045 \cdot (T - 273)]. \quad (19)$$

Оценим определяемый формулой (16) коэффициент температуропроводности χ для гексафторида урана при $T = 40^\circ\text{C}$ и при $\rho \approx 10^{-3} \text{ г/см}^3$: $c_p \approx 2,4 \cdot 10^5 \text{ эрг/(г} \cdot \text{К)}$ (из [1]); $\lambda \approx 7,2 \cdot 10^2 \text{ эрг/(см} \cdot \text{с} \cdot \text{К)}$ (из (19)), откуда $\chi \approx 0,2 \text{ см}^2/\text{с}$. Следовательно, в гексафториде урана тепловой волне для прохождения характерного для ОК ГЦ расстояния ($\sim 1 \text{ см}$) требуется $\sim 5 \text{ с}$.

Оцененные характерные времена процессов теплопереноса в ГЦ ($\sim 1 \text{ с}$) на 3 порядка больше, чем характерные времена газодинамических процессов ($\sim 10^{-3} \text{ с}$, что примерно соответствует одному обороту ГЦ). На самом деле, как следует из практических расчетов, характерное время установления стационарного режима процесса теплопроводности в ОК ГЦ может быть значительно более длительным: $\sim 1 \text{ мин}$ и более. Это обстоятельство требует особого подхода при осуществлении совместного расчета процессов газодинамики и теплопроводности на установление стационарного

режима в ОК ГЦ, поскольку газодинамические расчеты на столь длительное время (секунды и минуты) нереальны.

С этой целью применяется разбиение расчета полной задачи на три этапа.

1. Рассчитывается одна *вязкая* газодинамика без теплопроводности до момента времени, соответствующего 3–4 оборотам ГЦ (установление газодинамических процессов). На этом этапе рассчитываются поля плотности, давления, температуры газа, а также вязкое энерговыделение.
2. Рассчитывается *чистая* теплопроводность, т. е. решается отщепленное уравнение (14). Значение весового множителя α при расчете этого этапа полагается равным 0. Значения Q_T во всех пространственных ячейках рассчитываются на этапе расчета газодинамики, а на протяжении расчета уравнения теплопроводности (14) "замораживаются".

Расчет второго этапа проводится до установления стационарной температуры в ОК и может длиться от нескольких секунд до нескольких минут физического времени процесса ($\sim 10^3 \div 10^4$ шагов по времени для расчета отщепленного уравнения (14)). Расчет одного временного шага уравнения теплопроводности занимает в 3–4 раза меньше времени, чем расчет одного шага газодинамики.

3. Рассчитывается одна *вязкая* газодинамика до момента времени, соответствующего 1–2 оборотам ГЦ (окончательное установление газодинамических процессов).

Вообще, для уточнения стационарного режима можно несколько раз повторить цикл расчетов этапов 2, 3, однако для приемлемой точности достаточно, по-видимому, одного полного цикла этапов 1–3.

Заключение

В работе приводится описание модификаций программы газовой динамики и новых программ расчета теплопроводности, реализованных для расчета газодинамических течений в центрифуге с учетом теплопроводности на основе методики ЭГАК.

Верификация методики проведена на задачах, имеющих точное решение, а валидация — на известных задачах для центрифуги "Игуассу". Результаты тестирования будут приведены во второй части работы.

Список литературы

1. *Борисевич В. Д., Борман В. Д., Сулаберидзе Г. А., Тихомиров А. В., Токманцев В. И.* Физические основы разделения изотопов в газовой центрифуге. М.: МИФИ, 2005.
Borisevich V. D., Borman V. D., Sulaberidze G. A., Tikhomirov A. V., Tokmantsev V. I. Fizicheskie osnovy razdeleniya izotopov v gazovoy tsentrifuge. M.: MIFI, 2005.
2. *Ivanov A. G., Porodnov B. T., Seleznev V. D., Tokmantsev V. I., Glebov A. A., Khelnitskii A. V.* Calculation of binary uranium isotope mixture flow in Iguassu centrifuge model considering the rarified gas region in kinetic terms // Proc. 7th Workshop on Separation Phenomena in Liquids and Gases. Moscow, 2000. P. 54–63.
3. Iguassu Falls. Proc. 5th Workshop on Separation Phenomena in Liquids and Gases / Eds. by C. Shwab, N. A. S. Rodrigues, H. G. Wood. Brazil, 1996.
4. *Звонарёв К. В., Селезнёв В. Д., Токманцев В. И., Абрамов Ю. В.* Моделирование теплообмена между сверхзвуковым потоком и неподвижным телом в газовой центрифуге // Инженерно-физический журнал. 2012. Т. 85, № 6. С. 12–77.
Zvonaryev K. V., Seleznyev V. D., Tokmantsev V. I., Abramov Yu. V. Modelirovanie teploobmena mezhdru sverkhzvukovym potokom i nepodvizhnym telom v gazovoy tsentrifuge // Inzhenerno-fizicheskiy zhurnal. 2012. T. 85, № 6. S. 12–77.

5. Янилкин Ю. В., Беляев С. П., Бондаренко Ю. А., Гаврилова Е. С., Гончаров Е. А., Горбенко А. Д., Городничев А. В., Губков Е. В., Гужова А. Р., Дегтяренко Л. И., Жарова Г. В., Колобянин В. Ю., Софронов В. Н., Стадник А. Л., Ховрин Н. А., Чернышова О. Н., Чистякова И. Н., Шемяков В. Н. Эйлеровы численные методики ЭГАК и ТРЭК для моделирования многомерных течений многокомпонентной среды // Труды РФЯЦ-ВНИИЭФ. 2008. Вып. 12. С. 54—65.
Yanilkin Yu. V., Belyaev S. P., Bondarenko Yu. A., Gavrilova E. S., Goncharov E. A., Gorbenco A. D., Gorodnichev A. V., Gubkov E. V., Guzhova A. R., Degtyarenko L. I., Zharova G. V., Kolobyanin V. Yu., Sofronov V. N., Stadnik A. L., Khovrin N. A., Chernyshova O. N., Chistyakova I. N., Shemyakov V. N. Eylerovy chislennye metodiki EGAK i TREK dlya modelirovaniya mnogomernykh techeniy mnogokomponentnoy sredy // Trudy RFYaTs-VNIIEF. 2008. Vyp. 12. S. 54—65.
6. Обогащение урана: Пер. с англ. / Под ред. С. Виллани. М.: Энергоатомиздат, 1983.
Obogashchenie urana: Per. s angl. / Pod red. S. Villani. M.: Energoatomizdat, 1983.
7. Matsuda T., Tamura N., Sawada K. Three-dimensional numerical simulation of flows past scoops in a gas centrifuge // J. Fluid Mech. 1989. Vol. 201. P. 203—221.
8. Зельдович Я. Б., Райзер Ю. П. Физика ударных волн и высокотемпературных гидродинамических явлений. М.: Наука, 1966.
Zeldovich Ya. B., Rayzer Yu. P. Fizika udarnykh voln i vysokotemperaturnykh gidrodynamiceskikh yavleniy. M.: Nauka, 1966.
9. Самарский А. А. Теория разностных схем. М.: Наука, 1977.
Samarskiy A. A. Teoriya raznostnykh skhem. M.: Nauka, 1977.

Статья поступила в редакцию 02.02.2022.

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНОГО ТЕЧЕНИЯ ГАЗА В ГАЗОВОЙ ЦЕНТРИФУГЕ. ЧАСТЬ 2. РЕЗУЛЬТАТЫ РАСЧЕТОВ МОДЕЛЬНОЙ ЦЕНТРИФУГИ "ИГУАССУ"

Ю. В. Янилкин, В. П. Стаценко, Ю. Д. Чернышёв, С. П. Беляев,
В. Ю. Колобянин, А. В. Кондрашенко, О. О. Топорова, И. Ю. Турутина
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Данная статья является продолжением первой части работы авторов, посвященной трехмерному численному моделированию течений в газовой центрифуге. Во второй части приводятся описание шести трехмерных и двумерных тестовых задач для однокамерной и двухкамерной модельной центрифуги "Игуассу" и результаты их моделирования по трехмерной методике, описание которой дано в первой части работы. Рассмотрены следующие задачи: твердотельное вращение газа с постоянной температурой; установление твердотельного вращения; газодинамическая задача с учетом теплопроводности, для которой получено аналитическое решение; однокамерная модель ротора газовой центрифуги, для которой также получено аналитическое решение; задача с двухкамерной моделью ротора газовой центрифуги, в которой исследовалось влияние места сшивки двумерной и трехмерной областей решения и сходимости численного решения при измельчении счетной сетки; одномерная диффузионная задача с аналитическим решением. Для всех тестов получено приемлемое согласие с аналитическими решениями.

Ключевые слова: газовой центрифуга, центрифуга "Игуассу", разделительная камера, отборная камера, отвальная камера, вязкая жидкость, теплообмен, численное моделирование.

Введение

Численное моделирование течений в газовой центрифуге (ГЦ) представляет собой достаточно сложную задачу. В настоящее время известно несколько двумерных (2D) методов моделирования течений в разделительной камере [1–3]. Что касается трехмерных (3D) течений в отборной и отвальной камерах, то для их расчетов используют приближенные граничные условия на границе с разделительной камерой [3]. 3D расчет [4] был выполнен лишь для части отборной камеры и части отборника с использованием приближенных граничных условий на границе счетной области.

В первой части работы авторов [5] предложена физико-математическая модель и численная методика для сквозного 3D моделирования течений в ГЦ, в которой учитывается газодинамическое движение рабочего газа и конвективный перенос тепла в нем. Эта модель построена на основе полной (с учетом теплопроводности) модели теплопереноса в газе и отборнике, а также вязкого теплообмена между газом и стенками ГЦ. Аппроксимация уравнений производится в специальной декартово-цилиндрической системе координат.

Данная работа является продолжением работы [5] и содержит результаты 3D и 2D численного моделирования течения газа в однокамерной и двухкамерной моделях ротора ГЦ с параметрами, характерными для модельной центрифуги "Игуассу" [6].

Общая постановка расчетов центрифуги

В данной работе используются основные положения, принятые в [6]. У боковой стенки ротора давление гексафторида урана (UF_6) должно быть $p_0 \lesssim 100$ мм рт. ст. При радиусе ротора $R \lesssim 10$ см свободный пробег молекул на некотором радиусе $r = r_1$ становится сравним с радиусом ротора, если давление при $r = r_1$ на шесть порядков меньше p_0 . Это происходит при $r_1 \approx 0,8R$. В данной работе полагается (см. [5]), что уравнения газовой динамики, теплопроводности и диффузии применимы всюду. Именно так в [6] получено аналитическое выражение для профиля давления в поле центробежных сил (см. далее тест 1), а также приближенное аналитическое решение для движения газа вблизи боковой стенки длинного цилиндра (см. далее тест 2).

В расчетах данной работы коэффициент диффузии UF_6 определяется эмпирическим соотношением [6, 7] $\rho D = 1,117 \cdot 10^{-6} T^{0,925}$. Здесь и далее, где не указано, температура измеряется в К, единицы измерения остальных величин — в системе СГС (г, см, с). Коэффициент теплопроводности [6, 7] $\lambda = 6,07 \cdot 10^2 [1 + 0,0045(T - 273)]$; удельная теплоемкость $c_p = 3,857 \cdot 10^6 + 943,9T - \frac{3,814 \cdot 10^{10}}{T^2} \frac{\text{эрг}}{\text{г} \cdot \text{К}}$; коэффициент вязкости [6, 7] $\eta = 0,865 \cdot 10^{-6} T^{0,928}$. Уравнение состояния $p = \rho \varepsilon (\gamma - 1)$, где $\gamma = 1,0659 + 0,0067 \cdot 10^{-6} p$; для гексафторида урана более точным является выражение [6, 7] $p(1 + Ap) = \rho \frac{R_0}{M} T$, где $A = 1,217/T^3$; $M = 352$ г/моль — молярная масса гексафторида урана (для ^{238}U); $R_0 = 8,314 \cdot 10^7 \frac{\text{эрг}}{\text{К} \cdot \text{моль}}$ — универсальная газовая постоянная.

Все расчеты проводились на неподвижной цилиндрической счетной сетке, схематически показанной на рис. 2, б в [5]. В областях отборной и отвалной камер, где течение 3D, эта сетка занимает всю пространственную область, в области 2D течения в разделительной камере — лишь несколько ячеек по углу. При этом в угловом направлении в том и другом случае реализуются циклические граничные условия.

Тест 1. Твердотельное вращение газа с постоянной температурой

Расчетная область показана на рис. 1.

Все границы вращаются с угловой скоростью $\Omega_0 = 1,1 \cdot 10^4$ рад/с. Граничные условия по газовой динамике: на всех указанных границах — условие прилипания, а также условие непротекания (*жесткая стенка*).

В начальный момент ($t = 0$) компоненты скорости $u_z = 0$, $U_r = 0$, $U_\varphi = \Omega_0 r$; температура $T_0 = 312$ К. Начальные распределения давления и плотности газа брались из аналитического решения для газа с температурой T_0 , вращающегося как твердое тело [7]:

$$p(r) = p_0 F_0(r); \quad \rho(r) = \rho_0 F_0(r).$$

Здесь

$$F_0(r) \equiv \exp\left(A \frac{r^2 - a^2}{a^2}\right), \quad A = \frac{M \Omega_0^2 a^2}{2 R_0 T_0},$$

$M = 352$ г/моль — молярная масса $^{238}\text{UF}_6$; p_0 , ρ_0 — начальные значения давления и плотности на боковой стенке ротора ($r = a$) (см. рис. 1):

$$\rho_0 = \frac{M p_0}{R_0 T_0}; \quad p_0 = 1,6275 \cdot 10^5 \text{ дин/см}^2.$$

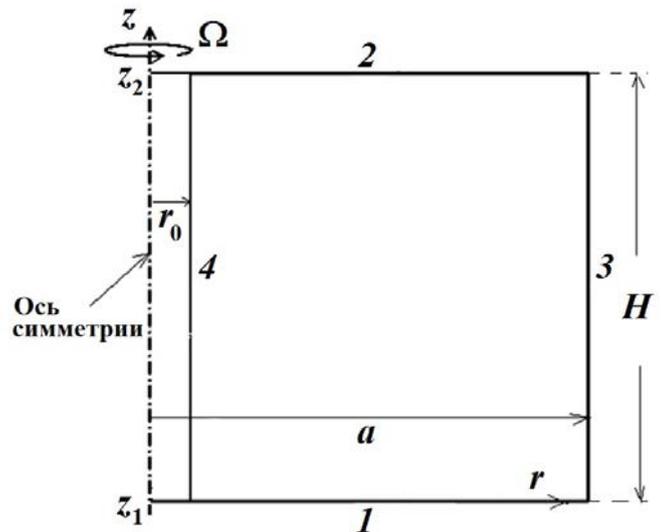


Рис. 1. Геометрия системы в тесте 1: 1 — нижний торец ротора; 2 — верхний торец; 3 — боковая стенка; 4 — внутренняя цилиндрическая поверхность

Рассматривается область высотой $H = 6$ с радиусами $r_0 = 1,5$, $a = 6,5$ и углом $\varphi = 6^\circ$. Расчеты проведены на двух сетках: $N_z \times N_r \times N_\varphi = 104 \times 72 \times 6$ (расчет N) и $340 \times 240 \times 6$ (расчет $2N$).

На рис. 2 приводятся профили плотности в логарифмическом масштабе и радиальной компоненты скорости для обоих расчетов. Как видно, они практически совпадают с начальным распределением независимо от сетки, т. е. имеется хорошее согласие между численными и аналитическим решениями.

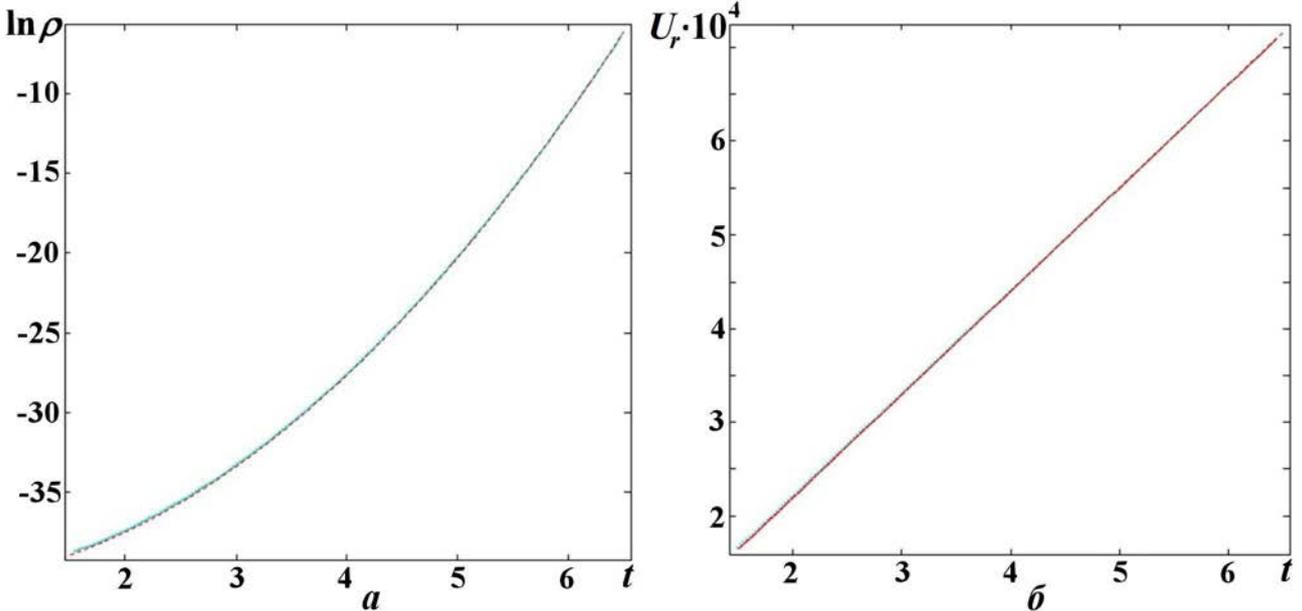


Рис. 2. Тест 1. Профили логарифма плотности (а) и радиальной компоненты скорости (б): $-\cdots-$ — $t = 0$; \cdots — $t = 0,1$ (расчет N); $—$ — $t = 0,04$ (расчет $2N$)

Тест 2. Установление твердотельного вращения

Отличие теста 2 от теста 1 — лишь в начальном профиле логарифма плотности: он — линейный по радиусу (рис. 3, а). Как видно из рис. 3, а, при расчете даже на самой грубой сетке (расчет N) уже к моменту $t = 0,02$ устанавливается профиль логарифма плотности, практически совпадающий с аналитическим решением для твердотельного вращения. Для более подробных сеток время установления еще меньше.

Задача была также смоделирована в двухобластной постановке, в которой тестируется способ разбиения задачи на две области [5] — 2D и 3D; в соответствии с этим рассматривались две счетные области. В 2D области в силу предполагаемой осевой симметрии использовалось лишь 6 ячеек по углу, т. е. по существу решалась 2D задача. 2D область имела длину $z = 0 \div 4,75$ с $r_0 = 1,5$, $a = 6,5$. Бралась счетная сетка $N_z \times N_r \times N_\varphi = 76 \times 72 \times 6$, при этом по углу она соответствовала сетке из 3D области. В 3D области ($z = 4,26 \div 6$) бралось 192 узла по углу φ , т. е. решалась полная 3D задача: $N_z \times N_r \times N_\varphi = 28 \times 72 \times 192$.

Как видно из рис. 3, б, в двухобластной постановке время установления аналитического решения несколько увеличивается по сравнению с однообластной постановкой, но к моменту $t = 0,04$ устанавливается профиль логарифма плотности, практически совпадающий с аналитическим решением в среднем сечении как первой, так и второй области.

Тест 3. Газодинамический тест с учетом теплопроводности

Постановка задачи. Газодинамическая постановка задачи соответствует тесту 1. Дополнительно в задаче учитывается теплопроводность. Граничные условия по теплу: на границе 1 (см. рис. 1)

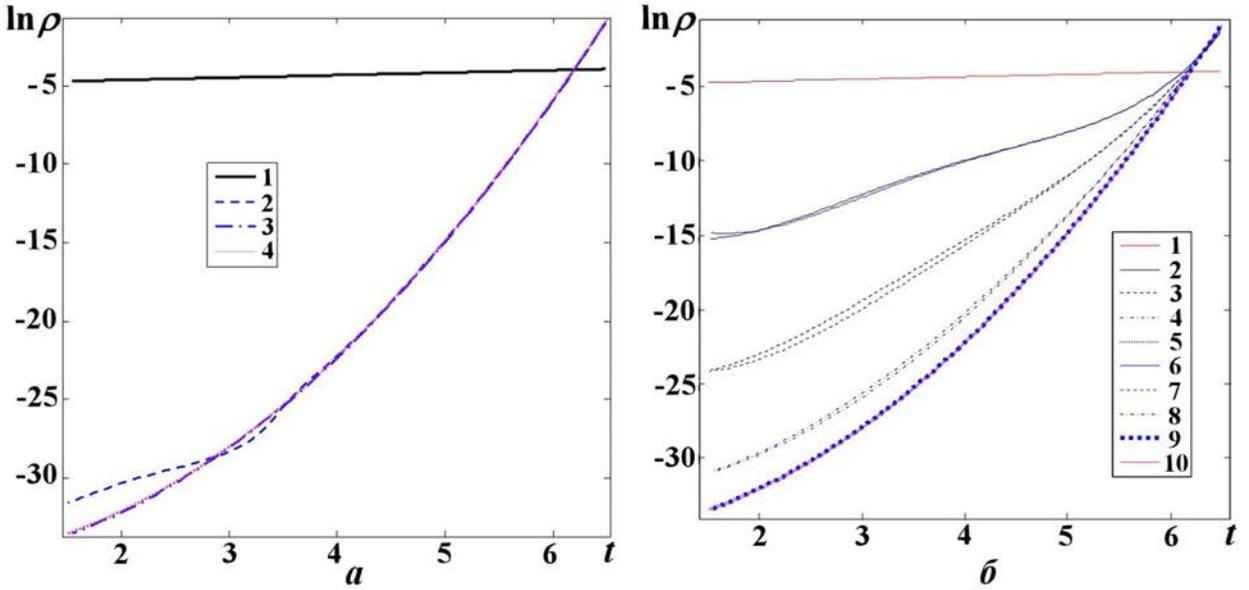


Рис. 3. Тест 2. Профили логарифма плотности: a — однообластной расчет N : 1 — $t = 0$; 2 — $t = 0,01$; 3 — $t = 0,02$; 4 — аналитическое решение; b — двухобластной расчет: 1 — $t = 0$; 2–5 — 2D область: 2 — $t = 0,005$; 3 — $t = 0,015$; 4 — $t = 0,0226$; 5 — $t = 0,04$; 6–9 — 3D область: 6 — $t = 0,005$; 7 — $t = 0,015$; 8 — $t = 0,0226$; 9 — $t = 0,04$; 10 — аналитическое решение

$T = T_1$, на границе 2 $T = T_2$, на границах 3 и 4

$$T(z) = T_1 + \Delta T \frac{z - z_1}{H}, \quad H = z_2 - z_1, \quad \Delta T = (T_2 - T_1).$$

В газе начальная температура $T_0(r, z) = T(z)$.

Начальные распределения плотности и давления газа брались из квазистационарного решения для твердотельного вращения.

В расчетах приняты значения $\rho_{00} = \rho_0 \left(\frac{z_1 + z_2}{2} \right) = 1,72574 \cdot 10^{-3}$ и $p_0 = 1,27 \cdot 10^5$, что соответствует средней температуре $T_0 = \frac{T_2 + T_1}{2} = 312$ К. Это дает газосодержание (для идеального газа и

$$A_0 \ll 1, \text{ где } A_0 \equiv A \left(\frac{z_1 + z_2}{2} \right))$$

$$m \approx \frac{\pi H \rho_{00} a^2}{A_0} \left(1 - \exp \left(A_0 \frac{r_0^2 - a^2}{a^2} \right) \right) \approx \frac{2\pi H \rho_{00}}{\Omega^2} \approx 0,35.$$

Приближенное аналитическое решение. При получении приближенного аналитического решения [3] для тонкого слоя (слой Стюартсона) вблизи боковой стенки ротора длинной центрифуги исходим из упрощений, описанных в [6]. При этом уравнение для осевой компоненты скорости u_z , которая не зависит от координаты z (решение рассматривается для точек вдали от торцов), имеет вид [6]

$$\frac{d^3 w}{d\xi^3} + \frac{d^2 w}{d\xi^2} = B e^{-\xi}, \quad (1)$$

где $w \equiv \frac{u_z}{\Omega a}$; $\xi \equiv A_0 \left(1 - \frac{r^2}{a^2} \right)$; $B \equiv \frac{b}{8A_0^3 E_k}$, $b \equiv \frac{a}{T_0} \frac{\Delta T}{H}$, $E_k = \frac{\mu}{\rho_{00} \Omega a^2}$ — число Экмана, μ — динамический коэффициент вязкости.

Как показано в [6], решение уравнения (1) имеет вид

$$w = B\xi e^{-\xi} + c_1(\xi - 1) + c_2 + c_3 e^{-\xi}. \quad (2)$$

Константы интегрирования находятся из трех граничных условий.

Первое условие — прилипания на боковой стенке 3 ротора (см. рис. 1) — означает, что

$$w(\xi = 0) = 0.$$

Тогда из (2) следует

$$c_1 = c_2 + c_3,$$

что позволяет записать (2) в виде

$$w = B\xi e^{-\xi} + c_1\xi + c_3(e^{-\xi} - 1). \quad (3)$$

Второе граничное условие отличается от принятого в [6], где предполагалось отсутствие конструкций на оси. Здесь учитывается наличие твердой трубки на оси, для поверхности 4 которой (см. рис. 1) запишем условие прилипания

$$w(\xi = \xi_0) = 0, \quad \xi_0 \equiv A_0 \left(1 - \frac{r_0^2}{a^2}\right).$$

С учетом (3) имеем

$$c_1 = \frac{c_3}{\xi_0} (1 - e^{-\xi_0}) - B e^{-\xi_0}. \quad (4)$$

Третье условие — закон сохранения массы в безотборном режиме, который имеет вид [6]

$$L = 0 = \int_{r_0}^a \rho(r) u_z 2\pi r dr. \quad (5)$$

Принятые приближения означают, что плотность $\rho(r)$ можно записать в виде

$$\rho(r) = \rho_{00} F_0(r),$$

где

$$F_0(r) \equiv \exp\left(A_0 \frac{r^2 - a^2}{a^2}\right) = e^{-\xi}.$$

Тогда (5) принимает вид

$$\int_{r_0}^a \rho_{00} e^{-\xi} u_z 2\pi r dr = \frac{\pi \rho_{00} \Omega a^3}{A_0} \int_0^{\xi_0} w e^{-\xi} d\xi = 0.$$

Отсюда и из выражений (3), (4) получаем формулу для c_3 :

$$c_3 = \frac{B \left[e^{-\xi_0} - e^{-2\xi_0} \left(\frac{\xi_0}{2} + \frac{3}{4} \right) - \frac{1}{4} \right]}{\frac{1 - e^{-\xi_0}}{\xi_0} \left[1 - e^{-\xi_0} (\xi_0 + 1) \right] - \frac{e^{-2\xi_0} - 1}{2} + e^{-\xi_0} - 1}. \quad (6)$$

Результаты расчетов. Далее приведены результаты расчетов по формулам (3), (4), (6). Для расчетов были выбраны характеристики модельной центрифуги типа "Игуассу": $a = 6,5$; $r_0 = 1,5$;

$H = 53$; разность температур на торцах варьировалась: $\Delta T = 1$ и 10 К. В расчетах варьировалась счетная сетка $N_z \times N_r \times N_\varphi$, при этом по углу $\varphi = 6^\circ$. В начальный момент $U_r = 0$, $U_\varphi = 0$, $u_z = \Omega_0 r$, $\Omega_0 = 1,1 \cdot 10^4$; плотность соответствовала твердотельному вращению.

Профили осевой компоненты скорости в расчетах на сетке $N_z \times N_r \times N_\varphi = 840 \times 72 \times 6$ показаны в сечении $z = H/2$ на различные моменты времени на рис. 4. Там же показаны полученное выше аналитическое решение, а также решение [3], приведенное в [6].

Как видно, уже к моменту $t = 0,05$ профиль осевой компоненты скорости практически устанавливается. Для $\Delta T = 1$ К он оказывается довольно близок к полученному выше аналитическому решению и лишь качественно согласуется с ним в случае $\Delta T = 10$ К. Подобное поведение численного решения представляется естественным, поскольку условие применимости такого решения $\Delta T \ll T$ во втором случае не выполняется.

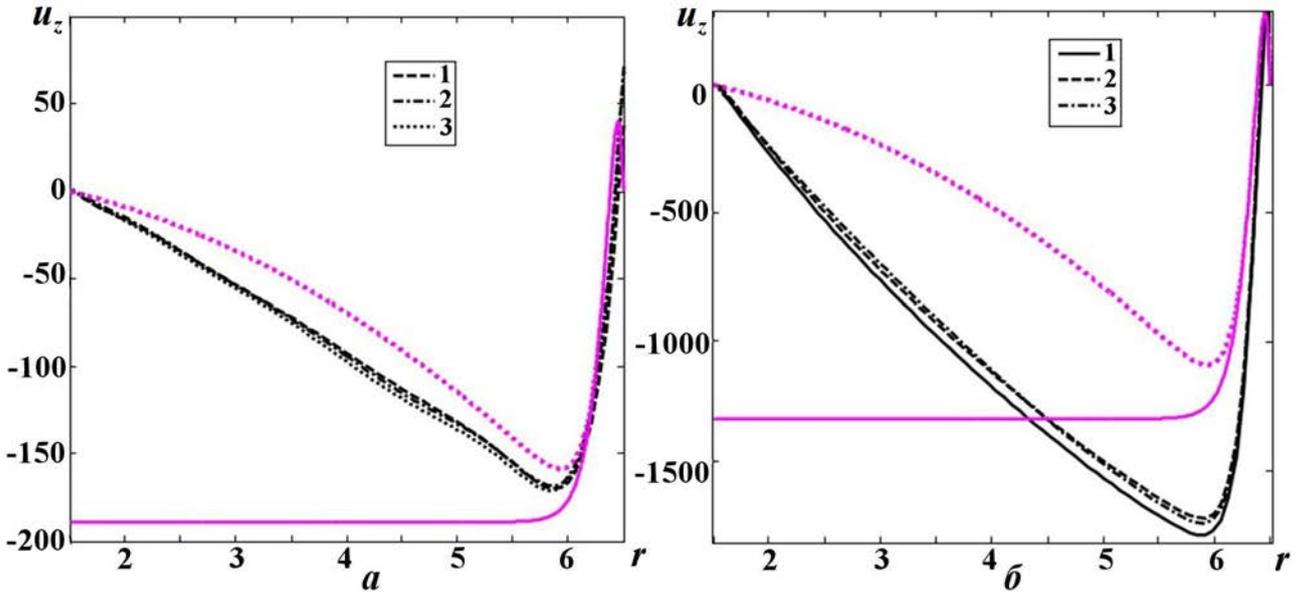


Рис. 4. Тест 3. Расчетные профили осевой компоненты скорости в задаче: $a - \Delta T = 1$ К: 1 - $t = 0,05$; 2 - $t = 0,1$; 3 - $t = 0,2$; $b - \Delta T = 10$ К: 1 - $t = 0,045$; 2 - $t = 0,1$; 3 - $t = 0,16$; — аналитическое решение из работы [3]; — аналитическое решение из данной работы

Тест 4. Однокамерная модель ротора ГЦ

Постановка задачи. На рис. 5 изображена однокамерная модель ротора центрифуги "Игуасу" [1]. Радиус ротора $R_0 = 6,5$ см, длина разделительной камеры $L_0 = 44$ см, частота вращения $\Omega = 1,1 \cdot 10^4$. С внутренней стороны область ограничена неподвижным коллектором $R_1 = 1$ см и источником питания с $R_g = 2$ см, $h_g = 2$ см. Расстояние от нижнего торца до источника питания $L_g = 0,6L_0$. Поток питания, равный

$$g_f = -2\pi \int_{r=1,7}^{r=2} \rho u_z r dr = 13 \cdot 10^{-3} \text{ г/с},$$

с массовой концентрацией легкого изотопа $c = c_0 = 0,5$ подается из кольцевого отверстия шириной $0,3$ см в нижней части источника питания. Поток питания разделяется на поток g_f^+ отбора легкой фракции и поток g_f^- отвала тяжелой фракции в пропорции $\theta = 0,5$:

$$g_f^+ = \theta g_f = -2\pi \int_{r_1}^{r_2} \rho c u_z r dr; \quad g_f^- = (1 - \theta) g_f = 2\pi \int_{r_5}^{r_6} \rho (1 - c) u_z r dr. \quad (7)$$

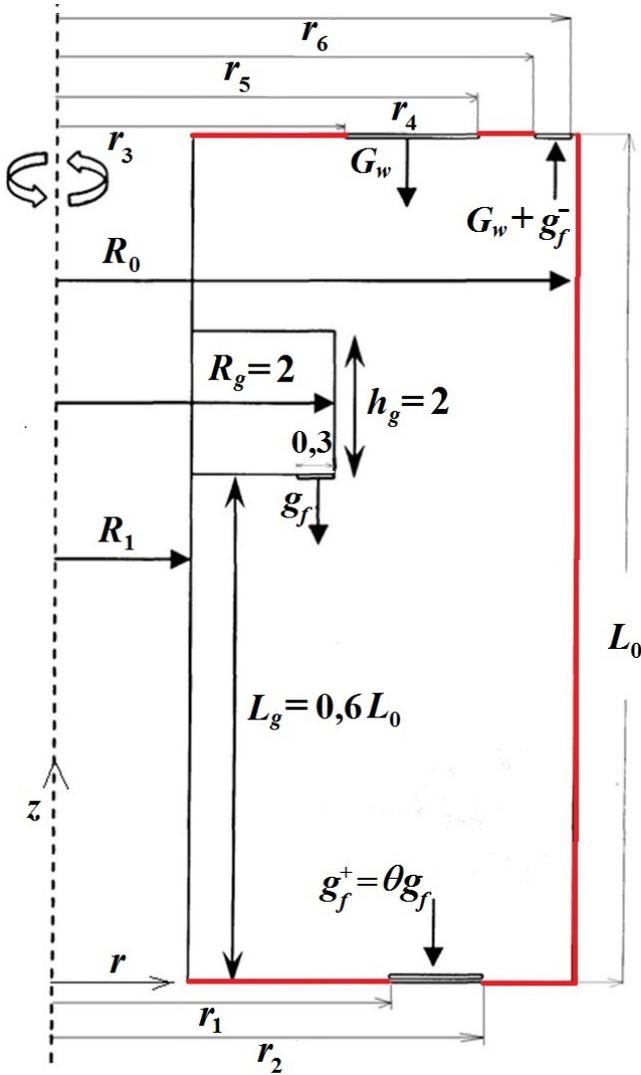


Рис. 5. Однокамерная модель ротора ГЦ

вой скоростью Ω . Граничные условия по газовой динамике: на всех границах задано условие прилипания. Граничные условия по теплопроводности: на нижнем торце ротора ($z = z_1$) $T = T_H = 305$ К; на верхнем торце ($z = z_2$) $T = T_B = T_H + \Delta T = 313$ К, $\Delta T = 8$ К; на вертикальных границах ($r = R_1$ и $r = R_0$) $T(z) = T_H + \Delta T(z - z_1)/L_0$.

Начальная температура газа $T^{(0)}(r, z) = T(z)$; распределения плотности и давления газа брались из квазистационарного твердотельного вращения с учетом того, что внутренняя граница центрифуги неподвижна.

Расчеты проводились на трех сетках: грубой ($N_z \times N_r \times N_\varphi = 34 \times 24 \times 6$), средней ($N_z \times N_r \times N_\varphi = 700 \times 80 \times 6$) и подробной ($N_z \times N_r \times N_\varphi = 1400 \times 160 \times 6$).

Аналитическое рассмотрение задачи. Рассмотрим стационарную задачу. Пусть r_1 — радиус внутренней поверхности цилиндра и блока питания, r_2 — радиус внешнего цилиндра, ω_1 и ω_2 — их угловые скорости; динамический коэффициент молекулярной вязкости $\eta = \text{const}$. Обозначим окружную скорость через u ($u \equiv U_\varphi$ зависит только от r). Система уравнений Навье—Стокса в цилиндрических координатах для такого течения примет вид

Для отбора легкой фракции g_f^+ на нижней границе разделительной камеры расположено кольцевое отверстие. На верхней границе вблизи боковой стенки ротора расположено кольцевое отвальное отверстие для прохождения газа из разделительной камеры в отвальную. Ближе к оси ротора расположено кольцевое отвальное отверстие для рециркуляционного прохождения газа из отвальной камеры в разделительную. Имеем

$$G'_w = G_w + g_f^-,$$

$$G_w = -2\pi \int_{r_3}^{r_4} \rho u_z r c dr,$$

где G_w описывает рециркуляционный (возвратный) поток газа из отвальной камеры в разделительную. Значение $G_w \sim 2 \div 4g_f^-$ должно подбираться исходя из условия максимального разделения изотопной смеси по потокам отбора. На первом этапе тестирования данной задачи для упрощения постановки задачи (сведения ее к двумерной) все отверстия брались кольцевыми.

Нижний торец ротора имеет температуру T_H , а верхний — T_B ($T_H < T_B$), температура боковой стенки ротора линейно изменяется от T_H до T_B . Для данной задачи $T_H = 305$ К, а температурный перепад должен составлять ~ 8 К. Давление газа на боковой стенке ротора $p_0 = 100$ мм рт. ст. Значения радиусов (см. рис. 5): $r_1 = 5,26$; $r_2 = 5,74$; $r_3 = 4,71$; $r_4 = 5,74$; $r_5 = 6,09$; $r_6 = 6,37$.

Внутренняя граница ($R_1 = 1$) имеет нулевую скорость, остальные границы вращаются с угло-

$$\rho \frac{u^2}{r} = \frac{dp}{dr}; \quad (8)$$

$$\frac{d^2u}{dr^2} + \frac{d}{dr} \left(\frac{u}{r} \right) = 0. \quad (9)$$

Граничные условия: $u = r_1\omega_1$ при $r = r_1$; $u = r_2\omega_2$ при $r = r_2$.

Проинтегрировав уравнение (9) при заданных граничных условиях, получим

$$u(r) = \frac{1}{r_2^2 - r_1^2} \left[r (\omega_2 r_2^2 - \omega_1 r_1^2) - \frac{r_1^2 r_2^2}{r} (\omega_2 - \omega_1) \right].$$

В рассматриваемом случае $\omega_1 = 0$, поэтому

$$u(r) = \frac{\omega_2 r_2^2}{r_2^2 - r_1^2} \left(r - \frac{r_1^2}{r} \right).$$

Дополнительно можно определить распределение давления, которое дается уравнением (8):

$$p = p_2 \exp \left(\int_{r_2}^r \frac{u^2}{RT r} dr \right),$$

где $p_2 = p(r_2)$. Обозначив

$$\frac{\omega_2 r_2^2}{r_2^2 - r_1^2} = a; \quad -\frac{\omega_2 r_2^2 r_1^2}{r_2^2 - r_1^2} = b,$$

получим

$$u(r) = ar + \frac{b}{r}; \quad u^2(r) = a^2 r^2 + 2ab + \frac{b^2}{r^2}.$$

Тогда давление

$$p = p_2 \exp \left(\frac{a^2 (r^2 - r_2^2)}{2RT} + \frac{2ab}{RT} \ln \frac{r}{r_2} + \frac{b^2}{2RT} \frac{r^2 - r_2^2}{r^2 r_2^2} \right).$$

Результаты расчетов. Поля осевой компоненты скорости показаны на рис. 6 для всех расчетов на два момента времени. Видно, что к определенному моменту времени течение устанавливается, так как картины течения на приведенные времена почти неотличимы. При этом время установления уменьшается с уменьшением размеров ячейки: на грубой сетке $t = 0,09$, на средней — $0,05$, на подробной — $0,01$.

На рис. 7 приводятся распределения температуры газа, полученные в расчете на подробной сетке на два момента времени, которые также демонстрируют выход течения на стационарный режим.

На рис. 8 приведены зависимости от времени величин $\zeta_F \equiv \frac{F - F_0}{E_{s0}}$, где F принимает значения внутренней энергии $E_{\text{вн}}$, кинетической энергии $E_{\text{кин}}$ и полной энергии $E_c \equiv E_{\text{вн}} + E_{\text{кин}}$; $F_0 \equiv \min(F)$. Как видно из рис. 8, устанавливается практически стационарное значение внутренней энергии, в то время как кинетическая энергия (а следовательно, и полная) не успевает установиться, хотя относительная погрешность при этом мала.

