

ISSN 2414-0171

ВОПРОСЫ
АТОМНОЙ
НАУКИ И
ТЕХНИКИ

Российский
федеральный
ядерный
центр –
ВНИИЭФ

СЕРИЯ

**Математическое моделирование
физических процессов**

2020

ВЫПУСК 4

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., Иванов Н. В., Кибзун А. И.,
Козелков А. С., Козманов М. Ю., Куркин А. А., 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/>



*29 сентября 2020 года
исполнилось 70 лет
доктору физико-математических наук, профессору
РАШИТУ МИРЗАГАЛИЕВИЧУ ШАГАЛИЕВУ*

*Редакционная коллегия сердечно поздравляет с юбилеем
Рашита Мирзагалиевича,
более десяти лет руководящего работой журнала,
и желает ему крепкого здоровья
и большой энергии для реализации
дальнейших творческих замыслов.*

ВОПРОСЫ
АТОМНОЙ НАУКИ И ТЕХНИКИ

СЕРИЯ

Математическое моделирование
физических процессов

НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

ВЫПУСК 4

Саров — 2020

Издается с 1978 г.

СОДЕРЖАНИЕ

<i>Софронов В. Н., Ветчинников М. В., Дёмина М. А.</i> Использование методов гамильтоновой динамики в численных расчетах задач механики сплошной среды	5
<i>Близнюк О. Г., Власова О. Е., Гичук А. В., Козелков А. С., Лялюшкина И. В.</i> Численное моделирование многофазных течений в эйлеровом приближении на основе метода SIMPLE	22
<i>Ерёменко А. Ю., Косарим С. С.</i> Пакет программ "Логос". Учет контактного взаимодействия при решении задач квазистатического деформирования	33
<i>Нуждин А. А.</i> Тестовая программа ПАУК как полигон для апробации алгоритмов и технологий параллельного программирования	48
<i>Лазарев В. В.</i> Эффективный алгоритм объединения согласованных по узлам фрагментов поверхностных сеток	62
<i>Арапова Е. Ю., Куделькин В. Г., Павлов Е. А., Полякова С. Ю., Тихонов А. В.</i> Генератор поверхностных неструктурированных четырехугольных сеток в препроцессоре пакета программ "Логос"	72
<i>Артёмова Н. А., Ушакова О. В.</i> О развитии алгоритма построения сеток в деформированных телах вращения для случая их деформации телами вращения, образованными несколькими поверхностями	86
Сведения об авторах	97
Перечень статей, опубликованных в 2020 г. в научно-техническом сборнике "Вопросы атомной науки и техники". Сер. "Математическое моделирование физических процессов"	99

CONTENTS

<i>Sofronov V. N., Vetchinnikov M. V., Dyemina M. A.</i> Use of Hamiltonian dynamics methods in computational continuum mechanics	5
<i>Bliznyuk O. G., Vlasova O. E., Gichuk A. V., Kozelkov A. S., Lyalyushkina I. V.</i> Eulerian multi-phase flow simulations based on the SIMPLE method	22
<i>Eryemenko A. Yu., Kosarim S. S.</i> "Logos" software package. Incorporation of contact interactions in quasi-static deformation simulations	33
<i>Nuzhdin A. A.</i> Test program PAUK as a testing range for parallel programming algorithms and technologies	48
<i>Lazarev V. V.</i> An efficient algorithm for merging node-matched fragments of surface grids	62
<i>Arapova E. Yu., Kudelkin V. G., Pavlov E. A., Polyakova S. Yu., Tikhonov A. V.</i> An unstructured quadrilateral surface grid generator in the preprocessor of the "Logos" software package	72
<i>Artyomova N. A., Ushakova O. V.</i> On the development of a grid generation algorithm for deformed solids of revolution formed by several surfaces	86
Information about authors	98
The list of papers published in the sci-tech collected edition "Voprosy atomnoi nauki i tekhniki". Ser. "Matematicheskoe Modelirovanie Fizicheskikh Protsessov" during 2020	99

Ответственный за выпуск Е. В. Соколовская

Редактор Е. Н. Старченко

Корректоры Н. Ю. Костюничева
Т. А. Меркушева
Е. А. Окальева
А. В. Федоренко

Подписано в печать 25.10.2020

Усл. печ. л. ~ 12

Тираж 1000 экз.

Зак. тип. 2029-2020

Формат 60×84/8

Уч.-изд. л. ~ 14

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 539.3

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ГАМИЛЬТОНОВОЙ ДИНАМИКИ В ЧИСЛЕННЫХ РАСЧЕТАХ ЗАДАЧ МЕХАНИКИ СПЛОШНОЙ СРЕДЫ

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

Представлен метод численного решения задач механики сплошной среды, основанный на дискретной гамильтоновой динамике. Для численного решения гамильтоновых уравнений используются симплектические разностные схемы. Приведены примеры моделирования бездиссипативных процессов.

Ключевые слова: дискретная гамильтонова динамика, фазовый объем, симплектические разностные схемы, динамические задачи упругости, пакет программ MoDyS.

Введение

Применение классических методов механики сплошной среды (МСС) привело к достижению больших успехов в решении прикладных задач. К сожалению, аппарат континуальной механики плохо приспособлен для расчета эффектов, связанных с большими деформациями и нарушением сплошности материала (дробление, пыление, диспергирование). Кроме того, для континуального подхода достаточно острой является проблема, связанная с генерацией расчетных сеток и их корректировкой в результате сильных деформаций и изменения геометрии.

В бессеточных методах (SPH [1], перидинамика PD [2], кластерная динамика KD [3], подвижные клеточные автоматы MSA [4]) сплошная среда заменяется системой взаимодействующих мезочастиц. Для методов этого класса не всегда удается сконструировать микроскопические взаимодействия так, чтобы получить заданные макроскопические свойства. Уравнения движения мезочастиц обычно записаны в интегродифференциальной форме, и не всегда устанавливается их связь с уравнениями гамильтоновой динамики.

Для построения дискретной модели в данной статье, как и в работе [5], использовались идеи гамильтоновой динамики. В отличие от [5] в расчетах были применены разностные методы с высоким порядком аппроксимации и сохранением симплектической структуры исходных дифференциальных уравнений.

Гамильтоновы методы, основанные на приведении уравнений к гамильтоновой форме, используются в различных разделах теоретической физики. В задачах МСС почти всегда присутствуют диссипативные процессы, разрушающие многие свойства гамильтоновых систем (например, сохранение фазового объема).

Предлагаемый метод в исходном виде не предполагает наличия диссипативных процессов и может использоваться только для решения ограниченного круга задач (акустики, сейсмологии, динамики гипотупругих сред). Учет диссипации возможен в рамках диссипативной динамики частиц DPD [6]. При моделировании упругопластических течений система уравнений дискретной модели должна быть дополнена соотношениями для вычисления тензора пластической дисторсии.

Программная реализация дискретной модели в рамках пакета молекулярно-динамического (МД) моделирования MoDyS [7] сводится к замене процедуры вычисления сил. Основные счетные модули, программы пре- и постобработки остаются без изменения. Использование дискретного метода (как и любого разностного метода) в практически важных расчетах невозможно без распараллеливания вычислительных процессов.

В результате реализации дискретного метода в комплексе программ MoDyS появляется возможность проведения расчетов задач МСС на новых суперЭВМ с использованием интерфейсов MPI, OpenMP и технологии CUDA с высокой (более 80 %) эффективностью распараллеливания.

1. Построение дискретной модели

Предполагается, что каждая частица (например, с номером i) является макрообъектом и наделена параметрами: массой m_i , плотностью ρ_i , начальными координатами $\vec{r}_i^0 (x_i^0, y_i^0, z_i^0)$ и компонентами скорости $\vec{u}_i (u_i^x, u_i^y, u_i^z)$. Кроме того, для каждой частицы заданы какие-либо два модуля упругости, например K_i — модуль объемного сжатия, G_i — модуль сдвига. Движение деформируемой среды характеризуется дискретным вектором смещений $\vec{w}_i (w_i^x, w_i^y, w_i^z)$. Верхний нулевой индекс будет использоваться для обозначения начального состояния.

Пусть N_i — список *sosedei* частицы с номером i . В этот список входят частицы, координаты которых удовлетворяют условию $|\vec{r}_i^0 - \vec{r}_k^0| < \delta_r$ (δ_r — горизонт взаимодействия).

Тензор* дисторсии $[C^i] = \left(\frac{\partial \vec{w}}{\partial \vec{r}} \right)^i_{k \in N_i}$ для частицы i на дискретном множестве частиц N_i может быть определен по методу наименьших квадратов. В этом случае он примет следующий вид [5]:

$$[C^i]^T = [A^i]^{-1} [B^i]. \quad (1)$$

Таким образом, для вычисления тензора дисторсии требуется определить элементы двух матриц (симметричной $[A^i]$ и несимметричной $[B^i]$):

$$[A^i] = \sum_{k \in N_i} (\vec{r}_i^0 - \vec{r}_k^0) \otimes (\vec{r}_i^0 - \vec{r}_k^0), \quad \text{т. е.}$$

$$A_{11}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (x_i^0 - x_k^0); \quad A_{12}^i = A_{21}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (y_i^0 - y_k^0);$$

$$A_{13}^i = A_{31}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (z_i^0 - z_k^0); \quad A_{23}^i = A_{32}^i = \sum_{k \in N_i} (y_i^0 - y_k^0) (z_i^0 - z_k^0);$$

$$A_{22}^i = \sum_{k \in N_i} (y_i^0 - y_k^0) (y_i^0 - y_k^0); \quad A_{33}^i = \sum_{k \in N_i} (z_i^0 - z_k^0) (z_i^0 - z_k^0);$$

$$[B^i] = \sum_{k \in N_i} (\vec{r}_i^0 - \vec{r}_k^0) \otimes (\vec{w}_i - \vec{w}_k), \quad \text{т. е.}$$

$$B_{11}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (w_i^x - w_k^x); \quad B_{12}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (w_i^y - w_k^y); \quad B_{13}^i = \sum_{k \in N_i} (x_i^0 - x_k^0) (w_i^z - w_k^z);$$

$$B_{21}^i = \sum_{k \in N_i} (y_i^0 - y_k^0) (w_i^x - w_k^x); \quad B_{22}^i = \sum_{k \in N_i} (y_i^0 - y_k^0) (w_i^y - w_k^y); \quad B_{23}^i = \sum_{k \in N_i} (y_i^0 - y_k^0) (w_i^z - w_k^z);$$

$$B_{31}^i = \sum_{k \in N_i} (z_i^0 - z_k^0) (w_i^x - w_k^x); \quad B_{32}^i = \sum_{k \in N_i} (z_i^0 - z_k^0) (w_i^y - w_k^y); \quad B_{33}^i = \sum_{k \in N_i} (z_i^0 - z_k^0) (w_i^z - w_k^z).$$

В (1) $[A^i]^{-1}$ — матрица, обратная к $[A^i]$:

$$[A^i]^{-1} = \frac{1}{|A^i|} \begin{pmatrix} A_{22}^i A_{33}^i - A_{23}^i A_{23}^i & A_{13}^i A_{23}^i - A_{12}^i A_{33}^i & A_{12}^i A_{23}^i - A_{13}^i A_{22}^i \\ A_{13}^i A_{23}^i - A_{12}^i A_{33}^i & A_{11}^i A_{33}^i - A_{13}^i A_{13}^i & A_{13}^i A_{12}^i - A_{11}^i A_{23}^i \\ A_{12}^i A_{23}^i - A_{13}^i A_{22}^i & A_{13}^i A_{12}^i - A_{11}^i A_{23}^i & A_{11}^i A_{22}^i - A_{12}^i A_{12}^i \end{pmatrix};$$

$$|A^i| = A_{11}^i A_{22}^i A_{33}^i + 2A_{12}^i A_{13}^i A_{23}^i - A_{11}^i A_{23}^i A_{23}^i - A_{22}^i A_{13}^i A_{13}^i - A_{33}^i A_{12}^i A_{12}^i.$$

* В данной статье в качестве тензоров всюду используются матрицы. Для облегчения восприятия обозначения матриц, в отличие от их элементов, будем заключать в квадратные скобки. Элементы же матриц, как обычно, снабжаются нижними индексами.