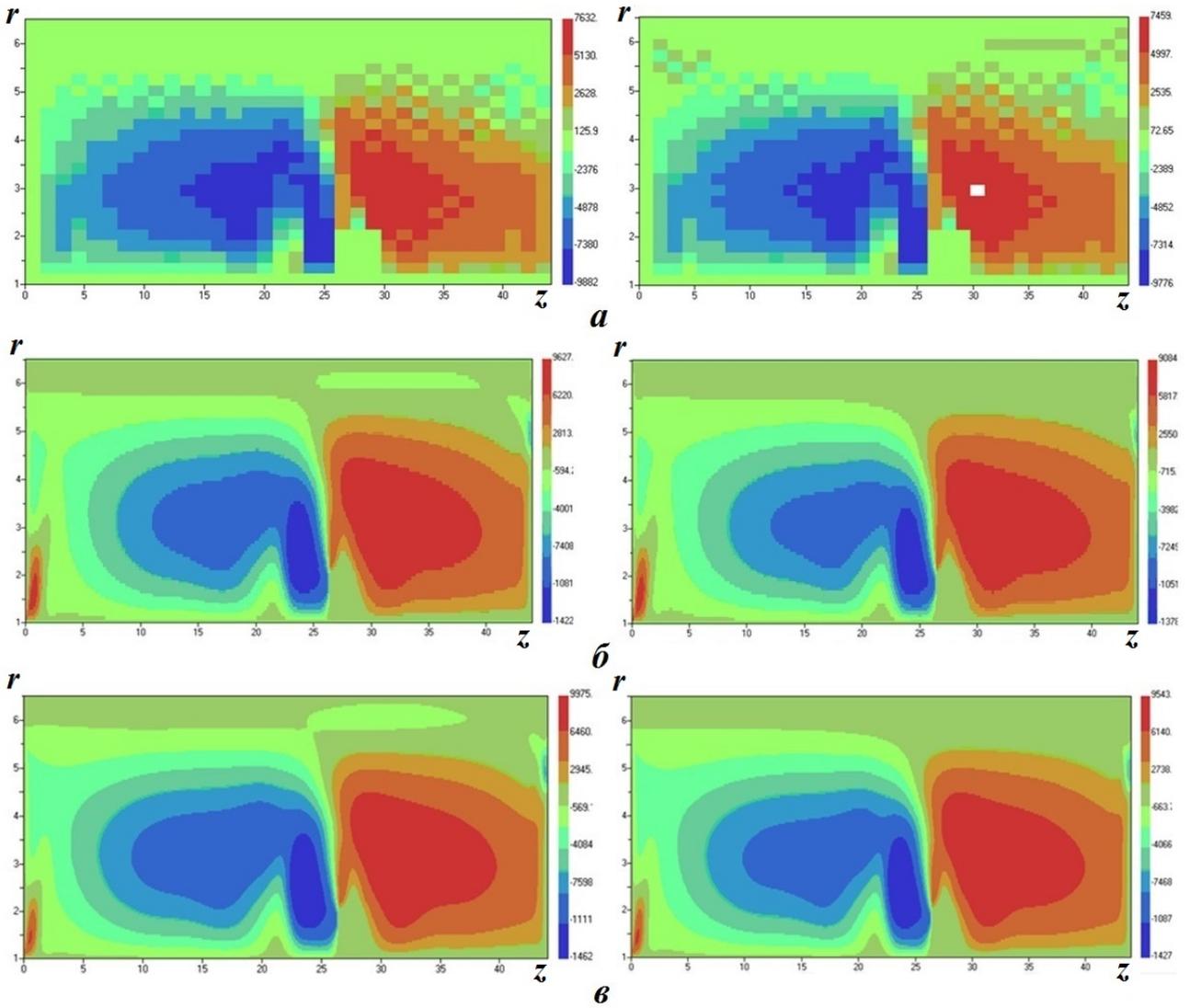


Рис. 6. Тест 4. Поля осевой компоненты скорости: *a* — грубая сетка, $t = 0,09; 0,2$; *б* — средняя сетка, $t = 0,01; 0,09$; *в* — подробная сетка, $t = 0,01; 0,04$

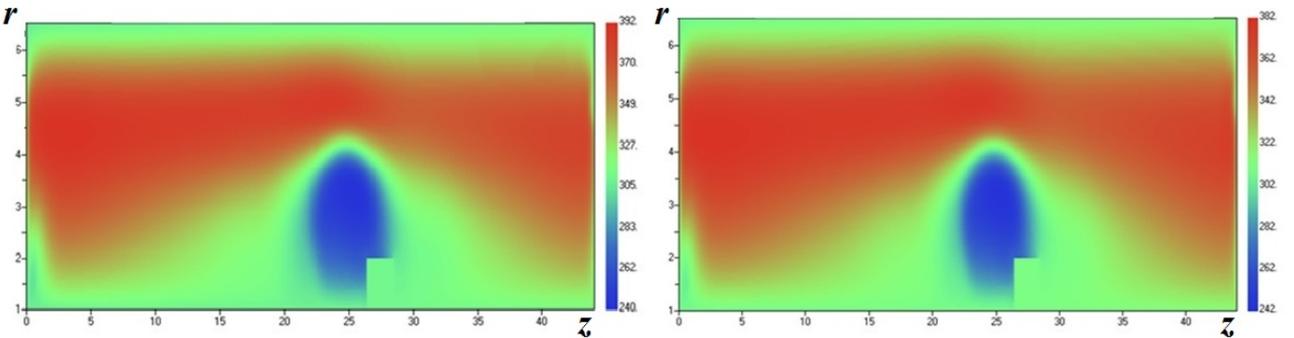


Рис. 7. Тест 4. Поля температуры в расчете на подробной сетке: $t = 0,01; 0,04$

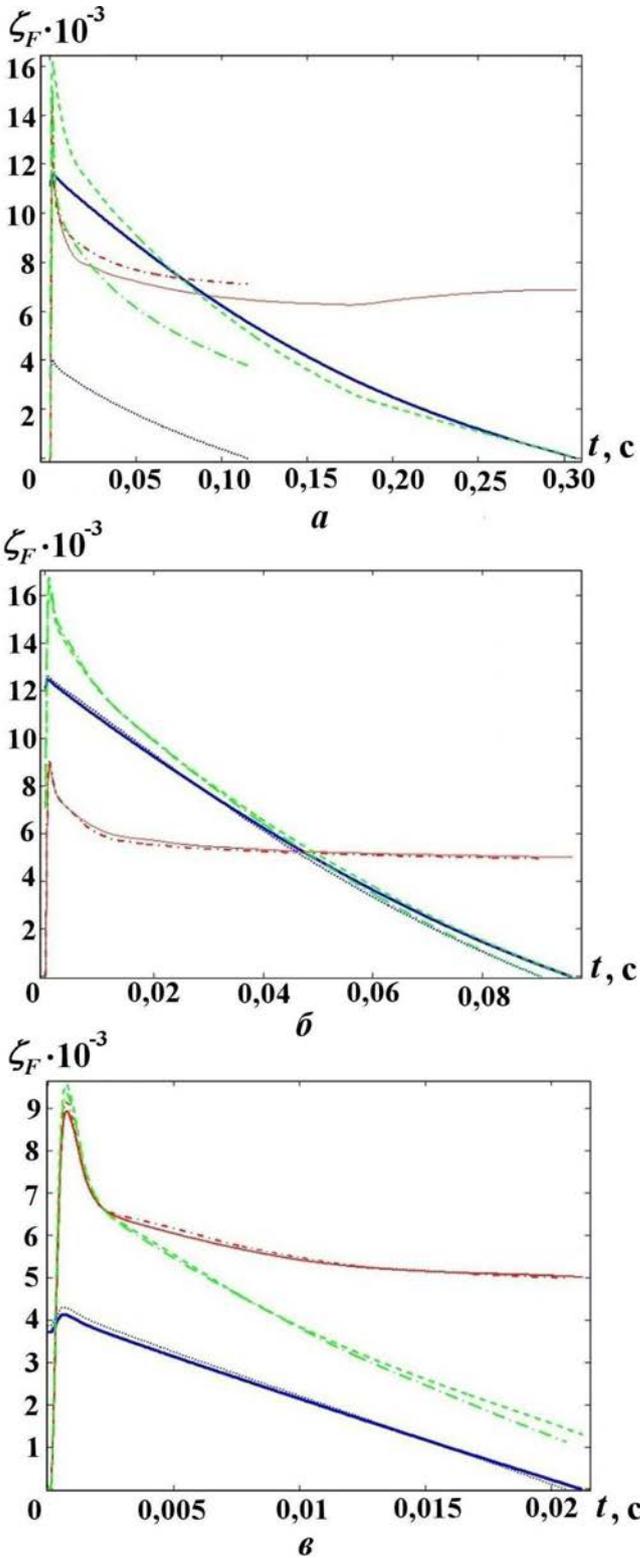


Рис. 8. Тест 4. Зависимости от времени энергий системы: *a* — грубая сетка; *b* — средняя сетка; *v* — подробная сетка: —, - - - — внутренняя энергия; —, — кинетическая энергия; - - - , - - - - — полная энергия; —, —, - - - — со сглаживанием скоростей; - - - , , - - - - — без сглаживания скоростей

Тест 5. Двухкамерная модель ротора ГЦ

Постановка расчета. Геометрия разделительной камеры (рис. 9) с источником питания и кольцевыми отверстиями в ней в тесте 5 совпадает с принятой для теста 4. Радиус ротора $R_0 = 6,5$ см; длина разделительной камеры $L_0 = 44,25$ см; $R_g = 2$ см; $h_g = 2$ см; $L_g = 0,6L_0$; $\Omega = 1,1 \cdot 10^4$. Поток питания $g_f = -2\pi \int_{r=1,7}^{r=2} \rho u_z r dr = 13 \cdot 10^{-3}$ г/с с массовой концентрацией легкого изотопа $c = c_0 = 0,5$ подается из кольцевого отверстия шириной 0,3 см в нижней части источника питания, $\theta = 0,5$. Для отбора легкой фракции (g_f^+) на нижней границе разделительной камеры расположено кольцевое

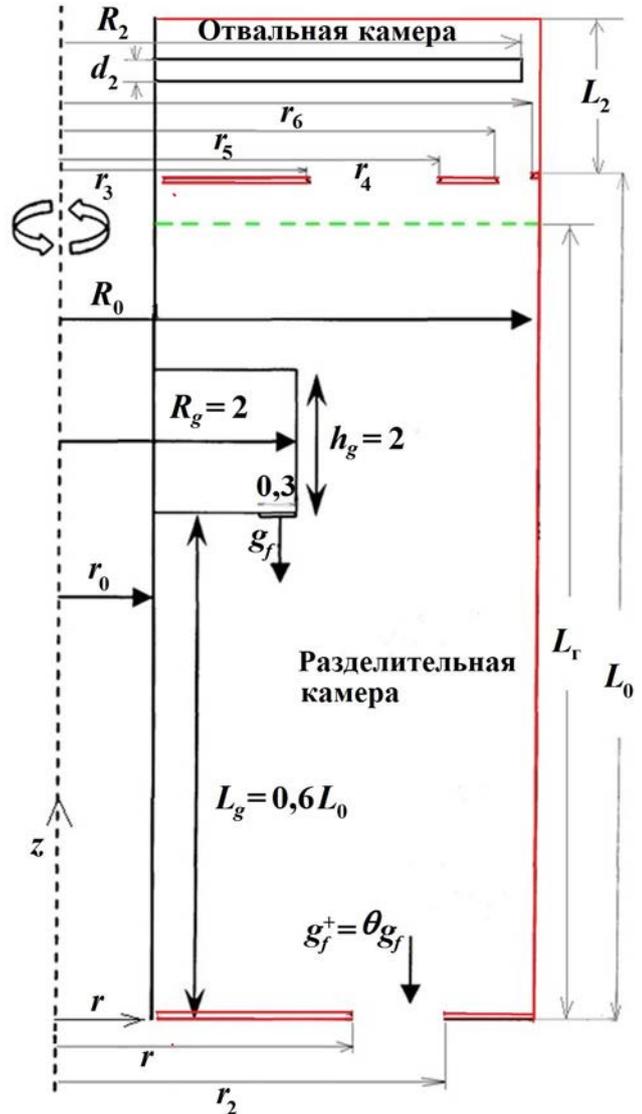


Рис. 9. Двухкамерная модель ротора ГЦ

отверстие. Поскольку в тесте 5 потоки отвала и отбора не учитываются (отборник цельнометаллический), то для отверстия нижнего торца полагается $c = 1$, $\theta = 1$. На верхней границе вблизи боковой стенки ротора расположено кольцевое отвальное отверстие для прохождения газа из разделительной камеры в отвальную. Ближе к оси ротора расположено кольцевое отверстие для рециркуляционного прохождения газа из отвальной камеры в разделительную. В центре (по оси) отвальной камеры, имеющей высоту $L_2 = 2$ см, расположен отборник с высотой (максимальный размер вдоль оси) $d_2 = 0,216$ см и вылетом $R_2 = 6,2855$ см. Нижняя крышка ротора имеет температуру T_H , а верхняя — T_B ($T_H < T_B$), температура боковой стенки ротора линейно изменяется от 305 до 313 К.

Номинальные значения радиусов кольцевых отверстий: $r_1 = 5,26$; $r_2 = 5,74$; $r_3 = 4,71$; $r_4 = 5,74$; $r_5 = 6,09$; $r_6 = 6,37$.

Расчеты проведены на двух сетках: $N_z \times N_r \times N_\varphi = 706 \times 72 \times 6 + 34 \times 72 \times 192$ (расчет N) и $N_z \times N_r \times N_\varphi = 1416 \times 144 \times 6 + 68 \times 144 \times 320$ (расчет $2N$). Поэтому реальные значения радиусов кольцевых отверстий в расчетах слегка отличались из-за различий в размерах ячеек: в N -расчете $r_1 = 5,28$; $r_2 = 5,74$; $r_3 = 4,67$; $r_4 = 5,66$; $r_5 = 6,04$; $r_6 = 6,35$; в $2N$ -расчете $r_1 = 5,28$; $r_2 = 5,74$; $r_3 = 4,67$; $r_4 = 5,82$; $r_5 = 6,23$; $r_6 = 6,42$.

Внутренняя граница (коллектор, устройство питания, отборник) имеет нулевую скорость, остальные границы вращаются с угловой скоростью $\Omega = 1,1 \cdot 10^4$.

Граничное условие по газовой динамике: на всех границах — условие прилипания. Граничные условия по теплу: для $z = z_1$ $T = T_H = 305$ К; для $z = z_2$ $T = T_B = T_H + \Delta T = 313$ К, $\Delta T = 8$ К; для $r = r_1$ и $r = R_0$ $T(z) = T_H + \Delta T(z - z_1)/L_0$.

Пунктирная линия на рис. 9 указывает место сшивки 2D и 3D областей. Расстояние $\Delta L = L_0 - L_\Gamma$ варьировалось от 0,7 до 2.

Результаты расчетов.

Влияние места сшивки 3D и 2D областей на течение в кольцевом отверстии. Проведены расчеты на грубой сетке с вариацией ΔL . На рис. 10, а для трех различных значений ΔL показаны профили осевой компоненты скорости по отрезку, проходящему через диафрагму между отвальной и разделительной камерами. Именно эта компонента в конечном счете отвечает за режим циркуляции между указанными камерами. Как видно, уже для значений $\Delta L = 1,4$ и $\Delta L = 2$ профили скорости практически совпадают. Аналогичное сравнение профилей логарифма плотности на вер-

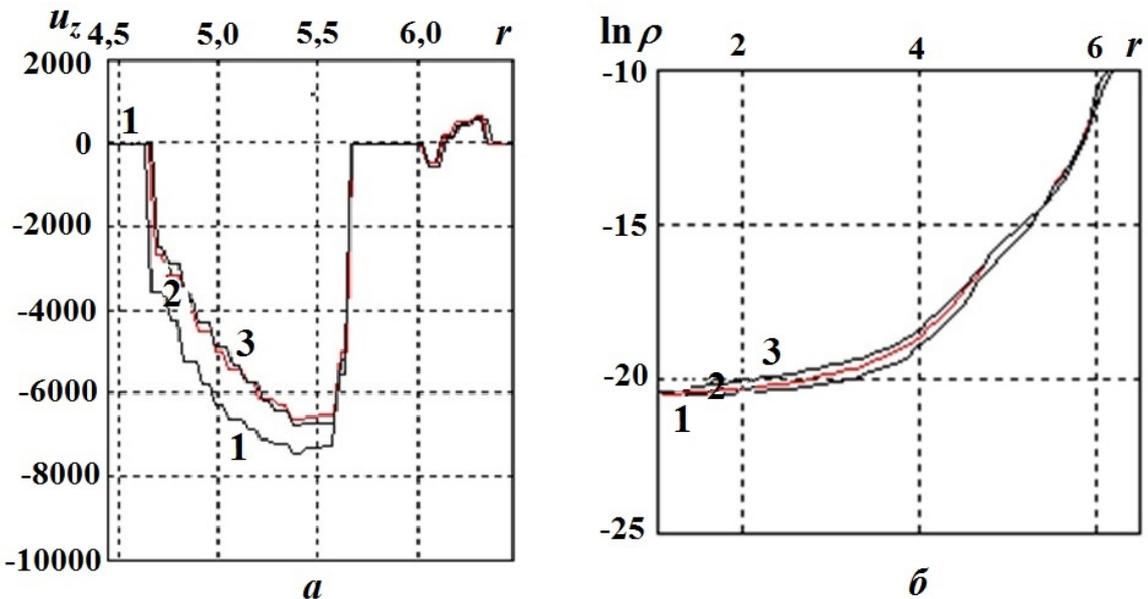


Рис. 10. Тест 5. Профили осевой компоненты скорости (а) и логарифма плотности (б) при $\Delta L = 0,7$ (1); 1,4 (2); 2 (3)

тикальном отрезке с координатой $z = 43,5$ показывает близость этих профилей для всех значений ΔL (рис. 10, б). Таким образом, удовлетворительные результаты получаются уже при $\Delta L = 1,4$.

Расчеты на разных сетках. Распределения температуры в сечении, проходящем через ось симметрии, полученные на грубой и подробной сетках, показаны на рис. 11. Видно, что зависимость результатов от сетки незначительна.

Установление течения. Для расчета на подробной сетке на рис. 12 показаны распределения температуры на три момента времени в сечении, проходящем через ось симметрии. На рис. 13 приводятся распределения температуры и осевой компоненты скорости в сечении, проходящем через центр отборника по нормали к оси симметрии. Отличие максимальных значений величин на моменты времени $t = 0,005$ и $0,01$ не превышает 2%, что свидетельствует о том, что к моменту $t = 0,01$ устанавливается течение, близкое к стационарному.

Изменение массы и энергии. На рис. 14, а показаны относительные изменения массы газа в зависимости от времени в 3D и 2D областях в расчете на подробной сетке. Суммарная в 2D и 3D областях масса к моменту $t = 0,01$ выходит на стационарное (с точностью до пульсаций) значение. Отметим, что в расчетах на разных сетках результаты близки.

Пульсации обусловлены нестационарностью данной задачи, связанной с наличием отборника. Это вносит в решение периодические возмущения с периодом около $0,6 \cdot 10^{-3}$, приблизительно соответствующие (как видно из рис. 14, а) возмущениям массы газа отдельно в каждой из областей — 3D и 2D.

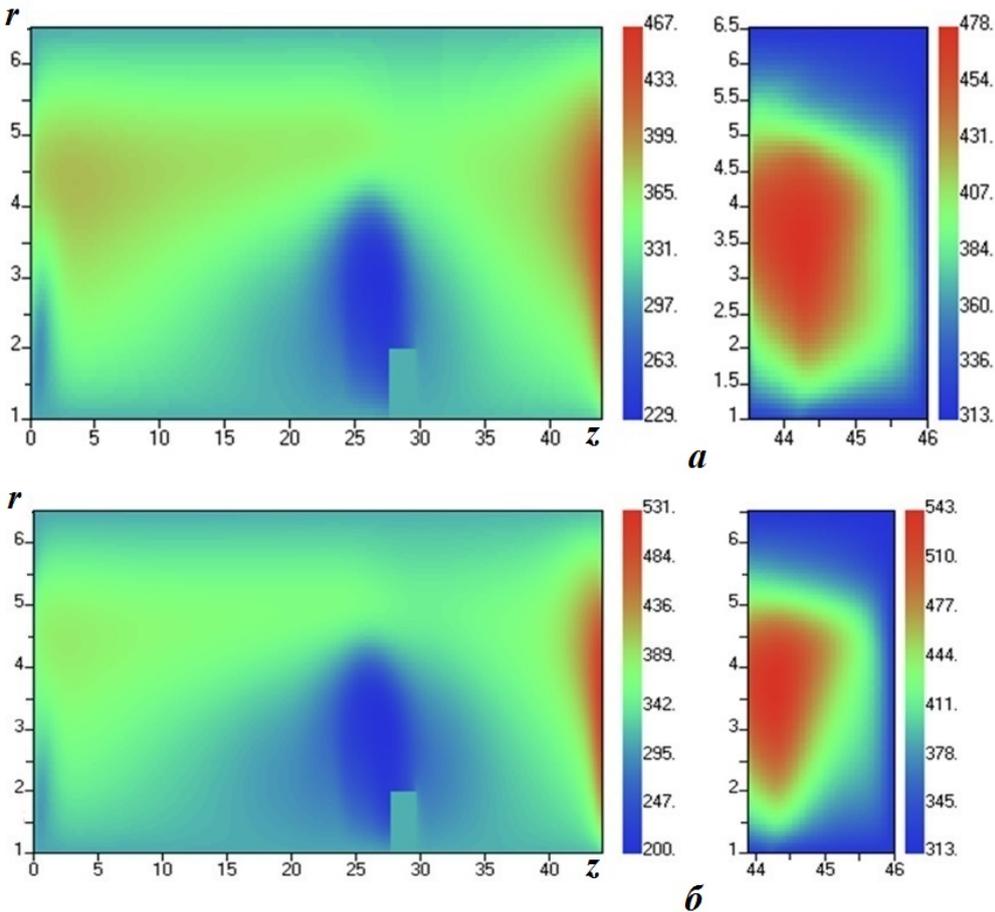


Рис. 11. Тест 5. Распределения температуры, $t = 0,01$: а — грубая сетка; б — подробная сетка; слева — 2D область; справа — 3D область

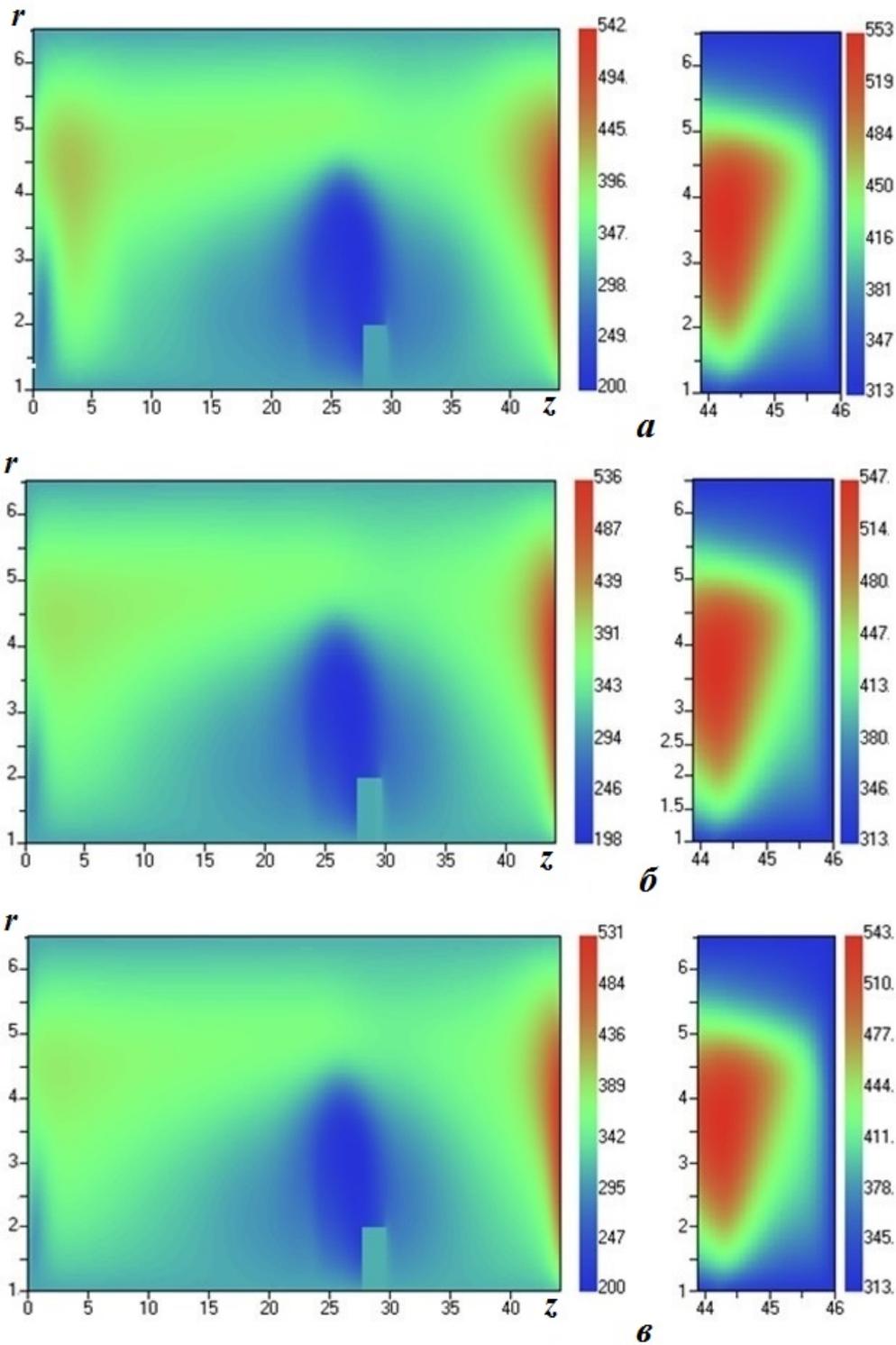


Рис. 12. Тест 5. Распределения температуры: $t = 0,0025$ (а); $0,005$ (б); $0,01$ (в)

Энергия 3D области (рис. 14, б) к моменту $t = 0,01$ выходит на стационарное (с точностью до пульсаций) значение, в то время как энергия 2D области (и соответственно суммарная в областях) продолжает уменьшаться, возвращаясь к начальному значению. Относительные изменения энергии при этом малы — меньше 3%.

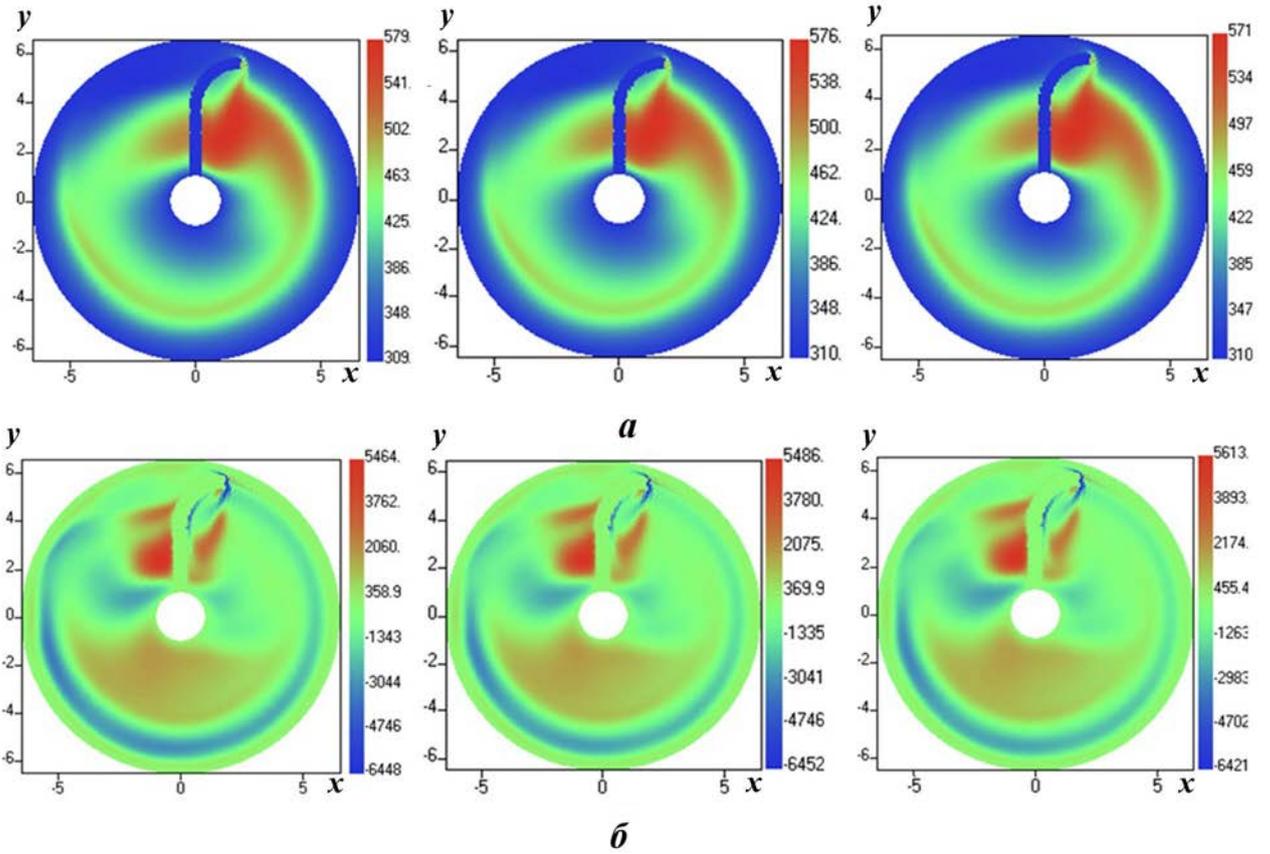


Рис. 13. Тест 5. Поля в сечении $z = \text{const}$: a — температура; b — осевая компонента скорости; слева — $t = 0,0025$; в центре — $t = 0,005$; справа — $t = 0,01$

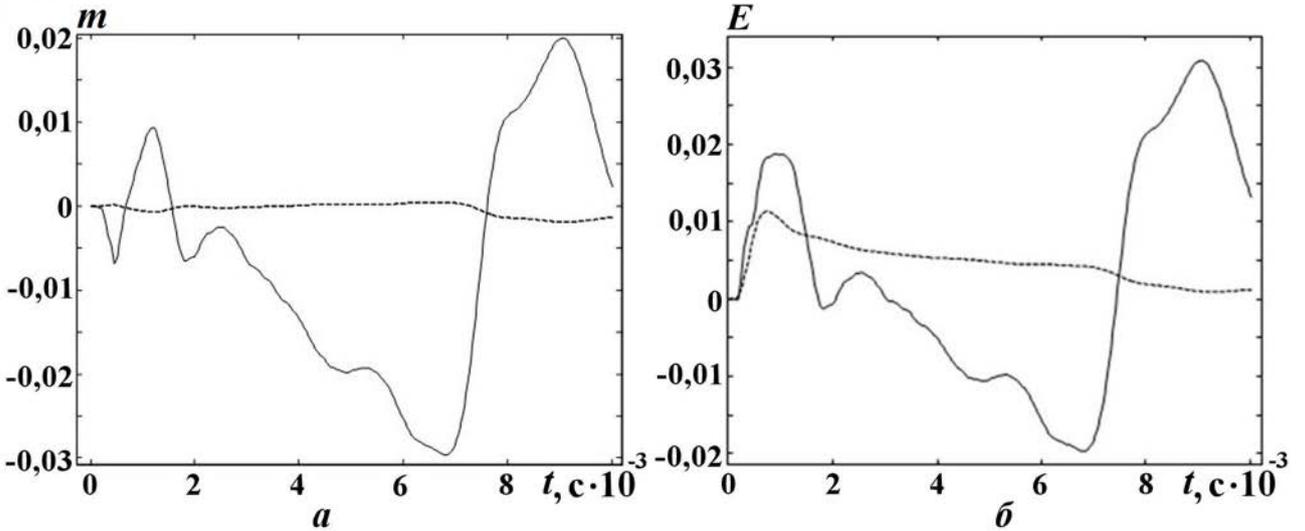


Рис. 14. Тест 5. Зависимости изменения относительной массы (a) и энергии (b) газа от времени в расчете на подробной сетке: — — в 3D области; — в 2D области

Тест 6. Одномерная диффузионная задача

Общая постановка задачи. Скорость j -го компонента газа равна сумме массовой скорости газа как целого \mathbf{u} и диффузионной скорости \mathbf{w}_j , которая для представляющего интерес случая бинарной

смеси удовлетворяет соотношению

$$\mathbf{J} \equiv \rho_1 \mathbf{w}_1 = -\rho_2 \mathbf{w}_2 = -\rho \frac{M_1 M_2}{(\rho/n)^2} D (\mathbf{d}_1 + k_T \nabla \ln T),$$

где M_1, M_2 — молярные массы легкого и тяжелого компонентов; D — коэффициент концентрационной диффузии; \mathbf{d}_j — так называемые диффузионные термодинамические силы:

$$\mathbf{d}_j = \nabla \left(\frac{n_j}{n} \right) + \left(\frac{n_j}{n} - \frac{\rho_j}{\rho} \right) \nabla \ln p \equiv \nabla N_j + (N_j - c_j) \nabla \ln p.$$

Согласно [6] в центрифуге вклад в диффузионный перенос термодиффузии мал, поэтому будем полагать $k_T = 0$. Далее обозначим

$$c = \frac{\rho_1}{\rho}; \quad N \equiv \frac{n_1}{n} = \frac{M_2 c}{M_2 c + M_1 (1 - c)},$$

где c — концентрация (массовая доля) легкого компонента. Тогда имеют место уравнения [6, 8]

$$\rho \left(\frac{\partial c}{\partial t} + \mathbf{u} \nabla c \right) = -\operatorname{div} \mathbf{J}; \quad (10)$$

$$\mathbf{J} = -\rho D (k_p \nabla \ln p + \nabla c), \quad (11)$$

$$k_p = \frac{\Delta M}{M} c (1 - c), \quad M = \frac{M_1 M_2}{(1 - c) M_2 + c M_1},$$

$$M_1 = M ({}^{235}\text{UF}_6) = 349 \text{ г/моль}, \quad M_2 = M ({}^{238}\text{UF}_6) = 352 \text{ г/моль}, \quad \Delta M = 3 \text{ г/моль}.$$

В изотопном приближении коэффициент диффузии (как и другие коэффициенты переноса) одинаков для обоих компонентов и определяется эмпирическим соотношением $\rho D = 1,117 \cdot 10^{-6} T^{0,925}$ [6]. Кроме того, можно принять $M \approx M_2$, тогда $\rho = nM$ не зависит от концентрации. Из граничного условия $\mathbf{J} \cdot \mathbf{n} = 0$ следует

$$k_p \frac{\partial \ln p}{\partial n} + \frac{\partial c}{\partial n} = 0.$$

Постановка и решение изотермической задачи. Рассматривается одномерная изотермическая задача, общая постановка и параметры которой приведены выше. Примем $k_p = 0$. В этом случае уравнения (10) и (11) принимают вид

$$\frac{\partial c}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (r j_r); \quad j_r = -D_\rho \frac{\partial c}{\partial r}.$$

Для данного теста принято также $D_\rho = \text{const} = 10^7 \rho D$.

В начальный момент профиль концентрации имеет вид

$$c(r, t = 0) = \begin{cases} c_0, & r = 1,5; \\ 0, & 1,5 < r < 6,5; \\ c_0, & r = R = 6,5, \end{cases}$$

где $c_0 = 0,5$.

От боковой границы $r = R = 6,5$ концентрация диффундирует близким к плоскому случаю образом. Это позволяет записать решение в виде решения (см. [9]) плоской задачи теплопроводности в случае бесконечного интервала:

$$c(r, t) = c_0 (1 - \operatorname{erf}(z)),$$

$$\text{где } \operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-\xi^2) d\xi, \quad z = \frac{R-r}{2\sqrt{Dt}}.$$

Сравнение результата расчета и аналитического решения для плоского случая показано на рис. 15. Видно, что расхождение возникает лишь на значительном расстоянии от боковой поверхности ротора, где течение все больше отличается от плоского.

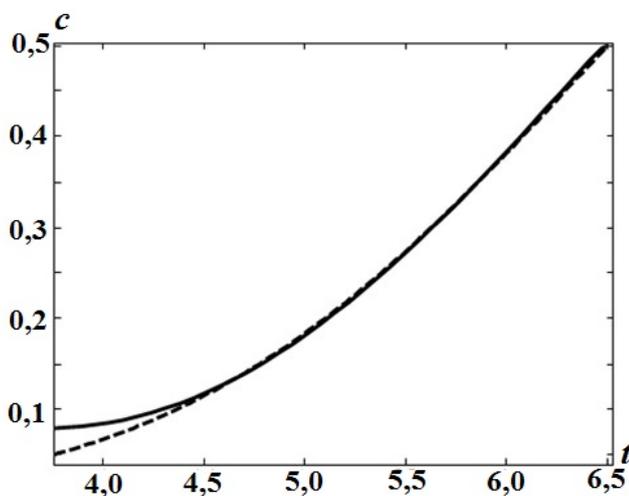


Рис. 15. Тест 6. Профили концентрации: — — расчет; ---- — аналитическое решение

Заключение

По разработанной авторами методике рассчитан ряд тестов, связанных с 3D газодинамическими течениями в однокамерной и двухкамерной моделях центрифуги типа "Игуассу". Данные тесты охватывают практически все режимы работы центрифуги. Для всех тестов получены приемлемые по точности результаты, что вселяет определенные надежды на то, что данная методика может оказаться эффективным средством для сквозного моделирования рассматриваемых течений.

Авторы считают своим приятным долгом выразить благодарность Е. В. Соколовской за большую работу по редакции текстов статей, представляющих обе части работы.

Список литературы

1. Wood H. G., Morton J. B. Onsager's pancake approximation for the fluid dynamics of a gas centrifuge // Fluid Mech. 1980. Vol. 101, part 1. P. 1–31.
2. Звонарёв К. В., Токманцев В. И. Оптимизация газовых потоков в разделительной центрифуге на основе аналитической модели конвекции-диффузии // Сб. докл. VIII Всерос. науч. конф. по физико-химическим процессам при селекции атомов и молекул. Звенигород, 2003. С. 43–47.
Zvonaryev K. V., Tokmantsev V. I. Optimizatsiya gazovykh potokov v razdelitelnoy tsentrifuge na osnove analiticheskoy modeli konveksii-diffuzii // Sb. dokl. VIII Vseros. nauch. konf. po fiziko-khimicheskim protsessam pri seleksii atomov i molekul. Zvenigorod, 2003. S. 43–47.
3. Von Halle E. US DOE Report KOA-4058, 1977.
4. Звонарёв К. В., Селезнёв В. Д., Токманцев В. И., Абрамов Ю. В. Моделирование теплообмена между сверхзвуковым потоком и неподвижным телом в газовой центрифуге // Инженерно-физический журнал. 2012. Т. 85, № 6. С. 1277–1284.
Zvonaryev K. V., Seleznyev V. D., Tokmantsev V. I., Abramov Yu. V. Modelirovanie teploobmena mezhdru sverkhzvukovym potokom i nepodvizhnym telom v gazovoy tsentrifuge // Inzhenerno-fizicheskiy zhurnal. 2012. T. 85, № 6. S. 1277–1284.
5. Янилкин Ю. В., Стаценко В. П., Чернышов Ю. Д., Беляев С. П., Колобянин В. Ю., Топорова О. О., Турутина И. Ю. Численное моделирование трехмерного течения газа в газовой центрифуге. Часть 1. Описание методики // См. настоящий выпуск. С. 3–19.

Yanilkin Yu. V., Statsenko V. P., Chernyshov Yu. D., Belyaev S. P., Kolobyanin V. Yu., Toporova O. O., Turutina I. Yu. Chislennoe modelirovanie trekhmernogo techeniya gaza v gazovoy tsentrifuge. Chast 1. Opisaniye metodiki // Sm. nastoyashchy vypusk. S. 3–19.

6. *Борисевич В. Д., Борман В. Д., Сулаберидзе Г. А., Тихомиров А. В., Токманцев В. И.* Физические основы разделения изотопов в газовой центрифуге. М.: МИФИ, 2005.
Borisevich V. D., Borman V. D., Sulaberidze G. A., Tikhomirov A. V., Tokmantsev V. I. Fizicheskie osnovy razdeleniya izotopov v gazovoy tsentrifuge. М.: MIFI, 2005.
7. Обогащение урана: Пер. с англ. / Под ред. С. Виллани. М.: Энергоатомиздат, 1983.
Obogashchenie urana: Per. s angl. / Pod red. S. Villani. М.: Energoatomizdat, 1983.
8. *Ландау Л. Д., Лифшиц Е. М.* Гидродинамика. М.: Наука, 1986.
Landau L. D., Lifshits E. M. Hidrodinamika. М.: Nauka, 1986.
9. *Корн Г., Корн Т.* Справочник по математике. М.: Наука, 1973.
Korn G., Korn T. Spravochnik po matematike. М.: Nauka, 1973.

Статья поступила в редакцию 15.03.2022.

УДК 519.6

КВАЗИСПЕКТРАЛЬНЫЙ МЕТОД РЕШЕНИЯ КИНЕТИЧЕСКОГО УРАВНЕНИЯ ПЕРЕНОСА ИЗЛУЧЕНИЯ В ДВУМЕРНОЙ ОСЕСИММЕТРИЧНОЙ ПОСТАНОВКЕ

А. Д. Хмельницкая, Н. Г. Карлыханов
(ФГУП "РФЯЦ-ВНИИТФ им. академ. Е. И. Забабахина",
г. Снежинск Челябинской области)

Предлагается квазиспектральный метод для решения уравнения переноса излучения в двумерном осесимметричном случае на прямоугольной сетке. Метод заключается в усреднении уравнения по спектру и угловой переменной и приведении усредненного кинетического уравнения переноса к уравнению лучистой теплопроводности. Достоинство метода в том, что вся сложность итерационного процесса переложена на уравнение лучистой теплопроводности, для которого задача существенно упрощается. Кроме того, метод позволяет организовать распараллеливание по спектральной переменной, а также по пространству для вычисления усредненных по энергии пробегов и эффективной энергии излучения. Показана работоспособность метода на второй задаче Флека, решение сравнивалось со спектральным расчетом в одномерном сферическом случае.

Ключевые слова: квазиспектральный метод, спектральное уравнение переноса излучения, уравнение квазидиффузии, квазиперенос, лучистая теплопроводность, лучевой эффект.

Введение

Данная работа является развитием метода для учета спектральных и кинетических эффектов при расчете задач переноса излучения в приближении лучистой теплопроводности, предложенного и реализованного в одномерном случае [1]. Суть метода заключается в усреднении по энергетической и угловой переменным уравнения переноса излучения и приведении усредненного уравнения к виду, максимально близкому к уравнению лучистой теплопроводности (ЛТ). Решение уравнения ЛТ обладает следующими достоинствами: простота построения монотонных разностных схем второго порядка точности, организация классического ньютоновского итерационного процесса по нелинейности, наличие большого количества рабочих методик, в которых можно применить рассматриваемый подход без существенных изменений в соответствующих программах. Кроме того, построение итерационного процесса, которое при решении нелинейного кинетического уравнения является чрезвычайно сложной задачей, для уравнения типа теплопроводности существенно упрощается. В этой работе метод реализован в двумерном осесимметричном случае в цилиндрической системе координат.

Постановка задачи

Для описания распространения теплового излучения в среде рассмотрим кинетическое уравнение переноса излучения без учета рассеяния в многогрупповом приближении для осесимметричной геометрии в цилиндрической системе координат [2]:

$$\frac{1}{c} \frac{\partial I_\varepsilon}{\partial t} + \xi \frac{\partial I_\varepsilon}{\partial r} + \mu \frac{\partial I_\varepsilon}{\partial z} - \frac{\eta}{r} \frac{\partial I_\varepsilon}{\partial \psi} + \alpha_\varepsilon I_\varepsilon = \alpha_\varepsilon I_{er}. \quad (1)$$

Здесь t — время; z, r — цилиндрические координаты (OZ — ось симметрии); $I_\varepsilon = \int_{\varepsilon_{g-1/2}}^{\varepsilon_{g+1/2}} I(\varepsilon, t, r, z, \mu, \psi) d\varepsilon$;

$I = I(\varepsilon, t, r, z, \mu, \psi)$ — интенсивность излучения фотонов с энергией ε , летящих в направлении $\vec{\Omega}(\mu, \xi, \eta)$, где $\mu = \cos \theta$ ($\mu \in [-1; 1]$); θ — угол между направлением $\vec{\Omega}$ и осью OZ , $\theta \in [0; \pi]$; $\xi = \sqrt{1 - \mu^2} \cos \psi$, $\eta = \sqrt{1 - \mu^2} \sin \psi$, где ψ — угол между проекцией $\vec{\Omega}$ на плоскость OXY и вектором \vec{r} , $\psi \in [0; 2\pi]$; $\alpha_\varepsilon(T)$ — коэффициент поглощения фотонов энергии ε при температуре среды T ; внешний источник излучения определяется равновесным излучением с интенсивностью

$$I_{er} = \frac{c}{4\pi} B_\varepsilon(T), \quad B_\varepsilon(T) = \frac{8\pi}{c^2 h^3} \frac{\varepsilon^3 \exp(-\varepsilon/T)}{1 - \exp(-\varepsilon/T)},$$

где h — постоянная Планка, c — скорость света.