В условиях однородной деформации вектор смещения имеет вид

$$\vec{w} = [\check{C}] \vec{r}, \quad [\check{C}] = \frac{\partial \vec{w}}{\partial \vec{r}}. \quad (2)$$

Здесь матрица дисторсии $[\check{C}]$ не зависит от координат.

Покажем, что элементы тензора дисторсии для поля $[C^i]$ смещений (2), вычисленные с помощью процедуры численного дифференцирования (1), совпадают с элементами матрицы $[\check{C}]$:

$$\begin{aligned} [B^i] &= \sum_{k \in N_i} (\vec{r}_i^0 - \vec{r}_k^0) \otimes (\vec{w}_i - \vec{w}_k) = \sum_{k \in N_i} (\vec{r}_i^0 - \vec{r}_k^0) \otimes [\check{C}] (\vec{r}_i^0 - \vec{r}_k^0) = \\ &= \left(\sum_{k \in N_i} (\vec{r}_i^0 - \vec{r}_k^0) \otimes (\vec{r}_i^0 - \vec{r}_k^0) \right) [\check{C}]^T = [A^i] [\check{C}]^T. \end{aligned}$$

Таким образом, $[C^i]^T = [A^i]^{-1} [B^i] = [\check{C}]^T$.

Тензор деформаций $[E^i]$ является симметричной частью тензора дисторсии $[C^i]$:

$$[E^i] = 0,5 \left([C^i]^T + [C^i] \right).$$

Как обычно, тензор деформаций может быть представлен в виде суммы шарового тензора и дивиа-тора:

$$[E^i] = [E_V^i] + [E_S^i], \quad [E_V^i] = \frac{1}{3} (\delta_{kl} [E^i]), \quad E_{jj}^i = \frac{\rho_i^0}{\rho_i} - 1,$$

где δ_{kl} — символ Кронеккера.

Энергия, приходящаяся на одну частицу i , с точностью до членов второго порядка имеет вид [8]

$$U_i = \frac{\lambda_i}{2} \sum_j E_{jj}^i E_{jj}^i + G_i \sum_{k,l} E_{kl}^i E_{kl}^i = \frac{K_i}{2} \text{Sp}^2 [E_V^i] + G_i [E_S^i] : [E_S^i]. \quad (3)$$

Здесь G_i и $\lambda_i = K_i - \frac{2}{3}G_i$ — коэффициенты Ламэ. Тензор напряжений Пиолы—Кирхгофа $[\sigma^i]$ определяется дифференцированием потенциала U_i , заданного формулой (3):

$$\begin{aligned} [\sigma^i] &= \left[\frac{\partial U_i}{\partial E_{kl}^i} \right] = K_i \delta_{kl} \sum_j E_{jj}^i + 2G_i \left(E_{kl}^i - \frac{1}{3} \delta_{kl} \sum_j E_{jj}^i \right), \quad \text{т. е.} \\ \sigma_{11}^i &= \left(K_i - \frac{2G_i}{3} \right) (E_{11}^i + E_{22}^i + E_{33}^i) + 2G_i E_{11}^i; \quad \sigma_{12}^i = \sigma_{21}^i = 2G_i E_{12}^i; \\ \sigma_{22}^i &= \left(K_i - \frac{2G_i}{3} \right) (E_{11}^i + E_{22}^i + E_{33}^i) + 2G_i E_{22}^i; \quad \sigma_{13}^i = \sigma_{31}^i = 2G_i E_{13}^i; \\ \sigma_{33}^i &= \left(K_i - \frac{2G_i}{3} \right) (E_{11}^i + E_{22}^i + E_{33}^i) + 2G_i E_{33}^i; \quad \sigma_{23}^i = \sigma_{32}^i = 2G_i E_{23}^i. \end{aligned}$$

Тензор напряжений Пиолы—Кирхгофа $[\sigma^i]$ отнесен к системе координат начального (до деформации) состояния. Переход к тензору напряжений Коши $[\Sigma^i]$, отнесенному к актуальному (текущему) состоянию, осуществляется по формулам $[\Sigma^i] = \frac{\rho_i}{\rho_i^0} [\sigma^i] ([I] + [C^i])$ [9–11].

С учетом (3) гамильтониан системы представим в виде

$$H = \sum_i \left(\frac{|\vec{p}_i|^2}{2m_i} + V_i^0 U_i \right) = \sum_i \left(\frac{|\vec{p}_i|^2}{2m_i} + \frac{m_i}{\rho_i^0} U_i \right), \quad (4)$$

где V_i^0, \vec{p}_i — начальный объем и импульс i -й частицы.

Динамические переменные \vec{r}_i, \vec{p}_i изменяются во времени согласно уравнениям Гамильтона:

$$\frac{d\vec{r}_i}{dt} = \frac{\partial H}{\partial \vec{p}_i} = \vec{u}_i, \quad \frac{d\vec{p}_i}{dt} = m_i \frac{d\vec{u}_i}{dt} = -\frac{\partial H}{\partial \vec{r}_i} = \vec{F}_i. \quad (5)$$

Сила, действующая на частицу, определяется дифференцированием гамильтониана:

$$\vec{F}_i = -\frac{\partial H}{\partial \vec{r}_i} = -V_i^0 \left[\frac{\partial U_i}{\partial \vec{E}^i} \right] \frac{\partial E^i}{\partial \vec{r}_i} = -\sum_{k \in N_i} V_i^0 [\sigma^i] [A^i]^{-1} (\vec{r}_i^0 - \vec{r}_k^0) = \sum_{k \in N_i} \vec{F}_{ik}. \quad (6)$$

В молекулярной (и кластерной) динамике тензор напряжений $[S^i]$ при заданной силе взаимодействия частиц i, k вычисляется по формуле

$$[S^i] V_i = \sum_{k \in N_i} \vec{F}_{ik} \otimes (\vec{r}_i - \vec{r}_k).$$

В результате преобразования с учетом (6) получаем, что так определенный тензор напряжений $[S^i]$ будет совпадать с тензором Коши:

$$\begin{aligned} [S^i] V_i &= \sum_{k \in N_i} V_i^0 [\sigma^i] [A^i]^{-1} (\vec{r}_i^0 - \vec{r}_k^0) \otimes (\vec{r}_i - \vec{r}_k) = V_i^0 [\sigma^i] [A^i]^{-1} ([A^i] + [B^i]) = \\ &= V_i^0 [\sigma^i] ([I] + [C^i]) = V_i^0 [\Sigma^i] \frac{\rho_i^0}{\rho_i} = \frac{m_i [\Sigma^i]}{\rho_i} = [\Sigma^i] V_i. \end{aligned}$$

Классическими следствиями уравнений (5) являются законы сохранения энергии, импульса и момента импульса. Гамильтоновы системы, кроме этих стандартных интегралов движения, имеют целую иерархию интегральных инвариантов [12]. Не вдаваясь в детали этой иерархической структуры, отметим лишь, что для гамильтоновых систем инвариантом движения является фазовый объем.

Пусть начальные данные принадлежат некоторой $6N$ -мерной области фазового пространства: $(\vec{r}^0, \vec{p}^0) \in \Omega_0$. В соответствии с теоремой Лиувилля [12] фазовый поток $\tilde{G} : R^{6N}(\vec{r}^0, \vec{p}^0) \rightarrow R^{6N}(\vec{r}, \vec{p})$ сохраняет фазовый объем: $\int d\vec{r}^0 d\vec{p}^0 = \int d\vec{r} d\vec{p}$.

Критерий симплектичности фазового потока \tilde{G} с матрицей перехода $[D] = \frac{\partial(\vec{r}, \vec{p})}{\partial(\vec{r}^0, \vec{p}^0)}$, которая является матрицей Якоби и выражается матричным соотношением [12]

$$[D]^T [J] [D] = [J], \quad [J] = \begin{pmatrix} 0 & [-E_{3N}] \\ [E_{3N}] & 0 \end{pmatrix}, \quad (7)$$

где $[E_{3N}]$ — единичная матрица порядка $3N$.

2. Симплектические разностные схемы численного решения натуральных гамильтоновых систем

Требования, предъявляемые к разностным схемам МСС, общеизвестны. Они заключаются в выполнении законов сохранения на дискретном уровне [13]. Так как гамильтоновы системы, кроме стандартных интегралов движения, имеют дополнительные интегральные инварианты, то и разностные методы решения таких систем должны наследовать все свойства соответствующих дифференциальных уравнений [14].

В соответствии с (7) одношаговый разностный метод называется симплектическим, если оператор перехода $[D^{n+1}] = \frac{\partial(\vec{r}^{n+1}, \vec{p}^{n+1})}{\partial(\vec{r}^n, \vec{p}^n)}$ осуществляет симплектическое преобразование динамических переменных на каждом временном шаге [14, 15]:

$$[D^{n+1}]^T [J] [D^{n+1}] = [J]. \quad (8)$$

Разностные схемы с выполнением условия (8) построены в работах Ю. Б. Суриса и Сан-Серны [14, 15]. Для систем с гамильтонианами общего вида не существует явных симплектических схем [16]. Такие схемы могут быть построены только для натуральных гамильтоновых систем (систем с распадающимися гамильтонианами):

$$H(\vec{r}, \vec{p}) = K(\vec{p}) + U(\vec{r}).$$

Дискретная модель, построенная в разд. 1, принадлежит именно к этому классу натуральных гамильтоновых систем. Для численного решения уравнений (4), (5) будем использовать разностные схемы с выполнением условия симплектичности (8). Можно надеяться, что подобные схемы будут лучше воспроизводить структуру решения в фазовом пространстве.

Для m -стадийных разностных схем Рунге—Кутты—Нистрема (RKN)

$$\begin{aligned} \vec{r}^{n+1} &= \vec{r}^n + \Delta t \vec{u}^n + \Delta t^2 \sum_{l=1}^m \beta_l \vec{F}(\vec{q}^l); \\ \vec{u}^{n+1} &= \vec{u}^n + \Delta t \sum_{l=1}^m \gamma_l \vec{F}(\vec{q}^l), \quad \vec{q}^l = \vec{r}^n + \Delta t \alpha_l \vec{u}^n + \Delta t^2 \sum_{k=1}^m a_{lk} \vec{F}(\vec{q}^k) \end{aligned}$$

условие симплектичности (8) будет выполнено при следующих ограничениях на коэффициенты разностной схемы [14, 15]:

$$\begin{aligned} \beta_l &= \gamma_l (1 - \alpha_l), \quad l = 1, 2, \dots, m; \\ \gamma_l a_{lk} - \gamma_k a_{kl} + \gamma_l \gamma_k (\alpha_k - \alpha_l) &= 0, \quad l, k = 1, 2, \dots, m. \end{aligned}$$

Для явных схем матрица коэффициентов $[a_{lk}]$, $l, k = 1, 2, \dots, m$, является нижней треугольной. В таком случае связь элементов этой матрицы с параметрами α_l , γ_l определяется однозначно:

$$a_{lk} = \begin{cases} 0, & 1 \leq l \leq k \leq m; \\ \gamma_k (\alpha_l - \alpha_k), & 1 \leq k < l \leq m. \end{cases}$$

Таким образом, m -стадийные симплектические схемы RKN полностью определяются заданием двух векторов: $\alpha = (\alpha_1, \dots, \alpha_m)$ и $\gamma = (\gamma_1, \dots, \gamma_m)$.

Следующим шагом в процессе построения схемы является выполнение условий порядка аппроксимации для коэффициентов схемы. Эти условия получаются приравнованием соответствующих коэффициентов в степенных разложениях точного и численного решений, что и обеспечивает нужный порядок аппроксимации.

Описание классов одно- и двухстадийных симплектических схем RKN не представляет трудностей. В классе одностадийных схем ($m = 1$) существует единственная симплектическая схема второго порядка — схема Верле [17] с параметрами $\alpha_1 = 0,5$, $\gamma_1 = 1$. Данная схема получила наибольшее распространение в МД-моделировании.

Порядок аппроксимации выше второго может быть получен только в том случае, если число стадий увеличивается до трех ($m \geq 3$). Проблема описания многообразия явных трехстадийных симплектических схем решена в [18–20]. Доказано, что в этом классе отсутствуют схемы пятого порядка, существует семь схем четвертого порядка [18–20], три из которых — с действительными коэффициентами:

– схема SIA4 [21]:

$$\alpha_1 = \frac{2^{1/3}}{6} + \frac{2^{2/3}}{12} + \frac{1}{3}; \quad \alpha_2 = \frac{1}{2}; \quad \alpha_3 = \frac{2}{3} - \frac{2^{1/3}}{6} - \frac{2^{2/3}}{12};$$

$$\gamma_1 = \frac{2^{1/3}}{3} + \frac{2^{2/3}}{6} + \frac{2}{3}; \quad \gamma_2 = -\frac{1}{3} - \frac{2 \cdot 2^{1/3}}{3} - \frac{2^{2/3}}{3}; \quad \gamma_3 = \frac{2^{1/3}}{3} + \frac{2^{2/3}}{6} + \frac{2}{3};$$

– два варианта схемы RKN4 [22] в исходном и сопряженном (в алгебраическом смысле – с изменением знаков при радикалах) виде:

$$\alpha_1 = \frac{3 \mp \sqrt{3}}{6}; \quad \alpha_2 = \frac{3 \pm \sqrt{3}}{6}; \quad \alpha_3 = \frac{3 \mp \sqrt{3}}{6}; \quad \gamma_1 = \frac{3 \pm 2\sqrt{3}}{12}; \quad \gamma_2 = \frac{1}{2}; \quad \gamma_3 = \frac{3 \mp 2\sqrt{3}}{12}.$$

Большая часть последующих тестовых расчетов проведена с использованием схемы Верле.

3. Программная реализация дискретной модели в рамках пакета программ MoDyS

Представленная дискретная модель отличается от МД-метода процедурой вычисления сил, и было естественным желание реализовать эту модель в рамках пакета программ MoDyS [7]. В этом МД-пакете для быстрого поиска взаимодействующих частиц используется сеточный подход, основанный на построении и обновлении на каждом счетном шаге списка Верле. Для дискретной модели связанный список строится один раз (после задания начальной геометрии) и остается неизменным на всем протяжении расчета.

Все программы пре- и постобработки пакета MoDyS остались практически без каких-либо изменений. Для параллельных вычислений потребовалась небольшая модификация программ, выполняющих обмены данными между MPI-процессами.

Блок-схема дискретной модели показана на рис. 1.

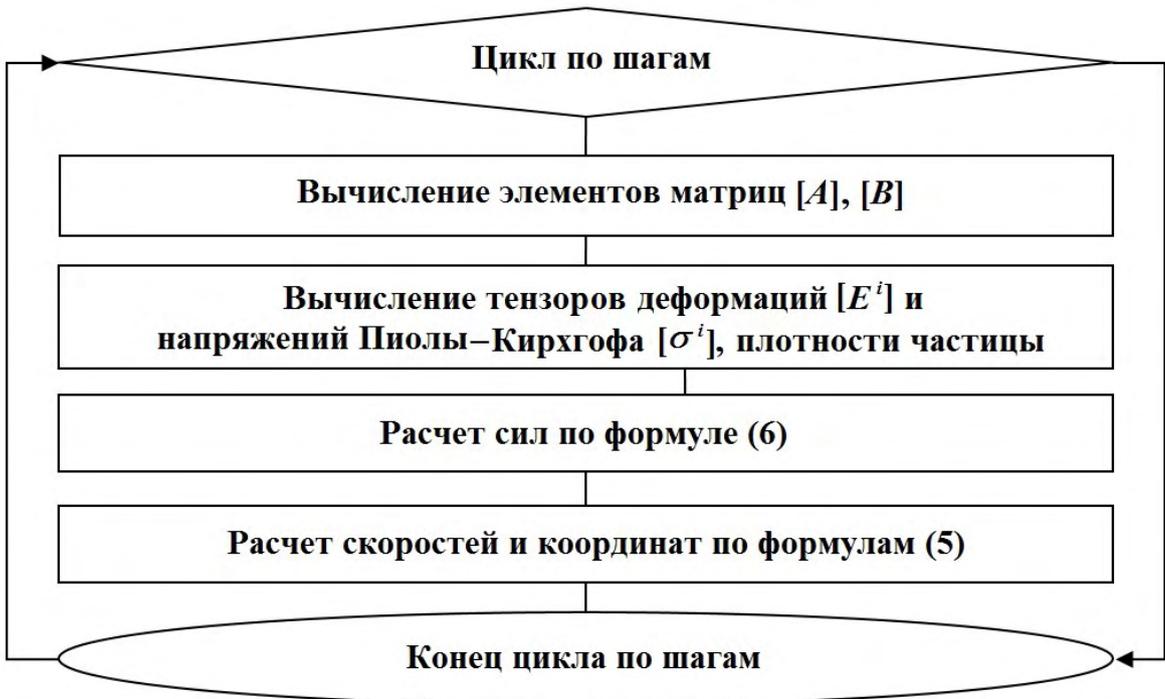


Рис. 1. Блок-схема дискретной модели

4. Особенности дискретной модели

Дискретная модель описывается нелокальной теорией, содержащей масштаб длины — горизонт взаимодействия (δ_r). Ответ на вопрос, что происходит с уравнениями дискретной модели при уменьшении масштаба длины, переходят ли эти уравнения в классические уравнения динамической теории упругости, получен в [23]. Было показано, что если деформации, определяющие модель, и какие-либо неоднородности достаточно гладкие, то тензор напряжений дискретной модели в предельном случае $\delta_r \rightarrow 0$ переходит в тензор Пиолы—Кирхгофа, а уравнения гамильтоновой механики превращаются в уравнения теории упругости. Соответственно предельные численные решения этих уравнений (при измельчении сетки для МСС и соответственно увеличении числа частиц для дискретной модели) также будут совпадать. Однако результаты расчетов при конечных размерах сетки и при конечном числе частиц могут различаться, так как для уравнений МСС и уравнений гамильтоновой механики используются совершенно разные численные методы. Хорошим качеством симплектических разностных схем является отсутствие диссипации — все характеристические числа матрицы перехода лежат на единичной окружности комплексной плоскости [12].

Для оценки точности численных методов будем использовать достаточно простые задачи, имеющие эталонные решения.

5. Эталонные решения тестовых задач

5.1. Прохождение упругих волн вдоль тонкого стержня. Рассмотрим прохождение упругих волн по цепочке частиц и соответственно прохождение волн по тонкому стержню [20]. Для плоских одномерных течений тензоры деформаций и напряжений имеют вид

$$[E] = \begin{pmatrix} \varepsilon & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad [\sigma] = \begin{pmatrix} (K + 4G/3)\varepsilon & 0 & 0 \\ 0 & (K - 2G/3)\varepsilon & 0 \\ 0 & 0 & (K - 2G/3)\varepsilon \end{pmatrix}.$$

Пусть $K = 2G/3$, тогда коэффициент Пуассона $\nu = 0$, модуль Юнга $E = 3K$, $\varepsilon = 0,001$. В этом случае в тонком стержне распространяются волны одного направления (вдоль оси X), так как поперечные напряжения отсутствуют: $\sigma_{22} = \sigma_{33} = 0$. Волны будут распространяться вдоль стержня со скоростью $C_E = \sqrt{E/\rho} = \sqrt{3K/\rho}$.

Постановка задачи для разрывных начальных данных взята из монографии по перидинамике [24]. Начальные условия (w — смещение; σ — нормальное напряжение; u — скорость; $2L$ — длина стержня):

$$w(x, t = 0) = \varepsilon \cdot \begin{cases} x, & 0 < x < L; \\ 2L - x, & L < x < 2L; \end{cases}$$

$$\sigma(x, t = 0) = E \frac{\partial w(x, t = 0)}{\partial t} = \begin{cases} E\varepsilon, & 0 < x < L; \\ -E\varepsilon, & L < x < 2L. \end{cases}$$

Граничные условия:

$$u(x = 0, t) = 0; \quad u(x = 2L, t) = 0. \quad (9)$$

Эталонное решение задачи в данной постановке имеет вид [24]

$$w(x, t) = \frac{8\varepsilon L}{\pi^2} \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2} \sin \frac{(2n+1)\pi x}{2L} \cos \sqrt{\frac{E}{\rho}} \frac{(2n+1)\pi t}{2};$$

$$u(x, t) = \frac{\partial w(x, t)}{\partial t}; \quad \sigma(x, t) = E \frac{\partial w(x, t)}{\partial x}.$$

В этой задаче представляет интерес исследование амплитудной и фазовой ошибок разностных схем на достаточно большом временном интервале.

Для гладких начальных данных

$$w(x, t = 0) = \varepsilon L \sin \frac{\pi x}{2L}; \quad \sigma(x, t = 0) = E \frac{\varepsilon \pi}{2} \cos \frac{\pi x}{2L}; \quad u(x, t = 0) = 0$$

при сохранении граничных условий (9) эталонное решение данной задачи имеет вид

$$w(x, t) = \varepsilon L \sin \frac{\pi x}{2L} \cos \sqrt{\frac{E}{\rho}} \frac{\pi t}{2L}; \quad u(x, t) = \frac{\partial w(x, t)}{\partial t}; \quad \sigma(x, t) = E \frac{\partial w(x, t)}{\partial x}.$$

5.2. Взаимодействие упругой волны с контактной границей. Важным качеством разностных схем является сохранение точности расчетов вблизи разрывов в решении или значениях параметров веществ [25]. В связи с этим рассмотрим задачу о взаимодействии гармонической волны с контактной границей, разделяющей счетную область $0 < x < 2L$ на две половины. В разд. 6.2 представлены результаты расчетов для веществ с сильно различающимися значениями акустической жесткости: $\frac{\rho_1 c_1}{\rho_2 c_2} \approx 3$ (Fe и Be с модулем Юнга E_1 и E_2 соответственно).

Начальные условия:

$$w(x, t = 0) = \varepsilon \cdot \begin{cases} \sin \frac{8\pi x}{L}, & 0 < x < L; \\ 0, & L < x < 2L; \end{cases} \quad \rho = \begin{cases} \rho_1, & 0 < x < L; \\ \rho_2, & L < x < 2L; \end{cases} \quad \varepsilon = 0,001;$$

$$\sigma(x, t = 0) = E \frac{\partial w(x, t = 0)}{\partial t} = \begin{cases} 8\pi E_1 \varepsilon \cos \frac{8\pi x}{L}, & 0 < x < L; \\ 0, & L < x < 2L; \end{cases} \quad E = \begin{cases} E_1, & 0 < x < L; \\ E_2, & L < x < 2L; \end{cases}$$

$$u(x, t = 0) = 0.$$

Граничные условия:

$$u(x = 0, t) = 0, \quad u(x = 2L, t) = 0.$$

Эталонное решение получено суперпозицией падающей, отраженной и преломленной волн [25]:

1) $x' = x - L < -c_1 t$:

$$\sigma(x, t) = 0,5 w_0 \rho_1 c_1 \omega \left[\cos \frac{\omega(x' + c_1 t)}{c_1} + \cos \frac{\omega(x' - c_1 t)}{c_1} \right];$$

$$u(x, t) = 0,5 w_0 \omega \left[\cos \frac{\omega(x' + c_1 t)}{c_1} - \cos \frac{\omega(x' - c_1 t)}{c_1} \right];$$

2) $-c_1 t < x' < 0$:

$$\sigma(x, t) = 0,5 w_0 \rho_1 c_1 \omega \left[\cos \frac{\omega(x' + c_1 t)}{c_1} + \cos \frac{\omega(x' - c_1 t)}{c_1} \right] + w_0 \frac{(\rho_1 c_1)^2}{\rho_1 c_1 + \rho_2 c_2} \cos \frac{\omega(x' + c_1 t)}{c_1};$$

$$u(x, t) = 0,5 w_0 \omega \left[\cos \frac{\omega(x' + c_1 t)}{c_1} - \cos \frac{\omega(x' - c_1 t)}{c_1} \right] - w_0 \frac{\rho_1 c_1}{\rho_1 c_1 + \rho_2 c_2} \cos \frac{\omega(x' + c_1 t)}{c_1};$$

3) $0 < x' < c_2 t$:

$$\sigma(x, t) = -w_0 \frac{\rho_1 c_1 \rho_2 c_2}{\rho_1 c_1 + \rho_2 c_2} \cos \frac{\omega(x' - c_2 t)}{c_2};$$

$$u(x, t) = -w_0 \frac{\rho_1 c_1}{\rho_1 c_1 + \rho_2 c_2} \cos \frac{\omega(x' - c_2 t)}{c_2}; \quad c_2 = \sqrt{\frac{E_2}{\rho_2}};$$

4) $c_2 t < x'$:

$$\sigma(x, t) = 0; \quad u(x, t) = 0.$$

Здесь всюду $\omega = \frac{8\pi c_1}{L}$, $c_1 = \sqrt{\frac{E_1}{\rho_1}}$.