Уравнение (1) решается совместно с уравнением баланса энергии для определения пространственно-временного распределения температуры среды $T(t, r, z)$:

$$\frac{\partial E(T)}{\partial t} = c \int \alpha_\varepsilon(T) (U_\varepsilon - B_\varepsilon(T)) d\varepsilon, \quad (2)$$

где $E(T)$ — внутренняя энергия вещества; $U_\varepsilon = \frac{1}{c} \int_{-1}^1 \int_0^{2\pi} I_\varepsilon d\psi d\mu$ — спектральная плотность энергии излучения.

Уравнение (1) дополняется граничными условиями:

- на границе $r = 0$ — условие *зеркального отражения*: $I(t, r = 0, z, \mu, \psi) = I(t, r = 0, z, \mu, \psi + \pi)$;
- на остальных границах — условие *свободной границы*: интенсивность с направлением внутрь системы равна нулю.

Условие периодичности по углу ψ определяется из уравнения (1) при подстановке в него $\psi = \pi$.

Для решения системы (1), (2) используем следующий алгоритм. На каждом временном шаге при известной температуре решается кинетическое уравнение (1) по схеме первого порядка. Используя полученное решение для интенсивности, вычисляем коэффициенты связи между исходным уравнением переноса и уравнением, усредненным по угловым переменным. Решаем усредненное уравнение по схеме второго порядка точности, затем по полученному решению находим коэффициенты, усредненные по энергии. После этого решаем систему нелинейных уравнений в приближении лучистой теплопроводности.

Усреднение по угловой переменной

Если проинтегрировать уравнение (1) трижды, сначала с весом 1, а затем с весами μ, ξ , то получим систему уравнений квазидиффузии (КД) [2]:

$$\begin{aligned} \frac{dU_\varepsilon}{dt} + \frac{\partial}{\partial r} S_{r\varepsilon} + \frac{\partial}{\partial z} S_{z\varepsilon} + \alpha_\varepsilon c U_\varepsilon &= c \alpha_\varepsilon B_\varepsilon; \\ \frac{1}{c^2} \frac{d}{dt} S_{r\varepsilon} + \frac{\partial}{\partial r} (D_{1\varepsilon} U_\varepsilon) + \frac{\partial}{\partial z} (D_{3\varepsilon} U_\varepsilon) + P_{r\varepsilon} U_\varepsilon + \frac{\alpha_\varepsilon}{c} S_{r\varepsilon} &= 0; \\ \frac{1}{c^2} \frac{d}{dt} S_{z\varepsilon} + \frac{\partial}{\partial r} (D_{2\varepsilon} U_\varepsilon) + \frac{\partial}{\partial z} (D_{3\varepsilon} U_\varepsilon) + P_{z\varepsilon} U_\varepsilon + \frac{\alpha_\varepsilon}{c} S_{z\varepsilon} &= 0. \end{aligned} \quad (3)$$

Здесь приняты обозначения: $S_{r\varepsilon} = \int_{-1}^1 \int_0^{2\pi} I_\varepsilon \xi d\psi d\mu$, $S_{z\varepsilon} = \int_{-1}^1 \int_0^{2\pi} I_\varepsilon \mu d\psi d\mu$ — компоненты спектрального потока энергии излучения; $D_{1\varepsilon} = (cU_\varepsilon)^{-1} \int_{-1}^1 \int_0^{2\pi} I_\varepsilon \xi^2 d\psi d\mu$, $D_{2\varepsilon} = (cU_\varepsilon)^{-1} \int_{-1}^1 \int_0^{2\pi} I_\varepsilon \mu^2 d\psi d\mu$, $D_{3\varepsilon} =$

$$= (cU_\varepsilon)^{-1} \int_{-1}^1 \int_0^{2\pi} I_\varepsilon \xi \mu d\psi d\mu, \quad P_{z\varepsilon} = \frac{1}{r} D_{3\varepsilon}, \quad P_{r\varepsilon} = \frac{1}{r} (2D_{1\varepsilon} + D_{2\varepsilon} - 1) — коэффициенты КД.$$

Из уравнений для потоков (второе и третье уравнения системы (3)) получим

$$\frac{1}{3} \frac{\partial U_\varepsilon}{\partial r} = -\frac{\alpha_{r\varepsilon}^{kin} + \widetilde{\alpha}_{r\varepsilon}}{c} S_{r\varepsilon}; \quad \frac{1}{3} \frac{\partial U_\varepsilon}{\partial z} = -\frac{\alpha_{z\varepsilon}^{kin} + \widetilde{\alpha}_{z\varepsilon}}{c} S_{z\varepsilon}. \quad (4)$$

Здесь аналогично работе [3] введен кинетический коэффициент с компонентами

$$\alpha_{r\varepsilon}^{kin} = \frac{c}{S_{r\varepsilon}} \left\{ \frac{1}{c^2} \frac{dS_{r\varepsilon}}{dt} + \frac{\partial}{\partial r} \left[\left(D_{1\varepsilon} - \frac{1}{3} \right) U_\varepsilon \right] + \frac{\partial(D_{3\varepsilon}U_\varepsilon)}{\partial z} + P_{r\varepsilon}U_\varepsilon \right\}, \quad S_{r\varepsilon} \ll cU \rightarrow \alpha_{r\varepsilon}^{kin} = 0;$$

$$\alpha_{z\varepsilon}^{kin} = \frac{c}{S_{z\varepsilon}} \left\{ \frac{1}{c^2} \frac{dS_{z\varepsilon}}{dt} + \frac{\partial}{\partial z} \left[\left(D_{2\varepsilon} - \frac{1}{3} \right) U_\varepsilon \right] + \frac{\partial(D_{3\varepsilon}U_\varepsilon)}{\partial r} + P_{z\varepsilon}U_\varepsilon \right\}, \quad S_{z\varepsilon} \ll cU \rightarrow \alpha_{z\varepsilon}^{kin} = 0;$$

$\widetilde{\alpha}_\varepsilon = (\widetilde{\alpha}_{r\varepsilon}, \widetilde{\alpha}_{z\varepsilon})$ — коэффициент поглощения на грани ячейки.

Объединив первое уравнение системы (3) и уравнения (4), получим систему уравнений квазипереноса (КП) [3]:

$$\begin{aligned} \frac{dU_\varepsilon}{dt} + \frac{\partial}{\partial r} S_{r\varepsilon} + \frac{\partial}{\partial z} S_{z\varepsilon} + \alpha_\varepsilon c U_\varepsilon &= c \alpha_\varepsilon B_\varepsilon; \\ S_{r\varepsilon} &= -\frac{ck_{r\varepsilon}}{3} \frac{\partial U_\varepsilon}{\partial r}; \\ S_{z\varepsilon} &= -\frac{ck_{z\varepsilon}}{3} \frac{\partial U_\varepsilon}{\partial z}, \end{aligned} \quad (5)$$

где $k_{r\varepsilon} = (\widetilde{\alpha}_{r\varepsilon} + \alpha_{r\varepsilon}^{kin})^{-1}$, $k_{z\varepsilon} = (\widetilde{\alpha}_{z\varepsilon} + \alpha_{z\varepsilon}^{kin})^{-1}$ имеют единицы измерения пробега, поэтому будем называть данные коэффициенты *пробегом*. Введем ограничение на кинетический коэффициент: $\alpha_{r\varepsilon}^{kin} < 0 \rightarrow k_{r\varepsilon}^{-1} = 0,1\widetilde{\alpha}_{r\varepsilon}$; $\alpha_{z\varepsilon}^{kin} < 0 \rightarrow k_{z\varepsilon}^{-1} = 0,1\widetilde{\alpha}_{z\varepsilon}$. Тогда пробег является неотрицательным, а система уравнений (5) — системой параболического типа, для которой имеются монотонные схемы второго порядка точности.

Алгоритм построения усредненного по энергии уравнения

С помощью уравнений (4) исключим поток из первого уравнения системы (3). Получим

$$\frac{\partial U_\varepsilon}{\partial t} + \frac{\partial}{\partial r} \left[\frac{c}{3(\alpha_{r\varepsilon}^{kin} + \widetilde{\alpha}_{r\varepsilon})} \frac{\partial U_\varepsilon}{\partial r} \right] + \frac{\partial}{\partial z} \left[\frac{c}{3(\alpha_{z\varepsilon}^{kin} + \widetilde{\alpha}_{z\varepsilon})} \frac{\partial U_\varepsilon}{\partial z} \right] = c \alpha_\varepsilon B_\varepsilon - \alpha_\varepsilon c U_\varepsilon. \quad (6)$$

На сетке по пространству определим температуру вещества и плотность энергии излучения в центрах ячеек, а потоки — на гранях. Тогда уравнение (6) в разностном виде запишется как

$$\begin{aligned} \frac{U_{\varepsilon_{i,j}}^{n+1} - U_{\varepsilon_{i,j}}^n}{\tau} - \frac{c}{3(r_i - r_{i-1})} \left[\left(\frac{U_{\varepsilon_{i+1,j}}^{n+1} - U_{\varepsilon_{i,j}}^{n+1}}{0,5(r_{i+1} - r_{i-1})} \right) k_{r\varepsilon_{i,j}}^{n+1} - \left(\frac{U_{\varepsilon_{i,j}}^{n+1} - U_{\varepsilon_{i-1,j}}^{n+1}}{0,5(r_i - r_{i-2})} \right) k_{r\varepsilon_{i-1,j}}^{n+1} \right] - \\ - \frac{c}{3(z_j - z_{j-1})} \left[\left(\frac{U_{\varepsilon_{i,j+1}}^{n+1} - U_{\varepsilon_{i,j}}^{n+1}}{0,5(z_{j+1} - z_{j-1})} \right) k_{z\varepsilon_{i,j}}^{n+1} - \left(\frac{U_{\varepsilon_{i,j}}^{n+1} - U_{\varepsilon_{i,j-1}}^{n+1}}{0,5(z_j - z_{j-2})} \right) k_{z\varepsilon_{i,j-1}}^{n+1} \right] = \\ = -c \alpha_{\varepsilon_{i,j}} U_{\varepsilon_{i,j}}^{n+1} + c \alpha_{\varepsilon_{i,j}} B_{\varepsilon_{i,j}}^{n+1}. \end{aligned}$$

Значения величин $U_{\varepsilon_{i+1,j}}^{n+1}$, $U_{\varepsilon_{i,j-1}}^{n+1}$, $U_{\varepsilon_{i,j+1}}^{n+1}$, $U_{\varepsilon_{i-1,j}}^{n+1}$ будем брать из решения системы КП (5). Тогда можно выразить значение $U_{\varepsilon_{i,j}}^{n+1}$ через $B_{\varepsilon_{i,j}}^{n+1}$:

$$U_{\varepsilon_{i,j}}^{n+1} = f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} + g_{\varepsilon_{i,j}}^p. \quad (7)$$

Здесь $f_{\varepsilon i,j}^p = \frac{c\tau\alpha_{\varepsilon i,j}^n}{L}$; $g_{\varepsilon i,j}^p = \frac{M}{L}$, где

$$L = 1 + \frac{c\tau}{3(r_i - r_{i-1})} \left[\frac{k_{r\varepsilon i,j}^n}{0,5(r_{i+1} - r_{i-1})} + \frac{k_{r\varepsilon i-1,j}^n}{0,5(r_i - r_{i-2})} \right] + \frac{c\tau}{3(z_j - z_{j-1})} \left[\frac{k_{z\varepsilon i,j}^n}{0,5(z_{j+1} - z_{j-1})} + \frac{k_{z\varepsilon i,j-1}^n}{0,5(z_j - z_{j-2})} \right] + c\tau\alpha_{\varepsilon i,j}^n;$$

$$M = U_{\varepsilon i,j}^n + U_{\varepsilon i+1,j}^p \frac{c\tau}{3(r_i - r_{i-1})} \frac{k_{r\varepsilon i,j}^n}{0,5(r_{i+1} - r_{i-1})} + U_{\varepsilon i-1,j}^p \frac{c\tau}{3(r_i - r_{i-1})} \frac{k_{r\varepsilon i-1,j}^n}{0,5(r_i - r_{i-2})} + U_{\varepsilon i,j+1}^p \times$$

$$\times \frac{c\tau}{3(z_j - z_{j-1})} \frac{k_{z\varepsilon i,j}^n}{0,5(z_{j+1} - z_{j-1})} + U_{\varepsilon i,j-1}^p \frac{c\tau}{3(z_j - z_{j-1})} \frac{k_{z\varepsilon i,j-1}^n}{0,5(z_j - z_{j-2})};$$

индекс p является признаком того, что значение рассчитано из решения КП. Очевидно, что если $k_{r\varepsilon} > 0$, $k_{z\varepsilon} > 0$, то

$$0 \leq f_{\varepsilon i,j}^p < 1; \quad g_{\varepsilon i,j}^p > 0. \quad (8)$$

Запишем уравнение баланса (2) в разностном виде, заменив плотность энергии излучения на выражение (7):

$$\frac{E_{i,j}^{n+1} - E_{i,j}^n}{\tau} = c \int_{\varepsilon} \alpha_{\varepsilon i,j}^{n+1} \left[\left(f_{\varepsilon i,j}^p - 1 \right) B_{\varepsilon i,j}^{n+1} + g_{\varepsilon i,j}^p \right] d\varepsilon.$$

После преобразования получим

$$E_{i,j}^{n+1} - E_{i,j}^n = -c\tau \int_{\varepsilon} \alpha_{\varepsilon i,j}^{n+1} \left[\left(1 - f_{\varepsilon i,j}^p \right) B_{\varepsilon i,j}^{n+1} \right] d\varepsilon + c\tau \frac{\int_{\varepsilon} \alpha_{\varepsilon i,j}^{n+1} g_{\varepsilon i,j}^p d\varepsilon}{\int_{\varepsilon} g_{\varepsilon i,j}^p d\varepsilon} \int_{\varepsilon} g_{\varepsilon i,j}^{n+1} d\varepsilon$$

и введем новые обозначения:

$$E_{i,j}^{n+1} - E_{i,j}^n = -c\tau \langle w_{i,j}^{n+1} \rangle + c\tau \langle w_{g i,j}^{n+1} \rangle G_{i,j}^{n+1}; \quad (9)$$

$$\langle w_{i,j}^{n+1} \rangle = \int_{\varepsilon} \alpha_{\varepsilon i,j}^{n+1} \left(1 - f_{\varepsilon i,j}^p \right) B_{\varepsilon i,j}^{n+1} d\varepsilon; \quad (10)$$

$$\langle w_{g i,j}^{n+1} \rangle = \frac{\int_{\varepsilon} \alpha_{\varepsilon i,j}^{n+1} g_{\varepsilon i,j}^p d\varepsilon}{\int_{\varepsilon} g_{\varepsilon i,j}^p d\varepsilon}. \quad (11)$$

Выразим из уравнения (9) функцию G — некую эффективную плотность энергии излучения:

$$G_{i,j}^{n+1} = \frac{E_{i,j}^{n+1} - E_{i,j}^n + c\tau \langle w_{i,j}^{n+1} \rangle}{c\tau \langle w_{g i,j}^{n+1} \rangle}. \quad (12)$$

Отметим, что усредненные пробеги $\langle w_{i,j}^{n+1} \rangle > 0$, $\langle w_{g i,j}^{n+1} \rangle > 0$ (в силу условий (8)).

Рассмотрим предельные случаи для $\langle w_{i,j}^{n+1} \rangle = \int_{\varepsilon} w_{i,j}^{n+1} d\varepsilon$; $\langle w_{g i,j}^{n+1} \rangle = \frac{\int_{\varepsilon} w_{g i,j}^{n+1} d\varepsilon}{\int_{\varepsilon} g_{\varepsilon i,j}^p d\varepsilon}$.

1. В случае оптически плотной среды ($\alpha_{\varepsilon} \rightarrow \infty$) имеем $g_{\varepsilon} \xrightarrow[\alpha_{\varepsilon} \rightarrow \infty]{k_{\varepsilon} \rightarrow 0} \frac{U_{\varepsilon i,j}^n}{1 + c\tau\alpha_{\varepsilon i,j}^n} \rightarrow 0$ и $f_{\varepsilon} \xrightarrow[\alpha_{\varepsilon} \rightarrow \infty]{} \frac{c\tau\alpha_{\varepsilon}}{1 + c\tau\alpha_{\varepsilon}} \rightarrow 1$, следовательно

$$w_{i,j}^{n+1} \xrightarrow[\alpha_{\varepsilon} \rightarrow \infty]{f_{\varepsilon} \rightarrow 1} \alpha_{\varepsilon} \left(1 - \frac{c\tau\alpha_{\varepsilon}}{1 + c\tau\alpha_{\varepsilon}} \right) B_{\varepsilon i,j}^n = \frac{B_{\varepsilon i,j}^n}{1/\alpha_{\varepsilon} + c\tau} = \frac{B_{\varepsilon i,j}^n}{c\tau};$$

$$w_{g i,j}^{n+1} \xrightarrow[\alpha_{\varepsilon} \rightarrow \infty]{g_{\varepsilon} \rightarrow 0} \alpha_{\varepsilon}^{n+1} g_{\varepsilon i,j}^p = \frac{\alpha_{\varepsilon i,j}^{n+1} U_{\varepsilon i,j}^n}{1 + c\tau\alpha_{\varepsilon i,j}^n} = \frac{U_{\varepsilon i,j}^n}{c\tau}.$$

2. Если $\alpha_\varepsilon \rightarrow 0$, то $w_{g_{i,j}}^{n+1} \xrightarrow{\alpha_\varepsilon \rightarrow 0} 0$, $w_{i,j}^{n+1} \xrightarrow{\alpha_\varepsilon \rightarrow 0} 0$. Во избежание деления на ноль ($\langle w_{g_{i,j}}^{n+1} \rangle = 0$) в (11) положим $c\tau \langle w_{g_{i,j}}^{n+1} \rangle = 10^{-5}$.

Если уравнение (6) проинтегрировать по энергии и сложить с уравнением баланса (2), то получим следующее уравнение:

$$\frac{\partial}{\partial t} (E(T) + U) + \frac{\partial S_r}{\partial r} + \frac{\partial S_z}{\partial z} = 0, \quad (13)$$

где $U = \int_{-\infty}^{\infty} U_\varepsilon d\varepsilon$; $S_r = \int_{-\infty}^{\infty} S_{r\varepsilon} d\varepsilon$; $S_z = \int_{-\infty}^{\infty} S_{z\varepsilon} d\varepsilon$. Разностный аналог уравнения (13) имеет следующий вид:

$$E_{i,j}^{n+1} - E_{i,j}^n + \int_{\varepsilon} U_{\varepsilon_{i,j}}^{n+1} d\varepsilon - \int_{\varepsilon} U_{\varepsilon_{i,j}}^n d\varepsilon + \tau \frac{\int_{\varepsilon} S_{1\varepsilon_{i,j}}^{n+1} d\varepsilon - \int_{\varepsilon} S_{1\varepsilon_{i-1,j}}^{n+1} d\varepsilon}{r_i - r_{i-1}} + \tau \frac{\int_{\varepsilon} S_{2\varepsilon_{i,j}}^{n+1} d\varepsilon - \int_{\varepsilon} S_{2\varepsilon_{i,j-1}}^{n+1} d\varepsilon}{z_j - z_{j-1}} = 0.$$

Заменяем плотность энергии излучения $U_{\varepsilon_{i,j}}^{n+1}$ формулой (7) и воспользуемся формулой (12):

$$E_{i,j}^{n+1} - E_{i,j}^n + \int_{\varepsilon} f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} d\varepsilon + \frac{E_{i,j}^{n+1} - E_{i,j}^n + c\tau \langle w_{i,j}^{n+1} \rangle}{c\tau \langle w_{g_{i,j}}^{n+1} \rangle} - U_{i,j}^n + \tau \frac{S_{1i,j}^{n+1} - S_{1i-1,j}^{n+1}}{r_i - r_{i-1}} + \tau \frac{S_{2i,j}^{n+1} - S_{2i,j-1}^{n+1}}{z_j - z_{j-1}} = 0,$$

где $U_{i,j}^n = \int_{\varepsilon} U_{\varepsilon_{i,j}}^n d\varepsilon$.

Проинтегрируем уравнения (4) по спектру с заменой плотности энергии излучения по формуле (7):

$$\begin{aligned} S_{r\varepsilon_{i,j}}^{n+1} &= -\frac{c}{3} \frac{f_{\varepsilon_{i+1,j}}^p B_{\varepsilon_{i+1,j}}^{n+1} + g_{\varepsilon_{i+1,j}}^p - f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} - g_{\varepsilon_{i,j}}^p}{0,5(r_{i+1} - r_{i-1})} k_{r\varepsilon_{i,j}}^{n+1}, \\ S_{z\varepsilon_{i,j}}^{n+1} &= -\frac{c}{3} \frac{f_{\varepsilon_{i,j+1}}^p B_{\varepsilon_{i,j+1}}^{n+1} + g_{\varepsilon_{i,j+1}}^p - f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} - g_{\varepsilon_{i,j}}^p}{0,5(z_{j+1} - z_{j-1})} k_{z\varepsilon_{i,j}}^{n+1}. \end{aligned} \quad (14)$$

Уравнения (14) можно переписать в следующем виде:

$$S_{r_{i,j}}^{n+1} = -\frac{c}{3 \cdot 0,5(r_{i+1} - r_{i-1})} (X_{r_{i+1,j}}^{n+1} - X_{r_{i,j}}^{n+1}); \quad S_{z_{i,j}}^{n+1} = -\frac{c}{3 \cdot 0,5(z_{j+1} - z_{j-1})} (X_{z_{i+1,j}}^{n+1} - X_{z_{i,j}}^{n+1}). \quad (15)$$

Здесь введены обозначения:

$$\begin{aligned} X_{r_{i+1,j}}^{n+1} &= N_{Tr_{i+1,j}}^{n+1} + N_{Gr_{i+1,j}}^{n+1} G_{i+1,j}^{n+1}; & X_{z_{i,j+1}}^{n+1} &= N_{Tz_{i,j+1}}^{n+1} + N_{Gz_{i,j+1}}^{n+1} G_{i,j+1}^{n+1}; \\ X_{r_{i,j}}^{n+1} &= N_{Tr_{i,j}}^{n+1} + N_{Gr_{i,j}}^{n+1} G_{i,j}^{n+1}; & X_{z_{i,j}}^{n+1} &= N_{Tz_{i,j}}^{n+1} + N_{Gz_{i,j}}^{n+1} G_{i,j}^{n+1}; \\ N_{Tr_{i+1,j}}^{n+1} &= \int k_{r\varepsilon_{i+1,j}}^{n+1} f_{\varepsilon_{i+1,j}}^p B_{\varepsilon_{i+1,j}}^{n+1} d\varepsilon; & N_{Tz_{i,j+1}}^{n+1} &= \int k_{z\varepsilon_{i,j+1}}^{n+1} f_{\varepsilon_{i,j+1}}^p B_{\varepsilon_{i,j+1}}^{n+1} d\varepsilon; \\ N_{Tr_{i,j}}^{n+1} &= \int k_{r\varepsilon_{i,j}}^{n+1} f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} d\varepsilon; & N_{Tz_{i,j}}^{n+1} &= \int k_{z\varepsilon_{i,j}}^{n+1} f_{\varepsilon_{i,j}}^p B_{\varepsilon_{i,j}}^{n+1} d\varepsilon; \\ N_{Gr_{i+1,j}}^{n+1} &= \frac{\int k_{r\varepsilon_{i+1,j}}^{n+1} g_{\varepsilon_{i+1,j}}^p d\varepsilon}{\int g_{\varepsilon_{i+1,j}}^p d\varepsilon}; & N_{Gz_{i,j+1}}^{n+1} &= \frac{\int k_{z\varepsilon_{i,j+1}}^{n+1} g_{\varepsilon_{i,j+1}}^p d\varepsilon}{\int g_{\varepsilon_{i,j+1}}^p d\varepsilon}; \\ N_{Gr_{i,j}}^{n+1} &= \frac{\int k_{r\varepsilon_{i,j}}^{n+1} g_{\varepsilon_{i,j}}^p d\varepsilon}{\int g_{\varepsilon_{i,j}}^p d\varepsilon}; & N_{Gz_{i,j}}^{n+1} &= \frac{\int k_{z\varepsilon_{i,j}}^{n+1} g_{\varepsilon_{i,j}}^p d\varepsilon}{\int g_{\varepsilon_{i,j}}^p d\varepsilon}. \end{aligned} \quad (16)$$

Таким образом, получаем нелинейное уравнение относительно температуры вещества, которое близко к уравнению ЛТ:

$$\tilde{E}_{i,j}^{n+1} - \tilde{E}_{i,j}^n + \tau \frac{S_{r_{i,j}}^{n+1} - S_{r_{i-1,j}}^{n+1}}{r_i - r_{i-1}} + \tau \frac{S_{z_{i,j}}^{n+1} - S_{z_{i,j-1}}^{n+1}}{z_j - z_{j-1}} = 0, \quad (17)$$

где

$$\tilde{E}_{i,j}^{n+1} = E_{i,j}^{n+1} + \int_{\varepsilon} f_{\varepsilon,i,j}^p B_{\varepsilon,i,j}^{n+1} d\varepsilon + \frac{E_{i,j}^{n+1} - E_{i,j}^n + c\tau \langle w_{i,j}^{n+1} \rangle}{c\tau \langle w_{g,i,j}^{n+1} \rangle}; \quad \tilde{E}_{i,j}^n = E_{i,j}^n + U_{i,j}^n;$$

$S_{r_{i,j}}^{n+1}$ и $S_{z_{i,j}}^{n+1}$ определяются по формулам (15), (16).

Для согласования граничных условий в кинетическом и рассматриваемом подходах воспользуемся приемом из работы [3]. Запишем поток на границе в виде двух слагаемых, введя мнимые ячейки за границей области и определив на них интенсивность ($I_{\text{ГР}}$):

$$S_{\text{ГР}}^{n+1} = \int_{\Omega} I^{\text{kin}} \mu d\Omega = \int_{\Omega>0} I_{\text{ВН}}^{\text{kin}} \mu d\Omega + \int_{\Omega<0} I_{\text{ГР}}^{\text{kin}} \mu d\Omega = \int_{\Omega<0} I_{\text{ГР}}^{\text{kin}} \mu d\Omega + \frac{\int_{\Omega>0} I_{\text{ВН}}^{\text{kin}} \mu d\Omega}{\frac{1}{c} \int_{\Omega} I^{\text{kin}} d\Omega} U_{\text{ГР}}^{n+1},$$

где индекс *kin* является признаком того, что I берется из решения кинетического уравнения.

Первое слагаемое полностью определяется заданной граничной интенсивностью излучения, а второе преобразовано в виде дроби. Введя обозначения $a_s = 1$; $a_u = -\frac{\int_{\Omega>0} I_{\text{ВН}}^{\text{kin}} \mu d\Omega}{\frac{1}{c} \int_{\Omega} I^{\text{kin}} d\Omega}$; $a = \frac{\int_{\Omega>0} I_{\text{ГР}}^{\text{kin}} \mu d\Omega}{\frac{1}{c} \int_{\Omega} I^{\text{kin}} d\Omega}$,

получим граничное условие для КД:

$$a_u U_{\varepsilon_{\text{ГР}}}^{n+1} + a_s S_{\varepsilon_{\text{ГР}}}^{n+1} = a,$$

где $U_{\varepsilon_{\text{ГР}}}^{n+1}$ относится к мнимой ячейке.

Чтобы реализовать граничное условие для системы уравнений в приближении ЛТ (17), необходимо исключить мнимую ячейку с помощью выражения $S^{n+1} = -\frac{ck_{\varepsilon} U_{\text{ГР}}^{n+1} - U_{\text{ВН}}^{n+1}}{3h}$:

$$S_{\varepsilon_{\text{ГР}}}^{n+1} + \tilde{a}_T^{n+1} U_{\varepsilon_{\text{ВН}}}^{n+1} = \tilde{a}^{n+1},$$

где $\tilde{a}_T^{n+1} = \mp \frac{ck_{\varepsilon}^{n+1}}{3h \mp cb_s k_{\varepsilon}^{n+1}}$, $b_s = \frac{a_s}{a_u}$; $\tilde{a}^{n+1} = \tilde{a}_T^{n+1} b$, $b = \frac{a}{a_u}$.

Подставив вместо плотности энергии излучения формулу (7) и проинтегрировав по спектру, получим

$$S_{\varepsilon_{\text{ГР}}}^{n+1} + \tilde{a}_T^{n+1} U_{\varepsilon_{\text{ВН}}}^{n+1} = \tilde{a}_T^{n+1} b; \quad S_{\text{ГР}}^{n+1} + \int \tilde{a}_T f^p B_{\text{ВН}}^{n+1} d\varepsilon + \frac{\int \tilde{a}_T g^p d\varepsilon}{\int g^p d\varepsilon} G_{\text{ВН}}^{n+1} = b \int \tilde{a}_T^{n+1} d\varepsilon.$$

На остальных границах условие согласуется аналогичным образом.

Еще раз кратко опишем схему метода для одного шага по времени:

- 1) для каждой группы по энергии:
 - решаем линейное уравнение переноса (1);
 - находим коэффициенты КД;
 - вычисляем кинетический коэффициент из уравнений (4);
 - решаем систему уравнений КП (5);
 - находим коэффициенты связи (7);
- 2) решаем нелинейную систему уравнений (17):
 - вычисляем интегралы по спектру, коэффициенты (10)–(12);
 - линеаризуем систему (17), (15) и решаем систему линейных алгебраических уравнений.

Линеаризация разностных уравнений

Для решения системы (17), (15) использовался метод Ньютона:

$$\begin{aligned} \tilde{E}_{i,j}^s + \frac{\partial \tilde{E}}{\partial T} \Big|_{T_{i,j}^s} \Delta T_{i,j}^{s+1} - \tilde{E}_{i,j}^n + \frac{\tau}{r_i - r_{i-1}} \left(S_{r_{i,j}}^s + \Delta S_{r_{i,j}}^{s+1} - S_{r_{i-1,j}}^s - \Delta S_{r_{i-1,j}}^{s+1} \right) + \\ + \frac{\tau}{z_j - z_{j-1}} \left(S_{z_{i,j}}^s + \Delta S_{z_{i,j}}^{s+1} - S_{z_{i,j-1}}^{n+1} \right) = 0; \\ S_{r_{i,j}}^s + \Delta S_{r_{i,j}}^{s+1} = -\frac{c}{3 \cdot 0,5(r_{i+1} - r_{i-1})} \left(X_{r_{i+1,j}}^s + \frac{\partial X_{r_{i+1,j}}^s}{\partial T_{i+1,j}^s} \Delta T_{i+1,j}^{s+1} - X_{r_{i,j}}^s - \frac{\partial X_{r_{i,j}}^s}{\partial T_{i,j}^s} \Delta T_{i,j}^{s+1} \right); \quad (18) \\ S_{z_{i,j}}^s + \Delta S_{z_{i,j}}^{s+1} = -\frac{c}{3 \cdot 0,5(z_{i+1} - z_{i-1})} \left(X_{z_{i+1,j}}^s + \frac{\partial X_{z_{i+1,j}}^s}{\partial T_{i,j+1}^s} \Delta T_{i,j+1}^{s+1} - X_{z_{i,j}}^s - \frac{\partial X_{z_{i,j}}^s}{\partial T_{i,j}^s} \Delta T_{i,j}^{s+1} \right), \end{aligned}$$

где s — номер ньютоновской итерации; $S_{r_{i,j}}^{s+1} = S_{r_{i,j}}^s + \Delta S_{r_{i,j}}^{s+1}$; $S_{z_{i,j}}^{s+1} = S_{z_{i,j}}^s + \Delta S_{z_{i,j}}^{s+1}$; $T_{i,j}^{s+1} = T_{i,j}^s + \Delta T_{i,j}^{s+1}$.

Преобразовав (18), получим систему линейных уравнений относительно изменения потоков и температуры:

$$\begin{aligned} A \Delta T_{i,j}^{s+1} + B \Delta S_{r_{i,j}}^{s+1} + C \Delta S_{r_{i-1,j}}^{s+1} + D \Delta S_{z_{i,j}}^{s+1} - D \Delta S_{z_{i,j-1}}^{s+1} = F; \\ \Delta S_{r_{i,j}}^{n+1} + K_1 \Delta T_{i,j}^{n+1} + H_1 \Delta T_{i+1,j}^{n+1} = M_1; \\ \Delta S_{z_{i,j}}^{n+1} + K_2 \Delta T_{i,j}^{n+1} + H_2 \Delta T_{i+1,j}^{n+1} = M_2. \end{aligned}$$

Здесь

$$\begin{aligned} A = \frac{\partial \tilde{E}}{\partial T} \Big|_{T_{i,j}^s}; \quad B = \frac{\tau}{r_i - r_{i-1}}; \quad C = -\frac{\tau}{r_i - r_{i-1}}; \quad D = \frac{\tau}{z_j - z_{j-1}}; \\ F = -E_{i,j}^s - \int_{\varepsilon} f_{\varepsilon i,j}^p B_{\varepsilon i,j}^s d\varepsilon - \frac{E_{i,j}^s - E_{i,j}^n + c\tau \langle w_{ij}^s \rangle}{c\tau \langle w_{ij}^s \rangle} + E_{i,j}^n + U_{i,j}^n - \frac{\tau}{r_i - r_{i-1}} \left(S_{1i,j}^s - S_{1i-1,j}^s \right) - \\ - \frac{\tau}{z_j - z_{j-1}} \left(S_{2i,j}^s - S_{2i,j-1}^{n+1} \right); \\ K_1 = -\frac{c}{3 \cdot 0,5(r_{i+1} - r_{i-1})} \frac{\partial X_{r_{i,j}}^s}{\partial T_{i,j}^s}; \quad K_2 = -\frac{c}{3 \cdot 0,5(z_{j+1} - z_{j-1})} \frac{\partial X_{z_{i,j}}^s}{\partial T_{i,j}^s}; \\ H_1 = \frac{c}{3 \cdot 0,5(r_{i+1} - r_{i-1})} \frac{\partial X_{r_{i+1,j}}^s}{\partial T_{i+1,j}^s}; \quad H_2 = \frac{c}{3 \cdot 0,5(z_{j+1} - z_{j-1})} \frac{\partial X_{z_{i,j+1}}^s}{\partial T_{i,j+1}^s}; \\ M_1 = -\frac{c}{3 \cdot 0,5(r_{i+1} - r_{i-1})} \left(X_{r_{i+1,j}}^s - X_{r_{i,j}}^s \right) - S_{r_{i,j}}^s; \\ M_2 = -\frac{c}{3 \cdot 0,5(z_{j+1} - z_{j-1})} \left(X_{z_{i,j+1}}^s - X_{z_{i,j}}^s \right) - S_{z_{i,j}}^s. \end{aligned}$$

Пролинеаризуем уравнение для граничных условий:

$$\begin{aligned} \Delta S_{\Gamma P}^{s+1} + B_T \Delta T_{\text{BH}}^{s+1} = B; \\ B_T = \int \frac{\partial \tilde{a}_T^s}{\partial T_{\text{BH}}} f^p B_{\text{BH}}^s d\varepsilon + \int \tilde{a}_T^s f^p \frac{\partial B_{\text{BH}}^s}{\partial T_{\text{BH}}} d\varepsilon + \frac{\int \tilde{a}_{Tz} g^p d\varepsilon}{\int g^p d\varepsilon} \frac{\partial G_{\text{BH}}^s}{\partial T_{\text{BH}}} + \frac{\int \partial \tilde{a}_{Tz} g^p d\varepsilon}{\int g^p d\varepsilon} G_{\text{BH}}^s - b \int \frac{\partial \tilde{a}_{Tz}}{\partial T_{\text{BH}}} d\varepsilon; \\ B = b \int \tilde{a}_{Tz} d\varepsilon - S_{\Gamma P}^s - \int \tilde{a}_T^s f^p B_{\text{BH}}^s d\varepsilon - \frac{\int \tilde{a}_{Tz} g^p d\varepsilon}{\int g^p d\varepsilon} G_{\text{BH}}^s; \\ \frac{\partial \tilde{a}_T^s}{\partial T_{\text{BH}}} = -\frac{3cdz}{(3h - ck_{\varepsilon}^s b_s)^2} \frac{\partial k_{\varepsilon}^s}{\partial T_{\text{BH}}}. \end{aligned}$$

Пример численного расчета

В качестве примера рассмотрим решение второй задачи Флека [4] в следующей постановке. Имеется сферически-симметричная система, состоящая из нескольких физических областей, ограниченных координатами по радиусу (рис. 1).

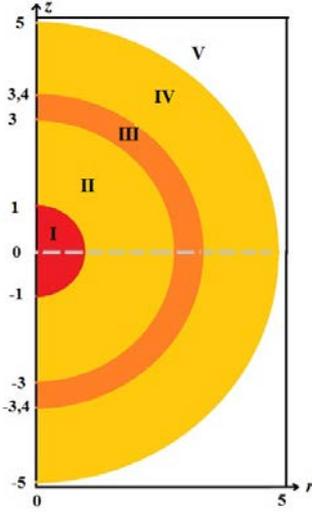


Рис. 1. Геометрия задачи

Коэффициент поглощения вычисляется по формуле $\alpha = \frac{1 - e^{-\varepsilon/T}}{\varepsilon^3}$; уравнение состояния вещества: $E = c_v T$, $c_v = 0,81$. В области I находится изотропный планковский источник с температурой 1 кэВ в виде абсолютно черного тела ($\alpha_0 = 10^4$; $c_v = 10^{10}$). В областях II и IV $\alpha_0 = 27$; в области III $\alpha_0 = 10^4$; область V — вакуум: $\alpha_0 = 0$.

Начальные условия: $T^0 = 10^{-3}$ кэВ; $I_{II,III,IV} = B(T^0)/4\pi$; $I_V = 0$.

Граничные условия: $T_I = 1$ кэВ, на границе $r = 0$ задано условие отражения, на остальных границах — условие свободной границы.

Сетка по спектру из 28 групп: $\varepsilon = 10^{-4}$; 0,02; 0,04; 0,06; 0,08; 0,1; 0,15; 0,2; 0,3; 0,4; 0,5; 0,7; 10,0; 1,4; 1,8; 2,2; 2,6; 3,0; 3,5; 4,0; 4,5; 5,0; 5,5; 6,0; 7,0; 8,0; 10,0; 12,0; 15,0.

В качестве эталонного решения взят результат расчета в спектральной квазидиффузионной постановке [5] в одномерном случае в сферических координатах со следующими параметрами: шаг по пространству $h_r = 0,05$; шаг по времени $\tau = 10^{-4}$; число разбиений по углу равно 16.

Сравнение результатов расчета по предлагаемому методу с эталонным решением представлено на рис. 2–4. Отклонение решений можно объяснить различием систем координат (сферическая и цилиндрическая). В целом полученные профили температуры качественно описывают эталонное решение. На рис. 4 можно наблюдать сходимость решения по пространству. Кроме того, поскольку пробеги $k_{r\varepsilon}$, $k_{z\varepsilon}$ могут принимать отрицательные значения, когда поток тепла и градиент температур

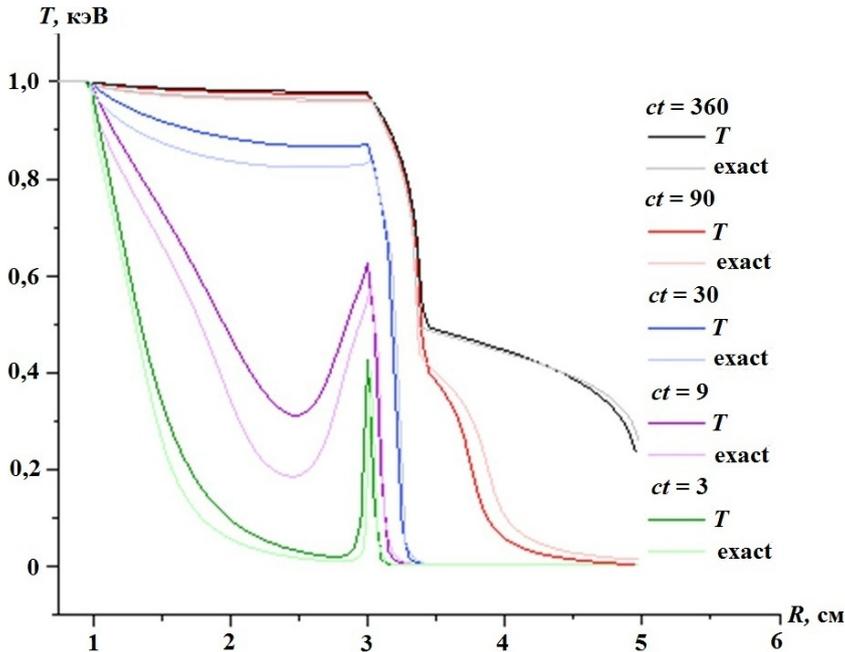


Рис. 2. Распределение температуры вещества вдоль $R = \sqrt{r^2 + z^2}$ в сечении под углом $\varphi = 45^\circ$ между \vec{R} и \vec{r} ; $h = 0,05$ см; 16 разбиений по каждому углу: T — расчет по предлагаемому методу; exact — эталонное решение

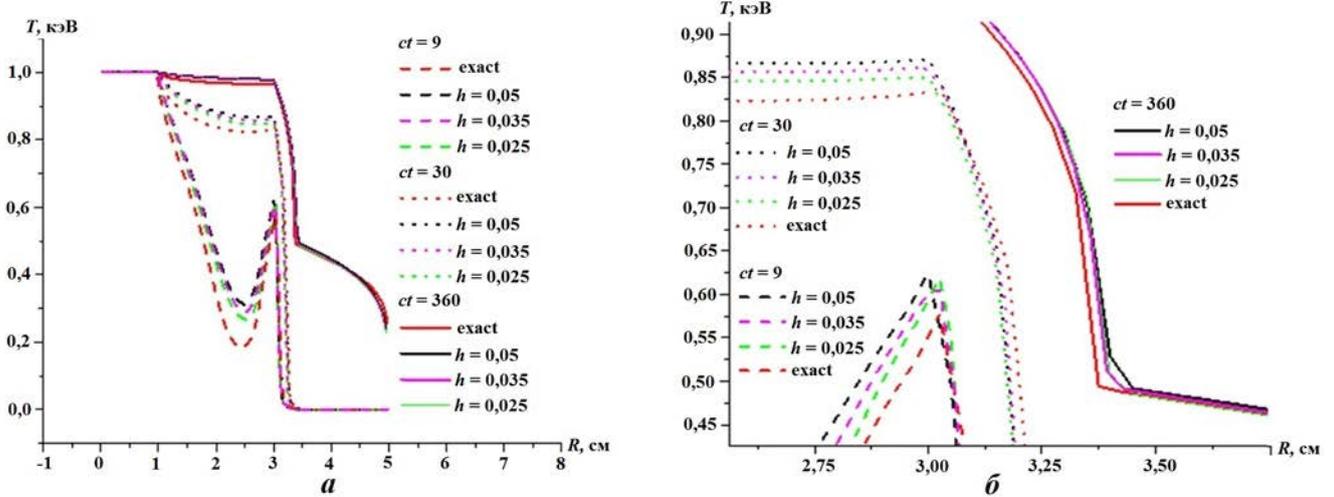


Рис. 3. Распределение температуры вещества вдоль $R = \sqrt{r^2 + z^2}$ в сечении под углом $\varphi = 45^\circ$ между \vec{R} и \vec{r} , 16 разбиений по каждому углу, $h = 0,025; 0,035; 0,05$ см (а), и его увеличенный фрагмент (б); exact — эталонное решение

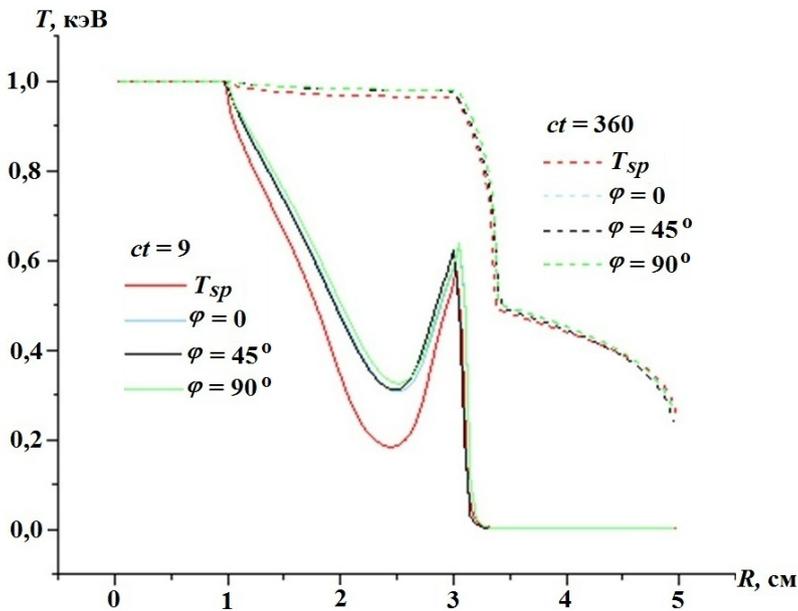


Рис. 4. Распределение температуры вещества вдоль $R = \sqrt{r^2 + z^2}$ в сечении под углом φ между \vec{R} и \vec{r} ; T_{sp} — решение, полученное в спектральном расчете

имеют разные знаки, удалось получить профиль температуры с нарушением принципа максимума (область III). То, что это не привело к развитию неустойчивости, — удивительный факт. Можно сделать вывод, что на данном этапе методика подтвердила работоспособность в двумерном случае.

Лучевой эффект

В решении уравнения переноса присутствует лучевой эффект — зависимость решения от сетки по углу. Выясним, как проявляется лучевой эффект в решении, полученном описанным методом. Из

рис. 5, где изображены профили температуры в сечениях под углами $0, 45, 90^\circ$ на моменты времени $ct = 9; 360$, видно отличие результатов при разных сечениях.

Для дальнейшего исследования лучевого эффекта была проведена серия расчетов с различным разбиением по углу (4, 8, 16, 32, 64 направления). Результаты представлены на рис. 5. Из рисунка можно сделать вывод, что сетки 16×16 по угловым переменным достаточно, дальнейшее разбиение в данном случае не сильно влияет на решение.

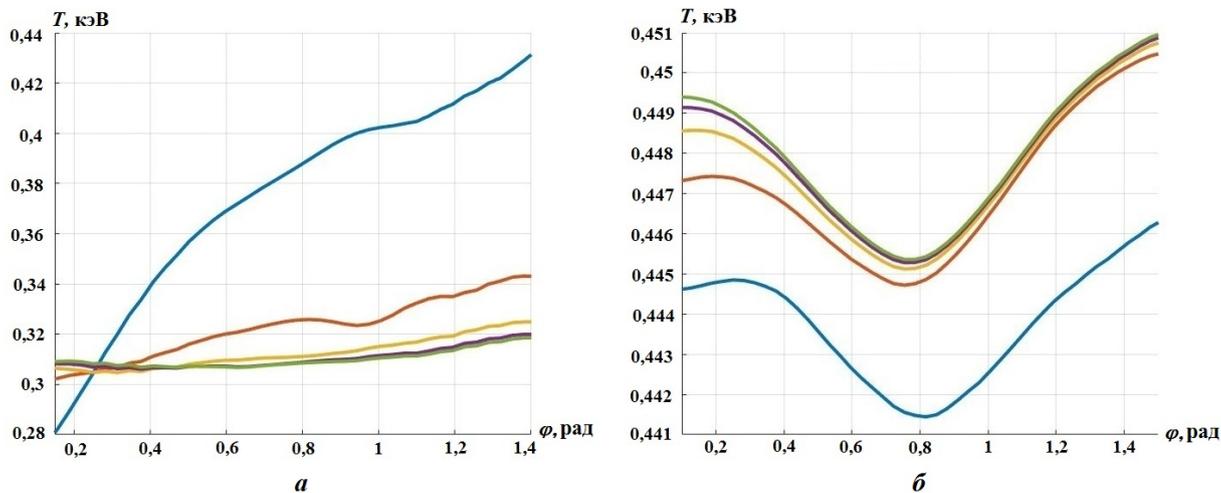


Рис. 5. Температура вещества в зависимости от угла $\varphi \in [0; \pi/2]$ при 4 (—), 8 (—), 16 (—), 32 (—), 64 (—) направлениях по углу: *a* — $ct = 9$; $R = 2,5$ см; *б* — $ct = 360$; $R = 4$ см

Заключение

В данной работе описан квазиспектральный метод для решения спектрального кинетического уравнения переноса излучения совместно с уравнением баланса энергии в двумерном осесимметричном случае на прямоугольной сетке. Достоинство метода заключается в том, что вся сложность итерационного процесса переложена на уравнение лучистой теплопроводности, а решение кинетического уравнения и уравнения спектральной диффузии при известной температуре не представляет большой проблемы. Кроме того, данный подход позволяет организовать распараллеливание по спектральной переменной, а также по пространству для вычисления усредненных по энергии пробегов и эффективной энергии излучения. На основе сравнения со спектральным расчетом в одномерном сферическом случае показана работоспособность метода в двумерном случае.

Список литературы

1. Карлыханов Н. Г. Учет спектральных и кинетических эффектов в приближении лучистой теплопроводности для расчета задач переноса излучения // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2019. Вып. 4. С. 21–32.
Karlykhanov N. G. Uchet spektralnykh i kineticheskikh effektov v priblizhenii luchistoy teploprovodnosti dlya rascheta zadach perenosa izlucheniya // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2019. Vyp. 4. S. 21–32.
2. Карлыханов Н. Г., Уракова А. В., Шнитко С. А. Применение метода квазидиффузии для решения задачи переноса излучения в спектральной кинетической постановке в двумерном осесимметричном случае на квадратной сетке // Там же. 2011. Вып. 2. С. 33–43.
Karlykhanov N. G., Urakova A. V., Shnitko S. A. Primenenie metoda kvazidiffuzii dlya resheniya zadachi perenosa izlucheniya v spektralnoy kineticheskoy postanovke v dvumernom osesimmetrichnom sluchae na kvadratnoy setke // Tam zhe. 2011. Vyp. 2. S. 33–43.

3. *Карлыханов Н. Г., Козманов М. Ю.* Учет кинетических эффектов в диффузионном приближении для расчета задач переноса излучения // Там же. 2010. Вып. 4. С. 3–8.
Karlykhanov N. G., Kozmanov M. Yu. Uchet kineticheskikh effektov v diffuzionnom priblizhenii dlya rascheta zadach perenosa izlucheniya // Там же. 2010. Вып. 4. С. 3–8.
4. *Fleck J. F., Cummings Jr. and J.D.* An implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport // J. Comp. Phys. 1971. Vol. 8. P. 313–342.
5. *Карлыханов Н. Г.* Применение метода квазидиффузии для решения задачи переноса излучения // Там же. Вып. 1. С. 32–38.
Karlykhanov N. G. Primenenie metoda kvazidiffuzii dlya resheniya zadachi perenosa izlucheniya // Там же. Вып. 1. С. 32–38.

Статья поступила в редакцию 06.12.21.

УДК 519.6

ИЕРАРХИЧЕСКИЙ АЛГОРИТМ РАСПАРАЛЛЕЛИВАНИЯ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ ТРЕХМЕРНОГО УРАВНЕНИЯ ПЕРЕНОСА НЕЙТРОНОВ НА ГИБРИДНЫХ СУПЕР-ЭВМ

А. А. Нуждин

(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Представлен иерархический алгоритм распараллеливания, который основан на выделении уровней архитектуры гибридной супер-ЭВМ с GPU и применении принципа геометрической декомпозиции на каждом из уровней. Внутри компактной группы нитей используется распараллеливание по элементам гиперплоскости фиксированного размера, состоящей из ячеек сетки. Между блоками одного GPU используется КВА-алгоритм на основе двумерной декомпозиции по столбцам и гиперплоскостям. Между различными GPU используется КВА-алгоритм на основе трехмерной декомпозиции. Программная реализация выполнена на примере тестовой программы ПАУК с помощью технологии CUDA. Эффективность адаптации теста ПАУК к одному GPU подтверждена результатами профилирования и сравнением с производительностью CPU-версии программы. Масштабируемость теста ПАУК в режиме multi-GPU исследована методом умножения.

Ключевые слова: GPU, CUDA, S_n -метод, алгоритм бегущего счета, КВА-алгоритм.

Введение

Графические ускорители (GPU) фирмы Nvidia уже более десяти лет активно используются в области суперкомпьютерных вычислений. В списке TOP-500 самых высокопроизводительных супер-ЭВМ за июнь 2020 г. суммарная пиковая производительность всех установленных GPU Nvidia различных поколений превысила рубеж в 10^{18} флпс, а в списке за ноябрь 2020 г. данная отметка была превышена уже одним поколением ускорителей — Nvidia Volta. За последнее десятилетие наибольший прирост производительности GPU был обеспечен увеличением числа потоковых мультипроцессоров (SM), что обусловлено совершенствованием архитектуры и технологического процесса. Число SM возросло с 14 для поколения Nvidia Kepler до 108 для Nvidia Ampere. При этом технология программирования CUDA [1] в части возможностей для взаимодействия процессов по данным и управлению развита сильнее для одного блока нитей (один SM), чем для сети блоков (множество SM).

Эволюционное увеличение числа ядер на кристалле универсального процессора (CPU) поддержано более совершенной экосистемой разработки параллельных приложений: модели памяти (разделенная, общая, PGAS), стандарты (MPI, OpenMP), инструментальные средства (компилятор, среда исполнения, библиотека MPI). Данная экосистема ориентирована на удобство выражения алгоритмов распараллеливания с учетом взаимодействия процессов как в части синхронизации, так и в части обмена данными.

В работе представлен иерархический алгоритм распараллеливания для решения уравнения переноса нейтронов разностным S_n -методом на трехмерной ортогональной пространственной сетке. Алгоритм характеризуется применением принципа геометрической декомпозиции на различных уровнях архитектуры гибридной супер-ЭВМ (1 SM, 1 GPU, множество GPU) и программной модели (компактная группа нитей, сетка блоков, множество MPI-процессов). На рис. 1 проиллюстрированы ключевые особенности алгоритма. Слева указаны выделяемые уровни архитектуры, справа —

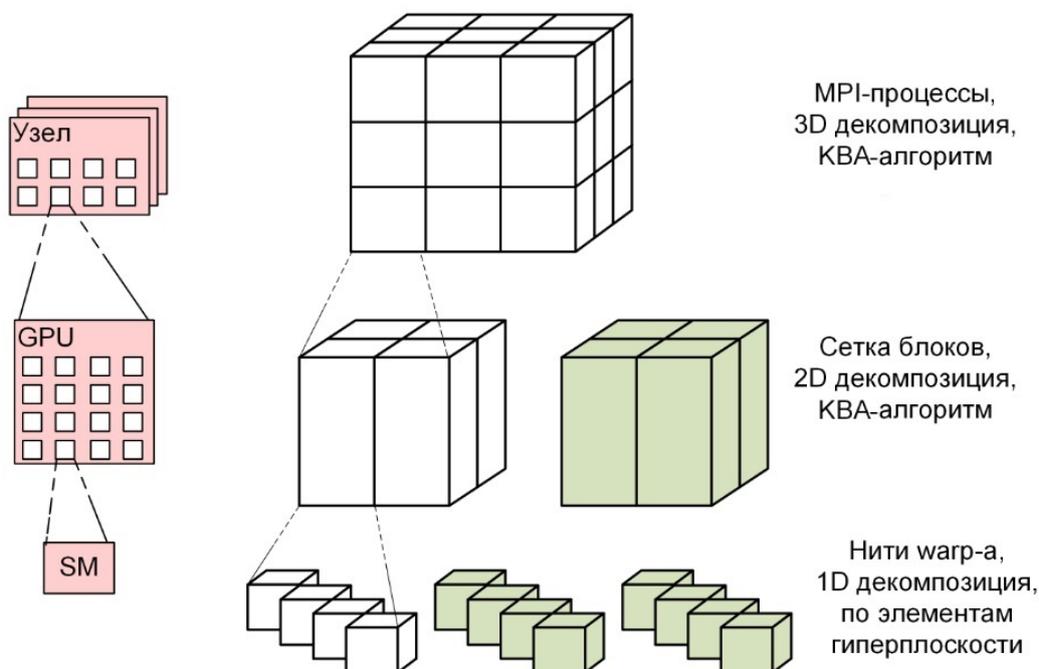


Рис. 1. Уровни архитектуры и типы пространственной декомпозиции на этих уровнях

тип пространственной декомпозиции на каждом из уровней. Здесь и далее warp обозначает компактную группу из 32 нитей одного блока.

На каждом из уровней реализуется распараллеливание для вложенного алгоритма бегущего счета, требующее взаимодействия между процессами. Каждый из рассматриваемых уровней обладает своими средствами для синхронизации и обмена данными, которые существенно отличаются масштабами накладных расходов. Это приводит к различной зернистости (granularity) алгоритмов распараллеливания.

Дополнительно на рис. 1 серо-зеленым цветом выделена вспомогательная декомпозиция по направлениям полета частиц (или группам), используемая для снижения давления на подсистему памяти ускорителя. В табл. 1 представлена иерархия алгоритмов распараллеливания (как основных, так и вспомогательных) на всех выделенных уровнях программной модели. Полуужирным шрифтом отмечены алгоритмы на основе геометрической декомпозиции.

Апробация предлагаемого иерархического алгоритма распараллеливания выполнена на примере адаптации тестовой программы ПАУК [2] к архитектуре гибридных супер-ЭВМ с GPU. Исследования производительности и масштабируемости выполнены на различных образцах вычислительной техники, в том числе из состава программно-аппаратного полигона Национального центра физики и математики (НЦФМ) г. Сарова для исследования архитектур супер-ЭВМ.

Таблица 1

Уровни иерархического алгоритма распараллеливания

Уровень программной модели	Алгоритм
MPI-процессы	КВА — 8 конвейеров по направлениям/группам, 3D декомпозиция
Группы блоков	По направлениям/группам, 1D декомпозиция
Блоки	КВА — 1 конвейер по слоям, 2D декомпозиция
warp-ы блока	По направлениям, 1D декомпозиция
Нити warp-а	По элементам гиперплоскости , 1D декомпозиция

Аналогичные работы

Схожие работы по реализации на GPU алгоритма бегущего счета можно классифицировать по уровню программной модели, на которой реализуется распараллеливание по элементам одной гиперплоскости [3]. После обработки каждой гиперплоскости требуются точка синхронизации и обмен данными между параллельными процессами. Базовые возможности технологии программирования CUDA обеспечивают два способа синхронизации: между нитями одного блока с помощью функции `_syncthreads` и между всеми нитями при выходе из ядра функции, выполняемой на GPU.

Распараллеливание бегущего счета на уровне нитей одного блока реализуется при первом способе синхронизации: обмен данными выполняется через разделяемую память, каждая нить рассчитывает множество ячеек пространственной сетки. По такой схеме выполнена адаптация к GPU кодов DENOVO [4], Minisweep [5] и Ardra [6]. В качестве технологии параллельного программирования использовались CUDA в DENOVO и Minisweep, OpenACC [7] в Minisweep, инфраструктура RAJA [8] в Ardra. Особенностью алгоритма в Ardra является фиксированный размер одного блока — 20×20 нитей.

Распараллеливание бегущего счета на уровне сети блоков реализуется при втором способе синхронизации: обмен данными выполняется через глобальную память, каждая нить рассчитывает одну ячейку пространственной сетки. Такой способ синхронизации использовался в работах по адаптации к GPU тестового приложения SNAP [9] с помощью технологии CUDA и стандарта OpenCL [10]. Особенностью OpenCL-реализации является "плоская" формулировка алгоритма, в которой определяется только общее число нитей, а размер одной рабочей группы или блока нитей задается не связанным с логикой алгоритма значением компилятора по умолчанию.

Комбинация двух способов синхронизации использовалась при исследовании вариантов распараллеливания для гнезда вложенных циклов с зависимостями на примере счетного прототипа [11], одним из применений которого может быть алгоритм бегущего счета на ортогональной сетке. В этом случае сетка логически разбивается на фрагменты и организуется вложенный бегущий счет — по фрагментам и по ячейкам в каждом из этих фрагментов. Первый способ синхронизации используется при распараллеливании по элементам гиперплоскости внутри одного фрагмента, второй способ — при распараллеливании между фрагментами, при этом размер сети блоков определяется числом фрагментов в текущей гиперплоскости.

Во всех перечисленных приложениях для увеличения степени параллелизма на GPU активно используются остальные переменные фазового пространства задачи: направления полета частиц и группы. Распараллеливание по указанным переменным не является алгоритмически сложным, однако означает исключение ресурса этих переменных из MPI-конвейеров, что негативно влияет на масштабируемость в режиме multi-GPU [12].

Постановка задачи

В тестовой программе ПАУК решается стационарное трехмерное уравнение переноса в однопроводном кинетическом приближении в декартовой системе координат на ортогональных пространственных сетках [2]:

$$\operatorname{div}(\vec{\Omega}N) + \alpha N = \frac{1}{4\pi} (\beta n^{(0)} + Q), \quad \operatorname{div}(\vec{\Omega}N) = \Omega_x \frac{\partial N}{\partial x} + \Omega_y \frac{\partial N}{\partial y} + \Omega_z \frac{\partial N}{\partial z}, \quad (1)$$

где α — коэффициент столкновения частиц; β — коэффициент размножения частиц; Q — независимый источник частиц; N — плотность потока частиц, летящих в направлении $\vec{\Omega}$ (для определенности скорость частиц $v = 1$); $\vec{\Omega}(\Omega_x, \Omega_y, \Omega_z)$ — единичный вектор направления полета частиц; $\Omega_x = \sqrt{1 - \mu^2} \cos \varphi$ — проекция вектора $\vec{\Omega}$ на ось OX ; $\Omega_y = \sqrt{1 - \mu^2} \sin \varphi$ — проекция $\vec{\Omega}$ на ось OY ; $\Omega_z = \mu$ — проекция $\vec{\Omega}$ на ось OZ (косинус угла между вектором $\vec{\Omega}$ и осью OZ); φ — угол между проекцией $\vec{\Omega}$ на плоскость OXY и осью OX ; $n^{(0)} = \int_{-1}^1 d\mu \int_0^{2\pi} N d\varphi$.

Уравнение (1) решается в области $d = \{(x, y, z) \in L, -1 \leq \mu \leq 1, 0 \leq \varphi \leq 2\pi\}$.

На внешней поверхности задаются граничные условия в виде потока частиц, входящих в тело при $\vec{\Omega} \cdot \vec{n} < 0$, где \vec{n} — внешняя нормаль к поверхности, ограничивающей область L .

Далее рассматривается конечно-разностная аппроксимация уравнения (1) в случае, когда пространственная сетка в области L состоит из прямоугольных параллелепипедов. Значения параметра μ выбираются из интервала $(-1, 1)$, значения параметра φ — из интервала $(0, 2\pi)$.

Уравнение баланса в счетной ячейке в конечно-разностной форме получается с помощью интегро-интерполяционного метода:

$$\operatorname{div}_h(\vec{\Omega}N) \equiv \Omega_x S_{yz}(N_2 - N_1) + \Omega_y S_{xz}(N_4 - N_3) + \Omega_z S_{xy}(N_6 - N_5) + V\alpha N_0 = V\bar{F}, \quad (2)$$

где N_i ($i = 1, 2, \dots, 6$) — средние значения искомой функции N на гранях ячейки; N_0 — значение функции N в центре ячейки; S_{yz}, S_{xz}, S_{xy} — площади граней ячейки; V — объем ячейки; $\bar{F} = \frac{1}{4\pi}(\beta\bar{n}^{(0)} + Q)$. Скалярный поток $n^{(0)}$ в каждой счетной ячейке вычисляется следующим образом: $\bar{n}^{(0)} = \sum_{w=1}^{n_w} N_0^w d\Omega_w$, $\sum_{w=1}^{n_w} d\Omega_w = 4\pi$. Здесь w — номер направления полета частиц; n_w — число направлений полета частиц; $d\Omega_w$ — телесный угол.

Для замыкания системы сеточных уравнений по пространственным переменным используется DD-схема:

$$N_0 = \frac{N_1 + N_2}{2} = \frac{N_3 + N_4}{2} = \frac{N_5 + N_6}{2}. \quad (3)$$

Система (2), (3) решается итерациями по источнику: $\operatorname{div}_h(\vec{\Omega}N^{s+1}) + V\alpha N_0^{s+1} = V\bar{F}^s$, где s — номер итерации.

Численное решение системы сеточных уравнений переноса нейтронов для одного направления полета частиц осуществляется с помощью алгоритма бегущего счета. Особенности данного алгоритма необходимо учитывать при организации параллельных вычислений на основе принципа пространственной декомпозиции.

Алгоритм бегущего счета

По сути алгоритм бегущего счета — это правило упорядочения ячеек пространственной сетки, которое обеспечивает преобразование разностного оператора к блочно-треугольному виду. Данное правило может быть построено не единственным способом.

На рис. 2, а показан пример двумерной сетки с 5 столбцами и 4 строками, на которой стрелками указана передача потоков частиц (далее просто потоков) между ячейками. Цифрами и цветом

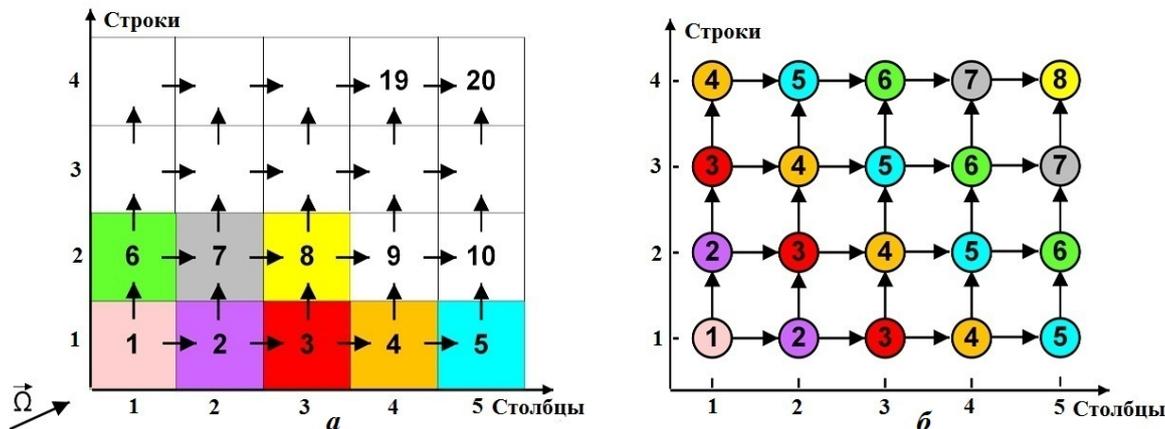


Рис. 2. Алгоритм бегущего счета на ортогональной сетке (а) и соответствующий ему орграф $G = (V, E)$ с выделенными гиперплоскостями (б)

отражено упорядочение ячеек по сеточным направлениям, которое чаще всего используется при организации бегущего счета на CPU. Циклы по столбцам и строкам имеют зависимость по данным при передаче потоков между ячейками, что мешает их параллельной обработке.

На рис. 2, б дано представление этого алгоритма в виде ациклического орграфа $G = (V, E)$, в котором каждой вершине $v \in \{V\}$ соответствует решение уравнения баланса в ячейке фазового пространства, а каждой дуге $e \in \{E\}$ соответствует передача потоков через грани смежных ячеек. Направление дуги определяется освещенностью грани. Цифрами и цветом в орграфе отмечены гиперплоскости [3]. Гиперплоскость — это подмножество вершин орграфа $\{H_{l_{\max}+1}\}$ с одинаковой максимальной длиной пути l_{\max} от истока. Цикл по элементам одной гиперплоскости не имеет зависимостей по данным, что обеспечивает возможность его параллельной реализации.

Распараллеливание между нитями одной группы

Для уровня внутри компактной группы из 32 нитей (warp) одного блока предназначен алгоритм распараллеливания по ячейкам пространственной сетки, принадлежащим одной гиперплоскости $\{H_{l_{\max}+1}\}$. Эффективную реализацию данного алгоритма затрудняют разное число элементов в разных гиперплоскостях ($|H_{l_{\max}+1}| \in [1, \min(n_x, n_y)]$, где n_x, n_y — соответственно число столбцов и строк) и формат представления данных, не обеспечивающий компактного размещения в памяти элементов одной гиперплоскости. Решением перечисленных проблем является алгоритм, основанный на выделении гиперплоскостей фиксированного размера и записи данных в специальных форматах [13].

Фиксированный размер гиперплоскости определяется числом нитей в группе и совпадает с первым размером блока. Номер нити жестко привязан к номеру столбца пространственной сетки.

Сеточные данные записываются в двух форматах, описываемых массивами вида $\left(n_x^{loc}, n_y + n_x^{loc} - 1, n_z, \frac{n_x}{n_x^{loc}} \right)$, где $n_x^{loc} \equiv 32$; $n_y + n_x^{loc} - 1$ — число гиперплоскостей; n_z — число слоев.

В зависимости от правила изменения сеточных индексов по столбцам и строкам бегущий счет в двумерном слое реализуется в виде одного из четырех различных орграфов. Для каждого из этих графов есть симметричная пара, полученная сменой направления у всех дуг. На рис. 3 симметричные пары графов обозначены разным цветом дуг. У каждой такой пары графов правило распределения вершин по гиперплоскостям одно и то же, т. е. достаточно одного формата данных. Поэтому для записи сеточных данных требуется всего два формата.

Алгоритм выделения гиперплоскостей фиксированного размера приводит к увеличению множества вершин в исходном графе. Для случая, показанного на рис. 2, алгоритм бегущего счета преобразуется к виду, представленному на рис. 3 слева. Здесь числа обозначают номера строк исходной сетки.

В новом орграфе первое сеточное направление совпадает с элементами одной гиперплоскости и не имеет зависимостей по данным. Вершины нового графа представляют собой объединение двух множеств: счетного $\{V\}$ ("цветные" вершины) и фиктивного $\{V_f\}$ (вершины белого цвета). Счетные вершины соответствуют ячейкам сетки, фиктивные — нет. Решение уравнения баланса выполняется в вершинах обоих типов $\{V\} \cup \{V_f\}$. Выходящий поток определяется с учетом типа вершины и по

условию $N^{\text{вых}} = \begin{cases} N^{\text{вых}}, & v \in \{V\} \\ N^{\text{вх}}, & v \in \{V_f\} \end{cases}$, где $v \in \{V\}$, если $i_y \in [1, n_y]$, иначе $v \in \{V_f\}$. Взаимосвязь

между индексами строк и гиперплоскостей: $i_y = i_{hyp} - i_x + 1$ — для орграфов на рис. 3 слева; $i_y = i_{hyp} - 32 + i_x$ — для орграфов на рис. 3 справа. Здесь i_x, i_y, i_{hyp} — индексы столбцов, строк и гиперплоскостей, $i_x \in [1, 32]$.

Каждая нить рассчитывает все гиперплоскости и все слои сетки для фиксированного номера столбца. После расчета каждой ячейки требуется взаимодействие нитей для передачи потоков только в первом топологическом направлении (между столбцами). Для этого используются CUDA-функции перестановки в пределах группы нитей `_shfl_up_sync` и `_shfl_down_sync`.

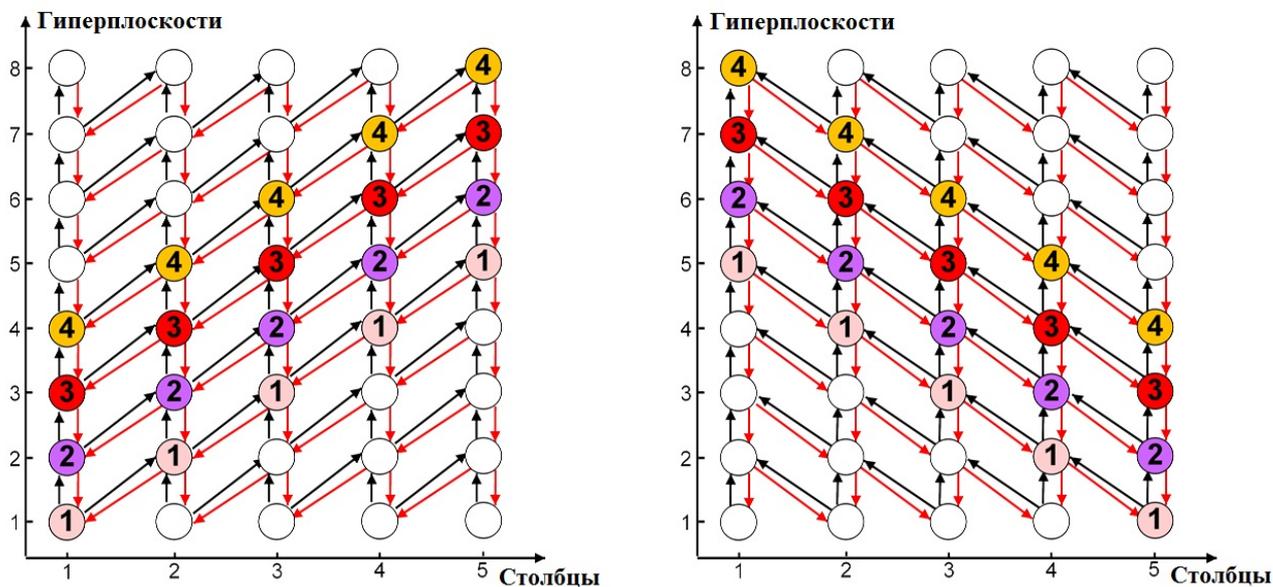


Рис. 3. Орграфы бегущего счета на новых форматах данных

Передача потоков в остальных направлениях не требует взаимодействия нитей, однако она реализована с учетом иерархии памяти на GPU. Для второго топологического направления (между гиперплоскостями одного столбца) используется регистровая память, для третьего — разделяемая память. При этом в алгоритм вводится порционность расчета циклов по гиперплоскостям (n_{hyp}^{loc}) и слоям (n_z^{loc}). Размеры порций определяются с учетом объема доступной разделяемой памяти.

Новые форматы данных требуют дополнительных затрат оперативной памяти. Все массивы увеличиваются в $\frac{n_y + n_x^{loc} - 1}{n_y}$ раз из-за перехода от строк к гиперплоскостям. Массивы данных, которые являются инвариантами относительно направлений полета частиц, дополнительно увеличиваются в два раза из-за записи данных в двух форматах (см. рис. 3 слева и справа). Двукратного увеличения общих затрат оперативной памяти не происходит при решении нестационарной задачи переноса нейтронов, т. е. когда в записи уравнения (1) присутствует производная по времени. В этом случае основной объем памяти приходится на массивы с зависимостью от направлений полета частиц, которые можно представить в нужном формате в зависимости от номера октанта.

Предлагаемый алгоритм распараллеливания по элементам гиперплоскости существенно отличается от аналогичных работ [4, 6]. Гиперплоскости выделены в одном трехмерном слое, фиксированный размер гиперплоскостей определен особенностью архитектуры GPU, данные записаны в специальных форматах, введена порционность при расчете циклов.

Зернистость алгоритма распараллеливания между нитями одной компактной группы составляет 1 ячейку фазового пространства задачи.

Распараллеливание между группами нитей одного блока

На уровне множества компактных групп нитей (warp) одного блока реализуется распараллеливание по направлениям полета частиц одного октанта. Этот уровень необходим для снижения нагрузки на подсистему памяти за счет использования разделяемой памяти. При решении уравнения переноса нейтронов часть данных не зависит от направлений полета частиц. Разделяемая память позволяет уменьшить число запросов в глобальную память для данных, инвариантных относительно направлений, в какой-то мере эмулируя работу кэш-памяти.

Распараллеливание между группами нитей одного блока организуется по порции направлений в октанте. Число направлений n_w^{loc} в порции определяет второй размер блока нитей. В тестовой

программе ПАУК используются три массива, которые не зависят от направлений. В разделяемой памяти заводится временный массив вида $(n_x^{loc}, 3)$. Данные читаются из глобальной памяти и пишутся в разделяемую, при этом каждая нить выполняет не более одной операции чтения. Такой подход в n_w^{loc} раз сокращает число запросов к глобальной памяти при работе с указанными массивами, если $n_w^{loc} \geq 3$.

Вычисление скалярного потока нейтронов $\bar{n}^{(0)}$ требует взаимодействия параллельных процессов и реализовано с помощью операции атомарного суммирования, которая эффективно кэшируется.

Зернистость алгоритма распараллеливания между компактными группами нитей одного блока составляет 1 ячейку фазового пространства задачи.

Исследование производительности выполнялось в режиме полной загрузки GPU Nvidia V100 16 GB SXM. Для этого в тесте ПАУК был реализован временный уровень распараллеливания между блоками — по порциям направлений одного октанта.

В табл. 2 представлены результаты исследования производительности и профилирования тестовой программы ПАУК в зависимости от формата представления данных и размера блока по направлениям. Метрика производительности вычислялась как число неизвестных фазового пространства, найденных за секунду, по формуле $R = 8n_x n_y n_z n_w^{oct} n_{it} / t$, где t — время в секундах, n_{it} — число итераций. Значения остальных метрик получены с помощью средства профилирования nvprof. Результаты получены при следующих параметрах теста: $n_x = 32$; $n_y = 169$; $n_{hyp} = 200$; $n_z = 4$; число направлений в октанте $n_w^{oct} = n_w^{loc} N_{SM} b_{SM} = 1\,600$, где $N_{SM} = 80$ — число SM, b_{SM} — число активных блоков на одном SM. Суммарное число нитей — 51 200.

Важной особенностью метрики R является расчет относительно числа строк, а не гиперплоскостей в задаче. Доля счетных вершин в графе бегущего счета $\frac{|V|}{|V| + |V_f|} = \frac{n_y}{n_y + n_x^{loc} - 1}$. Значения остальных метрик получены при вычислении всех вершин.

Результаты табл. 2 демонстрируют решающее влияние подсистемы памяти на производительность приложения. У данной модели GPU пропускная способность глобальной памяти составляет 900 Гбайт/с. В режиме одного направления в блоке не происходит сокращения числа запросов к глобальной памяти при работе с инвариантными массивами, что приводит к увеличенной нагрузке на память. В случае построчного формата соседние нити обращаются к данным, которые расположены в памяти со смещением в $n_x^{loc} \pm 1$ чисел в формате двойной точности, что также приводит к работе подсистемы памяти на пределе возможности — об этом свидетельствует значение метрики *Global Load Throughput*. Подсистема памяти не является ограничителем производительности только при записи данных в диагональных форматах и дополнительном распараллеливании по направлениям в блоке.

Таблица 2

Характеристики производительности вариантов программы ПАУК

Вариант программы	R , млрд. яч./с	FLOP Efficiency (Peak Double), %	Warp Execution Efficiency, %	Global Load Throughput, Гбайт/с
Диагональные форматы, 1 направление в блоке	14,9	16,1	89,4	860,4
Диагональные форматы, 4 направления в блоке	38,7	45,4	88,8	586,2
Построчный формат, 4 направления в блоке	17,3	18,1	83,6	937,6

Распараллеливание между блоками на основе геометрической декомпозиции

Для распараллеливания на следующем уровне архитектуры и программной модели используется КВА-алгоритм [14] на основе геометрической декомпозиции. Исходная пространственная сетка разбивается по столбцам и гиперплоскостям на геометрические фрагменты. Разбиение выполняется

регулярным образом, без перехлестов. Фиксированный размер гиперплоскостей учитывается при декомпозиции сетки. Предполагается жесткая привязка фрагмента к вычислительному ресурсу. Один блок выполняет расчет одного геометрического фрагмента.

Алгоритм бегущего счета в слое в случае геометрической декомпозиции имеет двухуровневую реализацию: по фрагментам и по ячейкам в каждом из этих фрагментов. На рис. 4, а представлен пример орграфа бегущего счета по ячейкам. Цифры в вершинах этого графа соответствуют номерам строк исходной сетки. Красными линиями выделены границы геометрических фрагментов.

На рис. 4, б представлен пример орграфа бегущего счета по фрагментам. Вершина здесь соответствует отдельному фрагменту и локальному бегущему счету по всем ячейкам этого фрагмента. Цифры в вершинах этого графа соответствуют номерам гиперплоскостей по фрагментам.

Орграфы на рис. 4 демонстрируют несколько важных особенностей в организации параллельного бегущего счета по фрагментам, которые обусловлены выбранными форматами представления данных. Эти особенности касаются объемов передаваемых данных и правила синхронизации.

Пусть n_x^{loc} , n_{hyp}^{loc} – размеры фрагментов по столбцам и гиперплоскостям. Передача потоков между фрагментами выполняется:

- при декомпозиции по столбцам для n_{hyp}^{loc} граней;
- при декомпозиции по гиперплоскостям для $2n_x^{loc} - 1$ граней.

Правило синхронизации между фрагментами учитывает максимальную длину пути в графе от источника до вершины, а также смещение гиперплоскостей через каждые n_x^{loc} столбцов. Длительность выполнения алгоритма параллельного бегущего счета по фрагментам можно оценить по формуле

$$T_{sweep} = n_{hyp}^{loc} (1 + (D_{hyp} - 1) + \left\lfloor \frac{n_x^{loc} + n_{hyp}^{loc}}{n_{hyp}^{loc}} \right\rfloor (D_x - 1)),$$

где "скобки" $\lfloor \rfloor$ используются для обозначения целой части числа; D_x , D_{hyp} – число фрагментов по столбцам и по гиперплоскостям:

$$D_x = \frac{n_x}{n_{nx}^{loc}}, \quad n_{nx}^{loc} \equiv 32; \quad D_{hyp} = \frac{n_y + n_x^{loc} - 1}{n_{hyp}^{loc}}.$$

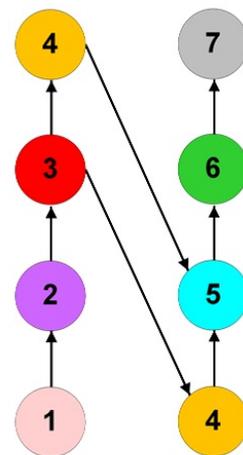
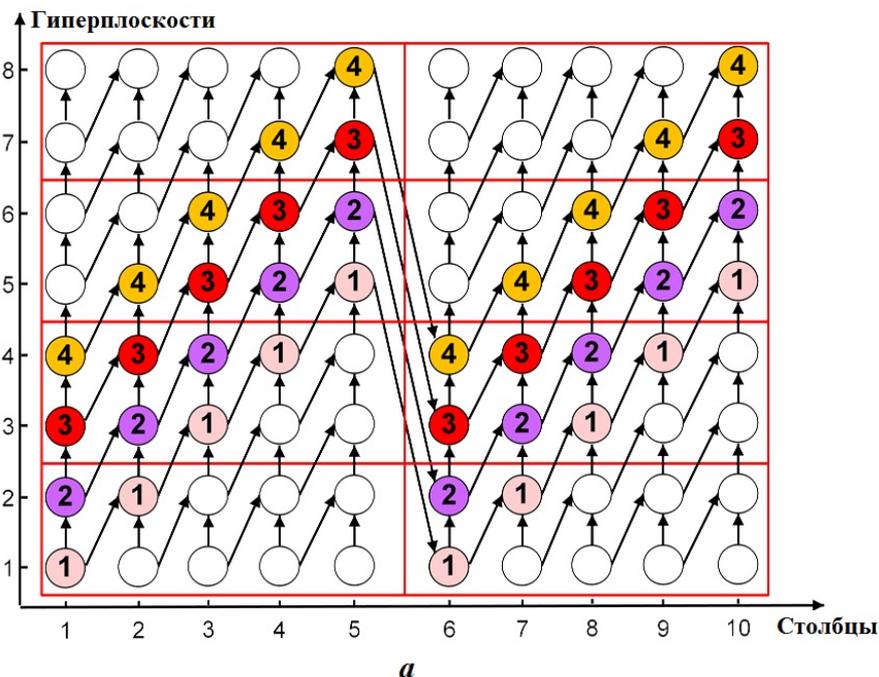


Рис. 4. Орграфы бегущего счета по ячейкам (а) и по фрагментам (б)

Параллельный бегущий счет можно организовать одновременно для порции в n_z^{loc} слоев ($n_z^{loc} \in [1, n_z]$), так как бегущий счет по ячейкам каждого слоя имеет одинаковую реализацию в виде орграфа. Размер порции по слоям определяет частоту межблочного взаимодействия и зернистость алгоритма распараллеливания.

Для эффективной загрузки всех блоков организуется конвейер по слоям пространственной сетки. В этом случае значение эффективности распараллеливания оценивается по формуле

$$E_{КВА} = \frac{\frac{n_z}{n_z^{loc}}}{\frac{n_z}{n_z^{loc}} + (D_{hyp} - 1) + \left[\frac{n_x^{loc} + n_{hyp}^{loc}}{n_{hyp}^{loc}} \right] (D_x - 1)} \cdot 100 \%, \quad (4)$$

где n_z — число слоев сетки, рассчитываемых на одном GPU.