5.3. Сферически-симметричная расходящаяся упругая волна. В однородной упругой среде с параметрами Ламэ λ , μ на сфере радиусом r^0 задан источник $w|_{r=r^0} = f(t)$. Рассматривается распространение расходящейся упругой волны [20].

Запишем уравнения динамической теории упругости при выполнении условий сферической симметрии:

$$\rho^0 \frac{\partial^2 w}{\partial t^2} = -\frac{\partial P_r}{\partial r} - 2\frac{P_r - P_\theta}{r}, \quad P_r = -\lambda \left(\frac{\partial w}{\partial r} + 2\frac{w}{r} \right) - 2\mu \frac{\partial w}{\partial r}, \quad P_\theta = -\lambda \left(\frac{\partial w}{\partial r} + 2\frac{w}{r} \right) - 2\mu \frac{w}{r}.$$

Исключив напряжения, получим волновое уравнение для потенциала

$$\frac{\partial^2 \Phi}{\partial t^2} = \frac{1}{C_l} \frac{\partial^2 \Phi}{\partial r^2},$$

где C_l — скорость продольной волны.

Общее решение для волны, распространяющейся от центра, запишем в виде

$$\Phi(r, t) = \Phi \left(t - \frac{r - r^0}{C_l} \right); \quad w = \frac{\partial}{\partial r} \left(\frac{1}{r} \Phi \left(t - \frac{r - r^0}{C_l} \right) \right); \quad C_l = \sqrt{\frac{\lambda + 2\mu}{\rho_0}}.$$

Пусть $\tau = t - (r - r^0)/C_l$. С учетом граничного условия решение для потенциала принимает вид

$$\Phi(\tau) = -r^0 C_l \int_0^\tau \exp \left(-\frac{C_l(\tau - \xi)}{r^0} \right) f(\xi) d\xi. \quad (10)$$

Пусть на сфере радиусом r^0 задан гармонический источник (w_0 , T — амплитуда и период колебаний)

$$f(t) = w_0 (1 - \cos \omega t), \quad \omega = \frac{2\pi}{T}.$$

Интегрирование (10) при выбранном источнике дает следующую формулу для вычисления потенциала:

$$\Phi(\tau) = \begin{cases} -\frac{r^{02} w_0}{1 + \alpha^2} \left(1 + \alpha^2 - \alpha^2 \exp \left(-\frac{C_l \tau}{r^0} \right) - \cos \omega \tau - \alpha \sin \omega \tau \right), & \tau \geq 0; \\ 0, & \tau < 0; \end{cases} \quad \alpha = \frac{\omega r^0}{C_l}.$$

Выпишем также значения производных:

$$\begin{aligned} \frac{d\Phi(\tau)}{d\tau} &= \begin{cases} -\frac{r^0 w_0 C_l}{1 + \alpha^2} \left(\alpha^2 \exp \left(-\frac{C_l \tau}{r^0} \right) + \alpha \sin \omega \tau - \alpha^2 \cos \omega \tau \right), & \tau \geq 0; \\ 0, & \tau < 0; \end{cases} \\ \frac{d^2\Phi(\tau)}{d\tau^2} &= \begin{cases} -\frac{w_0 C_l^2}{1 + \alpha^2} \left(-\alpha^2 \exp \left(-\frac{C_l \tau}{r^0} \right) + \alpha^2 \cos \omega \tau + \alpha^3 \sin \omega \tau \right), & \tau \geq 0; \\ 0, & \tau < 0; \end{cases} \\ \frac{d^3\Phi(\tau)}{d\tau^3} &= \begin{cases} -\frac{w_0 C_l^3}{(1 + \alpha^2) r^0} \left(\alpha^2 \exp \left(-\frac{C_l \tau}{r^0} \right) - \alpha^3 \sin \omega \tau + \alpha^4 \cos \omega \tau \right), & \tau \geq 0; \\ 0, & \tau < 0. \end{cases} \end{aligned}$$

Решение поставленной задачи имеет вид

$$w(r, t) = -\frac{1}{r^2} \left(\Phi(\tau) + \frac{r}{C_l} \frac{d\Phi}{d\tau} \right); \quad \frac{\partial w(r, t)}{\partial r} = \frac{1}{r^3} \left(2\Phi(\tau) + \frac{2r}{C_l} \frac{d\Phi}{d\tau} + \frac{r^2}{C_l^2} \frac{d^2\Phi}{d\tau^2} \right);$$

$$u(r, t) = \frac{\partial w(r, t)}{\partial t} = -\frac{1}{r^2} \left(\frac{d\Phi}{d\tau} + \frac{r}{C_l} \frac{d^2\Phi}{d\tau^2} \right); \quad \frac{\partial u(r, t)}{\partial t} = \frac{\partial^2 w(r, t)}{\partial t^2} = -\frac{1}{r^2} \left(\frac{d^2\Phi}{d\tau^2} + \frac{r}{C_l} \frac{d^3\Phi}{d\tau^3} \right);$$

$$\sigma_{rr}(r, t) = -\rho^0 C_l^2 \left(\frac{\partial w(r, t)}{\partial r} + \frac{w(r, t)}{r} \frac{2\nu}{1-\nu} \right); \quad \sigma_{\theta\theta}(r, t) = \sigma_{\varphi\varphi} = -\frac{\rho^0 C_l^2}{1-\nu} \left(\nu \frac{\partial w(r, t)}{\partial r} + \frac{w(r, t)}{r} \right).$$

Тестовые расчеты этой задачи проведены в декартовой системе координат, поэтому рассчитанные параметры для сравнения с эталоном были переведены в сферическую систему координат.

6. Расчеты тестовых задач

Расчеты тестовых задач из разд. 5 проведены с использованием консервативного варианта разностной схемы *крест* [26] и метода, основанного на дискретной гамильтоновой механике.

Малыми параметрами в этих расчетах являются размер счетной ячейки $\sim L/N$, а также объем, приходящийся на одну частицу, и соответствующий этому объему размер частицы $\sim L/N$. Здесь N — число ячеек либо число частиц вдоль одного из направлений.

Во всех расчетах выбрана следующая система единиц измерения: г, см, 10^{-5} с.

6.1. Прохождение гармонических волн вдоль тонкого стержня. Геометрия задачи:

$$0 < x < 2L; \quad -h < y < h; \quad -h < z < h; \quad L = 1\text{см}; \quad h = L/N; \quad N = 100.$$

Свойства материала: модуль Юнга $E = 200$, коэффициент Пуассона $\nu = 0$, плотность $\rho = 7,85$. Постановка расчетов для разрывных и гладких начальных данных приведена в разд. 5.1. В этой задаче представляет интерес исследование амплитудной и фазовой ошибок разностных схем на достаточно большом временном интервале. Конечное время счета $t_k = 250 t_0$, $t_0 = L/C_E$, соответствует $t_k C_E / (4L) \approx 60$ волновым периодам.

Результаты в интервале времени $240 < t/t_0 < 250$ вблизи конечного момента представлены на рис. 2 для двух вариантов расчетов: с использованием разностной схемы *крест* (МСС-расчетов) и дискретного метода (ДМ). Результаты соответствуют задаче с разрывными начальными данными.

Точность первого варианта расчетов определяется аппроксимационной и искусственной вязкостью. Во втором варианте счетная диссипация отсутствует. Волновой профиль в этих расчетах не

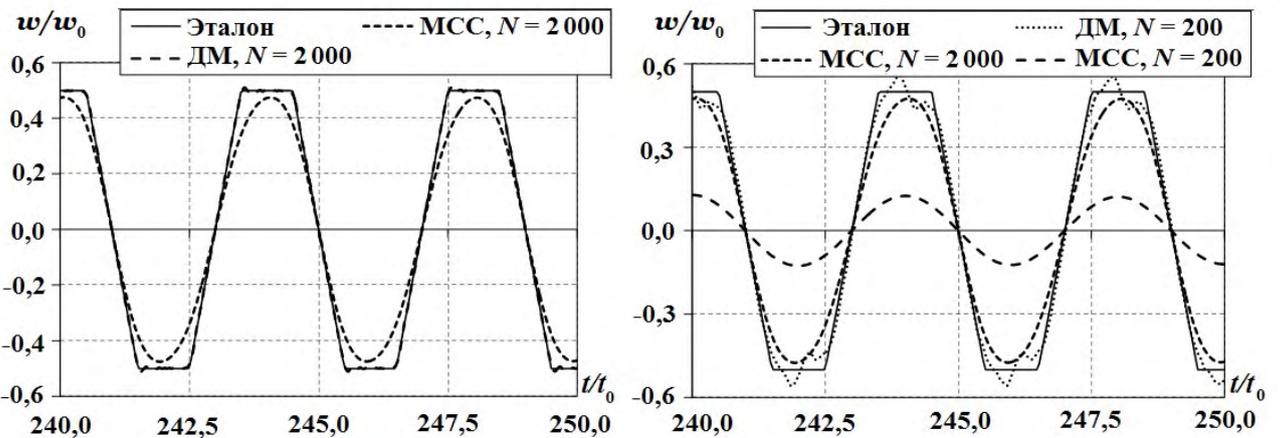


Рис. 2. Зависимости смещения от времени в сечении $x = L/2$, полученные в МСС-расчетах и расчетах с использованием ДМ с числом ячеек $N = 2000$ и 200

искажается в течение времени, равного 60 волновым периодам. Диаграмма смещения совпадает с эталоном с точностью до графического изображения.

Расхождение по точности между двумя вариантами расчетов еще более усиливается при уменьшении числа счетных ячеек и соответственно числа частиц до $N = 200$ (см. рис. 2, справа). В расчетах с ДМ приемлемая точность сохраняется и при таком уменьшении числа частиц, в то время как в МСС-расчетах амплитуда волны на конечный момент времени уменьшилась по сравнению с эталоном в 5 раз. Сравнительный анализ показывает, что результаты одинаковой точности на момент $t_k = 250t_0$ могут быть получены при использовании $N = 2000$ ячеек сетки в МСС-расчетах либо при использовании $N = 200$ частиц в ДМ. При этом следует учесть, что объем вычислительной работы в обоих случаях пропорционален N^2 .

В задачах с гладкими начальными данными преимущества методов гамильтоновой механики становятся еще более очевидными. В таких расчетах звуковые возмущения сохраняются без искажения на сколь угодно больших пространственных (либо временных) интервалах, если на длину волны приходится достаточное ($\gtrsim 20$) число частиц. В расчетах с использованием разностных схем МСС звуковые возмущения (как гладкие, так и негладкие) быстро затухают, и это обстоятельство не позволяет проводить расчеты на больших пространственных (либо временных) интервалах. Для устранения этого недостатка при построении разностных схем требуется учитывать характеристические свойства дифференциальных уравнений.