Передача потоков между блоками осуществляется через глобальную память GPU. Для указания компилятору особого режима работы с массивами для передачи потоков используется спецификатор `volatile`, запрещающий различные оптимизации при работе с такими массивами [1].

Параллельный конвейер требует организации вычислений в предопределенной последовательности, которая регулируется с помощью механизмов синхронизации. На программе ПАУК опробованы два механизма межблочной синхронизации: кооперативные группы и атомарные функции.

Для синхронизации с помощью кооперативных групп необходим режим отдельной компиляции. Такая компиляция приводит к увеличению нагрузки на регистровый файл и снижению значения параметра b_{SM} с 5 до 4 на GPU Nvidia Volta. Кроме того, синхронизация с помощью кооперативных групп реализуется по глобальной схеме — сразу по всей сети блоков. В конвейере по слоям необходимо столько точек глобальной синхронизации, сколько всего тактов в конвейере, что равно знаменателю дроби в формуле (4).

Синхронизация с помощью атомарных функций `atomicCAS` и `atomicExch` обеспечивает возможности, аналогичные замкам в стандарте OpenMP. Этот способ межблочной синхронизации реализуется по локальной схеме — для каждой дуги орграфа параллельного бегущего счета (см. рис. 4, б) в отдельности. В конвейере по слоям необходимо столько точек локальной синхронизации, сколько счетных тактов в конвейере, что равно числителю дроби в формуле (4).

Результаты исследований производительности программы ПАУК на GPU Nvidia V100 16 GB SXM для двух механизмов синхронизации приведены в табл. 3. Результаты получены при $n_x^{loc} = 32$; $n_{hyp}^{loc} = 8$; $n_z^{loc} = 4$; $n_z = 800$; $n_w^{loc} = 4$.

Из табл. 3 следует, что реализация с помощью атомарных функций обеспечивает более высокие характеристики производительности, так как требует меньше накладных расходов на синхронизацию и меньше регистров на счетное ядро.

Зернистость алгоритма распараллеливания между блоками составляет $n_{hyp}^{loc} n_z^{loc}$ ячеек фазового пространства задачи.

Предлагаемый алгоритм распараллеливания между блоками на основе геометрической декомпозиции существенно отличается от аналогичных алгоритмов из работ [10–12]. На GPU реализуется полноценный КВА-алгоритм на основе двумерной декомпозиции, а не только распараллеливание по элементам одной гиперплоскости. Для межблочной синхронизации используется механизм, не требующий завершения счетного ядра на GPU.

Таблица 3

Производительность тестовой программы ПАУК для двух способов синхронизации и различного числа блоков

Механизм синхронизации	Общее число блоков	R , млрд яч./с
Кооперативные группы	320	18,1
Атомарные функции	320	21,5
	400	24,5

Распараллеливание между группами блоков

Для уменьшения нагрузки на подсистему памяти в алгоритм распараллеливания между блоками введен дополнительный уровень распараллеливания по порциям направлений полета частиц одного октанта. Этот уровень позволяет логически разделить сеть блоков на равные группы, между которыми не требуется взаимодействия по данным и управлению в рамках конвейера по слоям.

Сеть блоков имеет размеры $\frac{n_x}{n_x^{loc}} \times \frac{n_y + n_x^{loc} - 1}{n_{hyp}^{loc}} \times \frac{n_w^{GPU}}{n_w^{loc}}$, где n_w^{GPU} обозначает общее число направлений для распараллеливания на двух уровнях — между группами нитей внутри блока и между группами блоков. Один блок имеет размеры $n_x^{loc} \times n_w^{loc}$.

Из табл. 4 видно, что алгоритм распараллеливания на основе геометрической декомпозиции обеспечивает достаточно эффективную загрузку GPU. В режиме комбинированного распараллеливания между блоками (по геометрии и направлениям) достигнута эффективность вычислений 30,6% от пиковой производительности при использовании всего 16 направлений полета частиц.

Зернистость алгоритма распараллеливания между группами блоков составляет $n_{hyp}^{loc} n_z^{loc} \left[\frac{n_z}{n_z^{loc}} + (D_{hyp} - 1) + \left[\frac{n_x^{loc} + n_{hyp}^{loc}}{n_{hyp}^{loc}} \right] (D_x - 1) \right]$ ячеек фазового пространства задачи.

Таблица 4

Производительность тестовой программы ПАУК на GPU Nvidia V100 16 GB SXM для алгоритмов распараллеливания между блоками ($n_x^{loc} = 32$; $n_{hyp}^{loc} = 8$; $n_w^{loc} = 4$; $n_z = 400$)

Алгоритм распараллеливания между блоками	Сеть блоков $D_x \times D_{hyp} \times n_w^{GPU} / n_w^{loc}$	n_w^{GPU}	$E_{КВА}$, %	R , млрд яч./с	FLOP Efficiency (Peak Double), %
По направлениям	$1 \times 1 \times 400$	1 600	100	38,9	45,4
По геометрии	$8 \times 50 \times 1$	4	82,6	24,5	—
Комбинированный	$4 \times 25 \times 4$	16	91,1	28,0	30,6

Распараллеливание между GPU

Для распараллеливания на уровне множества GPU предназначена модификация КВА-алгоритма. Исходная пространственная сетка разбивается по трем пространственным направлениям на геометрические подобласти. Декомпозиция выполняется регулярным образом, без перехлестов. На одном GPU рассчитывается одна геометрическая подобласть. Алгоритм реализуется в модели распределенной памяти с помощью средств стандарта MPI.

Из-за формата представления данных с выделением гиперплоскостей фиксированного размера были рассмотрены два способа разбиения по второму топологическому направлению:

1) декомпозиция по строкам: сначала выполняется разбиение сетки на геометрические подобласти, затем происходит смена формата в каждой подобласти (рис. 5, а). Число гиперплоскостей в задаче равно $N_y + P_2(n_x^{loc} - 1)$, где N_y — число строк; P_2 — число разбиений во втором топологическом направлении;

2) декомпозиция по гиперплоскостям: сначала происходит смена формата в задаче, а затем выполняется разбиение на подобласти (рис. 5, б). Число гиперплоскостей в задаче равно $N_y + n_x^{loc} - 1$.

Число вершин в графе, а значит, и объем вычислений меньше при декомпозиции по гиперплоскостям. Число гиперплоскостей, рассчитываемых в одном CUDA-блоке:

$$n_{hyp}^{loc} = \begin{cases} \frac{N_y + P_2(n_x^{loc} - 1)}{P_2 D_{hyp}} & \text{при декомпозиции по строкам;} \\ \frac{N_y + n_x^{loc} - 1}{P_2 D_{hyp}} & \text{при декомпозиции по гиперплоскостям.} \end{cases}$$

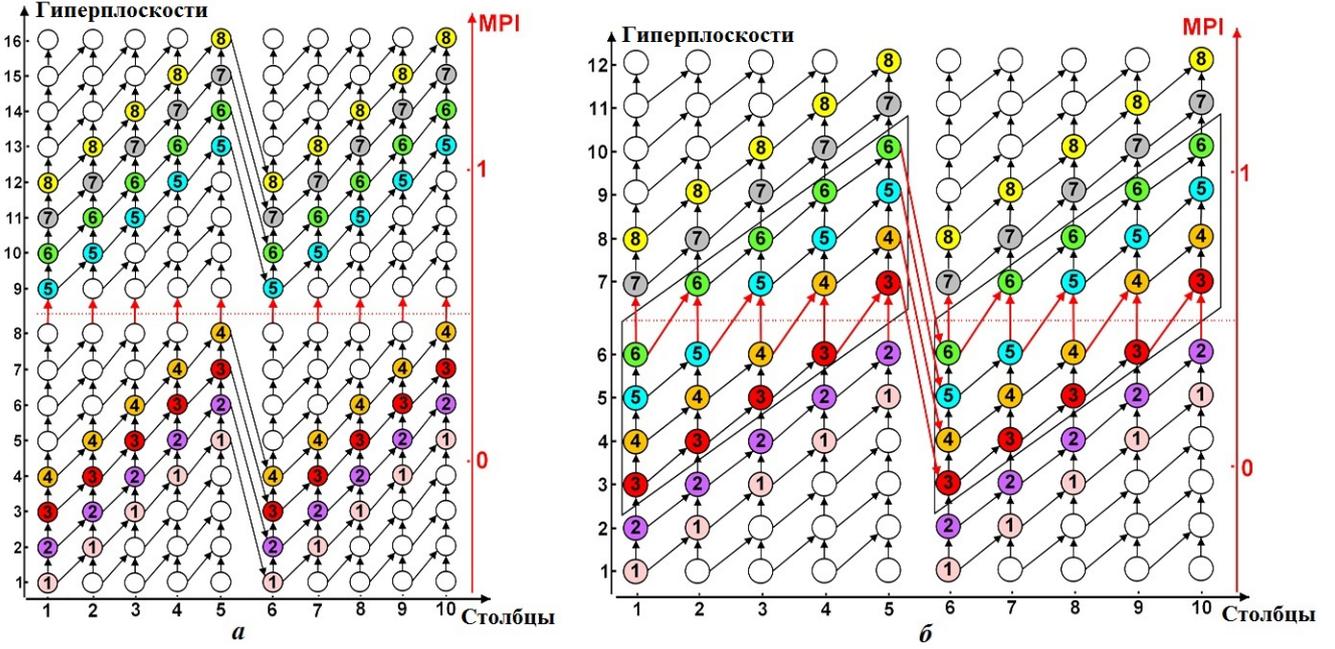


Рис. 5. Алгоритм бегущего счета при различных способах декомпозиции во втором топологическом направлении: *a* — по строкам; *b* — по гиперплоскостям

Число дуг в графе, которые соответствуют передаче потоков через MPI, значительно меньше при декомпозиции по строкам. На рис. 5, *a, b* такие дуги обозначены красным цветом. Объем данных (в байтах), передаваемых между двумя подобластями в рамках одного алгоритма параллельного бегущего счета:

$$V_{sweep} = \begin{cases} 8(n_x^{loc} D_x) n_z n_w^{GPU} & \text{при декомпозиции по строкам;} \\ 8[(2n_x^{loc} - 1)D_x + (D_x - 1)n_x^{loc}] n_z n_w^{GPU} & \text{при декомпозиции по гиперплоскостям.} \end{cases}$$

Нагрузка на коммуникационную подсистему при декомпозиции по гиперплоскостям практически в 3 раза больше, чем при декомпозиции по строкам. Кроме того, в силу необходимости представления данных в двух диагональных форматах (см. рис. 3) декомпозиция по гиперплоскостям приводит к частичному перехлесту подобластей. На рис. 5, *b* зона перехлеста обозначена штриховкой. Перехлест происходит из-за независимой декомпозиции орграфов для двух форматов представления данных: $i_{hyp} = i_x + i_y - 1$ и $i_{hyp} = n_x^{loc} - i_x + i_y$. Для каждой границы разбиения по гиперплоскостям потребуется дополнительный обмен информацией в объеме $V_{halo} = 8n_x^{loc} D_x (n_x^{loc} - 1) n_z$ байт. Этот обмен необходимо выполнить один раз на итерации по правой части решения уравнения переноса.

Еще одним недостатком разбиения по гиперплоскостям является усложнение графа параллельного бегущего счета и неполная загрузка GPU в случае $n_x > n_x^{loc}$ (см. рис. 5, *b*).

С учетом указанных обстоятельств для распараллеливания на рассматриваемом уровне архитектуры (множество GPU) выбрана декомпозиция по строкам, а не по гиперплоскостям.

Алгоритм параллельного бегущего счета по подобластям выполняется для всех слоев подобласти и для n_w^{GPU} направлений одного октанта. Конвейер организуется по направлениям углового октанта. В случае решения многогрупповой задачи количество тактов в конвейере может быть увеличено за счет групп, но не за счет слоев. Восемь (по числу угловых октантов) конвейеров обрабатываются в предопределенной последовательности и с пересечением, т. е. без точек глобальной синхронизации между ними.

Теоретическая эффективность алгоритма MPI-распараллеливания рассчитывается по формуле

$$E_{MPI} = \frac{8 \frac{n_w^{oct}}{n_w^{GPU}}}{8 \frac{n_w^{oct}}{n_w^{GPU}} + 4(P_1 - 1) + 4(P_2 - 1) + 2(P_3 - 1)} \cdot 100 \%, \quad (5)$$

где n_w^{oct} — число всех направлений в октанте; P_1, P_2, P_3 — параметры пространственной декомпозиции по столбцам, строкам и слоям.

Анализ формулы (5) показывает, что алгоритм MPI-распараллеливания обеспечивает теоретическую (без учета коммуникационной составляющей) масштабируемость до 70 000 GPU с эффективностью выше 60 % для задач с уровнем дискретизации *32 группы и квадратура* S_{16} .

Зернистость алгоритма MPI-распараллеливания можно определить двумя способами:

– относительно объема вычислений, выполняемых одной CUDA-нитью, —

$$n_{hyp}^{loc} n_z^{loc} \left[\frac{n_z}{n_z^{loc}} + (D_{hyp} - 1) + \left[\frac{n_x^{loc} + n_{hyp}^{loc}}{n_{hyp}^{loc}} \right] (D_x - 1) \right] \text{ячеек.}$$

– относительно объема вычислений, выполняемых всеми CUDA-нитьями за один вызов на GPU программы бегущего счета по ячейкам подобласти для n_w^{GPU} направлений, —

$$\frac{n_x (n_y + n_x^{loc} - 1) n_z n_w^{GPU}}{E_{КВА}/100 \%} \text{ячеек.}$$

Программная реализация MPI-обменов в тесте ПАУК выполнена по схеме прямой передачи информации между GPU, т. е. без промежуточного копирования в память универсального процессора.

В табл. 5 представлены результаты производительности и эффективности распараллеливания тестовой программы ПАУК в режиме Multi-GPU. Исследование выполнено методом слабой масштабируемости или методом увеличения задачи. Эффективность распараллеливания рассчитана по формуле

$$E_n = \frac{R_n}{nR_1} \cdot 100 \%,$$

где n — число GPU, т. е. MPI-процессов; $R_n = 8nn_x n_y n_z n_w^{oct} n_{it}/t$ — производительность, или скорость счета на n GPU; t — время счета в секундах.

При тестировании использовались сервер с двумя GPU Nvidia V100 16 GB SXM и сервер с четырьмя GPU Nvidia A100 40 GB SXM из состава программно-аппаратного полигона НЦФМ.

Переход с V100 на A100 потребовал увеличить размеры сети блоков из-за увеличения значений параметров N_{SM} и b_{SM} . При смене GPU прирост фактической производительности теста ПАУК в 1,325 раза оказался больше прироста в 1,244 раза пиковой производительности, что объясняется влиянием подсистемы памяти и особенностью расчета метрики R . Доля счетных вершин в тесте на GPU типа V100 составляет 0,845, на A100 — 0,871.

Эффективность распараллеливания на два MPI-процесса выше на сервере с A100, чем на сервере с V100, за счет двукратного увеличения пропускной способности интерфейса NVLink.

Таблица 5

Масштабируемость тестовой программы ПАУК ($n_x^{loc} = 32$; $n_{hyp}^{loc} = 8$; $n_w^{loc} = 4$; $n_w^{oct}/n_w^{GPU} = 10$)

Тип GPU	Сеть блоков	n_z	$E_{КВА}$, %	R_1 , млрд яч./с	E_2 , %	E_4 , %
V100	$4 \times 25 \times 4$	400	91,1	28,0	92,6–95,9	—
A100	$5 \times 30 \times 5$	500	91,1	37,1	94,8–96,5	89,3–93,0

Удельная характеристика производительности

Характерным критерием качества при оценке производительности GPU-версии приложения является сравнение с производительностью CPU-версии. Для тестовой программы ПАУК такое сравнение выполнено с помощью удельной характеристики. Вычислительная эффективность — это производительность приложения на один Гфлоп/с пиковой производительности задействованных вычислительных устройств.

В табл. 6 приведены результаты, полученные на универсальных процессорах по версии ПАУК с распараллеливанием MPI+MPI-3 SHM+OpenMP и векторизацией по направлениям полета частиц [13]. При тестировании использовался сервер с двумя CPU Intel Broadwell с суммарной пиковой производительностью 1,165 Тфлоп/с и сервер с двумя CPU Intel Skylake с суммарной пиковой производительностью 3,456 Тфлоп/с.

Для GPU указаны два значения вычислительной эффективности, рассчитанные относительно счетного и общего числа вершин (в скобках) в графах бегущего счета. На GPU Nvidia и CPU Intel Skylake получены очень близкие значения удельной производительности, что позволяет утверждать об эффективной адаптации тестовой программы ПАУК к архитектуре GPU.

Таблица 6

Вычислительная эффективность тестовой программы ПАУК, (млн яч.)/Гфлоп

2 CPU Intel Broadwell	2 CPU Intel Skylake	1 GPU Nvidia V100	1 GPU Nvidia A100
5,75	4,34	3,57 (4,22)	3,81 (4,38)

Заключение

В работе представлен иерархический алгоритм распараллеливания вычислений при решении трехмерного уравнения переноса нейтронов на гибридных супер-ЭВМ с GPU. Алгоритм основан на выделении уровней архитектуры и применении принципа геометрической декомпозиции. Программная реализация выполнена на примере тестовой программы ПАУК с помощью технологии CUDA.

Задача эффективной загрузки GPU решена при минимальном использовании ресурса направлений — 16 из 51 200 нитей на GPU Nvidia V100 и 20 из 96 000 нитей на GPU Nvidia A100. Зернистость алгоритмов распараллеливания по пространственным переменным (в ячейках фазового пространства) составила:

- для одной CUDA-нити на всех уровнях одного GPU:
 - 1 при распараллеливании по ячейкам между CUDA-нитями одной группы (warp);
 - 8 при распараллеливании по фрагментам между CUDA-блоками;
- для одного GPU на уровне MPI-процессов — $1,8 \cdot 10^8$ ($4,2 \cdot 10^8$) при распараллеливании по подобластям между различными GPU.

У вспомогательных алгоритмов распараллеливания по направлениям зернистость для одной CUDA-нити составила:

- 1 на уровне warp-ов одного блока;
- 3 512 (4 392) на уровне групп блоков.

Эффективность адаптации теста ПАУК к одному GPU подтверждена результатами профилирования и сравнением с производительностью CPU-версии программы. Для метрики *FLOP Efficiency (Peak Double)* получено значение 30,6 % от пиковой производительности GPU Nvidia V100. Для метрики *Вычислительная эффективность* (фактическая производительность приложения на 1 Гфлоп/с пиковой производительности) получены близкие результаты на GPU Nvidia V100, GPU Nvidia A100 и CPU Intel Skylake.

Масштабируемость тестовой программы ПАУК в режиме multi-GPU исследована методом умножения. В зависимости от способа декомпозиции эффективность распараллеливания составила 89,3–93,0% для четырех GPU. Алгоритм MPI-распараллеливания обеспечивает теоретическую (без учета коммуникационной составляющей) масштабируемость до 70 000 GPU с эффективностью выше 60% для задач с уровнем дискретизации *32 группы и квадратура S_{16}* .

Исследование выполнено в рамках научной программы Национального центра физики и математики (проект "Национальный центр исследования архитектур суперкомпьютеров").

Список литературы

1. Информация о CUDA. <https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>.
Informatsiya o CUDA. <https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>.
2. Бочков А. И., Нухудин А. А. Параллельный алгоритм решения трехмерного кинетического уравнения переноса. Программа ПАУК для тестирования многопроцессорных вычислительных систем // Межд. науч. конф. "Параллельные вычислительные технологии (ПАВТ'2008)": Тр. межд. науч. конф. (С.-Пб., 28 января — 1 февраля 2008 г.) Челябинск: ЮУрГУ, 2008. С. 39–47.
Bochkov A. I., Nuzhdin A. A. Parallelny algoritm resheniya trekhmernogo kineticheskogo uravneniya perenosa. Programma PAUK dlya testirovaniya mnogoprotsessornykh vychislitelnykh sistem // Mezhd. nauch. konf. "Parallelnye vychislitelnye tekhnologii (PAVT'2008)": Tr. mezhd. nauch. konf. (S.-Pb., 28 yanvarya — 1 fevralya 2008 g.) Chelyabinsk: YuUrGU, 2008.
3. Lamport L. The parallel execution of DO loops // Communications of the ACM. 1974. Vol. 17, No 2. P. 83–93.
4. Evans T. M., Joubert W., Hamilton S. P., Johnson S. R., Turner J. A., Davidson G. G., Pandya T. M. Three-dimensional discrete ordinates reactor assembly calculations on GPUs // ANS MC2015 — Joint Int. Conf. on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method. Nashville, Tennessee. April 19–23, 2015.
5. Searles R., Chandrasekaran S., Joubert W., Hernandez O. MPI + OpenACC: Accelerating radiation transport mini-application, minisweep, on heterogeneous systems // Comput. Phys. Commun. 2019. Vol. 236. P. 176–187.
6. Kunen A., Loffeld J., Black A., Chen R., Nowak P., Haut T., Bailey T., Brown P., Rennich S., Maginot P., Tagani B. Porting 3D discrete ordinates sweep algorithm in Ardra to CUDA // Proc. Int. Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering. Portland, USA, 2019. P. 2585–2598.
7. The OpenACC API Specification for Parallel Programming. <https://www.openacc.org/>.
8. Beckingsale D. A., Burmark J., Hornung R., Jones H., Killian W., Kunen A. J., Pearce O., Robinson P., Ryujin B. S., Scogland T. R. W. RAJA: Portable performance for large-scale scientific applications // 2019 IEEE/ACM Int. Workshop on Performance, Portability and Productivity in HPC (P3HPC). DOI 10.1109/P3HPC49587.2019.00012.
9. Zerr R. J., Baker R. S. SNAP: SN (Discrete Ordinates) Application Proxy: Description. Tech. Rep. LA-UR-13-21070. Los Alamos National Laboratories. 2013.
10. Deakin T., McIntosh-Smith S., Gaudin W. Many-core acceleration of a discrete ordinates transport mini-app at extreme scale // 31st Int. Conf. ISC High Performance 2016. Frankfurt, Germany, Proceedings. Cham: Springer International Publishing, 2016. P. 429–448.
11. Pennycook S., Mudalige G., Hammond S., Jarvis S. Parallelising wavefront applications on general-purpose GPU devices // 26th UK Performance Engineering Workshop (UKPEW10). 2010. P. 111–118.
12. Villa O., Johnson D. R., O'Connor M., Bolotin E., Nellans D., Luitjens J., Sakharnykh N., Wang P., Micikevicius P., Scudero A., Keckler S. W., Dally W. J. Scaling the power wall: a

path to exascale // Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis. IEEE Press, 2014. P. 830–841. <https://doi.org/10.1109/SC.2014.73>.

13. *Нуждин А. А.* Тестовая программа ПАУК как полигон для апробации алгоритмов и технологий параллельного программирования // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 48–61.
Nuzhdin A. A. Testovaya programma PAUK kak poligon dlya aprobatsii algoritmov i tekhnologiy parallelnogo programmirovaniya // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2020. Vyp. 4. S. 48–61.
14. *Koch K. R., Baker R. S., Alcouffe R. E.* Solution of the first-order form of the 3-D discrete ordinates equation on a massively parallel processor // Trans. of the Amer. Nuc. Soc. 1992. Vol. 65. P. 198.

Статья поступила в редакцию 16.03.22.

УДК 519.6

ОПТИМИЗАЦИЯ АЛГОРИТМОВ В ПРИКЛАДНОМ МЕТОДИЧЕСКОМ ТЕСТЕ MD ДЛЯ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ GPU

А. М. Ерофеев, М. В. Ветчинников
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Описываются алгоритмы тестовой программы молекулярной динамики (MD), позволившие осуществить полный перенос всех вычислений на GPU и тем самым избавиться от постоянной необходимости обмена между устройствами информацией о частицах. В результате взаимодействие между GPU и CPU необходимо только для организации пересылок граничной информации между отдельными GPU с помощью MPI на CPU, а это значительно меньше, чем пересылки в первоначальном коде, о чем свидетельствуют тестовые замеры эффективности. На разных по размеру задачах на одном GPU получено ускорение алгоритма относительно первоначального от 8,7 до 12,5 раза. При задействовании двух GPU такое ускорение составило от 6,6 до 12,5 раза. Эффективность распараллеливания на двух GPU V100 составила 76,3–79,6 % на задачах с числом частиц от 4 млн, на двух GPU A100 — 77,3–81,8 % на задачах с числом частиц от 13,5 млн.

Ключевые слова: молекулярная динамика, эффективность, GPU, CUDA, алгоритмы.

Введение

Впервые метод молекулярной динамики был применен в середине XX века [1]. Сразу стало очевидно, что без мощных вычислительных машин моделировать ансамбли микрочастиц (атомов и молекул) для получения полезной информации с помощью метода невозможно. Поэтому развитие молекулярной динамики неразрывно связано с развитием вычислительных средств: увеличение вычислительных мощностей открывает новые возможности для моделирования разных классов задач, что, в свою очередь, значительно расширяет сферу применения молекулярной динамики.

Существует несколько известных пакетов программ по классическому молекулярно-

динамическому моделированию, таких как DL-POLY [2], HOOMD [3], LAMMPS [4]. Основными целями разработчиков каждого из этих пакетов являлись не только написание алгоритмов, реализующих методы молекулярной динамики, но и их высокая эффективность для различного вида устройств. Из приведенных в [5] данных о вычислительной производительности разных пакетов программ на современных CPU и GPU видно, что, в частности, в пакете LAMMPS на GPU-устройствах достигаются значительные ускорения для различных потенциалов (табл. 1).

В РФЯЦ-ВНИИЭФ создаются вычислительные системы, которые позволяют решать поставленные задачи в более короткие сроки. Для того чтобы оценить эффективность работы вычислительных систем на реальных задачах, в

Таблица 1

Ускорения LAMMPS на GPU V100 для различных потенциалов [5]

Potential	Dual Cascade Lake 6240 (CPU-Only)	1 × V100 SXM2 32 GB	2 × V100 SXM2 32 GB
LJ 2.5	1x	3x	5x
EAM	1x	2x	5x
Tersoff	1x	5x	9x

РФЯЦ-ВНИИЭФ существует набор прикладных методических тестов.

Одним из таких тестов является тест MD [6], реализующий метод молекулярной динамики, который применяется для решения широкого класса задач, связанных:

- с моделированием свойств твердых тел и жидкостей;
- развитием и поведением дефектов (в том числе трещин) внутри твердого тела, на его поверхности, в тонких пленках и микроэлектронике;
- изучением свойств поверхностей и межфазных границ;
- изучением свойств атомных и молекулярных кластеров и их взаимодействием с поверхностями;
- моделированием процессов в живых клетках и т. д.

Благодаря тому, что тест MD обладает хорошей масштабируемостью, большим количеством независимых арифметических вычислений и относительно небольшими количеством обменов и объемом передаваемых при этом данных, он позволяет высокоэффективно использовать графические ускорители.

Основы теста MD

Тест MD создан на базе программы молекулярной динамики [7] для исследования производительности и эффективности распараллеливания многопроцессорных вычислительных систем. Он представляет собой численное интегрирование уравнений движения Ньютона для систем частиц с заданным законом межчастичного взаимодействия:

$$\begin{aligned} \frac{d\vec{r}_i}{dt} &= \vec{v}_i; \\ m_i \frac{d\vec{v}_i}{dt} &= - \sum_{j \neq i} \nabla_r u(r_{ij}); \\ \vec{F}_{ij} &= -\nabla_r u(r_{ij}), \end{aligned} \quad (1)$$

где \vec{r}_i , \vec{v}_i , m_i — соответственно координаты, скорость и масса i -й частицы; $\nabla_r u(r_{ij})$ — градиент межчастичного потенциала взаимодействия, зависящего от расстояния между частицами r_{ij} ; \vec{F}_{ij} — сила, действующая на частицу с номером i со стороны частицы с номером j ; dt — временной интервал.

Законы взаимодействия пары частиц или многочастичные взаимодействия определены в виде потенциалов взаимодействия, по которым находятся компоненты сил для каждой конкретной частицы. По результирующей силе рассчитываются скорость частицы и ее смещение в пространстве за небольшой временной промежуток — шаг по времени dt .

Наиболее трудоемкая часть в организации молекулярно-динамических вычислений содержится в блоке вычисления сил взаимодействия между частицами, а именно при нахождении потенциалов взаимодействия. При этом критичны по времени как перебор частиц с многократным доступом к памяти, так и сами расчеты потенциалов по довольно сложным формулам. Если в системе из N взаимодействующих друг с другом частиц каждая частица испытывает на себе действие любой другой частицы, то необходимо рассчитать N^2 таких взаимодействий. Но, как правило, начиная с какого-то расстояния между частицами, потенциалы взаимодействия имеют значение, близкое к нулю. Поэтому в большинстве программ молекулярной динамики вводится эффективный радиус взаимодействия (его еще называют радиусом обрезания потенциала), дальше которого потенциал принимает нулевое значение.

Вычисление потенциала не производится, если частицы находятся на расстоянии, большем радиуса обрезания. Это первое допущение, позволяющее уменьшить количество расчетов потенциалов. Но при этом расстояние между частицами все же рассчитывается, а для этого необходимо извлечь из памяти данные о соседней частице.

Для уменьшения такого перебора существует несколько алгоритмов. Один из них — это создание так называемых списков Верле [8]: для каждой частицы создается список частиц, взаимодействующих с ней в течение определенного количества шагов. Данные списки требуют дополнительной памяти, и, кроме того, их необходимо периодически обновлять.

Другой алгоритм, который как раз реализован в тесте MD, основан на сеточном подходе. Область моделируемого пространства разбивается на параллелепипеды (ячейки) со стороной, равной или чуть большей радиуса обрезания. Поэтому потенциал взаимодействия ищется между частицами, находящимися в своей ячейке и в двадцати шести соседних ячейках. В этом случае количество операций поиска взаимодействующих частиц пропорционально n^2 , где n — коли-

чество частиц в ячейке, что значительно меньше, чем общее число частиц N .

В памяти информация о частицах располагается последовательно в виде списка, когда есть адрес начала списка для ячейки и у каждой частицы (соответствующего ей элемента списка) есть указатель на следующую частицу в списке. Для ячейки также хранится количество частиц, которые она содержит, и адрес последней частицы для удобства изменения списка. Последняя частица в списке для ячейки ссылается на пустой адрес *null*.

При перелете частицы часто возникает ситуация, когда она покидает одну ячейку и переходит в другую. Здесь необходимо скорректировать информацию о принадлежности частиц ячейкам, изменив соответствующие элементы массивов указателей.

Правильно выбранная стратегия развития программы молекулярной динамики позволила создать комплекс программ MoDyS [9] для проведения численных экспериментов по получению различных характеристик исследуемых материалов [10] с высокой эффективностью распараллеливания.

Перенос программ на GPU

Первоначальная адаптация. Адаптация алгоритмов молекулярной динамики для графических ускорителей впервые была выполнена в 2008—2009 гг. [11]. В тот момент только началось использование данного типа архитектуры для моделирования физических процессов и задач. После анализа возможности портирования кода комплекса программ MoDyS на GPU-устройство первой для этой цели была выбрана самая трудоемкая часть кода — модуль расчета сил. Расчет сил занимает в среднем от 80 до 95 % общего времени выполнения счетного шага и зависит от типа потенциала взаимодействия, атомной структуры и радиуса обрезания в расчете потенциала взаимодействия.

Схема счетного шага с использованием GPU стала выглядеть, как показано на рис. 1. В начале шага с CPU, или хоста (host), на устройство (device) GPU копируется информация с координатами и типами частиц. На GPU осуществляется расчет потенциалов и по ним — сил взаимодействия. Расчет сил взаимодействия между частицами может выполняться независимо, поэтому данный процесс на GPU хорошо распа-



Рис. 1. Схема счетного шага с использованием GPU

раллеливается. Для расчета потенциалов было написано счетное ядро нахождения потенциала одной частицы. При этом число запускаемых параллельных процессов, или нитей, на GPU равняется количеству частиц, рассчитываемых на MPI-процессе. После расчета потенциалов компоненты сил отсылаются обратно на CPU. Следующее действие на шаге — это расчет на CPU скоростей частиц и по ним — новых координат. Применение сеточного подхода при реализации алгоритмов молекулярной динамики вызвало также необходимость расчета перехода частиц из одной ячейки в другую и между процессами.

Оказалось, что первоначальная адаптация к использованию вычислительных мощностей GPU имеет право на существование и показывает неплохие ускорения по отношению к CPU. При этом не потребовалось дорабатывать сервисную часть комплекса. Кроме того, у данной адаптации остался еще резерв ускорения.

Дело в том, что устройство GPU имеет смешанную архитектуру параллелизма: в нем не все нити соответствуют принципу SIMD (Single Instruction Multiple Data), физически одновременно выполняются лишь только нити в пределах одного так называемого *warp* (группа из 32 нитей). Нити различных *warp* могут быть на разных этапах выполнения кода программы. Такой метод обработки данных производитель GPU Nvidia называет SIMT (Single Instruction Multiple Threads).

Таким образом, очередная цель — получить ускорение кода, в том числе по возможности за счет SIMD-составляющей.

Для достижения данной цели прежде всего нужно иметь в виду, что у первоначальной адаптации к GPU есть большой недостаток — необходимость обмена информацией, находящейся на всем устройстве, т. е. пересылки координат частиц и сил (см. рис. 1). Так как по вычислительной мощности GPU превосходит CPU и на него передается много данных, то это становится фактором, уменьшающим эффективность использования GPU.

Одним из путей совершенствования кода молекулярной динамики становится полный перенос всех вычислений на GPU, который избавляет от постоянной необходимости обмена информацией о частицах между счетными устройствами. Тогда обмен между устройством и хостом необходим только для организации пересылок граничной информации между отдельными GPU с

помощью MPI на CPU, а это значительно меньше, чем пересылки в первоначальном варианте кода с адаптацией к GPU (*старом* алгоритме).

Реализация нового алгоритма. Основная задача, которая решалась авторами, — избавиться от необходимости пересылки большого объема информации. Для этого все счетные модули, выполняемые на CPU, были последовательно адаптированы к устройству GPU.

В качестве ускорительных устройств используются графические ускорители фирмы Nvidia. Инструментом портирования кода на GPU является программно-аппаратная архитектура параллельных вычислений CUDA.

CUDA-программа может запускаться как на CPU, так и на GPU. Параллельный код можно перенести на GPU, где он будет выполняться множеством нитей (threads), менее мощных по сравнению с ядрами CPU, но число которых больше. Достижимое на GPU ускорение обусловлено возможностью одновременного запуска на обработку большого количества информации: выигрыш относительно CPU получается как за счет большего количества нитей, так и за счет большей ширины векторного регистра.

Будем считать, что область решения задачи содержит *npart* частиц и разбита на *ncell* ячеек.

Проводится адаптация решения уравнений (1) для GPU, которое ранее выполнялось на CPU. Кроме решения самих уравнений, на GPU переводятся все вспомогательные модули, отвечающие за отслеживание информации о частицах в памяти, формирование списков частиц и т. п.

После того как силы рассчитаны, решение уравнений можно выполнить независимо, параллельно запустив *npart* нитей (по числу частиц). При этом для каждой частицы по ее новому местоположению можно узнать, осталась ли она в той же ячейке или мигрировала в другую. Если частица покинула ячейку, то возможны два варианта: она осталась на своем MPI-процессе или перешла на другой MPI-процесс.

Сначала разберем вариант, когда частица остается на своем MPI-процессе, но переходит из одной ячейки в другую. В таком случае программы, обрабатывающие ячейки, будут запускаться на GPU с числом нитей, равным числу ячеек — *ncell*. Каждая нить будет обрабатывать цепочку частиц, содержащихся в "ее" ячейке.

Вариант с параллельным запуском нитей по числу частиц проблематичен, так как возникают конфликты по указателям для ячеек и ча-

стиц. Однако таких конфликтов можно избежать с помощью атомарных операций `atomic`, которые блокируют нити при одновременном обращении к памяти, образуя при этом очередь.

Было реализовано два варианта кода: с числом нитей, равным числу частиц (с применением атомарных операций), и с числом нитей, равным числу ячеек (без атомарных операций). Вариант с обработкой по ячейкам оказался несколько быстрее варианта с применением `atomic`. На одном GPU прием с запуском нитей по числу ячеек отработан на модуле расчета переходов в другую ячейку. Дальше такой способ будет применяться и для других целей.

Теперь рассмотрим случай, когда частица переходит на другой MPI-процесс. Выполняемый в этом случае алгоритм схематично показан на рис. 2 и заключается в следующем.

На GPU формируется буфер частиц, покидающих связанный с устройством MPI-процесс. Процесс формирования буфера разделен на две части. Первая часть — это отбор частиц, покидающих MPI-процесс. Такой отбор запускается с числом нитей, равным числу ячеек, но фактически полезная информация поступает из ячеек, находящихся на границе с другими MPI-процессами. Вторая часть запускается с тем же числом нитей, равным числу ячеек, — она и фор-



Рис. 2. Алгоритм пересылки частиц на другой MPI-процесс

мирует окончательно буфер, записывая информацию о пересылаемых частицах. На каждую частицу приходится восемь элементов: номер ячейки, куда частица переходит, номер материала частицы и по три координаты и компоненты скорости.

Далее сформированный буфер копируется с устройства на хост, и на нем осуществляются обмены средствами MPI.

После того как осуществлена пересылка между MPI-процессами, хост копирует на устройство принятый буфер, который разбирается на GPU в два приема. Сначала выполняется копирование информации из буфера в массивы частиц, которое запускается с числом нитей, равным числу пришедших частиц. Затем запускаются нити на тех ячейках, в которых идет добавление прилетевших частиц, и формируются новые цепочки частиц для этих ячеек.

Следующий модуль, который нужно было адаптировать, — это модуль работы с *несчетными* ячейками (содержащими частицы, которые рассчитываются на соседнем процессоре). Здесь также возможны два варианта: 1) когда есть граница с периодическими условиями, но нет распараллеливания в этом направлении; 2) многопроцессорный случай. Во втором случае наличие периодических условий на границе приводит лишь к дополнительным пересылкам, так как границы, на которых заданы связанные условия периодичности, рассчитываются разными MPI-процессами.

В варианте 1 происходит дублирование информации из счетных ячеек на границе в несчетные ячейки на смежной по условиям периодичности границе (координаты частицы при этом изменяются на размер смещения по условиям периодичности). Это осуществляется запуском на GPU числа нитей, равного числу ячеек на границе.

В варианте 2 (многопроцессорный случай) сначала на GPU запускается программа формирования буфера с числом нитей, равным числу ячеек для пересылки. Так как количество частиц в ячейках известно, то буфер сразу формируется для плотного заполнения. Для передачи необходимо всего пять элементов: номер ячейки, куда частица переходит, номер материала частицы и три координаты. Сформированный буфер копируется устройством на хост. Там осуществляется обмен информацией между процессами, и полученный в результате буфер хост возвращает на устройство. После этого копирование информации о частицах и ячейках за-

пускается на нитях, число которых равно числу несчетных ячеек.

В результате адаптации всех счетных и вспомогательных программ получен счетный код, полностью работающий на GPU и использующий CPU только для обмена информацией между процессами. Тем самым пропала необходимость копирования большого объема информации с GPU на CPU и обратно.

Модификация работы с памятью. При проведении тестирования нового алгоритма было отмечено, что первый счетный шаг проходит быстрее, чем последующие. Было проведено исследование данного явления и определено, что списочная структура хранения частиц в ячейках в процессе моделирования приводит к запутанности списков. После перехода частиц из ячейки в ячейку может оказаться, что частицы, расположенные рядом в списке, в физической памяти устройства находятся далеко друг от друга. На рис. 3 показано, как запутаны списки и как в процессе расчета сил взаимодействия постоянно идет обращение к памяти с далеко отстоящими друг от друга адресами, что сильно замедляет счет.

Для решения этой проблемы новый алгоритм был модифицирован: в конце каждого шага была введена сортировка частиц в ячейках на устройстве, учитывающая все переходы как между ячейками внутри GPU, так и между ячейками на разных GPU. Тем самым на следующем

шаге для каждой ячейки достигнуто последовательное размещение всех частиц в физической памяти устройства (рис. 4).

Результаты тестирования и анализ эффективности новых алгоритмов

Для тестирования эффективности реализованных алгоритмов были выбраны следующие начальные данные: образец меди с границей кубической решеткой, характерный шаг атомной решетки $a_0 = 3,615 \text{ \AA}$; радиус обрезания $7,23 \text{ \AA}$; атомная масса $63,546 \text{ а.е.м.}$; $T_0 = 300 \text{ К}$. Граничные условия — периодические. Рассматривались образцы разных размеров с количеством частиц от 442 тыс. до 210 млн. Используемый потенциал взаимодействия — парный потенциал Морзе. Количество счетных шагов в каждом расчете равно 100.

При верификации реализованных алгоритмов полученные по ним результаты расчетов полностью совпали с результатами расчетов на CPU.

Тестирование проводилось на сервере с двумя графическими ускорителями Nvidia Tesla V100 и двумя CPU Intel Skylake 6132.

Ускорение счета алгоритмов рассчитывалось по формуле

$$k = \frac{T_{old}}{T_{new}},$$

где T_{old} , T_{new} — время счета на GPU со старым и новым алгоритмом соответственно.