6.2. Взаимодействие гармонической волны с контактной границей. Геометрия задачи:

$$0 < x < 2L; \quad -h < y < h; \quad -h < z < h; \quad L = 1; \quad h = L/N; \quad N = 100.$$

Счетная область $0 < x < 2L$ разделена контактной границей на две половины. Выбраны вещества (Fe и Be) общей массой M с сильно различающимися значениями акустической жесткости: $\frac{\rho_1 c_1}{\rho_2 c_2} \approx 3$ (таблица).

Параметры веществ в задаче о взаимодействии гармонической волны с контактной границей

Fe	$0 < x < L$	$\rho_1 = 7,85$	$\nu_1 = 0$	$E_1 = 200$	$c_1 = \sqrt{E_1/\rho_1} = 5,0475$
Be	$L < x < 2L$	$\rho_2 = 1,85$	$\nu_2 = 0$	$E_2 = 106,13$	$c_2 = \sqrt{E_2/\rho_2} = 7,5741$

Начальные условия:

$$w(x, t = 0) = \varepsilon \cdot \begin{cases} \sin \frac{8\pi x}{L}, & 0 < x < L; \\ 0, & L < x < 2L; \end{cases} \quad \varepsilon = 0,001;$$

$$\rho = \begin{cases} \rho_1, & 0 < x < L; \\ \rho_2, & L < x < 2L; \end{cases} \quad u(x, t = 0) = 0.$$

Граничные условия:

$$u(x = 0, t) = 0; \quad u(x = 2L, t) = 0.$$

Профиль нормального напряжения в момент времени $t = 0,1$ представлен на рис. 3. Расчет с хорошей точностью воспроизводит амплитуды и длины отраженной и преломленной волн. Вместе с тем в ДМ отсутствует счетная диссипация, что приводит к появлению слабой немонотонности вблизи разрывов.

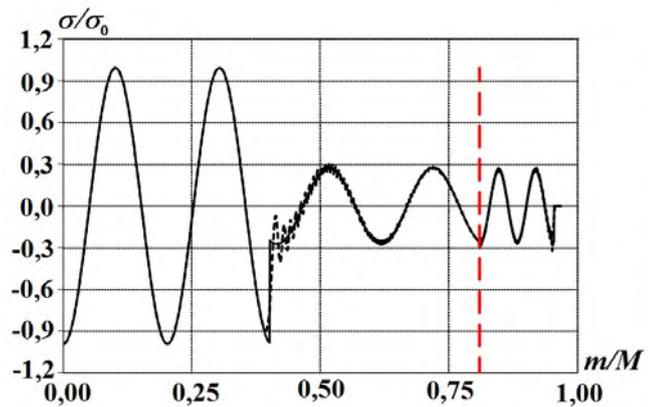


Рис. 3. Профили нормального напряжения в задаче о взаимодействии гармонической волны с контактной границей Fe–Be, $N = 2000$: - - - - расчет ДМ; — — — — эталонное решение; — — — — положение контактной границы

6.3. Сферически-симметричная расходящаяся упругая волна. Счетная область: $r^0 < |x| < r_1$, $r^0 < |y| < r_1$, $r^0 < |z| < r_1$, $r^0 = 1$, $r_1 = 40$. Граничные условия задаются на гранях внутреннего куба: $(|x| = r^0; |y| = r^0; |z| = r^0)$:

$$w_x(x, y, z, t) = \frac{x}{r}w(r, t); \quad w_y(x, y, z, t) = \frac{y}{r}w(r, t); \quad w_z(x, y, z, t) = \frac{z}{r}w(r, t).$$

Параметры источника: $T = 1$; $\omega = 2\pi$; $w_0 = 0,01$. Конечное время счета $t = 4$.

Профили радиальных компонент смещения и скорости приведены на рис. 4. В качестве амплитуды скорости выбрана величина, связанная с параметрами источника: $u_0 = w_0\omega$. В силу сферической симметрии все параметры течения зависят от одной пространственной координаты $r = \sqrt{x^2 + y^2 + z^2}$, поэтому на приведенных графиках нанесены значения сеточных функций для всех частиц.

Растровые картины в нелинейной шкале (шпирен-визуализация) изображены на рис. 5, 6. Оттенки серого цвета в шпирен-визуализации соответствуют величине $S = 0,8 \exp(-5|\vec{u}|/|\vec{u}|_{\max})$.

Отметим хорошее сохранение сферической симметрии в представленном расчете.

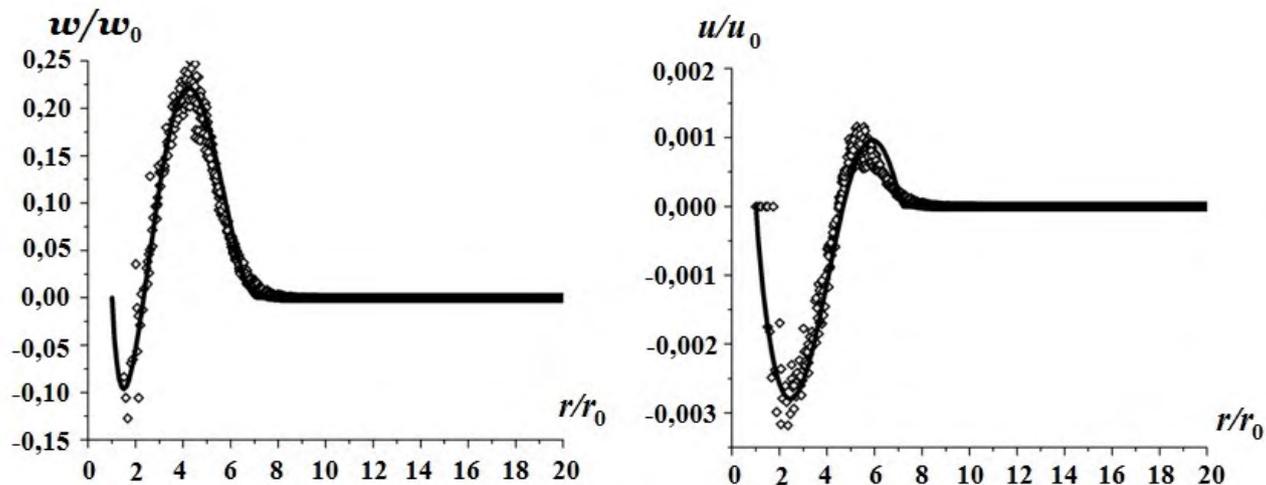
Заключение

Достоинствами метода, основанного на дискретной гамильтоновой динамике, являются:

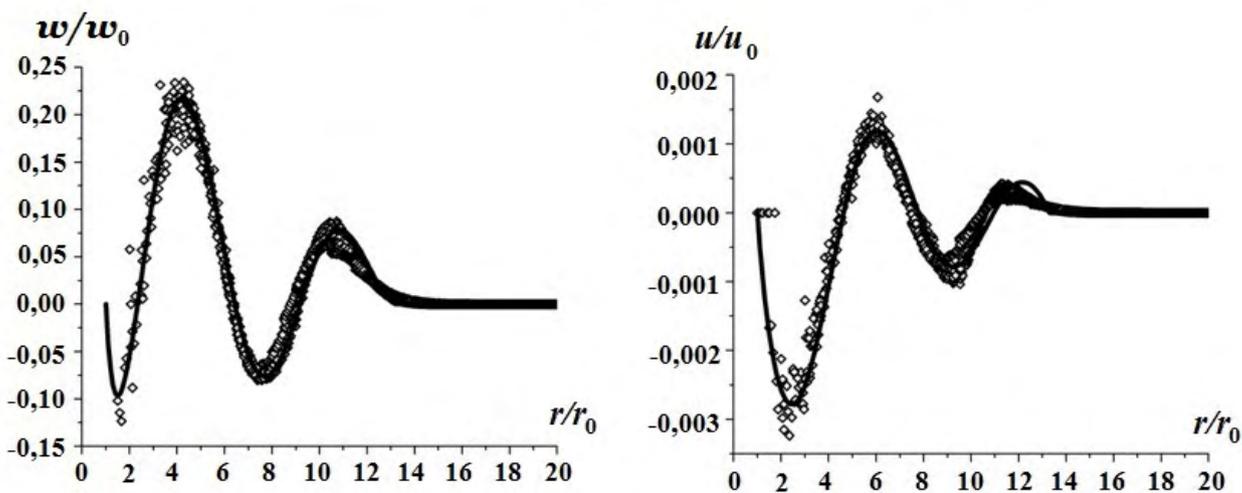
- отсутствие неопределенностей, возникающих при решении задач, содержащих разрывы и нарушение сплошности;
- отсутствие проблем, связанных с генерацией высококачественных расчетных сеток, так как метод является бессеточным;
- отсутствие этапов, связанных с корректировкой и перестроением расчетных сеток в результате сильных деформаций и изменения геометрии;
- наличие независимых арифметических вычислений, позволяющих рассчитывать на высокую эффективность использования GPU.

Список литературы

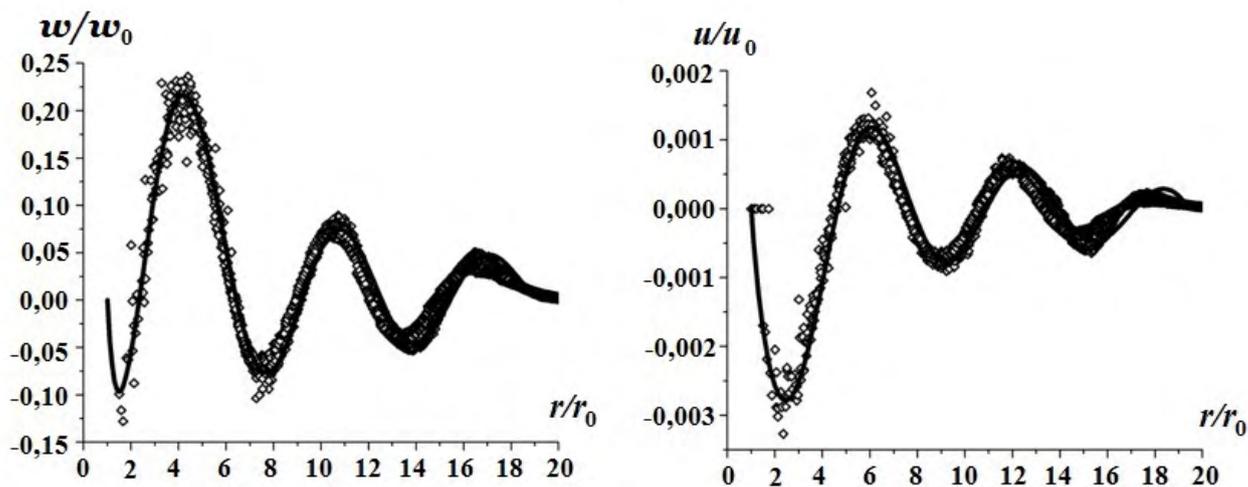
1. *Gingold R. A., Monaghan J. J.* Kernel estimates as a basis for general particle methods in hydrodynamics // J. Comp. Phys. 1982. Vol. 82. P. 1–15.
2. *Silling S. A., Epton M., Weckner O., Xu J., Ascory E.* Peridynamic states and constitutive modeling // J. Elasticity. 2007. Vol. 88. P. 151–184.
3. *Пискунов В. Н., Давыдов И. А.* Метод кластерной динамики и его применение в задачах импульсного нагружения твердых тел // Деформация и разрушение материалов. 2008. № 1. С. 3–9.
Piskunov V. N., Davydov I. A. Metod klasternoy dinamiki i ego primenenie v zadachakh impulsnogo nagruzheniya tverdykh tel // Deformatsiya i razrushenie materialov. 2008. № 1. S. 3–9.
4. *Псахье С. Г., Остермайер Г. П., Дмитриев А. И., Шилько Е. В., Смолин А. Ю., Коростелёв С. Ю.* Метод подвижных клеточных автоматов как новое направление дискретной вычислительной механики. I. Теоретическое описание // Физическая мезомеханика. 2000. Т. 3, № 2. С. 5–13.
Psakhye S. G., Ostermayer G. P., Dmitriev A. I., Shilko E. V., Smolin A. Yu., Korostelyev S. Yu. Metod podvizhnykh kletochnykh avtomatov kak novoe napravlenie diskretnoy vychislitel'noy mekhaniki. I. Teoreticheskoe opisanie // Fizicheskaya mezomekhanika. 2000. T. 3, № 2. S. 5–13.
5. *Надарейшвили А. И., Петушков В. А.* Моделирование нелинейной динамики гамильтоновых систем в биомеханике с использованием образов компьютерной томографии // Математическое моделирование. 2014. Т. 26, № 1. С. 109–121.
Nadareyshvili A. I., Petushkov V. A. Modelirovanie nelineynoy dinamiki gamiltonovykh sistem v biomekhanike s ispolzovaniem obrazov kompyuternoy tomografii // Matematicheskoe modelirovanie. 2014. T. 26, № 1. S. 109–121.



a



б



в

Рис. 4. Профили смещения (слева) и скорости (справа): а — $t = 1$; б — $t = 2$; в — $t = 3$; \diamond — расчет ДМ; — — эталонное решение

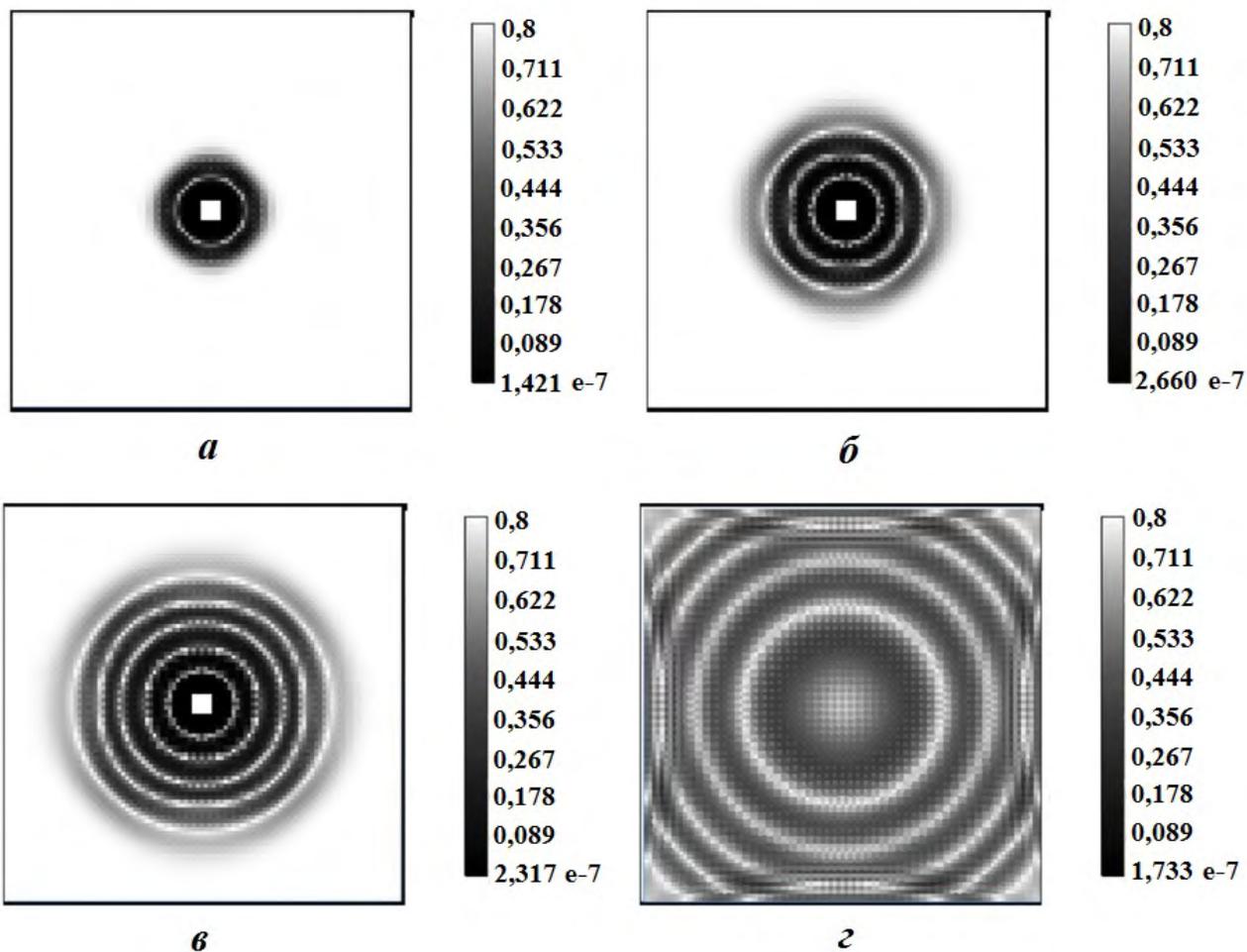


Рис. 5. Растровая картина модуля скорости (шпирен-визуализация в плоскости $z = 0$) в задаче со сферически-симметричной расходящейся упругой волной, $N = 1030274$: $a - t = 1$; $б - t = 2$; $в - t = 3$; $г - t = 4$

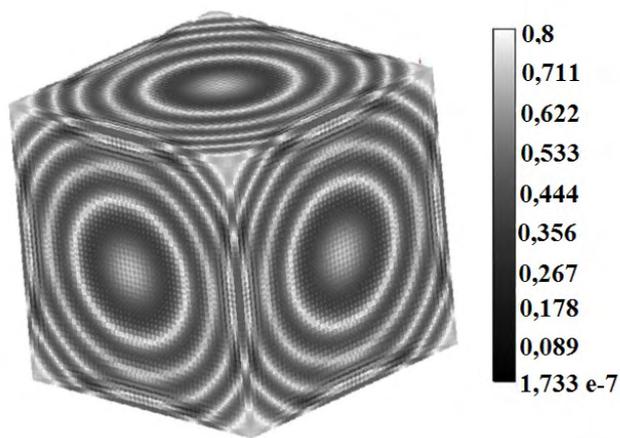


Рис. 6. Растровая картина модуля скорости (шпирен-визуализация на внешней поверхности куба) в задаче со сферически-симметричной расходящейся упругой волной, $t = 4$

6. Франкель Д., Смит Б. Принципы компьютерного моделирования молекулярных систем. М.: Научный мир, 2013.
Frankel D., Smit B. Printsipy kompyuternogo modelirovaniya molekulyarnykh sistem. M.: Nauchny mir, 2013.
7. 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., Dyemin D. A., Dyemina M. A., Zdorova M. V., Vetchinnikov M. V., Ericheva N. S., Kovalenko N. O., Kryuchkov I. A., Kechin A. G., Degtyaryev 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.
8. Ландау Л. Д., Лифшиц Е. М. Теория упругости. Теоретическая физика. Т. VII. М.: Наука, 1987.
Landau L. D., Lifshits E. M. Teoriya uprugosti. Teoreticheskaya fizika. T. VII. M.: Nauka, 1987.
9. Куликовский А. Г., Свешникова Е. И. Нелинейные волны в упругих средах. М.: Московский лицей, 1998.
Kulikovskiy A. G., Sveshnikova E. I. Nelineynye volny v uprugikh sredakh. M.: Moskovskiy litsey, 1998.
10. Блэнд Д. Р. Нелинейная динамическая теория упругости. М.: Мир, 1972.
Blend D. R. Nelineynaya dinamicheskaya teoriya uprugosti. M.: Mir, 1972.
11. Седов Л. И. Механика сплошной среды. М.: Наука, 1994.
Sedov L. I. Mekhanika sploshnoy sredy. M.: Nauka, 1994.
12. Арнольд В. И. Математические методы классической механики. М.: Наука, 1979.
Arnold V. I. Matematicheskie metody klassicheskoy mekhaniki. M.: Nauka, 1979.
13. Самарский А. А., Попов Ю. А. Разностные методы решения задач газовой динамики. М.: Наука, 1992.
Samarskiy A. A., Popov Yu. A. Raznostnye metody resheniya zadach gazovoy dinamiki. M.: Nauka, 1992.
14. Сурис Ю. Б. О сохранении симплектической структуры при численном решении гамильтоновых систем // Численное решение обыкновенных дифференциальных уравнений. М.: ИПМ АН СССР, 1988. С. 148—160.
Suris Yu. B. O sokhranenii simplekticheskoy struktury pri chislennom reshenii gamiltonovykh sistem // Chislennoe reshenie obyknovennykh differentsialnykh uravneniy. M.: IPM AN SSSR, 1988. S. 148—160.
15. Sanz-Serna J. M. Runge—Kutta schemes for Hamiltonian system // BIT. 1988. Vol. 22, No 4. P. 877—883.
16. Еленин Г. Г., Шляхов П. И. Геометрическая структура пространства параметров трехстадийных симплектических методов Рунге—Кутты // Математическое моделирование. 2011. Т. 23, № 5. С. 16—34.
Elenin G. G., Shlyakhov P. I. Geometricheskaya struktura prostranstva parametrov trekhstadiynykh simplekticheskikh metodov Runge—Kutty // Matematicheskoe modelirovanie. 2011. T. 23, № 5. S. 16—34.
17. Verlet L. Computer "experiments" on classical fluids. Thermodynamical properties of Lennard—Jones molecules // Phys. Rev. 1967. Vol. 159, No 1. P. 98—103.

18. *Софронов В. Н., Шемарулин В. Е.* Классификация явных трехстадийных симплектических разностных схем численного решения натуральных гамильтоновых систем. Сравнительное исследование схем высокого порядка на задачах молекулярной динамики // Журнал вычисл. мат. и мат. физ. 2016. Т. 56, № 45. С. 551–571.
Sofronov V. N., Shemarulin V. E. Klassifikatsiya yavnykh trekhstadiynykh simplekticheskikh raznostnykh skhem chislennoy resheniya naturalnykh gamiltonovykh sistem. Sravnitelnoye issledovanie skhem vysokogo poryadka na zadachakh molekulyarnoy dinamiki // Zhurnal vychisl. mat. i mat. fiz. 2016. T. 56, № 45. S. 551–571.
19. *Shemarulin V. E., Sofronov V. N.* Three — stage symplectic difference schemes for molecular dynamics. Description, classification, analytical and numerical investigation // Proc. of the 10th Seminar on New Models and Hydro-codes for Shock Wave Processes on Condensed Matter. Pardubice, Czech Republic. July 27 — August 1, 2014. P. 125–137.
20. *Sofronov V. N., Shemarulin V. E., Vetchinnikov M. V.* High-order symplectic difference schemes for numerical solution of natural Hamiltonian system. Applications to molecular dynamics and continuum mechanics problems // Horizons in World Physics. 2019. Vol. 301. P. 1–79.
21. *Candy J., Rozmus W.* A symplectic integration algorithm for separable Hamiltonian function // J. Comp. Phys. 1991. Vol. 92. P. 230–256.
22. *Софронов В. Н., Мокина К. С., Шемарулин В. Е.* Разностные схемы молекулярной динамики. 1. Сравнительный анализ точности, устойчивости и экономичности // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2011. Вып. 2. С. 18–32.
Sofronov V. N., Mokina K. S., Shemarulin V. E. Raznostnye skhemy molekulyarnoy gazovoy dinamiki. 1. Sravnitelny analiz tochnosti, ustoychivosti i ekonomichnosti // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2011. Vyp. 2. S. 18–32.
23. *Silling S. A., Lehoucq R. B.* Convergence of Peridynamics to Classical Elasticity Theory. Report SAND 2007-7242. Sandia National Laboratories, 2007.
24. *Madenci E., Oterkus E.* Peridynamics. Theory and its Applications. New York: Springer, 2014.
25. *Бондаренко Ю. А., Софронов В. Н.* Взаимодействие звука с границей разных веществ в разностных схемах одномерной лагранжевой газовой динамики. Зависимость точности счета от способов согласования сетки // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2015. Вып. 2. С. 12–27.
Bondarenko Yu. A., Sofronov V. N. Vzaimodeystvie zvuka s granitsej raznykh veshchestv v raznostnykh skhemakh odnomernoy lagranzhevoy gazovoy dinamiki. Zavisimost tochnosti scheta ot sposobov soglasovaniya setki // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2015. Vyp. 2. S. 12–27.
26. *Бондаренко Ю. А.* Консервативное расщепление уравнения энергии в разностных схемах типа "крест" для лагранжевой газодинамики // Журнал вычисл. мат. и мат. физ. 1997. Т. 37, № 8. С. 1020–1023.
Bondarenko Yu. A. Konservativnoye rasshcheplenie uravneniya energii v raznostnykh skhemakh tipa "krest" dlya lagranzhevoy gazodinamiki // Zhurnal vychisl. mat. i mat. fiz. 1997. T. 37, № 8. S. 1020–1023.