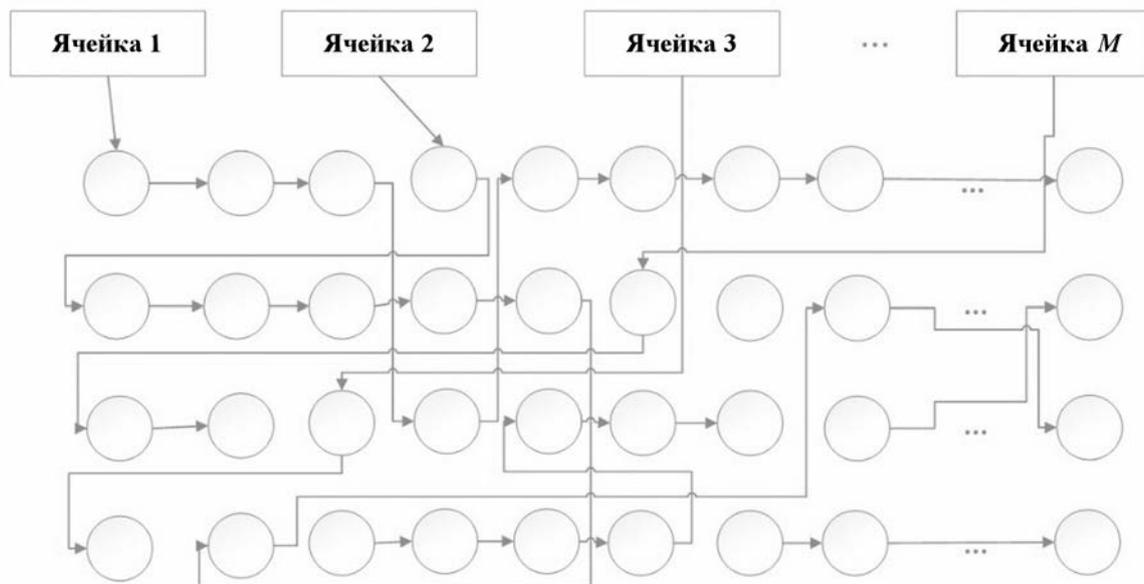


Рис. 3. Запутанность списков частиц в ячейках

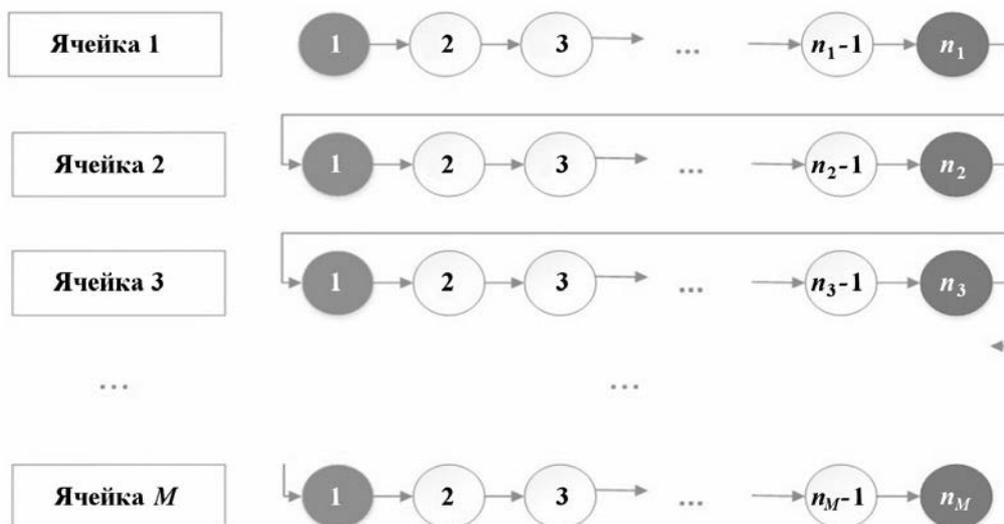


Рис. 4. Непрерывный список ячеек

Тест 1. Выполнена профилировка реализованного нового алгоритма на GPU и сравнение ее с данными замеров этапов выполнения старого алгоритма (табл. 2).

Как видно из таблицы, время копирования информации в старом алгоритме занимало около 50 % времени счетного шага. Причина — в передаче большого объема данных о всех рассчитываемых частицах.

Алгоритмы расчета уравнений движения и обновления связанных списков в старом алгоритме выполнялись на одном ядре CPU, а в новом — на GPU. Из-за наличия большого количества логических операций, выполняющихся в определенной последовательности, получено небольшое ускорение относительно CPU (в 5 раз). Тем не менее перенос всех вычислений на GPU позволил полностью избавиться от копирования информации между устройствами и обеспечил значительный прирост общей производительности тестовой программы.

Тесты 2, 3. Было также проведено тестирование нового алгоритма на разных по объему задачах. Результаты сравнительных расчетов приведены в табл. 3. Видно, что на одном GPU получено ускорение нового алгоритма относительно старого от 4,7 до 7,5 раза.

В табл. 3 также приведены результаты исследования производительности нового алгоритма после модификации работы с памятью, которые свидетельствуют о том, что удалось ускорить код программы в 1,41–1,85 раза. Из табл. 3 также видно, что ускорение модифицированного кода относительно старого составляет 8,7–12,5 раза.

Тест 4. Проведено сравнительное тестирование нового и старого алгоритмов в режиме multi-GPU на двух устройствах. Результаты показаны в табл. 4. Видно, что в случае использования двух GPU ускорение нового алгоритма относительно старого составляет от 4,7 до 9,1 раза.

Таблица 2

Профилировка счетного шага при расчете на одном GPU (время — в секундах)

Алгоритм на GPU	Копирование информации на GPU	Расчет сил	Копирование информации на CPU	Расчет уравнений движения и обновление связанных списков	Время счетного шага
Старый	$1,19 \cdot 10^{-1}$	$4,48 \cdot 10^{-2}$	$7,09 \cdot 10^{-2}$	$2,25 \cdot 10^{-1}$	0,47
Новый	Только на первом шаге	$4,51 \cdot 10^{-2}$	—	$4,4 \cdot 10^{-2}$	0,09

Таблица 3

Время (в секундах) и ускорение выполнения 100 шагов при расчете на одном GPU

Алгоритм GPU	Размер образца, a_0 (число частиц)						
	48 (442,368 тыс.)	86 (2,544 млн)	100 (4 млн)	150 (13,5 млн)	172 (20,353 млн)	200 (32 млн)	210 (37,044 млн)
Старый	4,281	28,87	33,87	110,69	162,40	252,02	289,9
Новый	0,635	3,87	6,76	20,75	34,14	47,84	61,49
Модифицированный	0,449	2,313	3,711	12,21	18,38	28,23	33,23
Ускорение:							
новый/старый	6,8	7,5	5,0	5,3	4,8	4,8	4,7
модифицированный/новый	1,41	1,67	1,82	1,7	1,86	1,7	1,85
общее	9,53	12,48	9,13	9,07	8,84	8,93	8,72

Таблица 4

Время (в секундах) и ускорение выполнения 100 шагов при расчете на двух GPU

Алгоритм GPU	Размер образца, a_0 (число частиц)						
	48 (442,368 тыс.)	86 (2,544 млн)	100 (4 млн)	150 (13,5 млн)	172 (20,353 млн)	200 (32 млн)	210 (37,044 млн)
Старый	2,62	13,09	29,55	61,29	90,37	138,19	159,97
Новый	0,51	1,99	3,25	11,4	18,34	29,47	32,78
Ускорение	5,13	6,58	9,1	5,37	4,93	4,7	4,88

Тест 5. Исследование эффективности реализованных алгоритмов проводилось на серверах с двумя GPU Nvidia Tesla V100 и четырьмя GPU Nvidia A100 40 GB SXM из состава программно-аппаратного полигона Национального центра физики и математики (НЦФМ) г. Сарова. Результаты приведены в табл. 5, 6, где t_1 , t_2 — время 100 шагов на одном и двух GPU соответственно. Эффективность распараллеливания вычисляется по формуле

$$E2 = \frac{R2}{2 \cdot R1} \cdot 100 \%,$$

где $R1$ — метрика производительности, равная числу обработанных частиц за секунду на одном GPU, $R2$ — на двух GPU.

Из табл. 5, 6 видно, что GPU V100 полностью загружается для задач с числом частиц от 4 млн, а полная загрузка GPU A100 начинается для задач с числом частиц от 13,5 млн. Эффективность распараллеливания на двух GPU V100 для задач с числом частиц от 4 млн составляет 76,3–79,6%, на двух GPU A100 для задач с числом частиц от 13,5 млн — 77,3–81,8%.

Таблица 5

Результаты тестирования на V100

Характеристика	Размер образца, a_0 (число частиц)						
	48 (442,368 тыс.)	86 (2,544 млн)	100 (4 млн)	150 (13,5 млн)	172 (20,353 млн)	200 (32 млн)	210 (37,044 млн)
t_1 , с	0,449	2,313	3,711	12,21	18,38	28,23	33,23
t_2 , с	0,398	1,586	2,366	7,67	11,67	18,49	21,31
$R1$, частиц/с	$9,85 \cdot 10^7$	$1,10 \cdot 10^8$	$1,08 \cdot 10^8$	$1,11 \cdot 10^8$	$1,11 \cdot 10^8$	$1,13 \cdot 10^8$	$1,11 \cdot 10^8$
$R2$, частиц/с	$1,11 \cdot 10^8$	$1,60 \cdot 10^8$	$1,69 \cdot 10^8$	$1,76 \cdot 10^8$	$1,74 \cdot 10^8$	$1,73 \cdot 10^8$	$1,74 \cdot 10^8$
$E2$, %	56,4	72,9	78,4	79,6	78,7	76,3	78,0

Результаты тестирования на A100

Характеристика	Размер образца, a_0 (число частиц)						
	48 (442,368 тыс.)	86 (2,544 млн)	100 (4 млн)	150 (13,5 млн)	172 (20,353 млн)	200 (32 млн)	210 (37,044 млн)
$t1, c$	0,434	1,928	3,081	10,56	16,54	25,45	30,04
$t2, c$	0,435	1,533	2,179	6,6	10,11	16,46	18,72
$R1, \text{ частиц}/c$	$1,02 \cdot 10^8$	$1,32 \cdot 10^8$	$1,3 \cdot 10^8$	$1,28 \cdot 10^8$	$1,23 \cdot 10^8$	$1,26 \cdot 10^8$	$1,23 \cdot 10^8$
$R2, \text{ частиц}/c$	$1,02 \cdot 10^8$	$1,66 \cdot 10^8$	$1,84 \cdot 10^8$	$2,05 \cdot 10^8$	$2,01 \cdot 10^8$	$1,94 \cdot 10^8$	$1,98 \cdot 10^8$
$E2, \%$	49,9	62,9	70,7	80,0	81,8	77,3	80,2

Заключение

В результате переноса всех счетных и вспомогательных модулей на GPU и оптимизации работы с памятью получена новая версия теста MD. Таким образом, при избавлении от копирования большого объема информации с GPU на CPU и обратно на одном GPU получено ускорение нового алгоритма относительно старого от 8,7 до 12,5 раза в зависимости от размеров задач. При использовании двух GPU получено ускорение нового алгоритма относительно старого от 6,6 до 12,5 раза. Эффективность распараллеливания на двух GPU V100 составила 76,3—79,6 % для задач с числом частиц от 4 млн, на двух GPU A100 — 77,3—81,8 % для задач с числом частиц от 13,5 млн. Версии программ сравнивались между собой, результаты тестовых расчетов по разным кодам совпадают.

Для повышения эффективности MPI-распараллеливания теста в режиме multi-GPU планируется реализовать MPI-обмены данными, расположенными в памяти GPU, т. е. не использовать CPU для обмена информацией.

Все наработки по ускорению кода, сделанные в тесте MD, будут перенесены в комплекс программ MoDyS, для которого предполагается дальнейшее развитие с целью моделирования задач не только классической молекулярной динамики, но и механики сплошной среды на основе разрабатываемого метода гамильтоновой динамики [12].

В данной статье описываются подходы, которые были применены для GPU, но их также можно использовать для оптимизации вычислений на CPU. Исследование таких подходов будет проведено в следующей работе.

Настоящее исследование выполнено в рамках научной программы НЦФМ г. Сарова (проект "Национальный центр исследования архитектур суперкомпьютеров").

Список литературы

1. Alder B. J., Wainwright T. E. Studies in Molecular Dynamics. I. General Method // J. Chem. Phys. 1957. Vol. 27. P. 1208.
2. Сайт DL_POLY. https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx.
Сайт DL_POLY. https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx.
3. Сайт HOOMD. <http://glotzerlab.engin.umich.edu/hoomd-blue>.
Сайт HOOMD. <http://glotzerlab.engin.umich.edu/hoomd-blue>.
4. Сайт LAMMPS. <https://www.lammps.org>.
Сайт LAMMPS. <https://www.lammps.org>.
5. Официальный сайт Nvidia. <https://developer.nvidia.com/hpc-application-performance>.
Официальный сайт Nvidia. <https://developer.nvidia.com/hpc-application-performance>.
6. Алексеев А. В., Беляев С. П., Бочков А. И., Быков А. Н., Ветчинников М. В., Залялов А. Н., Нурждин А. А., Огнев С. П., Самсонова Н. С., Сапронов И. Д., Чистякова И. Н., Шемякина Т. В., Шагалиев Р. М., Янилкин Ю. В. Методические прикладные тесты РФЯЦ-ВНИИЭФ для численного исследования параметров высокопроизводительных вычислительных систем // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 2. С. 86—100.
Alekseev A. V., Belyaev S. P., Bochkov A. I., Bykov A. N., Vetchinnikov M. V., Zalyalov A. N., Nuzhdin A. A., Ognev S. P., Samsonova N. S., Sapronov I. D., Chistyakova I. N., Shemyakina T. V., Shagaliev R. M., Yanilkin Yu. V.

- kin Yu. V. Metodicheskie prikladnye testy RFYaTs-VNIIEF dlya chislennogo issledovaniya parametrov vysokoproizvoditelnykh vychislitelnykh sistem // *Voprosy atomnoy nauki i tekhniki*. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2020. Vyp. 2. S. 86–100.
7. *Алейников А. Ю., Барабанов Р. А., Бутнев О. И., Быков А. Н., Веселов Р. А., Воронин Б. Л., Ганчук Н. С., Делов В. И., Ерофеев А. М., Пронин В. А., Рудько Н. М., Селезнев А. А., Скрыпник С. И.* Программа МДП-СОВЦ решения задач молекулярной динамики на параллельных ЭВМ с распределенной памятью // Там же. 2001. Вып. 1. С. 3–13.
- Aleynikov A. Yu., Barabanov R. A., Butnev O. I., Bykov A. N., Veselov R. A., Voronin B. L., Ganchuk N. S., Delov V. I., Erofeev A. M., Pronin V. A., Rudko N. M., Seleznev A. A., Skrypnik S. I.* Programma MDP-SOVTs resheniya zadach molekulyarnoy dinamiki na parallelnykh EVM s raspredelennoy pamyatyu // Там же. 2001. Vyp. 1. S. 3–13.
8. *Verlet L.* Computer "experiments" on classical fluids. Thermodynamical properties of Lennard-Jones molecules // *Phys. Rev.* 1967. Vol. 159. P. 98–103.
9. *Анисимов А. Н., Грушин С. А., Воронин Б. Л., Копкин С. В., Ерофеев А. М., Дёмин Д. А., Дёмина М. А., Здорова М. В., Ветчинников М. В., Еричева Н. С., Коваленко Н. О., Крючков И. А., Кечин А. Г., Дегтярёв В. А., Урм В. Я.* Комплекс программ молекулярно-динамического моделирования (MoDyS). Свидетельство о государственной регистрации программы для ЭВМ № 2010614974 // *Электронный бюллетень*. 2010. http://www.fips.ru/wps/wcm/connect/content_ru/ru.
- Anisimov A. N., Grushin S. A., Voronin B. L., Kopkin S. V., Erofeev A. M., Dyemina M. A., Zdorova M. V., Vetchinnikov M. V., Elicheva N. S., Kovalenko N. O., Kryuchkov I. A., Kechin A. G., Degtyarev V. A., Urm V. Ya.* Kompleks programm molekulyarno-dinamicheskogo modelirovaniya (MoDyS). Svidetelstvo o gosudarstvennoy registratsii programmy dlya EVM № 2010614974 // *Elektronny byulleten*. 2010. http://www.fips.ru/wps/wcm/connect/content_ru/ru.
10. *Ветчинников М. В., Дёмина М. А., Анисимов А. Н., Грушин С. А., Кечин А. Г., Фомин В. П., Дегтярёв В. А.* Молекулярно-динамическое моделирование процесса самозатравивания ударника из W при проникании в мишень из Fe // *Вопросы атомной науки и техники*. Ser. Теоретическая и прикладная физика. 2017. Вып. 3. С. 23–34.
- Vetchinnikov M. V., Dyemina M. A., Anisimov A. N., Grushin S. A., Kechin A. G., Fomin V. P., Degtyarev V. A.* Molekulyarnodinamicheskoe modelirovanie protsesssa samozatrichivaniya udarnika iz W pri pronikanii v mishen iz Fe // *Voprosy atomnoy nauki i tekhniki*. Ser. Teoreticheskaya i prikladnaya fizika. 2017. Vyp. 3. S. 23–34.
11. *Воронин Б. Л., Ерофеев А. М., Копкин С. В., Крючков И. А., Рыбкин А. С., Степаненко С. А., Южаков В. В.* Применение графических арифметических ускорителей для расчета задач молекулярной динамики по программному комплексу МД // Там же. Ser. Математическое моделирование физических процессов. 2009. Вып. 2. С. 55–61.
- Voronin B. L., Erofeev A. M., Kopkin S. V., Kryuchkov I. A., Rybkin A. S., Stepanenko S. A., Yuzhakov V. V.* Primenenie graficheskikh arifmeticheskikh uskoriteley dlya rascheta zadach molekulyarnoy dinamiki po programmnomu kompleksu MD // Там же. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2009. Vyp. 2. S. 55–61.
12. *Ветчинников М. В., Софронов В. Н., Дёмина М. А.* Использование методов гамильтоновой динамики в численных расчетах задач механики сплошной среды // Там же. 2020. Вып. 4. С. 5–21.
- Vetchinnikov M. V., Sofronov V. N., Dyemina M. A.* Ispolzovanie metodov gamiltonovoy dinamiki v chislennykh raschetakh zadach mekhaniki sploshnoy sredy // Там же. 2020. Vyp. 4. S. 5–21.

УДК 519.6

УЛУЧШЕНИЕ КАЧЕСТВА ГРАНЕЙ ОБЪЕМНЫХ ЯЧЕЕК ПРИ ГЕНЕРАЦИИ НЕСТРУКТУРИРОВАННЫХ СЕТОК В ПАКЕТЕ ПРОГРАММ "ЛОГОС"

О. Н. Борисенко, М. В. Кузьменко, М. В. Черенкова, А. Г. Гиниятуллина,
Н. В. Чухманов, Д. Н. Смолкина, Т. Е. Тимаева, К. А. Блажнова
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Приводится описание способов улучшения качества граней объемных ячеек при автоматической генерации неструктурированных сеток в пакете программ "Логос", используемом при численном моделировании задач аэро- и гидродинамики. Это модификация ячеек с частично совпадающими гранями, объединение граней с учетом их формы, разрезание искривленных граней, разрезание невыпуклых граней с добавлением нового узла с возможностью удаления малого ребра, разрезание искривленных граней на множество плоских в зонах перехода негладких областей в гладкую область, удаление лишних узлов. Применение совокупности указанных способов позволяет улучшить форму граней ячеек, полученных методом отсечения, и, как следствие, построенного на их основе слоя ячеек, для которых характерен сильный градиент скорости потока. В итоге повышается качество расчетной сетки в целом.

Ключевые слова: пакет программ "Логос", неструктурированные сетки, улучшение качества сетки, поверхностные грани, пограничный слой.

Введение

Построение качественной расчетной сетки для проведения расчетов трехмерных задач является одним из наиболее трудоемких и затратных по времени этапов подготовки численной модели, поскольку от этого напрямую зависит успешность дальнейшего расчета.

Для моделирования задач аэро- и гидродинамики требуются сетки специального вида с неоднородной структурой, состоящие из ячеек двух типов:

- образующих пограничный слой — слой ячеек в форме многогранных призм на поверхности обтекаемого тела вблизи моделируемого объекта, где проявляется эффект вязкости, характеризуемый сильным градиентом скорости потока;
- заполняющих остальную область моделирования и имеющих форму куба, тетраэдра или, в общем случае, произвольного многогранника.

Такая структура сетки позволяет упростить моделирование течения жидкости/газа путем

разделения потока на две области — тонкого вязкого пограничного слоя и невязкого течения.

В пакете программ "Логос" [1] для подготовки расчетных сеток при решении задач аэро- и гидродинамики применяются автоматические генераторы поверхностных треугольных [2] и объемных сеток [3–6].

При генерации объемных сеток чаще всего применяется автоматический генератор неструктурированных сеток, использующий метод отсечения [3], поскольку он позволяет получать расчетные сетки с требуемыми характеристиками при задании минимального количества параметров для геометрических моделей произвольной конфигурации. Исходными данными для генератора является поверхностная сетка, состоящая из треугольных ячеек. Она аппроксимирует поверхность геометрической модели с учетом кривизны поверхностей и характерных кривых. Характерные кривые — это ломаные, проходящие по особенностям геометрической модели (негладкие места стыковки поверхностей, острые ребра геометрии).

При генерации сеток методом отсечения вся область моделирования заполняется шаблонной сеткой, ячейки которой имеют форму куба. Вблизи моделируемого объекта формируется пограничный слой ячеек в форме многоугольных призм. Ячейки на границе шаблонной сетки и пограничного слоя являются многогранниками произвольной формы. Они формируются в результате отсечения ячеек шаблонной сетки треугольниками поверхностной сетки, поэтому будем называть их отсеченными.

Ячейки объемной сетки содержат два типа граней — внутренние и поверхностные. Грани, разделяющие две ячейки, назовем внутренними, а грани, аппроксимирующие поверхностную сетку из треугольников на стыке отсеченных ячеек и пограничного слоя, — поверхностными. Качество отсеченных ячеек напрямую влияет на качество ячеек пограничного слоя, которые строятся на основе поверхностных граней отсеченных ячеек.

Построение многогранной ячейки методом отсечения имеет свои особенности. Наиболее существенная из них — в том, что узлы отсеченной грани получаются из точек пересечения ребер шаблонной ячейки и набора треугольников, а также характерных кривых. Поэтому в некоторых случаях, например при обрыве характерной кривой внутри ячейки, отсеченная грань получается сильно неплоской и невыпуклой, что существенно затрудняет дальнейшую работу с такими ячейками.

Для разбиения невыпуклых областей обычно применяются алгоритмы, основанные на триангуляции [7—10]. Но такие подходы не могут быть использованы при генерации расчетных сеток методом отсечения, так как при численном моделировании в пограничном слое предпочтительно использование шестигранников, которые, в свою очередь, формируются на основе четырехугольных граней, полученных при отсечении.

Вопросы разбиения невыпуклых многоугольников и разрезание невыпуклых ячеек также рассматриваются в статьях [11—14], методы по улучшению качества ячеек сетки представлены в работах [15, 16]. Но в данных работах в рамках операций разрезания и улучшения качества ячеек не рассматриваются вопросы минимизации разницы площадей граней, соседних по ребру и узлу, и объемов ячеек, соседних по грани, ребру и узлу. Учет и корректировка разницы площадей граней и объемов ячеек в сетке являются важными критериями успешности проведе-

ния численного моделирования.

Анализ проблем, возникающих при моделировании задач аэро- и гидродинамики на построенных сеточных моделях, позволил выработать критерии качества ячеек сетки для успешного проведения расчета. Основные критерии качества ячеек сетки:

- ненулевой объем ячеек;
- допустимое соотношение объемов ячеек и площадей поверхностных граней;
- ненулевая площадь граней ячейки;
- выпуклость ячеек и поверхностных граней ячеек.

Для получения требуемого качества ячеек сетки в генераторе был реализован соответствующий программный модуль [3].

Улучшение качества поверхностных граней отсеченных ячеек

Для улучшения качества граней объемных ячеек при генерации сетки методом отсечения первоначально были реализованы операции объединения граней и разрезания невыпуклых граней. В настоящее время в связи с постоянным усложнением геометрических моделей, повышением их детализации, а также ростом требований к качеству расчетных сеток дополнительно реализованы следующие способы для улучшения качества поверхностных граней отсеченных ячеек:

- удаление лишних узлов;
- модификация ячеек с частично совпадающими гранями;
- объединение граней с учетом их формы;
- разрезание искривленных граней;
- разрезание невыпуклых граней с добавлением нового узла и возможностью удаления нового ребра;
- разрезание неплоских граней на множество плоских при изменении кривизны поверхности.

Под разрезанием грани будем понимать разделение одной грани на две путем добавления ребра, сформированного по двум несмежным точкам грани, под объединением — формирование одной грани из нескольких путем удаления смежных ребер.

Все приведенные выше способы улучшения качества граней объемных ячеек направлены

на улучшение формы, уменьшение соотношения площадей соседних граней и сокращение количества невыпуклых и искривленных граней. Численные пороговые значения критериев, используемые в методах улучшения качества граней объемных ячеек, получены в результате оценки качества построенных расчетных сеток для базы моделей с различными значениями соответствующих параметров.

Удаление лишних узлов. Для улучшения структуры сетки, формы боковых граней ячеек пограничного слоя и уменьшения соотношения площадей соседних граней реализован способ улучшения качества граней с помощью удаления узлов с валентностью 2 на ребрах, разделяющих две грани объемной сетки.

Термин *валентность 2* для узла сетки означает, что узел принадлежит только двум граням либо одной, либо соседних ячеек. Как правило, такие узлы возникают в результате модификации отсеченных ячеек и могут быть удалены из сетки без потери качества ячеек и точности аппроксимации поверхности.

Данный способ позволяет улучшить форму граней отсеченных ячеек, как показано на рис. 1,

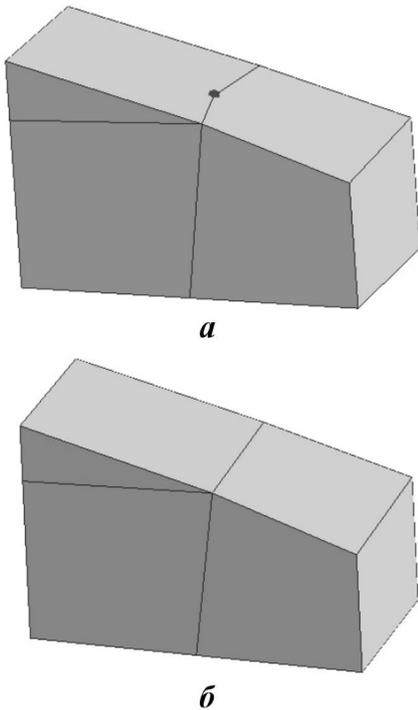


Рис. 1. Пример удаления лишнего узла: *a* — исходные ячейки до удаления узла; *б* — ячейки после удаления узла

сократить количество ребер сетки, в некоторых случаях избавиться от невыпуклых граней, что в дальнейшем позволяет повысить качество ячеек пограничного слоя.

Модификация ячеек с частично совпадающими гранями. В результате применения некоторых способов улучшения качества сетки, в частности удаления малых ребер, могут появиться ячейки с частично совпадающими гранями. Частично совпадающие грани — это грани одной ячейки, имеющие более двух общих узлов, соединенных между собой и не лежащих на одной прямой. В большинстве случаев грань частично совпадает с другой неплоской гранью ячейки. На рис. 2 приведен пример ячейки с частично совпадающими гранями $ABCD$ и $AECB$: узлы A, B, C принадлежат общим для этих граней ребрам AB и BC .

Определение ячеек с частично совпадающими гранями состоит из двух этапов:

- поиск в ячейках граней, имеющих больше двух общих узлов, соединенных между собой ребрами;
- проверка взаимного расположения граней, найденных на первом этапе, для исключения случаев ложного наложения, когда общие узлы лежат на одной прямой.

Если в ячейке установлено наличие частично совпадающих граней, то проводится проверка возможности корректного объединения рассматриваемой ячейки с соседней через частично совпадающую грань с образованием выпуклой ячейки. Если такое объединение возможно, то одна частично совпадающая грань удаляется, а грани объединенной ячейки, смежные между собой по ребру и разделяющие одни и те же ячейки, объединяются при условии, что новая грань будет выпуклой. На рис. 3 показан пример объединения ячейки, имеющей частично совпадающие грани, с соседней ячейкой.

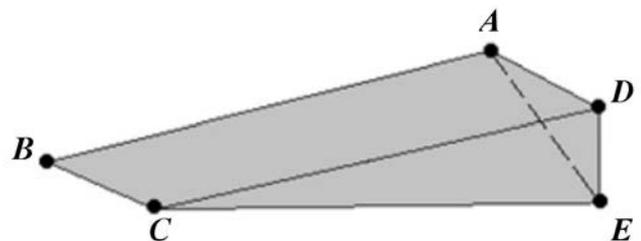


Рис. 2. Ячейка с частично совпадающими гранями

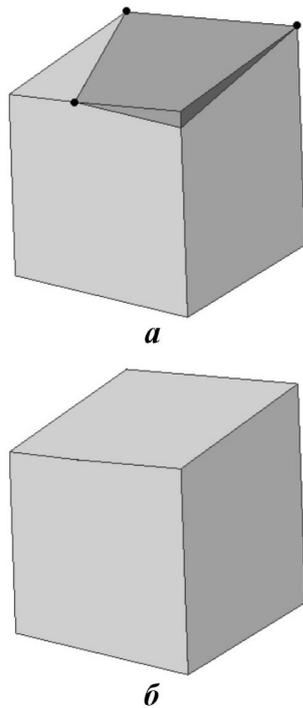


Рис. 3. Пример объединения ячейки с частично совпадающими гранями с соседней: *a* — ячейки до объединения; *б* — результат объединения

В случае, когда корректное объединение ячеек невозможно, от частично совпадающих граней отделяются фрагменты, сформированные по набору общих ребер, один из которых удаляется, а другой становится новой гранью.

На рис. 4 показан пример, в котором ячейку нельзя объединить с соседней через частично совпадающую грань с образованием выпуклой ячейки, поэтому ситуация разрешается благодаря разрезанию частично совпадающих граней.

Объединение граней с учетом их формы.

Грани объемных ячеек после отсечения ячеек шаблонной сетки треугольниками смещенной поверхности, как правило, имеют форму треугольников или четырехугольников. Возникают ситуации, когда грань не является малой, но ее форма может быть улучшена путем объединения с соседними гранями ячейки.

На рис. 5, *a* приведены примеры отсеченных ячеек, содержащих две грани, форма которых может быть улучшена объединением с соседними гранями, что позволит получить грани более высокого качества.

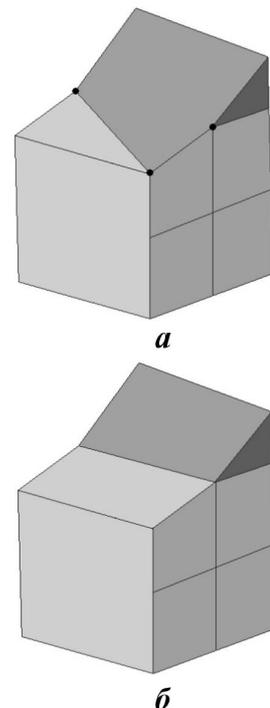


Рис. 4. Пример удаления общей части частично совпадающих граней: ячейки до (*a*) и после (*б*) выполнения операции

Для корректировки формы граней вычисляется величина качества формы грани по формуле [7]

$$Q_{face} = \frac{S_{face}}{P_{face}^2},$$

где S_{face} — площадь грани; P_{face} — периметр грани.

Использование вычисляемого по формуле критерия позволяет при корректировке грани приблизить ее форму к правильному многоугольнику, в частности форму четырехугольной грани — к квадрату.

Минимальное пороговое значение качества формы грани составляет 0,0481, что несколько больше значения качества формы равностороннего треугольника. Таким образом, при указанном пороговом значении все треугольные грани считаются плохими по форме и по возможности должны быть объединены с другими гранями ячейки, что приведет к уменьшению количества треугольных призм в пограничном слое.

Каждая найденная с помощью данного критерия грань объединяется с соседними гранями данной ячейки при выполнении следующих условий:

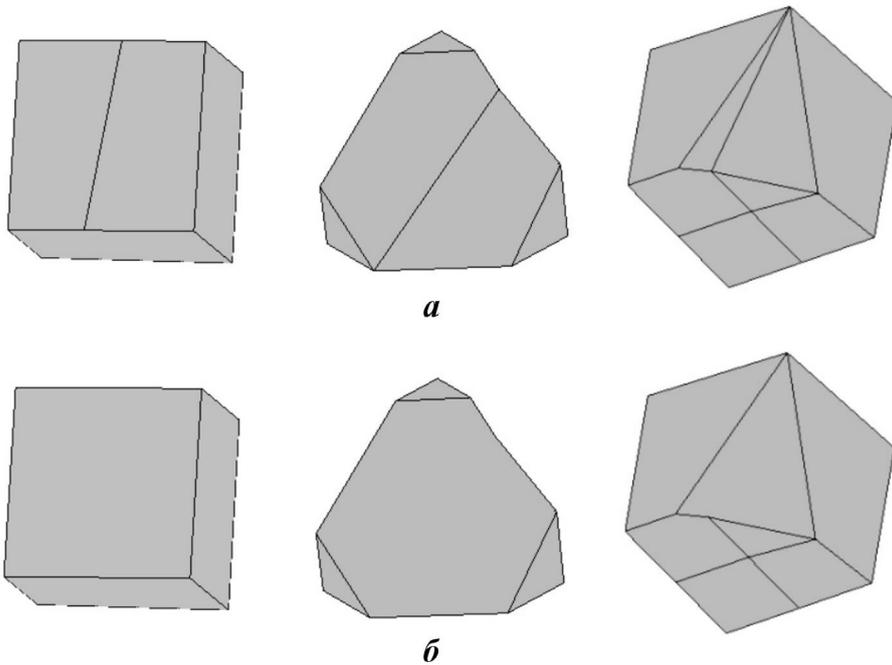


Рис. 5. Примеры ячеек до (а) и после (б) объединения граней по критерию формы

- объединяемые грани не являются смежными по ребру, принадлежащему характерной кривой;
- полученная в результате объединения грань не является невыпуклой или искривленной;
- качество формы объединенной грани выше, чем у исходной грани, предложенной для объединения по критерию формы.

На рис. 5, б показан результат объединения граней по критерию формы в ячейках, изображенных на рис. 5, а.

Критерии диагностики невыпуклых и искривленных граней. Грани отсеченной многогранной ячейки могут иметь простую или сложную форму в зависимости от особенностей геометрии модели. Если поверхностная сетка не содержит характерных кривых, грани ячеек имеют простую форму, поскольку формируются только из точек пересечения ребер ячейки шаблонной сетки с треугольниками поверхностной сетки. В областях, содержащих характерные кривые, грани могут быть невыпуклыми и искривленными, с большим количеством узлов, а их форма — достаточно сложной, поскольку при построении граней также учитываются характерные особенности модели в ячейке. Причинами формирования искривленных и невыпуклых граней в отсеченных ячейках также

могут стать грубая треугольная сетка, крупные ячейки шаблонной сетки в областях с высокой кривизной поверхности или обрыв характерной кривой внутри ячейки. Наличие таких граней в сетке может привести к построению некорректных ячеек пограничного слоя и, как следствие, уменьшению точности получаемого решения или некорректным результатам расчетов.

При генерации сеток методом отсечения для определения невыпуклой грани ищется *узел невыпуклости* грани по формуле [13] (рис. 6)

$$\beta = \begin{cases} \frac{1}{\cos(-\alpha)}, & \text{если } E \cdot N \times N_A < 0; \\ 2\pi - \frac{1}{\cos(-\alpha)}, & \text{если } E \cdot N \times N_A \geq 0, \end{cases}$$

где E — нормаль к грани $ABCDGF$, рассчитанная по методу Ньюэлла [13]; N, N_A — нормали к вспомогательным граням $AGG'A'$ и $GFF'G'$; α — угол между этими нормальными; узлы G', A', F' получены путем переноса узлов A, G, F грани $ABCDGF$ в направлении нормали к этой грани.

Если $\beta \geq \pi$, то грань $ABCDGF$ выпуклая. Если $\beta < \pi$, то $ABCDGF$ невыпуклая и узел G является узлом невыпуклости.

Для вычисления степени искривленности грани при генерации сеток методом отсечения используется формула

$$W = 2 \operatorname{arctg} \frac{d^2}{S},$$

где d — произведение максимального расстояния между соседними узлами грани и единичного вектора нормали грани; S — площадь грани.

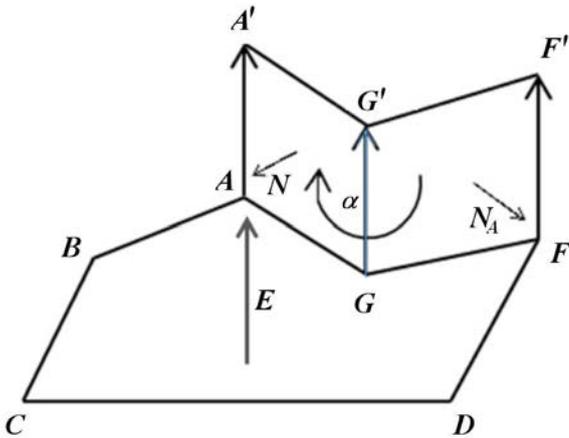


Рис. 6. Схема вычисления угла невыпуклости

Грань считается сильно искривленной, если $W > 50^\circ$. Примеры ячеек с искривленными гранями приведены на рис. 7.

Разрезание искривленных граней. Для корректировки формы искривленных и невыпуклых граней и улучшения их качества реализован способ разрезания, основанный на триангуляции граней.

Алгоритм состоит из двух этапов. На первом этапе строится набор треугольников, максимально близких к поверхностной сетке. Для оценки близости используется косинус угла между нормалью рассматриваемого треугольника и средней нормалью к треугольникам поверхностной сетки, которым принадлежит вершины рассматриваемого треугольника. Предельный случай — это точное совпадение треугольника с участком поверхностной сетки, тогда значение косинуса равно 1. Таким образом, чем ближе значение косинуса угла к 1, тем точнее треугольник будет аппроксимировать участок поверхностной сетки.

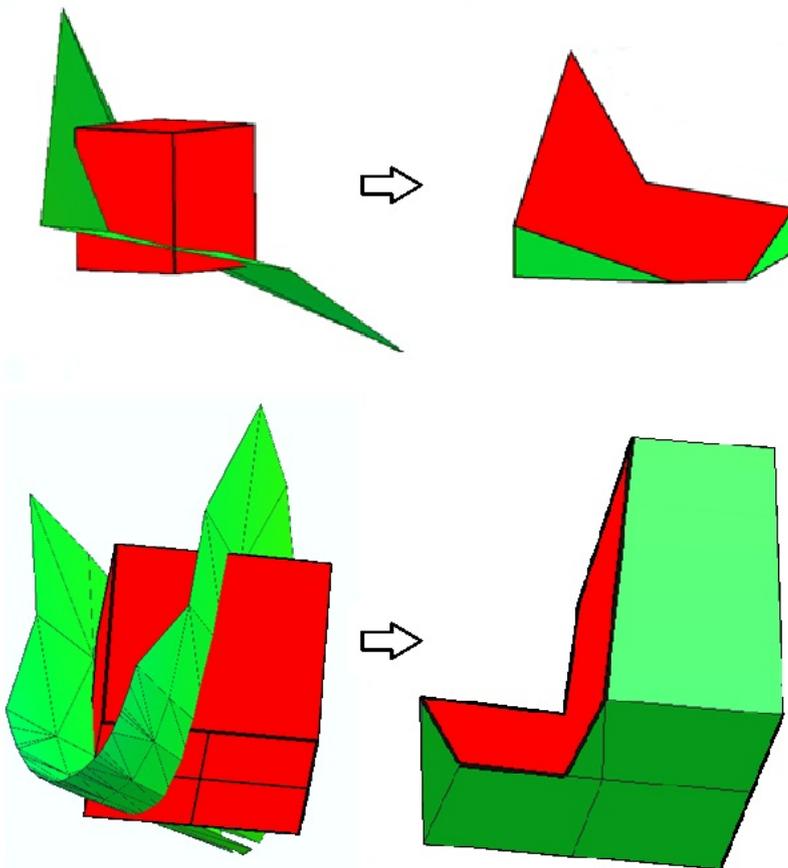


Рис. 7. Примеры ячеек с искривленными гранями, полученных отсечением шаблонных ячеек треугольниками поверхностной сетки

Для построения триангуляции используется модифицированный для неплоского многоугольника метод *отрезания ушей* [17], который основан на том, что граф триангуляции без дополнительных вершин для простого многоугольника всегда является деревом. Отсюда следует, что любой простой n -угольник с $n > 3$ имеет по меньшей мере два "уха", т. е. два треугольника, у каждого из которых две стороны являются сторонами многоугольника, а третья полностью находится внутри него (рис. 8).

Так как целью является построение совокупности треугольников, максимально приближенных к треугольникам поверхностной сетки, то организован итерационный процесс, на каждой итерации которого выбирается оптимальный на данном участке грани треугольник.

Для выбора такого треугольника строится набор "кандидатов", которыми становятся все возможные "уши" многоугольника. Из полученного набора выбирается оптимальный треугольник, т. е. имеющий максимальный косинус угла между нормальными и удовлетворяющий условиям корректности грани (выделение треугольника в отдельную грань не приведет к возникновению пересечений между соседними гранями или ячейками призматического слоя). На рис. 9 показан пример формирования некорректного треугольника (выделен пунктиром), поскольку одно из его ребер пересекает контур грани.

Выбранный оптимальный треугольник-ухо отрезается от многоугольника. После этого итерация по выбору треугольника выполняется с оставшимся многоугольником до тех пор, пока не останется один треугольник.

Если на какой-то итерации невозможно выделить треугольник приемлемого качества, то та-

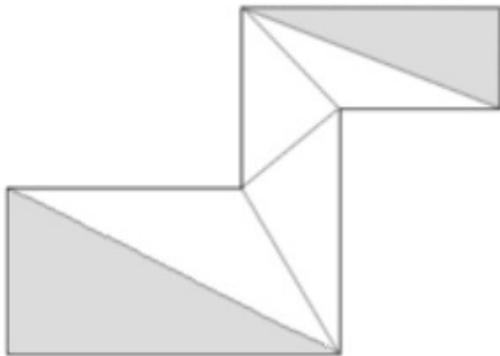


Рис. 8. Многоугольник и его два "уха"

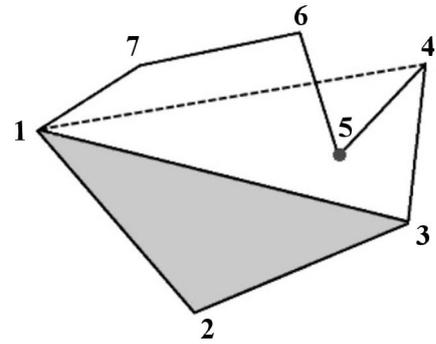


Рис. 9. Некорректный треугольник, сформированный по вершинам 1, 3, 4

кой многоугольник выделяется в самостоятельную грань.

Для решения задачи корректировки формы искривленных граней нельзя ограничиться только разбиением грани на треугольники, это приведет к значительному увеличению числа ячеек пограничного слоя. Чтобы этого избежать, на втором этапе алгоритма выполняется объединение треугольников, для которых угол между нормальными меньше порогового значения в 20° .

На рис. 10 показаны исходная ячейка, ячейка после этапа формирования треугольников и результат операции объединения треугольников.

Разрезание грани путем добавления нового узла с возможностью удаления малого ребра. Для невыпуклых граней с одним узлом невыпуклости реализован метод разрезания грани без добавления новых узлов в сетку [3], но данный способ дает удовлетворительный результат не во всех случаях. На рис. 11, а представлена грань, для которой этот метод не может быть применен, поскольку в результате каждого из разрезов один из фрагментов является невыпуклым. Пунктирными линиями показаны два возможных варианта построения разреза.

Для решения данной проблемы реализован способ разрезания грани с добавлением нового узла. Схематичное представление грани с возможными вариантами формирования узла представлены на рис. 11, б.

Рассмотрим невыпуклую грань $ABCDEF$, у которой узел F является узлом невыпуклости. Узел P образуется после пересечения продолжения ребра AF с одним из несмежных ребер грани $ABCDEF$. При этом образуются две части грани — $ABPF$ и $PCDEF$. Обе полученные части грани должны быть выпуклыми, и площадь каж-

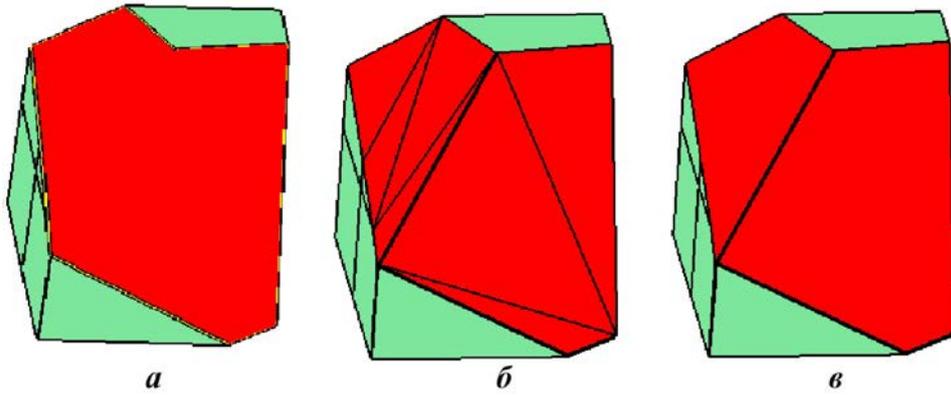


Рис. 10. Разрезание грани: *a* — исходная грань; *б* — разбиение на треугольники; *в* — объединение треугольников

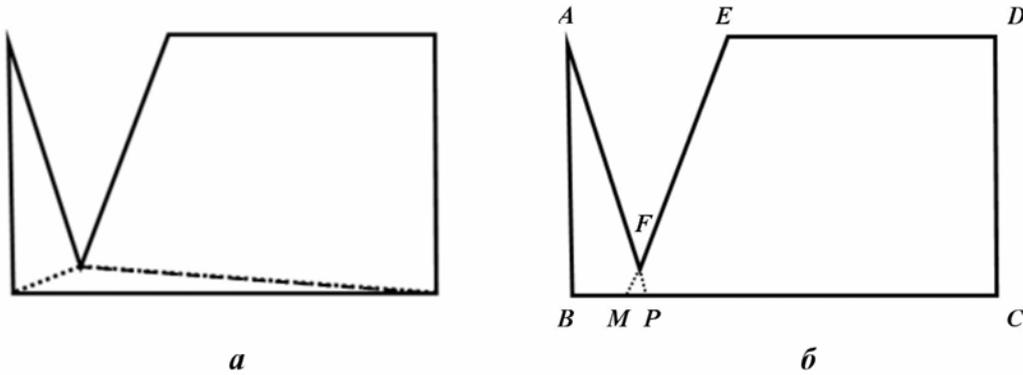


Рис. 11. Пример грани с двумя некорректными разрезами без добавления новых узлов (*a*) и с возможными вариантами разрезания (*б*)

дой части должна быть больше, чем 10% площади исходной грани. Если это условие не выполняется, то находим другой узел M как точку пересечения продолжения ребра EF с одним из других несмежных ребер грани $ABCDEF$.

После того как нужный узел, например P , найден, происходит разрезание исходной грани

на две (в данном случае $ABPF$ и $CDEFP$). Найденный узел P проецируется на поверхностную треугольную сетку и добавляется в описание сетки.

На рис. 12 представлен результат разрезания невыпуклых граней с добавлением нового узла A .

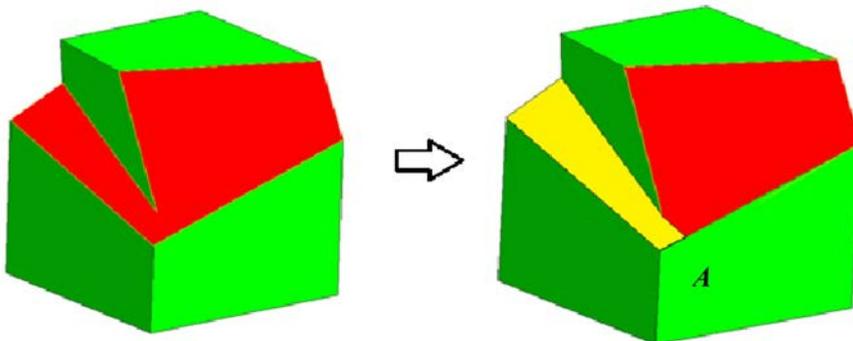


Рис. 12. Пример применения способа разрезания невыпуклых граней с добавлением нового узла сетки

Если после проецирования найденного узла на поверхностную треугольную сетку в новых гранях возникают самопересечения, то дополнительно выполняется поиск ребра (например, BC на рис. 13), у которого расстояние до узла выпуклости (узел E) меньше 10 % длины этого ребра.

Чтобы избежать появления малых ребер, найденное ребро BC заменяется двумя близко лежащими к нему ребрами BE и EC . На всех гранях, содержащих первоначальное ребро (BC), проводится его замена на новую пару ребер BE и EC . Все соседние по ребру грани проверяются на выпуклость. В результате грань $ABCDEF$ разбивается на две грани — $ABEF$ и ECD .

Пример удаления нового ребра при разрезании грани показан на рис. 14.

Разрезание искривленных граней на множество плоских в зонах перехода областей с характерными особенностями в гладкую область. Подходы к выявлению характерных особенностей поверхностной сетки описаны в работах [18–20]. Если характерная

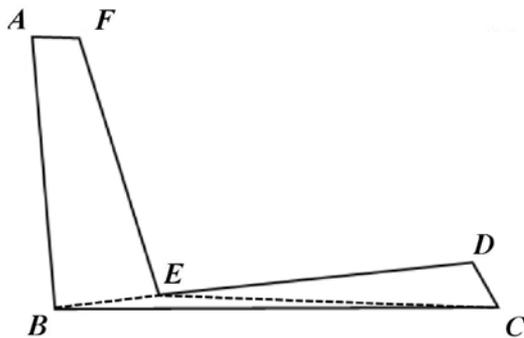


Рис. 13. Пример невыпуклой грани

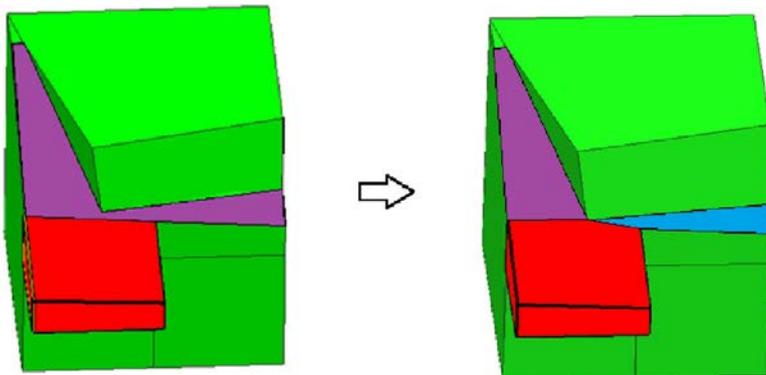


Рис. 14. Пример удаления нового ребра при разрезании грани

кривая переходит в гладкую область, как показано на рис. 15, то могут появляться ячейки, поверхностные грани которых имеют степень искривления больше порогового значения. Ячейки пограничного слоя, восстановленные с таких граней, также будут иметь искривленные основания.

Для коррекции таких граней реализован отдельный метод. Каждую поверхностную грань можно разбить на множество выпуклых фрагментов различными способами [21]. Из всех возможных вариантов разбиения выбирается тот, у которого разница между площадями фрагментов минимальна. На основе этих фрагментов формируется набор плоских граней. На рис. 16, *a* показана ячейка с поверхностной гранью $P_1P_2P_3P_4$ до разрезания, на рис. 16, *б* — после разрезания. В результате разрезания сформированы две грани — $P_1P_2P_3$ и $P_1P_3P_4$.

Для оценки эффективности применения способов улучшения качества граней ячеек выполнено сравнение по основным критериям качества расчетных сеток для модели летательного аппарата, построенных с использованием программного модуля улучшения качества сетки и без него. Общий вид модели представлен на рис. 17. Модель содержит около 5 млн ячеек.

Результаты сравнения приведены в таблице.

Из таблицы видно, что использование модуля улучшения качества граней ячеек позволяет получить более высокое качество сетки. Для каждой построенной сетки эффективность работы модуля улучшения качества по критериям будет различаться, поскольку формы ячеек и граней

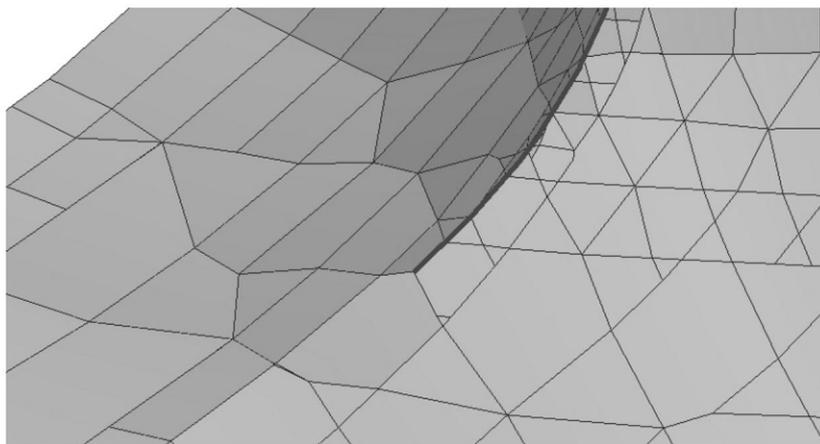


Рис. 15. Фрагмент сетки, в котором характерная кривая переходит в гладкую область

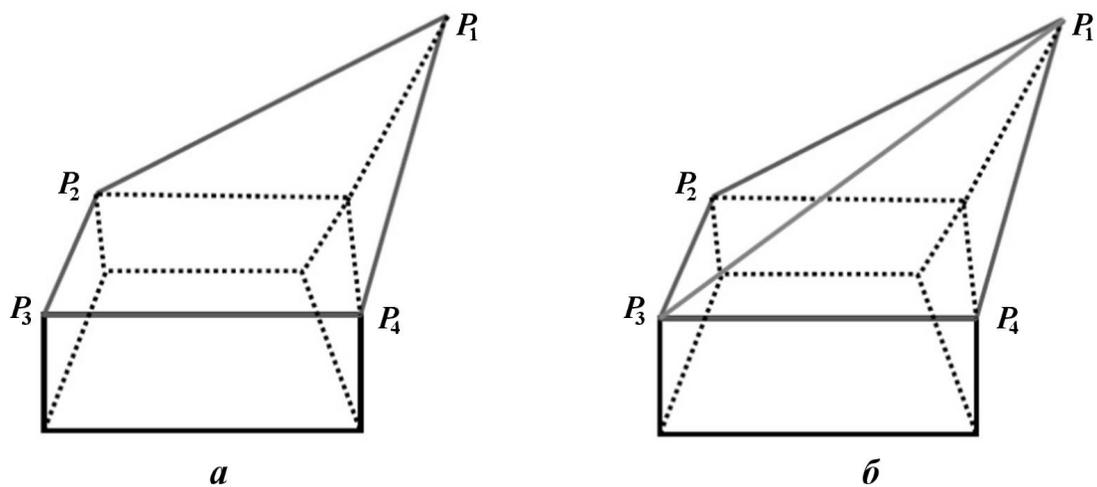


Рис. 16. Пример многогранной ячейки до (а) и после (б) разрезания грани

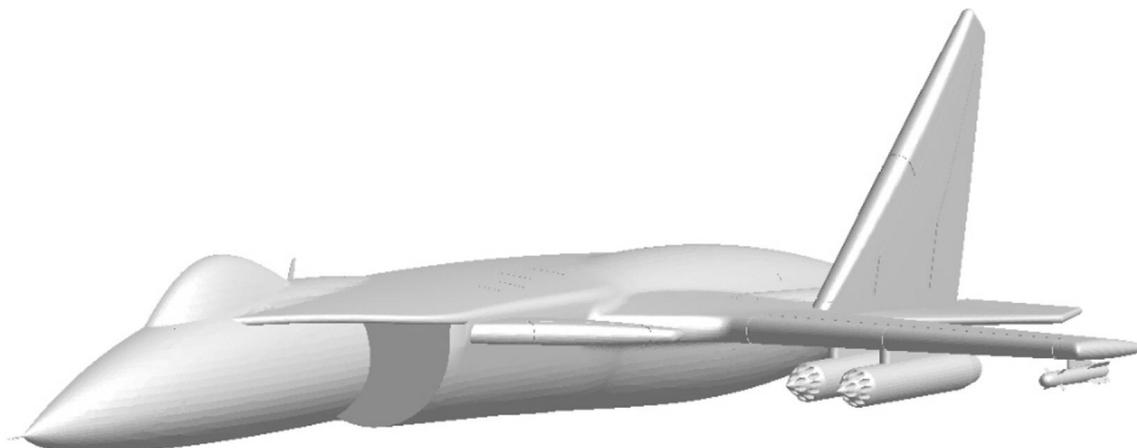


Рис. 17. Общий вид модели летательного аппарата

Сравнение качества сеток, построенных с использованием и без использования модуля улучшения качества граней ячеек

Критерий	Численные характеристики качества сеток		
	без улучшений	с улучшениями	Улучшение, %
Невыпуклые грани	11	4	64
Соотношение площадей граней	142	97	32
Совпадающие грани	620	69	89
Искривление граней	5 850	318	95

значительно зависят от геометрии задачи. Например, для данной задачи наиболее значительное улучшение было получено по критерию *Искривление граней* (95%), наименьшее — по критерию *Соотношение площадей граней* (~ 32%).

Заключение

В связи с постоянно возрастающей сложностью и детализацией геометрических моделей, а также расширением областей применимости пакета программ "Логос" при расчете задач аэро- и гидродинамики повышаются требования к качеству расчетных сеток, построенных с использованием генератора на основе метода отсечения.

Анализ причин некорректного моделирования ряда задач различной сложности позволил выработать критерии численной оценки качества расчетной сетки. На основе этих критериев разработаны и реализованы способы повышения качества многогранных ячеек. Использование их при построении объемной сетки существенно повышает качество ячеек, тем самым увеличивая точность проведения дальнейшего расчета на построенной объемной сетке.

Кроме того, методы повышения качества граней многогранных ячеек используются с целью адаптации расчетной сетки под особенности протекающих физических процессов для корректного численного моделирования задач аэро- и гидродинамики, в частности расчета вязких турбулентных течений в ячейках приграничного слоя. Этап улучшения качества граней ячеек является одним из ключевых в процессе построения расчетной сетки методом отсечения, поскольку результат его работы оказывает значительное влияние на качество ячеек пограничного слоя.

В статье приведены критерии диагностики невыпуклых и искривленных граней, рассмотрены способы улучшения качества граней отсеченных ячеек, реализованные в генераторе сеток на

основе метода отсечения, который используется в пакете программ "Логос"

Описанные в статье методы позволяют существенно расширить применимость генератора, обеспечивая построение качественных расчетных сеток для моделирования задач аэро- и гидродинамики.

Список литературы

1. 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.
Kozelkov A. S., Zhuchkov R. N., Utkina A. A., Volodchenkova K. B. Modelirovanie turbulentnykh techeniy na setkakh gibridnoy struktury s ispolzovaniem skhem vysokogo poryadka tochnosti // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2014. Vyp. 3. S. 18–31.
2. Борисенко О. Н., Лукичев А. Н., Евстифеева Е. О., Панкратов Д. М., Цалко Т. В., Гиниятуллина А. Г. Алгоритмы обработки особенностей геометрических моделей при построении поверхностных треугольных сеток в препроцессоре пакета программ "Логос" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 3. С. 40–51.
Borisenko O. N., Lukichev A. N., Evstifeeva E. O., Pankratov D. M., Tsalko T. V., Giniyatullina A. G. Algoritmy obrabotki osobennostey geometricheskikh modeley pri postroenii poverkhnostnykh treugolnykh setok

- v preprotessore paketa programm "Logos" // *Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov.* 2020. Vyp. 3. S. 40–51.
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 // Там же. 2018. Вып. 2. С. 25–39.
Smolkina D. N., Borisenko O. N., Cherenkova M. V., Giniyatullina A. G., Kuzmenko M. V., Chukhmanov N. V., Potekhina E. V., Popova N. V., Turusov M. R. Avtomaticheskii generator nestrukturirovannykh mnogogrannykh setok v preprotessore paketa program LOGOS // Там же. 2018. Vyp. 2. S. 25–39.
 4. *Попова Н. В., Борисенко О. Н., Корнеева И. И., Чухманов Н. В., Потехина Е. В., Лазарев В. В., Гиниятуллина А. Г.* Автоматический генератор неструктурированных тетраэдральных сеток с призматическими слоями в препроцессоре пакета программ ЛОГОС // Там же. 2020. Вып. 1. С. 43–57.
Popova N. V., Borisenko O. N., Korneeva I. I., Chukhmanov N. V., Potekhina E. V., Lazarev V. V., Giniyatullina A. G. Avtomaticheskii generator nestrukturirovannykh tetraedralnykh setok s prizmaticheskimi sloyami v preprotessore paketa program LOGOS // Там же. 2020. Vyp. 1. S. 43–57.
 5. *Попова Н. В.* Автоматический генератор неструктурированных многогранных сеток на основе тетраэдральных сеток с призматическими слоями // Там же. 2021. Вып. 3. С. 70–83.
Popova N. V. Avtomaticheskii generator nestrukturirovannykh mnogogrannykh setok na osnove tetraedralnykh setok s prizmaticheskimi sloyami // Там же. 2021. Vyp. 3. S. 70–83.
 6. *Лазарев В. В.* Распараллеливание и оптимизация построения блочных расчетных сеток в препроцессоре пакета программ ЛОГОС // Там же. 2018. Вып. 1. С. 54–63.
Lazarev V. V. Rasparallelivanie i optimizatsiya postroeniya blochnykh raschetnykh setok v preprotessore paketa programm LOGOS // Там же. 2018. Vyp. 1. S. 54–63.
 7. *El Gindy H., Everett H., Toussaint G. T.* Slicing an ear using prune-and-search // *Pattern Recognition Letters: journal.* 1993. Vol. 14, No 9. P. 719–722.
 8. *Воропинов А. А., Соколов С. С., Шмельева А. К.* Построение трехмерной сетки на основе диаграммы вороного в невыпуклых областях // *Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов.* 2018. Вып. 2. С. 40–54.
Voropinov A. A., Sokolov S. S., Shmelyeva A. K. Postroenie trekhmernoy setki na osnove diagrammy Voronogo v nevyuklykh oblastiakh // *Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov.* 2018. Vyp. 2. S. 40–54.
 9. *Garimella R. V., Kim J., Berndt M.* Polyhedral mesh generation and optimization for non-manifold domains // *Proc. 22nd Int. Meshing Roundtable.* Springer International Publishing, 2014. P. 313–330.
 10. *Balafas G.* Polyhedral Mesh Generation for CFD-Analysis of Complex Structures. Master Thesis. Technische Universitat Munichen, 2014.
 11. *Гаранжа В. А., Кудрявцева Л. Н., Цветкова В. О.* Построение гибридных расчетных сеток Вороного. Алгоритмы и нерешенные проблемы // *Журнал вычисл. мат. и мат. физ.* 2019. Т. 59, № 12. С. 2024–2044.
Garanzha V. A., Kudryavtseva L. N., Tsvetkova V. O. Postroenie gibridnykh raschetnykh setok Voronogo. Algoritmy i nereshennyye problemy // *Zhurnal vychisl. mat. i mat. fiz.* 2019. T. 59, № 12. S. 2024–2044.
 12. *Dumetrescu A., Toth C. D.* Minimum Weight Convex Steiner Partitions. Department of Computer Science and Hausdorff Center for Mathematics University of Bonn, 2009.
 13. *Owen S. J.* Non-Simplicial Unstructured Mesh Generation. USA, Pittsburg, 1999.
 14. *Handbook of Grid Generation / Ed. by J. F. Thompson, B. K. Soni, N. P. Weatherill.* Boca Raton, Florida, USA: CRC Press, 1999.
 15. *Jasak H., Tukovi Z.* Automatic mesh motion for the unstructured finite volume method //

- Transactions of FAMENA. 2006. Vol. 30 (2). P. 1–20.
16. *Чернышенко А. Ю.* Технология построения адаптируемых многогранных сеток и численное решение эллиптических уравнений 2-го порядка в трехмерных областях и на поверхностях: Автореф. дис. канд. физ.-мат. наук. М.: ИВМ РАН. 2013.
Chernyshenko A. Yu. Tekhnologiya postroyeniya adaptiruemykh mnogogrannykh setok i chislennoe reshenie ellipticheskikh uravneniy 2-go poryadka v trekhmernykh oblastiakh i na poverkhnostyakh: Avtoref. dis. kand. fiz.-mat. nauk. M.: IVM RAN. 2013.
17. *Klincsek G. T.* Minimal triangulations of polygonal domains // Proc. Annals of Discrete Mathematics. 1980. Vol. 9. P. 121–123.
18. *Jiao X., Heath M. T.* Feature detection for surface meshes // Proc. 8th Int. Conf. on Numerical Grid Generation in Computational Field Simulations. Honolulu, Hawaii, USA, 2002. P. 705–714.
19. *Jiao X., Alexander P.* Parallel feature-preserving mesh smoothing // Proc. Int. Conf. on Comput. Sci. and Appl. Singapore, 2005. P. 1180–1189.
20. *Baker T.* Identification and preservation of surface features // Proc. 13th Int. Meshing Roundtable. Williamsburg, Virginia, USA, 2004. P. 299–310.
21. *Eberly D.* The Minimal Cycle Basis for a Planar Graph. <https://www.geometrictools.com/>.

Статья поступила в редакцию 22.12.21.

УДК 004.9

РЕАЛИЗАЦИЯ В ПРОГРАММНОМ КОМПЛЕКСЕ "ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР" ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ВЕКТОРНЫХ ШРИФТОВ ДЛЯ АДДИТИВНОГО ПРОИЗВОДСТВА

М. А. Титов
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Представлена возможность применения векторных шрифтов в интегрирующей оболочке программного комплекса "Виртуальный 3D-принтер". Она обеспечивает нанесение текста, символов или логотипа предприятия на конструкции или детали, получаемые в процессе аддитивного производства методом послойного лазерного спекания металлической порошковой смеси. Рассмотрен SVG-формат компьютерных масштабируемых векторных шрифтов. Представлено его краткое описание на языке команд двумерной векторной графики Scalable Vector Graphics. Предложен подход получения координат точек, лежащих на границе обводки символа векторного шрифта. Описаны реализованные для этого синтаксический SVG-анализатор — парсер глифов символов шрифта — и библиотека растеризации квадратичных и кубических кривых Безье. Приведен пример использования разработанных программ для формирования трехмерной сцены из строки символов, выполненной векторным шрифтом. Представлены фотографии полученного в процессе аддитивного производства трехмерного текста.

Ключевые слова: SVG-формат векторных шрифтов, кривые Безье, парсер глифов символов, библиотека растеризации, аддитивное производство.

Введение

В РФЯЦ-ВНИИЭФ с 2019 г. разрабатывается комплекс программного обеспечения "Виртуальный 3D-принтер" [1] для моделирования физических процессов, протекающих при селективном лазерном сплавлении, с целью прогнозирования структуры, свойств материалов, получения изделий с заданными свойствами, а также для проведения топологической оптимизации изделий. Разрабатываемый комплекс обеспечивает:

- создание и редактирование геометрий деталей;
- оптимизацию формы деталей на основе методов топологической оптимизации, в том числе с использованием решетчатых структур;
- подготовку деталей к печати;
- выбор оптимальных параметров работы оборудования;
- прогнозирование свойств деталей на базе многомасштабного математического моделирования на макро- и мезоуровнях.

Важной частью комплекса "Виртуальный 3D-принтер" является функциональный блок "Интегрирующая оболочка" (далее просто интегрирующая оболочка), объединяющий модули комплекса в единую программу с графическим интерфейсом и регламентирующий процесс подготовки изделий к печати посредством пользовательских диалогов. Статья посвящена развитию возможностей интегрирующей оболочки — реализации на языке программирования C++ и с помощью библиотек STL и QT возможности применения векторных шрифтов для обеспечения нанесения текста (например

марки металла изделия, поясняющих знаков и символов или логотипа предприятия) на выпекаемую конструкцию или деталь (рис. 1).

В своей ранней версии интегрирующая оболочка уже обеспечивала поддержку собственного растрового шрифта (рис. 2). Этот шрифт ограничен буквенно-цифровым набором символов, геометрии частей его *кегельных площадок (глифов)* имеют равную ширину и эстетически несовершенны. Файл шрифта требует ручного вмешательства в случае необходимости его пополнения нестандартными символами. Перечисленные недостатки актуализировали задачу реализации возможности применения более совершенных *масштабируемых* векторных шрифтов.

Для обеспечения поддержки векторных шрифтов в первую очередь нужно сформировать формы фигур требуемых буквенно-цифровых символов — получить координаты точек, лежащих на контурах обводки глифов этих знаков. Один из подходов к решению поставленной задачи — получение нужных данных из компьютерных векторных шрифтов.

Компьютерный шрифт — это файл, содержащий в себе описание набора буквенных, цифровых, служебных и псевдографических символов, используемый для отображения этих символов программой или операционной системой. В векторных шрифтах символы представляют собой криволинейные контуры, описываемые математическими формулами. Каждый знак описан с помощью серии векторных команд, определяющих координаты опорных точек, которые соединены прямыми или кривыми и образуют контур знака без привязки к абсолютному размеру или разрешению. Такое описание обеспечивает увеличение масштаба изображения без потери качества. Обеспечение чтения векторного файла шрифта позволит получить серии команд, описывающих криволинейные контуры обводки символов, а реализация программ растеризации этих команд — требуемые координаты точек, лежащих на границах криволинейных контуров. В интегрирующей оболочке имеется возможность построения трехмерных моделей из плоских фигур, представленных координатами точек, лежащими на их контурах. Используем эту возможность для построения трехмерных моделей символов и печати сформированной сцены на принтере.



Рис. 1. Металлическая деталь с нанесенными символами



Рис. 2. Растровый шрифт интегрирующей оболочки

SVG-формат векторных шрифтов

Существует несколько форматов шрифта, различающихся способом хранения и представления информации, — SVG, EMF, CDR, CGM, DXF, OPENVG, GXL и т. д. Для данной работы был выбран открытый стандартный формат SVG. Этот формат текстовый, а значит, наиболее понятный для изучения. Он реализует язык разметки Scalable Vector Graphics [2] описания двумерной векторной графики. Некоторые (free software) редакторы векторной графики, поддерживающие SVG-формат: Inkscape [3], SK1 [4], Scribus [5].

Обводка глифа любого символа в этом формате представляется компактной SVG-строкой, описывающей путь от начальной точки до конечной через любые промежуточные координаты. Строка с данными задается атрибутом `d` тега `glyph` и содержит серию команд, закодированных набором букв

и чисел. Буква определяет тип команды, числа — ее параметры (чаще всего это координаты). Так, например, тег `glyph`, описывающий обводку глифа цифры "5" векторного шрифта `DejaVuSansMono-BoldOblique.svg`, выглядит следующим образом:

```
<glyph glyph-name="five" unicode="5"
d="M340 1493 h811 l-51 -260 h-572l-53 -277q33 13 69 19t77 6q197 0 313.5 -
116.5 t116.5 -313.5q0 -267 -178 -423.5t-486 -156.5q-95 0 -190.5 16t -190.5
48151 266q70 -40 155.5 -61t180.5 -21q172 0 271.5 79t99.5 214 q0104 -71
164.5t-193 60.5q-78 0 -160.5 -20 t-161.5 -58z" />
```

Описание обводки глифа может состоять из отрезков прямых, дуг, а также квадратичных и кубических кривых Безье (рис. 3).

Кривые Безье [6] были представлены в 60-х годах XX века независимо друг от друга П. Безье из автомобилестроительной компании "Рено" и П. де Кастельжо из компании "Ситроен" и применялись для проектирования кузовов автомобилей. Кривые относятся к частному классу алгебраических кривых третьего и второго порядков и являются одним из важнейших инструментов современных систем автоматизированного проектирования и программ компьютерной графики.

Квадратичная кривая Безье задается тремя опорными точками \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 и описывается формулой

$$\mathbf{B}(t) = (1 - t)^2 \mathbf{P}_0 + 2t(1 - t) \mathbf{P}_1 + t^2 \mathbf{P}_2, \quad t \in [0, 1]. \quad (1)$$

Параметр t определяет, на каком расстоянии от \mathbf{P}_0 до \mathbf{P}_2 находится точка $\mathbf{B}(t)$. Например, при $t = 0,25$ значение функции $\mathbf{B}(t)$ соответствует четверти расстояния между точками \mathbf{P}_0 и \mathbf{P}_2 .

Кубическая кривая Безье задается четырьмя опорными точками \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 , \mathbf{P}_3 и описывается формулой

$$\mathbf{B}(t) = (1 - t)^3 \mathbf{P}_0 + 3t(1 - t)^2 \mathbf{P}_1 + 3t^2(1 - t) \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0, 1]. \quad (2)$$

Опорные точки \mathbf{P}_0 , \mathbf{P}_1 , \mathbf{P}_2 и \mathbf{P}_3 , заданные в двух- или трехмерном пространстве, определяют форму кривой (см. рис. 3, б). Линия берет начало из точки \mathbf{P}_0 , направляясь к \mathbf{P}_1 , и заканчивается в точке \mathbf{P}_3 , подходя к ней со стороны \mathbf{P}_2 . То есть кривая не проходит через точки \mathbf{P}_1 и \mathbf{P}_2 , они используются лишь для указания ее направления. Длина отрезка между \mathbf{P}_0 и \mathbf{P}_1 определяет, как скоро кривая повернет к \mathbf{P}_3 . Точки \mathbf{P}_0 и \mathbf{P}_3 называются конечными, а \mathbf{P}_1 и \mathbf{P}_2 — управляющими.

В табл. 1 приведено описание основных команд и параметров атрибута `d` тега `glyph` на языке Scalable Vector Graphics, где `x_prev` и `y_prev` — координаты предыдущей конечной точки, а `xs_prev` и `ys_prev` — координаты предыдущей управляющей точки.

Все команды существуют в двух вариантах: команда, начинающаяся заглавной буквой, использует абсолютные координаты, а начинающаяся строчной буквой — относительные (например, команда `l 31 26` означает перемещение на 31 пиксель вверх и 26 пикселей влево от последней точки).

На рис. 4, а показан пример построения одноконтурного символа "С". Контур состоит как из отрезков, так и из кривых. В терминах языка разметки масштабируемой векторной графики для

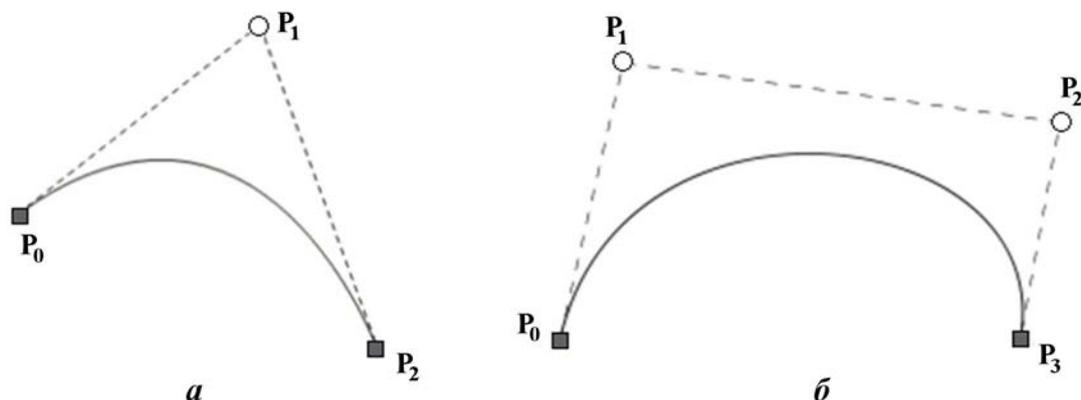


Рис. 3. Квадратичная (а) и кубическая (б) кривые Безье

Команды и параметры атрибута `d` тега `glyph`

Команда	Описание
<code>M x y</code>	Переместить перо в точку (x, y)
<code>m x y</code>	Переместить перо в точку $(x_{prev}+x, y_{prev}+y)$
<code>V y</code>	Провести вертикальную линию из точки (x_{prev}, y_{prev}) в (x_{prev}, y)
<code>v y</code>	Провести вертикальную линию из точки (x_{prev}, y_{prev}) в $(x_{prev}, y_{prev}+y)$
<code>H x</code>	Провести горизонтальную линию из (x_{prev}, y_{prev}) в (x, y_{prev})
<code>h x</code>	Провести горизонтальную линию из (x_{prev}, y_{prev}) в $(x_{prev}+x, y_{prev})$
<code>l x y</code>	Провести линию из (x_{prev}, y_{prev}) в $(x_{prev}+x, y_{prev}+y)$
<code>Q xc yc x y</code>	Построить квадратичную кривую Безье: начальная точка (x_{prev}, y_{prev}) , конечная точка (x, y) , управляющая точка (xc, yc)
<code>q xc yc x y</code>	Построить квадратичную кривую Безье: начальная точка (x_{prev}, y_{prev}) , конечная точка $(x_{prev}+x, y_{prev}+y)$, управляющая точка $(x_{prev}+xc, y_{prev}+yc)$
<code>T x y</code>	Построить симметричную кривую Безье: начальная точка (x_{prev}, y_{prev}) , конечная точка (x, y) , управляющая точка $(2*x_{prev}-xc_{prev}, 2*y_{prev}-yc_{prev})$
<code>t x y</code>	Построить симметричную кривую Безье: начальная точка (x_{prev}, y_{prev}) , конечная точка $(x_{prev}+x, y_{prev}+y)$, управляющая точка $(2*x_{prev}-xc_{prev}, 2*y_{prev}-yc_{prev})$
<code>Z</code>	Замкнуть контур

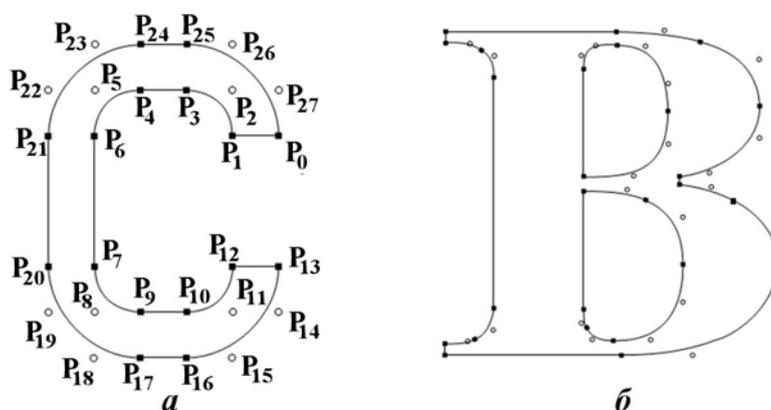


Рис. 4. Пример построения одноконтурного (а) и трехконтурного (б) векторных символов

построения этого символа использовались точки, лежащие на кривых контура (*on-curve*): $P_1, P_3, P_4, P_6, P_7, P_9, P_{10}, P_{12}, P_{13}, P_{16}, P_{17}, P_{20}, P_{21}, P_{24}, P_{25}$, и управляющие точки, не лежащие на кривых контура (*off-curve*): $P_2, P_{26}, P_{27}, P_{11}, P_{14}, P_{15}, P_8, P_{18}, P_{19}, P_5, P_{22}, P_{23}$. 25-я точка соединяется с 0-й; 26-я и 27-я точки являются управляющими (*off-curve*) и служат для построения кривой $P_{25}-P_0$.

На рис. 4, б показан пример построения трехконтурного символа "B". В соответствии с SVG-форматом обход точек внешнего контура происходит по часовой стрелке, а двух внутренних — против часовой стрелки.

Парсер глифов символов и библиотека растеризации

Итак, в файлах SVG-шрифтов строки данных с атрибутом `d` содержат геометрию обводки глифов символов. Глифы имеют векторный формат описания данных, а принтер работает с растровыми данными — координатами позиционирования лазерного луча. Следовательно, задача получения обводки глифа сводится к формированию набора SVG-команд и вычислению из набора отрезков и кривых Безье векторного формата координат точек, лежащих на контуре обводки, т. е. к растеризации данных — процессу, обратному векторизации. Для этого были реализованы:

- синтаксический анализатор, или *парсер*, тега **glyph** для получения серии SVG-команд векторного формата, описывающих обводку глифа;
- библиотека подпрограмм растеризации SVG-команд для нахождения множества точек кривой по координатам ее опорных точек.

Схема взаимодействия парсера с библиотекой растеризации следующая (рис. 5): парсер получает в качестве аргумента строку данных тега **glyph** и разбивает ее на серию SVG-команд, из которых выделяет аргументы и активирует соответствующие им подпрограммы библиотеки растеризации, при необходимости вычисляя абсолютное смещение координат (если команды начинаются со строчной буквы). Подпрограммы библиотеки вычисляют координаты точек обводки глифа и сохраняют их в динамической структуре хранения данных.

Например, если в качестве аргумента парсера используется строка данных глифа цифры "2" векторного шрифта `DejaVuSansMono-BoldOblique.svg` (`M354 260h617l-51 -260h-949l49 252l277 248q51 46 145 128q402 354 402 507q0 72 -56 112.5t-155 40.5q-82 0 -186 -29.5t-222 -87.5l51 267q109 40 216 61t208 21q193 0 314 -95t121 -245q0 -249 -454 -638l-40 -34z`), то результат его работы можно представить в виде табл. 2.

Реализация адаптеров, изображенных на схеме взаимодействия парсера с библиотекой растеризации (см. рис. 5), обусловлена форматом языка разметки масштабируемой векторной графики Scalable Vector Graphics: в его командах построения линий/кривых начальная точка не задается явно, а определяется из конечных координат предыдущей команды.

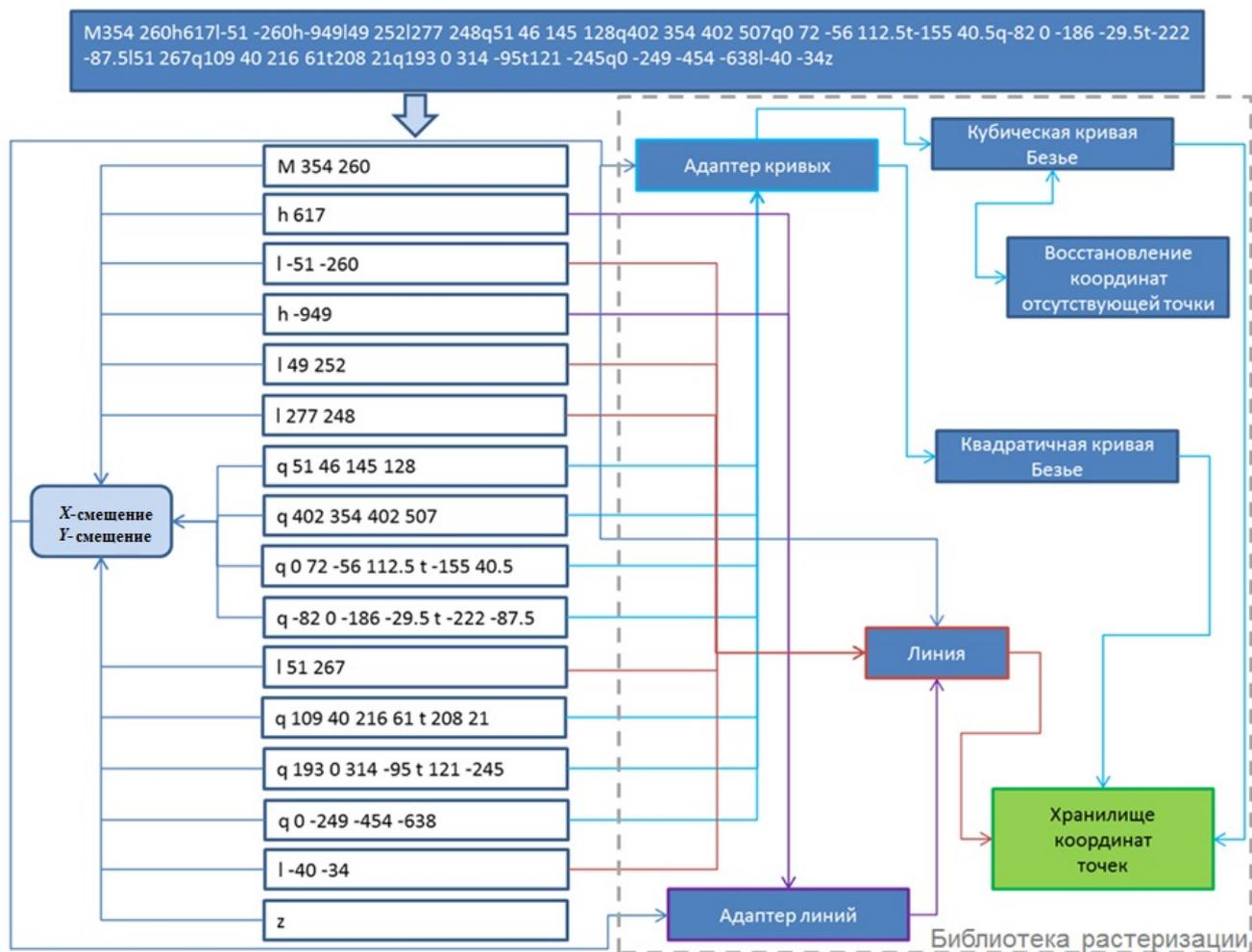


Рис. 5. Схема автоматического взаимодействия парсера с библиотекой растеризации

Результат парсинга данных тэга glyph цифры "2" векторного шрифта

SVG-команда тэга glyph	Вызываемая подпрограмма библиотеки растеризации и значения ее аргументов	Координаты абсолютного смещения
M 354 260	wraperM(354,260)	354,260
h 617	wraperL(P, 617, 0)	971,260
l -51 -260	wraperL(P, -51, -260)	920,0
h -949	wraperL(P, -949, 0)	-29,0
l 49 252	wraperL(P, 49, 252)	20,252
l 277 248	wraperL(P, 277, 248)	297,500
q 51 46 145 128	wraperBezierPlusMultiT(P,51,46,145,128, NULL)	225,1169
q 402 354 402 507	wraperBezierPlusMultiT(P,402,354,402,507, NULL)	442,628
q 0 72 -56 112.5t-155 40.5	wraperBezierPlusMultiT(P, 0,72,-56,112.5, Points)	844,1135
q -82 0 -186 -29.5t-222 -87.5	wraperBezierPlusMultiT(P, 82,0,-186,-29.5, Points)	633,1287
l 51 267	wraperL(P, 51, 267)	276,1436
q 109 40 216 61t208 21	wraperBezierPlusMultiT(P, 109,40,216,61, Points)	700,1518
q 193 0 314 -95t121 -245	wraperBezierPlusMultiT(P, 193,0,314,-95, Points)	1 135,1178
q 0 -249 -454 -638	wraperBezierPlusMultiT(P, 0,-249,-454,-638, NULL)	618,540
l -40 -34	wraperL(P, -40, -34)	641,506
z	wraperM(354,260)	354,260

Поддержка векторных шрифтов в модуле задания 2D-контуров

Для того чтобы отобразить сформированный контур символа и хранить координаты его точек, парсер глифов и библиотека растеризации были интегрированы в разработанный автором модуль задания 2D-контуров интегрирующей оболочки. Этот модуль обеспечивает импорт и экспорт DXF-файлов САПР-систем [7], а также позволяет создавать и редактировать наборы замкнутых контуров, описывающие комплекты геометрий плоскостных срезов трехмерной детали в едином (уникальном) формате хранения данных со счетным модулем программного комплекса "Виртуальный 3D-принтер".