Статья поступила в редакцию 13.05.20.

USE OF HAMILTONIAN DYNAMICS METHODS IN COMPUTATIONAL CONTINUUM MECHANICS / V. N. Sofronov, M. V. Vetchinnikov, M. A. Dyemina (FSUE "RFNC-VNIIEF", Sarov, Nizhniy Novgorod region).

The paper presents a computational continuum mechanics method based on discrete Hamiltonian dynamics. Hamilton's equations are numerically solved using symplectic difference schemes. Examples of dissipation-free process simulations are provided.

Keywords: discrete Hamiltonian dynamics, phase volume, symplectic difference schemes, dynamic elasticity problems, MoDyS software package.

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ МНОГОФАЗНЫХ ТЕЧЕНИЙ В ЭЙЛЕРОВОМ ПРИБЛИЖЕНИИ НА ОСНОВЕ МЕТОДА SIMPLE

О. Г. Близнюк, О. Е. Власова, А. В. Гичук, А. С. Козелков, И. В. Лялюшкина
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Описаны особенности применения метода SIMPLE для численного моделирования течений многофазных сред с учетом межфазных взаимодействий. Используется многожидкостная модель, в которой каждая фаза описывается своими плотностью, скоростью и температурой. Для алгоритма, реализованного в пакете программ "Логос", приведены результаты расчета некоторых тестовых задач.

Ключевые слова: многофазные течения, численное моделирование, метод SIMPLE, пакет программ "Логос".

Введение

Физические процессы, сопровождающиеся изменением агрегатного состояния вещества, лежат в основе функционирования многих промышленных установок и бытовых приборов. По этой причине вопросы численного моделирования таких процессов особо актуальны, а востребованность программных продуктов для расчетов задач гидродинамики многофазных течений только растет.

Развитие математических моделей и методов расчета многофазных течений закономерно опирается на уже имеющиеся наработки в части гидродинамики однофазных течений. Действительно, запись основных уравнений относительно осредненных по объему фазовых величин позволяет получить модели, по виду соотношений и свойствам очень близкие к моделям однофазных течений [1–4]. В качестве алгоритма решения получающейся системы уравнений удобно использовать успешно применяемый при численном моделировании однофазных течений метод SIMPLE [5].

Практика расчетного моделирования многофазных течений показала, что при значительном изменении характера течения (пузырьковое, снарядное, кольцевое, капельное и др.) по пространству или времени, а также при наличии интенсивных межфазных обменов массой, импульсом и теплом подходы с осреднением фазовых величин (скоростей, температур) не могут обеспечить удовлетворительного решения. В таких случаях математическая модель должна формулироваться в многожидкостном приближении [1], когда каждая фаза имеет свои плотность, скорость и температуру. Система уравнений для каждой из фаз подобна однофазному случаю, т. е. включает в себя уравнения неразрывности, движения и энергии. Существенное отличие – правые части этих уравнений содержат члены, описывающие межфазные обмены. Расчет обменов по явной схеме, т. е. по известным с предыдущей итерации значениям, допускает использование обычного алгоритма SIMPLE, применяемого в расчетах однофазных течений. Но в случаях, когда межфазные обмены имеют определяющее значение в моделируемом процессе, для получения правильного решения потребуется учесть их в неявном алгоритме.

В настоящей работе описана модель многофазного течения в многожидкостном приближении, реализованная в пакете программ "Логос" [6, 7], отмечены особенности построения алгоритма на основе метода SIMPLE [5, 6].

Модель многофазных течений

Рассматриваемая модель многофазности основана на многожидкостном описании течения и опирается на концепцию многоскоростного континуума [1], где каждая из рассматриваемых фаз имеет свои плотность, скорость и температуру [1, 4, 8]. Данный подход также называется эйлеровым приближением [9, 10]. Соответственно для каждой фазы записываются уравнение неразрывности, уравнение движения, а также уравнение энергии. В правых частях этих уравнений учитываются межфазные обмены массой, импульсом и энергией.

Пусть рассматривается n -фазное течение. Фазами могут быть одно или несколько веществ в жидком или газообразном состоянии, а также дисперсные частицы (пузырьки или капли) разного размера одного вещества. Уравнения неразрывности для данного течения имеют вид

$$\frac{\partial}{\partial t} (\alpha_q \rho_q) + \operatorname{div} (\alpha_q \rho_q \mathbf{v}_q) = \sum_{p=1}^n (m_{pq} - m_{qp}). \quad (1)$$

Здесь t [с] — время; α_q — объемная концентрация фазы q , $\sum_{p=1}^n \alpha_p = 1$; ρ_q [кг/м³] — плотность фазы q ; $\mathbf{v}_q = (v_{qx}, v_{qy}, v_{qz}) = (u_q, v_q, w_q)$ [м/с] — скорость фазы q ; m_{pq} [кг/(м³·с)] — удельная интенсивность массообмена, идущего от фазы p к фазе q .

Уравнения движения для n -фазного течения запишутся как

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_q \rho_q \mathbf{v}_q) + \nabla (\alpha_q \rho_q \mathbf{v}_q \cdot \mathbf{v}_q) + \alpha_q \nabla P = \\ = \nabla (\alpha_q \tau_q) + \alpha_q \rho_q \mathbf{g} + \sum_{p=1}^n (\mathbf{R}_{pq} + m_{pq} \mathbf{v}_{pq} - m_{qp} \mathbf{v}_{qp}) + \mathbf{F}_q. \end{aligned} \quad (2)$$

Здесь P [Па] — давление; τ_q — тензор вязких напряжений для фазы q ; $\mathbf{g} = (g_1, g_2, g_3)$ [м/с²] — ускорение свободного падения; \mathbf{R}_{pq} [Н/м³] — объемная сила межфазного трения; \mathbf{F}_q [Н/м³] — прочие силы (виртуальной массы и пр.); \mathbf{v}_{pq} и \mathbf{v}_{qp} таковы, что

$$\mathbf{v}_{pq} = \begin{cases} \mathbf{v}_p, & m_{pq} \geq 0; \\ \mathbf{v}_q, & m_{pq} < 0; \end{cases} \quad \mathbf{v}_{qp} = \begin{cases} \mathbf{v}_q, & m_{qp} \geq 0; \\ \mathbf{v}_p, & m_{qp} < 0. \end{cases}$$

Уравнения энергии для многофазного течения запишем относительно энтальпии [1]:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_q \rho_q h_q) + \operatorname{div} (\alpha_q \rho_q \mathbf{v}_q h_q) - \alpha_q \frac{dP}{dt} = \\ = \nabla \cdot (\alpha_q \lambda_q \nabla T_q) + \alpha_q \tau_q : \nabla \mathbf{v}_q + \sum_{p=1}^n (Q_{pq} + m_{pq} h_{pq} - m_{qp} h_{qp}), \end{aligned} \quad (3)$$

где h_q [Дж/кг] — удельная энтальпия фазы q ; λ_q [Вт/(м·К)] — коэффициент теплопроводности фазы q ; T_q [К] — температура фазы q ; Q_{pq} [Вт/м³] — объемная мощность теплопереноса из фазы p в фазу q ; h_{pq} , h_{qp} [Дж/кг] — энтальпия межфазной поверхности, которая зависит от направления межфазного массообмена и для каждой фазы определяется как энтальпия насыщения при входящем массовом потоке и энтальпия фазы — при выходящем.

Для замыкания системы уравнений (1)–(3) требуется описать все межфазные взаимодействия (массообмен, теплообмен, межфазное трение) с использованием переменных, присутствующих в этих уравнениях, а также определить зависимость плотностей, энтальпий и их производных от температур и давления. Функция энтальпии в уравнении энергии линейризуется по температуре, затем, после учета всех замыкающих соотношений, получаем систему относительно независимых переменных $(P, \alpha_2, \dots, \alpha_n, \mathbf{v}_1, \dots, \mathbf{v}_n, T_1, \dots, T_n)$.

Межфазные взаимодействия

Для описания межфазных взаимодействий разработано множество моделей, опирающихся преимущественно на экспериментальные данные. В данном разделе приведены общие выражения [1, 11–15] без уточнения эмпирических констант.

Межфазный массообмен учитывается добавлением в уравнения неразрывности (1), движения (2) и энергии (3) удельной интенсивности массообмена m_{pq} . В простейшем случае может быть задана постоянная скорость r увеличения/уменьшения дисперсных частиц, тогда

$$m_{pq} = \max(0, \lambda_{pq}) - \max(0, -\lambda_{pq}),$$

где $\lambda_{pq} = r\alpha_p\rho_q$.

Межфазный теплообмен учитывается добавлением в правую часть уравнения энергии (3) для фазы q и вычитанием из правой части уравнения для фазы p потока тепла $Q_{pq} = -Q_{qp}$ от фазы p к фазе q , который находится по формулам

$$Q_{pq} = \kappa_{pq}a_i(T_p - T_q), \quad \kappa_{pq} = \frac{\lambda_q \text{Nu}_p}{d_p}, \quad a_i = \frac{6\alpha_p}{d_p},$$

где κ_{pq} [Вт/м²К] — коэффициент теплопередачи между фазами p и q ; a_i [м⁻¹] — удельная площадь межфазной поверхности; Nu_p — число Нуссельта фазы p ; d_p [м] — диаметр дисперсной частицы фазы p . Разные модели расчета межфазного теплообмена в основном отличаются формулой расчета числа Нуссельта, в простейшем случае оно может быть задано константой.

Межфазное трение учитывается добавлением в правую часть уравнения движения (2) для фазы q члена

$$\sum_{p=1}^n \mathbf{R}_{pq} = \sum_{p=1}^n K_{pq}(\mathbf{v}_p - \mathbf{v}_q).$$

Предполагается, что фаза q является преобладающей (называется несущей или первичной), а дисперсные частицы образованы вторичной фазой p . Здесь K_{pq} ($= K_{qp}$) — коэффициент межфазного обмена импульсом. В общем виде формула для этого коэффициента записывается как

$$K_{pq} = \frac{\rho_p f}{6\tau_p} d_p a_i.$$

Здесь f — функция трения; τ_p — время релаксации:

$$f = \frac{C_D \text{Re}}{24}; \quad \tau_p = \frac{\rho_p d_p^2}{18\mu_q},$$

где μ_q [Па · с] — динамическая вязкость несущей фазы q ; Re — относительное число Рейнольдса, для несущей фазы q и вторичной фазы p равное

$$\text{Re} = \text{Re}_p = \frac{\rho_q |\mathbf{v}_p - \mathbf{v}_q| d_p}{\mu_q};$$

C_D — коэффициент трения, в простейшем случае задаваемый константой. В известной модели трения Шиллера—Науманна [11, 16] коэффициент трения рассчитывается по формуле

$$C_D = \begin{cases} \frac{24}{\text{Re}} (1 + 0,15\text{Re}^{0,687}), & \text{Re} < 1000; \\ 0,44, & \text{Re} \geq 1000. \end{cases}$$

Подъемная сила действует на дисперсную частицу фазы p в несущей фазе q перпендикулярно направлению ее движения и может быть найдена по формуле

$$\mathbf{F}_{lift} = -C_l \rho_q \alpha_p (\mathbf{v}_q - \mathbf{v}_p) \times (\nabla \times \mathbf{v}_q),$$

где C_l – коэффициент подъемной силы, который добавляется в правые части уравнений движения для обеих фаз с противоположным знаком. Как и для остальных параметров, в простейшем случае этот коэффициент может быть задан константой.