На рис. 6 представлен контур глифа цифры "2", полученный с использованием библиотеки растеризации и отображенный графическим интерфейсом модуля задания 2D-контуров. Из рисунка видно, что сформированный контур растеризован — он состоит только из отрезков.

Схема работы программ поддержки векторных шрифтов в модуле задания 2D-контуров, представленная на рис. 7, отображает два пути получения контуров обводки глифов:

- вывод текста, заданного стандартными буквенно-цифровыми символами (от "0" до "9", от "A" до "Я", от "a" до "я", от "А" до "Z", от "a" до "z", ",", ".", "+", "*", "-"), данные тэгов глифов которых интегрированы в код программы;
- вывод любого нестандартного символа по его юникоду.

На рис. 8 показан интерфейс модуля задания 2D-контуров для ввода текста и получения координат контуров обводки глифов символов. Программа запрашивает

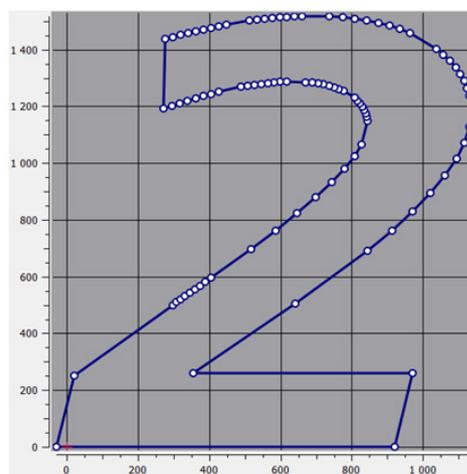


Рис. 6. Построенный контур цифры "2"



Рис. 7. Схема работы программ поддержки векторных шрифтов в модуле задания 2D-контуров

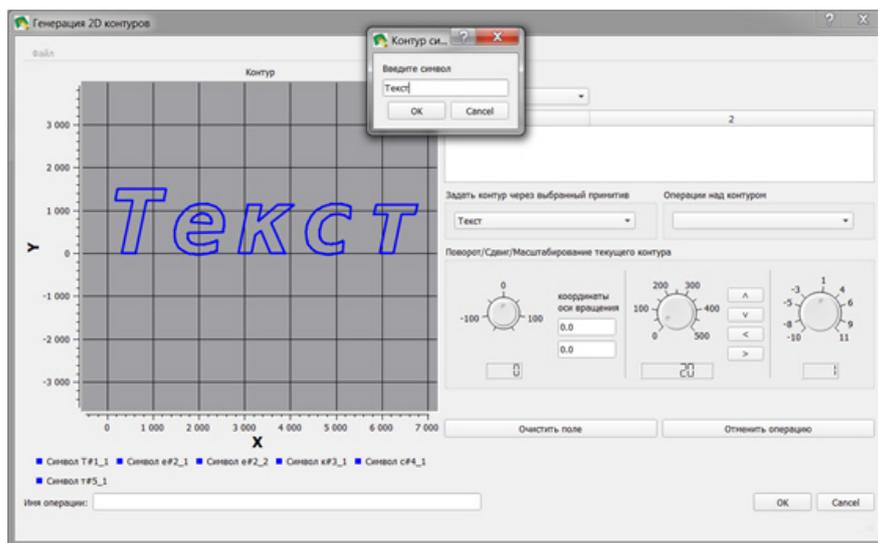


Рис. 8. Интерфейс ввода текста для получения контуров обводки глифов символов

физические размеры символов и автоматически вычисляет межсимвольный интервал введенного текста, т. е. смещение координат точек контуров глифов, следующих по отношению к первому контуру. С помощью инструментов этого модуля полученные контуры можно вращать, перемещать, масштабировать, редактировать координаты его точек (рис. 9) и т. д.

Для получения координат контура любого символа из произвольного SVG-файла векторного шрифта достаточно знать его юникод (Unicode). Так, указанием с помощью интерфейса модуля пути к файлу шрифта fontawesome-webfont.svg, а также юникода *xf1cc* его символа (рис. 10) был получен контур символа бесконечности (рис. 11).

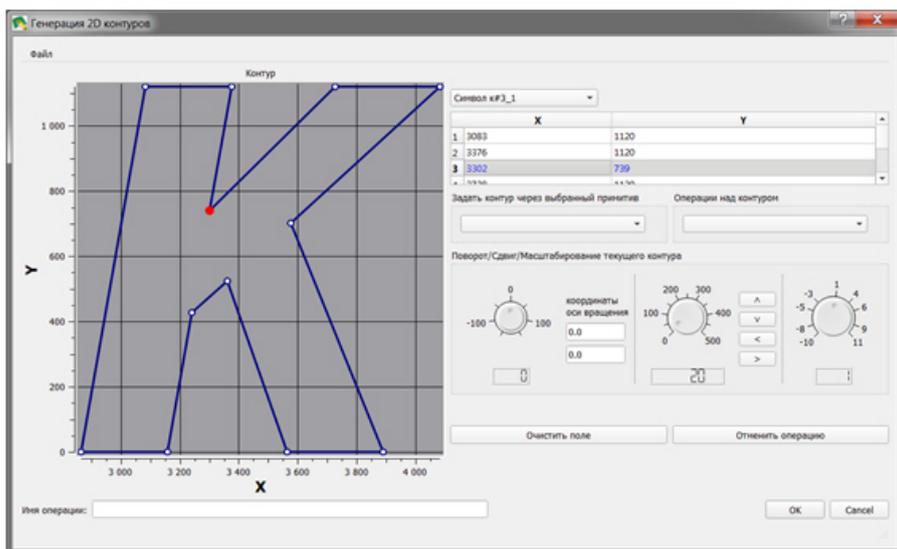


Рис. 9. Выделение точки контура символа инструментами модуля

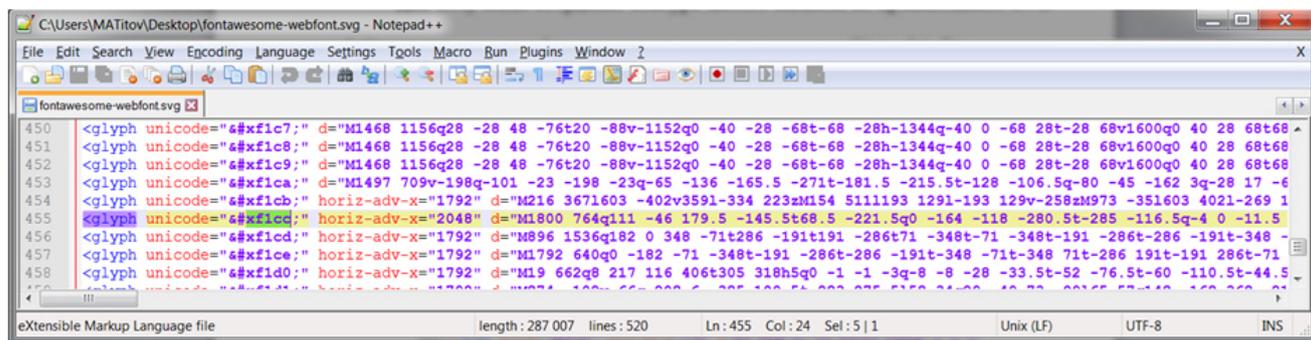


Рис. 10. Фрагмент SVG-файла шрифта fontawesome-webfont.svg с выделенной строкой с юникодом *xf1cc*

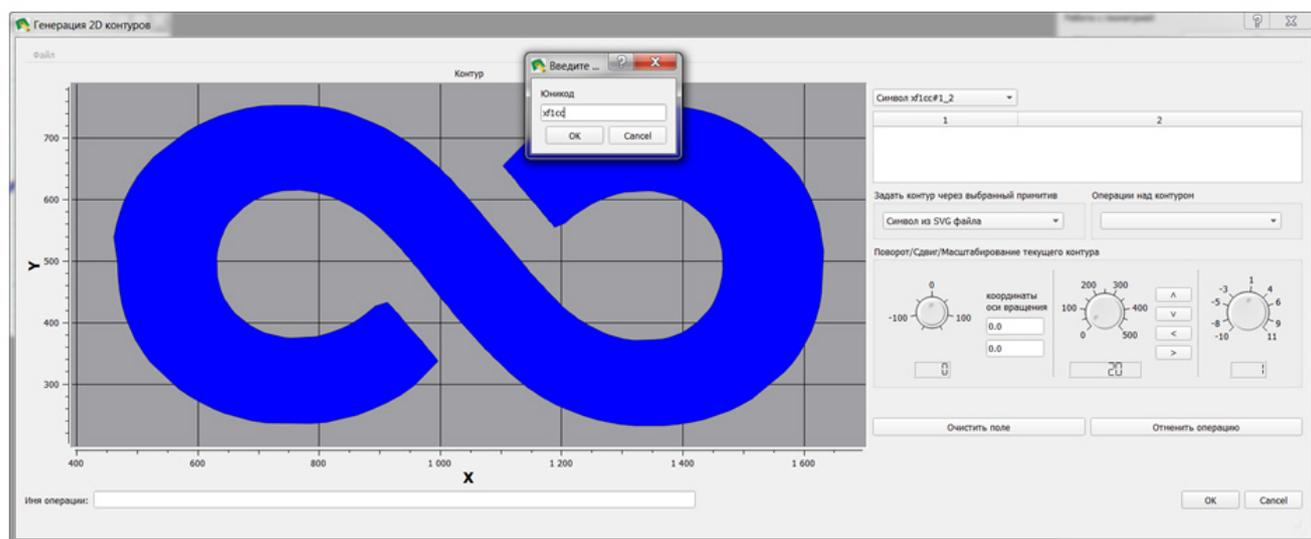


Рис. 11. Сформированный по юникоду *xf1cc* глиф символа бесконечности

Печать текста векторным шрифтом

Проиллюстрируем все вышесказанное практическим примером работы пользователя в интегрирующей оболочке. С помощью инструментов модуля задания 2D-контуров сформируем контуры обводки глифов символов, составляющих фразу "virtual 3d printer" (рис. 12). Затем способом экструзии (путем "выдавливания" двумерного компонента как поперечного сечения объекта вдоль оси OZ), реализованным в модуле создания твердотельной геометрии из 2D-контуров, преобразуем двумерные контуры символов в трехмерные объекты (рис. 13). Далее посредством диалога *Генерация управляющего (G) кода* сгенерируем управляющий файл.

В программном комплексе "Виртуальный 3D-принтер" на данный момент реализована возможность генерации файла управляющего кода для принтеров производства ОАО НПО "ЦНИИТМАШ" линейки "MeltMaster" [8]. Управляющие файлы для аддитивной машины этой линейки имеют текстовый формат и состоят из команд собственного языка управления: выбор режимов работы лазеров, перемещение луча лазера и платформы, насыпание слоя металлической порошковой смеси.

Используя сгенерированный управляющий файл, напечатаем на принтере "MeltMaster 250" сформированную трехмерную сцену. Результат печати представлен фотографиями на рис. 14, 15.

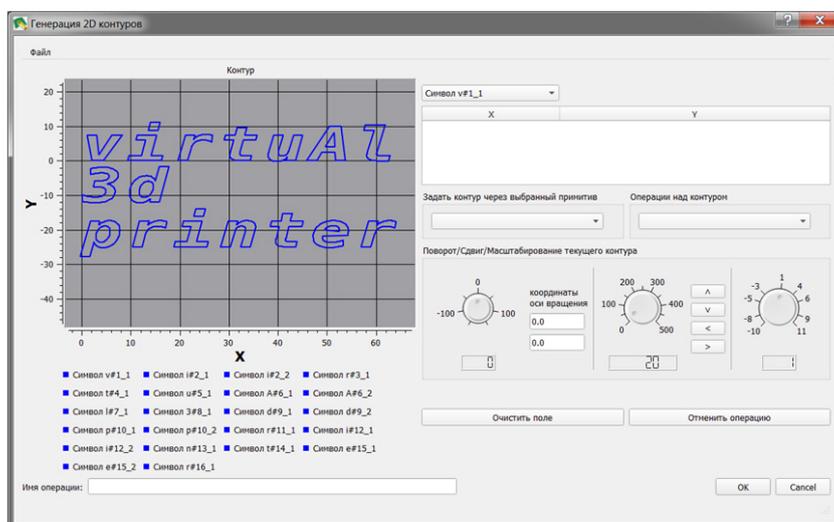


Рис. 12. Сформированные контуры обводки глифов символов

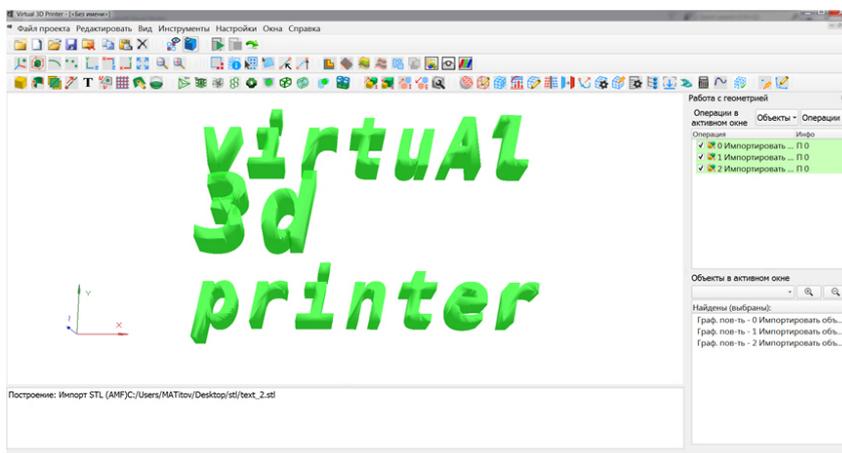


Рис. 13. Трехмерная сцена, подготовленная к печати способом экструзии

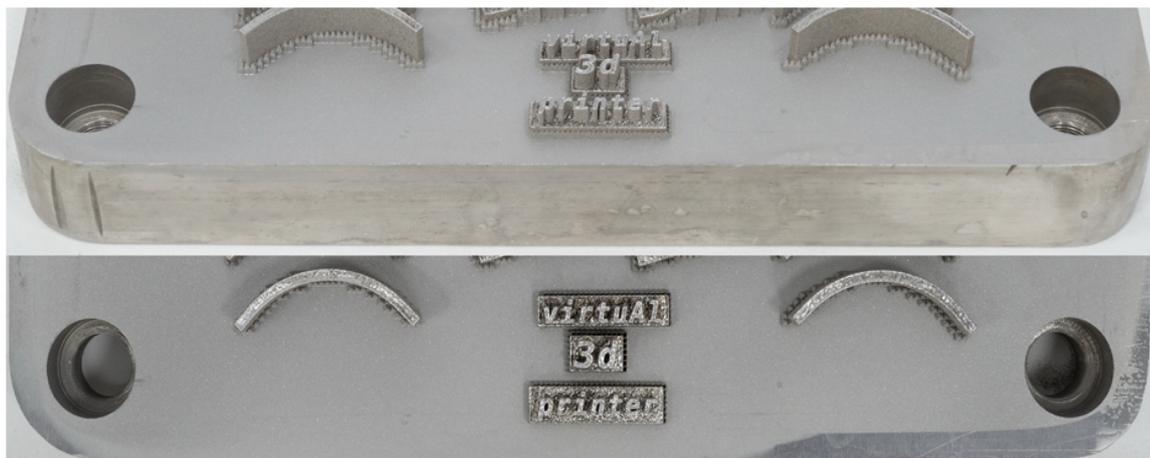


Рис. 14. Фрагменты фотографий платформы размером 250 × 250 мм с напечатанной сценой (материал – сталь 03X17Н14М2), виды сбоку и сверху



Рис. 15. Фотография выполненной векторным шрифтом надписи, полученной селективным лазерным спеканием металлической порошковой смеси

Заключение

В результате проделанной работы в функциональном блоке "Интегрирующая оболочка" программного комплекса "Виртуальный 3D-принтер" реализована возможность применения масштабируемых векторных шрифтов для нанесения текста на конструкцию или деталь, получаемую в процессе послойного лазерного спекания металлической порошковой смеси. Предложенный подход не ограничивается стандартным буквенно-цифровым набором символов, он позволяет получать кон-

туры любого символа любого файла шрифта SVG-формата, обеспечивая масштабирование символов без потери качества.

Реализованная возможность способствовала приобретению нового отличительного свойства выпекаемыми изделиями. В дальнейших планах развития интегрирующей оболочки — обеспечение нанесения текста на криволинейные поверхности изделий.

Список литературы

1. Дьянов Д. Ю., Медведкина М. В., Быков А. Н., Попов В. В. Методы топологической оптимизации в программном комплексе 3D Printer // Математическое моделирование. 2019. Т. 19, № 7. С. 75–90.
Dyanov D. Yu., Medvedkina M. V., Bykov A. N., Popov V. V. Metody topologicheskoy optimizatsii v programmnom komplekse 3D Printer // Matematicheskoe modelirovanie. 2019. T. 19, № 7. S. 75–90.
2. Фрэйн Б. Использование SVG для достижения независимости от разрешения // HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств: Пер. с англ. 2 изд. С.-Пб.: Питер, 2017. С. 175–204.
Freygn B. Ispolzovanie SVG dlya dostizheniya nezavisimosti ot razresheniya // HTML5 i CSS3. Razrabotka saytov dlya lyubykh brauzerov i ustroystv: Per. s angl. 2 izd. S.-Pb.: Piter, 2017. S. 175–204.
3. *Bah T. Inkscape: Guide to a Vector Drawing Program. Prentice Hall, 2011.*
4. Редактор sK1 для работы с векторной графикой. <https://sk1project.net>.
Redaktor sK1 dlya raboty s vektornoj grafikoju. <https://sk1project.net>.
5. Программа Scribus для визуальной верстки документов. <https://www.scribus.net>.
Programma Scribus dlya vizualnoy vyerstki dokumentov. <https://www.scribus.net>.
6. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. М.: Мир, 2001.
Rodzhers D., Adams Dzh. Matematicheskie osnovy mashinnoy grafiki: Per. s angl. M.: Mir, 2001.
7. Малух В. Н. Введение в современные САПР: Курс лекций. М.: ДМК Пресс, 2010.
Malyukh V. N. Vvedenie v sovremennye SAPR: Kurs lektsiy. M.: DMK Press, 2010.
8. Сайт ОАО НПО "ЦНИИТМАШ". <https://www.cniitmash.ru>.
ОАО НПО "TsNIITMASH". <https://www.cniitmash.ru>.

Статья поступила в редакцию 01.03.22.

СВЕДЕНИЯ ОБ АВТОРАХ

Блажнова Ксения Алексеевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, математик, *e-mail*: kablazhnova@vniief.ru

Борисенко Ольга Николаевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, *e-mail*: ONBorisenko@vniief.ru

Ветчинников Максим Владимирович — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, *e-mail*: MVVetchinnikov@vniief.ru

Гиниятуллина Анастасия Геннадьевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: AGGiniyatullina@vniief.ru

Ерофеев Алексей Михайлович — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, заместитель начальника научно-исследовательского отдела — начальник научно-исследовательской лаборатории, *e-mail*: AMErofeev@vniief.ru

Карлыханов Николай Григорьевич — ФГУП "РФЯЦ-ВНИИТФ им. академ. Е. И. Забабахина", г. Снежинск Челябинской области, ведущий научный сотрудник, *e-mail*: n.g.karlykhanov@vniitf.ru

Колобянин Вадим Юрьевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательского отдела, *e-mail*: VYuKolobyanin@vniief.ru

Кондрашенко Алексей Валерьевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, заместитель начальника научно-исследовательского отделения, *e-mail*: AVKondrashenko@vniief.ru

Кузьменко Милана Владимировна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: MVKuzmenko@vniief.ru

Нуждин Андрей Алексеевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, *e-mail*: AANuzhdin@vniief.ru

Смолкина Дина Николаевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: DNSmolkina@vniief.ru

Стаценко Вячеслав Павлович — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, ведущий научный сотрудник, *e-mail*: VPStatsenko@vniief.ru

Тимаева Татьяна Евгеньевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, математик, *e-mail*: tetimaeva@vniief.ru

Топорова Олеся Олеговна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: OOToporova@vniief.ru

Турутина Инна Юрьевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник

Титов Михаил Александрович — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: MATitov@vniief.ru

Хмельницкая Алена Дмитриевна — ФГУП "РФЯЦ-ВНИИТФ им. академ. Е. И. Забабахина", младший научный сотрудник, *e-mail*: KhmelnitskayaAD@vniitf.ru

Черенкова Марина Витальевна — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник

Чернышёв Юрий Дмитриевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, ведущий научный сотрудник, *e-mail*: ydchern@md08.vniief.ru

Чухманов Николай Васильевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, ведущий научный сотрудник, *e-mail*: NVChukhmanov@vniief.ru

Янилкин Юрий Васильевич — ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области, главный научный сотрудник, *e-mail*: YVYanilkin@vniief.ru

INFORMATION ABOUT AUTHORS

Blazhnova Kseniya Alekseevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, mathematician, *e-mail*: kablazhnova@vniief.ru

Borisenko Olga Nikolaevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of the research laboratory, *e-mail*: ONBorisenko@vniief.ru

Vetchinnikov Maksim Vladimirovich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of the research laboratory, *e-mail*: MVVetchinnikov@vniief.ru

Giniyatullina Anastasiya Gennadyevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist, *e-mail*: AGGiniyatullina@vniief.ru

Erofeev Aleksey Mikhailovich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, deputy head of the research department — head of the research laboratory, *e-mail*: AMErofeev@vniief.ru

Karlykhanov Nikolay Grigoryevich — FSUE "RFNC-VNIITF named after Academician E. I. Zababakhin", Snezhinsk, Chelyabinsk region, leading scientist, *e-mail*: n.g.karlykhanov@vniitf.ru

Khmelnitskaya Alena Dmitrievna — FSUE "RFNC-VNIITF named after Academician E. I. Zababakhin", Snezhinsk, Chelyabinsk region, junior scientist, *e-mail*: KhmelnitskayaAD@vniitf.ru

Kolobyanin Vadim Yuryevich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of the research department, *e-mail*: VYuKolobyanin@vniief.ru

Kondrashenko Aleksey Valeryevich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, deputy head of the research department, *e-mail*: AVKondrashenko@vniief.ru

Kuzmenko Milana Vladimirovna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist, *e-mail*: MVKuzmenko@vniief.ru

Nuzhdin Andrey Alekseevich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of the research laboratory, *e-mail*: AANuzhdin@vniief.ru

Smolkina Dina Nikolaevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist, *e-mail*: DNSmolkina@vniief.ru

Statsenko Vyacheslav Pavlovich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, leading scientist, *e-mail*: VPStatsenko@vniief.ru

Timaeva Tatyana Evgenyevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, mathematician, *e-mail*: tetimaeva@vniief.ru

Toporova Olesya Olegovna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist, *e-mail*: OOToporova@vniief.ru

Turutina Inna Yuryevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist

Titov Mikhail Aleksandrovich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist, *e-mail*: MATitov@vniief.ru

Cherenkova Marina Vitalyevna — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist

Chernyshev Yuriy Dmitrievich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, leading scientist, *e-mail*: ydchern@md08.vniief.ru

Chekhmanov Nikolay Vasilyevich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, leading scientist, *e-mail*: NVChukhmanov@vniief.ru

Yanilkin Yuriy Vasilyevich — FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, chief scientist, *e-mail*: YVYanilkin@vniief.ru

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНОГО ТЕЧЕНИЯ ГАЗА В ГАЗОВОЙ ЦЕНТРИФУГЕ. ЧАСТЬ 1. ОПИСАНИЕ МЕТОДИКИ / Ю. В. Янилкин, В. П. Стаценко, Ю. Д. Чернышёв, С. П. Беляев, В. Ю. Колобянин, А. В. Кондрашенко, О. О. Топорова, И. Ю. Турутина // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 3–19.

Статья представляет собой первую часть работы, посвященной трехмерному численному моделированию разреженного газа в реальной газовой центрифуге с учетом физической вязкости и теплопроводности. Приведено описание численной методики, реализация которой проведена на основе методики ЭГАК на цилиндрической сетке в специальной системе координат на параллельной ЭВМ. Часть процессов аппроксимируется в неподвижной декартовой системе координат, а часть — в инерциальной цилиндрической системе. Методика предназначена для сквозного моделирования всего процесса работы центрифуги без применения приближенных методов "сшивки" результатов расчетов в разных областях центрифуги. С этой целью используется приближение, основанное на том, что уравнения газовой динамики, теплопроводности и диффузии применимы всюду. Используется подход, основанный на решении трехмерных уравнений Навье—Стокса во всей центрифуге, в том числе в области двумерного течения в разделительной камере. Для улучшения эффективности программы используется декомпозиция по процессам, в которой на область двумерного течения приходится всего один процесс. Это позволяет экономить вычислительные ресурсы на порядки. Результаты тестирования методики приводятся во второй части работы (рис. — 8, список лит. — 9).

Ключевые слова: газовая центрифуга, разделительная камера, отборная и отвальная камеры, методика ЭГАК, вязкая жидкость, теплообмен, численная методика.

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ТРЕХМЕРНОГО ТЕЧЕНИЯ ГАЗА В ГАЗОВОЙ ЦЕНТРИФУГЕ. ЧАСТЬ 2. РЕЗУЛЬТАТЫ РАСЧЕТОВ МОДЕЛЬНОЙ ЦЕНТРИФУГИ "ИГУАССУ" / Ю. В. Янилкин, В. П. Стаценко, Ю. Д. Чернышёв, С. П. Беляев, В. Ю. Колобянин, А. В. Кондрашенко, О. О. Топорова, И. Ю. Турутина // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 20—36.

Данная статья является продолжением первой части работы авторов, посвященной трехмерному численному моделированию течений в газовой центрифуге. Во второй части приводятся описание шести трехмерных и двумерных тестовых задач для однокамерной и двухкамерной модельной центрифуги "Игуассу" и результаты их моделирования по трехмерной методике, описание которой дано в первой части работы. Рассмотрены следующие задачи: твердотельное вращение газа с постоянной температурой; установление твердотельного вращения; газодинамическая задача с учетом теплопроводности, для которой получено аналитическое решение; однокамерная модель ротора газовой центрифуги, для которой также получено аналитическое решение; задача с двухкамерной моделью ротора газовой центрифуги, в которой исследовалось влияние места сшивки двумерной и трехмерной областей решения и сходимость численного решения при измельчении счетной сетки; одномерная диффузионная задача с аналитическим решением. Для всех тестов получено приемлемое согласие с аналитическими решениями (рис. — 15, список лит. — 9).

Ключевые слова: газовая центрифуга, центрифуга "Игуассу", разделительная камера, отборная камера, отвальная камера, вязкая жидкость, теплообмен, численное моделирование.

УДК 519.6

КВАЗИСПЕКТРАЛЬНЫЙ МЕТОД РЕШЕНИЯ КИНЕТИЧЕСКОГО УРАВНЕНИЯ ПЕРЕНОСА ИЗЛУЧЕНИЯ В ДВУМЕРНОЙ ОСЕСИММЕТРИЧНОЙ ПОСТАНОВКЕ / А. Д. Хмельницкая, Н. Г. Карлыханов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 37—47.

Предлагается квазиспектральный метод для решения уравнения переноса излучения в двумерном осесимметричном случае на прямоугольной сетке. Метод заключается в усреднении уравнения по спектру и угловой переменной и приведении усредненного кинетического уравнения переноса к уравнению лучистой теплопроводности. Достоинство метода в том, что вся сложность итерационного процесса переложена на уравнение лучистой теплопроводности, для которого задача существенно упрощается. Кроме того, метод позволяет организовать распараллеливание по спектральной переменной, а также по пространству для вычисления усредненных по энергии пробегов и эффективной энергии излучения. Показана работоспособность метода на второй задаче Флека, решение сравнивалось со спектральным расчетом в одномерном сферическом случае (рис. — 5, список лит. — 5).

Ключевые слова: квазиспектральный метод, спектральное уравнение переноса излучения, уравнение квазидиффузии, квазиперенос, лучистая теплопроводность, лучевой эффект.

УДК 519.6

ИЕРАРХИЧЕСКИЙ АЛГОРИТМ РАСПАРАЛЛЕЛИВАНИЯ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ ТРЕХМЕРНОГО УРАВНЕНИЯ ПЕРЕНОСА НЕЙТРОНОВ НА ГИБРИДНЫХ СУПЕР-ЭВМ / А. А. Нуждин // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 48–62.

Представлен иерархический алгоритм распараллеливания, который основан на выделении уровней архитектуры гибридной супер-ЭВМ с GPU и применении принципа геометрической декомпозиции на каждом из уровней. Внутри компактной группы нитей используется распараллеливание по элементам гиперплоскости фиксированного размера, состоящей из ячеек сетки. Между блоками одного GPU используется КВА-алгоритм на основе двумерной декомпозиции по столбцам и гиперплоскостям. Между различными GPU используется КВА-алгоритм на основе трехмерной декомпозиции. Программная реализация выполнена на примере тестовой программы ПАУК с помощью технологии CUDA. Эффективность адаптации теста ПАУК к одному GPU подтверждена результатами профилирования и сравнением с производительностью CPU-версии программы. Масштабируемость теста ПАУК в режиме multi-GPU исследована методом умножения (рис. — 5, табл. — 6, список лит. — 14).

Ключевые слова: GPU, CUDA, S_n -метод, алгоритм бегущего счета, КВА-алгоритм.

УДК 519.6

ОПТИМИЗАЦИЯ АЛГОРИТМОВ В ПРИКЛАДНОМ МЕТОДИЧЕСКОМ ТЕСТЕ MD ДЛЯ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ GPU / А. М. Ерофеев, М. В. Ветчинников // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 63–72.

Описываются алгоритмы тестовой программы молекулярной динамики (MD), позволившие осуществить полный перенос всех вычислений на GPU и тем самым избавиться от постоянной необходимости обмена между устройствами информацией о частицах. В результате взаимодействие между GPU и CPU необходимо только для организации пересылки граничной информации между отдельными GPU с помощью MPI на CPU, а это значительно меньше, чем пересылки в первоначальном коде, о чем свидетельствуют тестовые замеры эффективности. На разных по размеру задачах на одном GPU получено ускорение алгоритма относительно первоначального от 8,7 до 12,5 раза. При задействовании двух GPU такое ускорение составило от 6,6 до 12,5 раза. Эффективность распараллеливания на двух GPU V100 составила 76,3–79,6% на задачах с числом частиц от 4 млн, на двух GPU A100 — 77,3–81,8% на задачах с числом частиц от 13,5 млн (рис. — 4, табл. — 6, список лит. — 12).

Ключевые слова: молекулярная динамика, эффективность, GPU, CUDA, алгоритмы.

УДК 519.6

УЛУЧШЕНИЕ КАЧЕСТВА ГРАНЕЙ ОБЪЕМНЫХ ЯЧЕЕК ПРИ ГЕНЕРАЦИИ НЕСТРУКТУРИРОВАННЫХ СЕТОК В ПАКЕТЕ ПРОГРАММ "ЛОГОС" / О. Н. Борисенко, М. В. Кузьменко, М. В. Черенкова, А. Г. Гиниятуллина, Н. В. Чухманов, Д. Н. Смолкина, Т. Е. Тимаева, К. А. Блажнова // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 73—85.

Приводится описание способов улучшения качества граней объемных ячеек при автоматической генерации неструктурированных сеток в пакете программ "Логос", используемом при численном моделировании задач аэро- и гидродинамики. Это модификация ячеек с частично совпадающими гранями, объединение граней с учетом их формы, разрезание искривленных граней, разрезание невыпуклых граней с добавлением нового узла с возможностью удаления малого ребра, разрезание искривленных граней на множество плоских в зонах перехода негладких областей в гладкую область, удаление лишних узлов. Применение совокупности указанных способов позволяет улучшить форму граней ячеек, полученных методом отсечения, и, как следствие, построенного на их основе слоя ячеек, для которых характерен сильный градиент скорости потока. В итоге повышается качество расчетной сетки в целом (рис. — 17, табл. — 1, список лит. — 21).

Ключевые слова: пакет программ "Логос", неструктурированные сетки, улучшение качества сетки, поверхностные грани, пограничный слой.

УДК 004.9

РЕАЛИЗАЦИЯ В ПРОГРАММНОМ КОМПЛЕКСЕ "ВИРТУАЛЬНЫЙ 3D-ПРИНТЕР" ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ВЕКТОРНЫХ ШРИФТОВ ДЛЯ АДДИТИВНОГО ПРОИЗВОДСТВА / М. А. Титов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2022. Вып. 3. С. 86—96.

Представлена возможность применения векторных шрифтов в интегрирующей оболочке программного комплекса "Виртуальный 3D-принтер". Она обеспечивает нанесение текста, символов или логотипа предприятия на конструкции или детали, получаемые в процессе аддитивного производства методом послойного лазерного спекания металлической порошковой смеси. Рассмотрен SVG-формат компьютерных масштабируемых векторных шрифтов. Представлено его краткое описание на языке команд двумерной векторной графики Scalable Vector Graphics. Предложен подход получения координат точек, лежащих на границе обводки символа векторного шрифта. Описаны реализованные для этого синтаксический SVG-анализатор — парсер глифов символов шрифта — и библиотека растеризации квадратичных и кубических кривых Безье. Приведен пример использования разработанных программ для формирования трехмерной сцены из строки символов, выполненной векторным шрифтом. Представлены фотографии полученного в процессе аддитивного производства трехмерного текста (рис. — 15, табл. — 2, список лит. — 8).

Ключевые слова: SVG-формат векторных шрифтов, кривые Безье, парсер глифов символов, библиотека растеризации, аддитивное производство.

NUMERICAL SIMULATION OF 3D GAS FLOW IN GAS CENTRIFUGA. PART 1. DESCRIPTION OF THE METHOD / Yu. V. Yanilkin, V. P. Statsenko, Yu. D. Chernyshev, S. P. Belyaev, V. Yu., Kolobyanin, A. V. Kondrashenko, O. O. Toporova, I. Yu. Turutina // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 3–19.

This is the first part of the work describing the effort on 3D numerical simulation of the rarefied gas in the real gas centrifuga with the account for the physical viscosity and thermal conductivity. A numerical method is described; its realization is carried out on the basis of EGAK on a cylinder mesh in a special coordinate system using a parallel computer. A part of processes is approximated in an immobile Cartesian coordinate system, and the other part — in the inertial cylindrical system. The method is meant for the end-to-end simulation of the whole process of centrifuga operation without implementation of approximated methods of linking the computation results in different regions of the centrifuga. Aiming at this, we use an approximation based on the fact that the equations of gas dynamics, thermal conductivity and diffusion are implemented everywhere. An approach based on the solution of 3D Navier-Stokes equations is used in all centrifuga including the regions of 2D flow in the separating chamber. Decomposition by processes is used to improve the efficiency of the program where only one process comes for the region of the 2D flow. This allows saving computational resources by orders of magnitude. Testing results for the method are given in the second part of the publication.

Key words: gas centrifuga, separating chamber, sample and stripping chamber, EGAK method, viscous liquid, heat exchange, numerical method.

NUMERICAL SIMULATION OF 3D GAS FLOW IN GAS CENTRIFUGA. PART 2. COMPUTATION RESULTS OF MODEL "IGUASSU" CENTRIFUGA / Yu. V. Yanilkin, V. P. Statsenko, Yu. D. Chernyshev, S. P. Belyaev, V. Yu., Kolobyanin, A. V. Kondrashenko, O. O. Toporova, I. Yu. Turutina // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 20–36.

This is the continuation of the first part of the effort on 3D numerical simulation of flows in gas centrifuga by the group of authors. The second part describes test problems on the "Iguassu" model centrifuga and the outcomes of their simulation using the 3D method described in the first part. The following problems were analyzed: solid-state rotation of gas with constant temperature; establishing solid-state rotation; gas-dynamic problem with account for thermal conductivity, for which analytical solution has been produced; one-chamber model of the gas centrifuga rotor, for which analytical solution has been produced; a problem with two-chamber model of the gas centrifuga rotor, where the influence of the point of linking 2D and 3D areas of the solution was studied as well as convergence of numerical solution when refining the mesh; 1D diffusion problem with an analytical solution. Acceptable agreement with analytical solutions was produced for all tests.

Key words: gas centrifuga, "Iguassu" centrifuga, separating chamber, sample chamber, viscous liquid, heat exchange, numerical simulation.

QUASI-SPECTRAL METHOD TO SOLVE THE KINETIC EQUATION OF RADIATION TRANSFER IN 2D AXIALLY SYMMETRIC SETUP / A. D. Khmel'nitskaya, N. G. Karlykhanov // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 37–47.

A quasi-spectral method is proposed to solve radiation transfer equation in 2D axially symmetric case using a rectangular mesh. The method lies in averaging the equation by spectrum and angle variables and reduction of the averaged kinetic equation of transfer to the equation of radiation heat conductivity. The value of the method is in the fact that all the complexity of the iteration is shifted to the equation of the radiation thermal conductivity, for which the problem is considerably simplified. Besides, the method allows arranging parallelization by spectrum variable and by space to compute energy-averaged paths and active radiation energy. The performance of the method is demonstrated in the second Fleck problem; the solution is compared to the spectrum computation in 1D spherical case.

Key words: quasi-spectral method, spectral equation of radiation transfer, quasi-diffusion equation, quasi-transfer, radiation thermal conductivity, radiation effect.

HIERARCHICAL PARALLELIZATION ALGORITHM WHEN COMPUTING 3D NEUTRON TRANSFER EQUATION ON HYBRID SUPER-COMPUTERS / A. A. Nuzhdin // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 48–62.

A hierarchical parallelization algorithm is described in the paper, which is based on distinguishing the levels of the architecture of the hybrid super-computer with GPU and implementation of the principle of geometric decomposition at every level. Inside a compact group of threads parallelization by the elements of the hyper-plane of a fixed size is used; it consists of mesh cells. KBA-algorithm on the basis of 2D decomposition by columns and hyper-planes is used between the blocks of one GPU. KBA-algorithm on the basis of 3D decomposition is used between different GPU. The program is realized on the example of PAUK mini-app using CUDA technology. The efficiency of adaptation of the PAUK mini-app for one GPU is confirmed with the results of profiling and comparison with the performance of CPU-version of the program. Scalability of PAUK mini-app in the mode of multi-GPU is studied with multiplication method.

Key words: GPU, CUDA, Sn-method, sweep algorithm, KBA-algorithm.

OPTIMIZATION OF ALGORITHMS IN APPLIED TECHNICAL MD TEST FOR EFFICIENT GPU IMPLEMENTATION / A. M. Erofeev, M. V. Vetchinnikov // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 63–72.

Algorithms of molecular dynamics (MD) benchmark are described that allow completing the transfer of all computations on GPU in such a way that we get rid of a constant need of particle information exchange between the devices. As a result, interaction between GPU and CPU is necessary only to send boundary information between separate GPU using MPI to CPU, and this is much less than sending in the initial code; that is proven by the test measurement of the performance. The speedup of the algorithm on one GPU compared to the initial one is from 8,7 to 12,5 times in case of problems different in their size. When two GPUs were involved in this work, the speedup was from 6,6 to 12,5 times. The parallel efficiency on two GPUs V100 was 76,3–79,6 % in case of problems with the number of particles from 4 mln; on two GPUs A100 it was 77,3–81,8 % in case of problems with the number of particles from 13,5 mln.

Key words: molecular dynamics, performance, GPU, CUDA, algorithms.

QUALITY IMPROVEMENT OF THE FACETS OF 3D CELLS WHEN GENERATING UNSTRUCTURED MESHES IN "LOGOS" SOFTWARE PACKAGE / O. N. Borisenko, M. V. Kuzmenko, M. V. Cherenkova, A. G. Giniyatullina, N. V. Chukhmanov, D. N. Smolkina, T. E. Timaeva, K. A. Blazhnova // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 73–85.

The ways to improve the quality of the facets of 3D cells in case of automatic generation of unstructured meshes in "Logos" software package are described; they are used in numerical simulation of the problems of aero- and hydrodynamics. This is a modification of cells with partially coinciding facets, integration of facets with the account for their shape, cutting of the curved facets, cutting of not-convex facets with adding a new node having a possibility to delete a small edge, cutting of the curved facets into the multiple plane ones in the transfer regions of the not-smooth regions into the smooth area, and deleting extra nodes. Implementation of the integrity of these ways allows improving the shape of the facets of the cells produced by the cutting-off method and, as a result, the layer of the cells generated on their basis, where a strong flow velocity gradient is characteristic. Eventually, the quality of the computational mesh in general is improved.

Key words: "Logos" software package, unstructured meshes, mesh quality improvement, surface facets, boundary layer.

POSSIBLE IMPLEMENTATION OF VECTOR FONTS FOR ADDITIVE PRODUCTION REALIZED IN "VIRTUAL 3D-PRINTER" SOFTWARE COMPLEX / M. A. Titov // VANT. Ser.: Mat. Mod. Fiz. Proc. 2022. No 3. P. 86—96.

A possibility to use vector fonts in the integrating frame of the "Virtual 3D-printer" software package is shown. It provides application of the text, symbols or a logotype of the company on the structures and items produced in the process of additive production using the method of layer-by-layer laser sintering of a metal powder mixture. SVG-format of computer scalable vector fonts is considered. Its brief description is provided in the language of commands of 2D Scalable Vector Graphics. An approach is proposed to produce the coordinates of the points that lie on the boundary of delineating the symbol of the vector font. A syntactical SVG-analyzer (a parser of glyphs of the font symbols) and the rasterization library of quadratic and cubic Bezier curves realized for these purposes are described. There is an example of using developed programs to form a 3D scene from the line of symbols produced with a vector font. There are photos of the 3D text produced in the process of additive production.

Key words: SVG-format of vector fonts, Bezier curves, a parser of glyphs of symbols, rasterization library, additive production.