Сила виртуальной массы возникает при ускорении дисперсной фазы относительно несущей и заключается в силе инерции, противодействующей ускорению части несущей фазы, увлекаемой дисперсной частицей. Она добавляется в правые части уравнений движения для обеих фаз с противоположным знаком и вычисляется по формуле

$$\mathbf{F}_{VM} = C_{VM}\alpha_p\rho_q \left[\frac{\partial \mathbf{v}_q}{\partial t} + (\mathbf{v}_q \cdot \nabla)\mathbf{v}_q - \left(\frac{\partial \mathbf{v}_p}{\partial t} + (\mathbf{v}_q \cdot \nabla)\mathbf{v}_p \right) \right],$$

где C_{VM} – коэффициент виртуальной массы.

Сила пристеночного скольжения действует на дисперсную фазу по направлению от стенки, для несущей фазы q и дисперсной фазы p она выражается формулой

$$\mathbf{F}_{wl} = C_{wl}\rho_q\alpha_p \left| (\mathbf{v}_q - \mathbf{v}_p)_{\parallel} \right|^2 \mathbf{n},$$

где C_{wl} – коэффициент пристеночного скольжения; $(\dots)_{\parallel}$ – тангенциальная компонента вектора; \mathbf{n} – нормаль к стенке. Подмодели расчета силы пристеночного скольжения отличаются друг от друга методом расчета коэффициента C_{wl} .

Алгоритм метода SIMPLE для многофазного случая

После дискретизации исходной системы (1)–(3) с помощью метода конечных объемов [17] получим систему линейных алгебраических уравнений. При неявном учете межфазных обменов матрица этой системы в части уравнений движения и энергии приобретает блочную структуру с размерами блока $n \times n$. Сформулируем алгоритм метода SIMPLE для системы линейных уравнений такого вида. Данный подход описан для ортогональных сеток в работе [18] и представляет собой расширение метода решения однофазных задач [5, 6, 19] с переходом от скалярных величин к векторам и матрицам.

Пусть известно решение на некотором слое по времени, ищем решение на следующем слое. Для этого организуется цикл по итерациям с индексом m . Обозначим символом N ячейку пространственной сетки, а F_N – грани этой ячейки. Рассмотрим уравнение движения, в котором новое значение скорости обозначим индексом m^* :

$$A_N \mathbf{v}_N^{m*} = \mathbf{Q}_v^{m-1} + \mathbf{B}_v - \sum_{F_N} A_{F_N} \mathbf{v}_{F_N}^{m*} - \sum_{F_N} \mathbf{P}_{F_N} P_{F_N}^{m-1}. \quad (4)$$

Здесь $\mathbf{v}_P = (v_{1x}, \dots, v_{nx}, v_{1y}, \dots, v_{ny}, v_{1z}, \dots, v_{nz})$ – вектор фазовых скоростей; A_N – матрица коэффициентов при скоростях в центре ячейки N , неявный учет межфазных обменов в общем случае делает ее недиагональной; \mathbf{Q}_v^{m-1} – вектор, образованный значениями величин с предыдущей, $(m-1)$ -й итерации; \mathbf{B}_v – вектор, образованный значениями величин с предыдущего слоя по времени; A_{F_N} – матрица коэффициентов при скоростях на гранях; \mathbf{P}_{F_N} – вектор коэффициентов при давлении с предыдущей итерации на гранях. Система уравнений (4) решается независимо по координатам, т. е. отдельно относительно векторов (v_{1x}, \dots, v_{nx}) , (v_{1y}, \dots, v_{ny}) и (v_{1z}, \dots, v_{nz}) , поскольку матрица A_N является блочно-диагональной и имеет вид

$$A_N = \begin{pmatrix} A_N^x & 0 & 0 \\ 0 & A_N^y & 0 \\ 0 & 0 & A_N^z \end{pmatrix}.$$

Скорости \mathbf{v}^{m*} , найденные после решения системы (4), вообще говоря, не удовлетворяют уравнению неразрывности, поэтому найдем такие новые значения скоростей

$$\mathbf{v}^m = \mathbf{v}^{m*} + \mathbf{v}' \quad (5)$$

и давления

$$P^m = P^{m-1} + P', \quad (6)$$

что, с одной стороны, будет выполняться уравнение неразрывности, а с другой стороны, не нарушится баланс импульсов в уравнении движения.

Для дальнейших построений используется сумма по фазам нормированных по средней плотности ρ_{rq} уравнений неразрывности (1):

$$\sum_{q=1}^n \frac{1}{\rho_{rq}} \left(\frac{\partial}{\partial t} (\alpha_q \rho_q) + \operatorname{div} (\alpha_q \rho_q \mathbf{v}_q) \right) = \sum_{q=1}^n \frac{1}{\rho_{rq}} \sum_{p=1}^n (m_{pq} - m_{qp}).$$

Это уравнение можно переписать в матричной форме относительно скоростей:

$$\sum_{F_N} C_{F_N} \mathbf{v}_{F_N}^m = \mathbf{Q}_\alpha^{m-1} + \mathbf{B}_\alpha, \quad (7)$$

где \mathbf{Q}_α^{m-1} — вектор, образованный значениями величин с предыдущей итерации $m-1$; \mathbf{B}_α — вектор, образованный значениями величин с предыдущего слоя по времени. При этом скорость и давление на итерации m таковы, что удовлетворено уравнение движения

$$A_N \mathbf{v}_N^m = \mathbf{Q}_v^{m-1} + \mathbf{B}_v - \sum_{F_N} A_{F_N} \mathbf{v}_{F_N}^{m*} - \sum_{F_N} \mathbf{P}_{F_N} P_{F_N}^m. \quad (8)$$

Конвективный член в правой части уравнения (8) для упрощения расчетов аппроксимирован с привлечением скоростей \mathbf{v}^{m*} , как это делается в традиционном варианте алгоритма SIMPLE.

Применим в алгоритме введенные в (5), (6) поправки скорости и давления. В результате из (8) и (4) получим соотношение

$$A_N \mathbf{v}'_N = - \sum_{F_N} \mathbf{P}_{F_N} P'_{F_N}.$$

Обратив теперь матрицы с размерами $n \times n$ в левой части (A_N образована тремя такими блоками), запишем

$$\mathbf{v}'_N = -A_N^{-1} \sum_{F_N} \mathbf{P}_{F_N} P'_{F_N}.$$

Обращенная матрица, очевидно, также является блочно-диагональной.

При интерполяции скоростей на грани применяется подход Ри—Чоу [20], позволяющий подавлять осцилляции решения при использовании схем с совмещенным шаблоном, когда скорости и давление определены в центрах ячеек. Для этого требуется скорректировать значения скалярных произведений скоростей на гранях F ячейки N и нормалей \mathbf{n} к этим граням:

$$\mathbf{v}_{\mathbf{n}F}^{m*} = \overline{(\mathbf{v}_{\mathbf{n}}^{m*})}_F - \overline{(\tilde{A}_N^{-1} \boldsymbol{\alpha})}_F \left((\nabla P \cdot \mathbf{n})_F^{m-1} - \overline{(\nabla P \cdot \mathbf{n})}_F^{m-1} \right); \quad (9)$$

$$\mathbf{v}'_{\mathbf{n}F} = - \overline{(\tilde{A}_N^{-1} \boldsymbol{\alpha})}_F (\nabla P' \cdot \mathbf{n})_F, \quad (10)$$

Здесь $\mathbf{v}_{\mathbf{n}F} = ((\mathbf{v}_{1F} \cdot \mathbf{n}), (\mathbf{v}_{2F} \cdot \mathbf{n}), \dots, (\mathbf{v}_{nF} \cdot \mathbf{n}))$; \tilde{A}_N — матрица, получаемая осреднением блоков A_N^x , A_N^y и A_N^z ; $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$. Черта сверху в формулах (9) и (10) означает интерполяцию соответствующих значений из центров ячеек с общей рассматриваемой гранью.

Алгоритм одной итерации метода SIMPLE для многофазного случая формулируется следующим образом:

1. Уравнения движения (4) решаются относительно скорости отдельно по координатам, но совместно по фазам.
2. Найденные скорости корректируются на гранях по формуле (9), затем с учетом (5) из (7) и (10) получается система для отыскания поправки давления, введенной в (6).

3. Градиент найденной поправки давления используется в (10) для корректировки потока.
4. Новые значения потока и давления P^m используются для решения уравнений энергии (3) относительно температур фаз, причем уравнения решаются согласованно по фазам, образуя одну систему.
5. Последовательно решаются уравнения неразрывности относительно объемных концентраций всех фаз, кроме несущей, концентрация которой находится из балансного соотношения (сумма концентраций всех фаз равна единице).

Описанные действия повторяются до сведения итераций по независимым переменным с требуемой точностью.

При числе фаз, равном единице, этот алгоритм сводится к обычному методу SIMPLE.

Тестовые расчеты

Алгоритм моделирования многофазных течений достаточно сложен, поэтому верификация реализованного программного кода должна быть многоэтапной. В силу того, что он допускает редукцию до однофазного алгоритма, первичная отладка включает проведение тестовых расчетов однофазных течений. Следующим этапом является введение в рассмотрение особенностей, присущих только многофазному случаю: межфазные взаимодействия, многоскоростной характер течения. Ниже приведены результаты расчетов модельных задач, описывающих изотермические многофазные течения с учетом межфазных обменов импульсом.

Задача 1. В движущийся с постоянной скоростью поток водяного пара впрыскиваются капли воды. Целью данной задачи является исследование ускорения потока водяных капель паром. Предполагается наличие капель двух размеров — 0,0005 и 0,005 м. Пар представляет собой несущую фазу, а капли — две дисперсные фазы (у каждой фазы свой размер капель). В соответствии с описанной выше математической моделью каждая из трех фаз имеет свою скорость и объемную концентрацию. Скорость входящего в счетную область пара полагается равной 5 м/с, а скорость вводимых капель — 1 м/с. Объемные концентрации дисперсных фаз на входе равны 1%. Свойства водяного пара: вязкость 10^{-5} кг/(м·с), плотность 0,5 кг/м³; свойства воды: вязкость 0,001 кг/(м·с), плотность 1000 кг/м³. Учитывается межфазное трение между несущей фазой (паром) и каждой из дисперсных фаз (каплями), коэффициент трения при этом полагается равным единице.

Из результатов расчета, показанных на рис. 1, видно, что скорости дисперсных фаз монотонно возрастают. Капли большего размера приобретают меньшую скорость, что соответствует используемой модели трения. Действительно, дисперсные фазы имеют одинаковую концентрацию, в результате у фазы с меньшим размером капель площадь межфазной поверхности будет больше, следовательно, больше и сила трения между этой фазой и несущей.

Дополнительно рассмотрена постановка аналогичной задачи с тремя дисперсными фазами: вода с диаметром капель 0,0005 м, вода с диаметром капель 0,005 м и нефть с диаметром капель 0,0005 м. Свойства нефти: вязкость 0,081 кг/(м·с), плотность 920 кг/м³. Концентрации всех дисперсных фаз равны 1%. Коэффициент межфазного трения между каждой из дисперсных фаз и несущей фазой равен единице.

Как видно из результатов, показанных на рис. 2, скорости дисперсных фаз монотонно возрастают, как и в предыдущем случае. Максимальная скорость нефтяной дисперсной фазы превышает скорость водяной фазы с каплями такого же размера на 1,5%. Это объясняется разностью плотностей нефти и воды на 8%. Простая оценка ускорения двух объектов с аналогичной разницей масс под действием одинаковой силы показывает, что разность их скоростей составила бы 3,6%. В используемой модели сила межфазного трения пропорциональна плотности дисперсной фазы ($\rho = \alpha_p \rho_p$) и квадрату разности скоростей несущей и дисперсной фаз, в силу чего перепад скоростей получается меньше приведенной оценки. Таким образом, найденное распределение скоростей дисперсных фаз демонстрирует правильные тенденции и физически непротиворечиво.

Задача 2. Целью данной задачи является изучение отстаивания двухфазной смеси в прямоугольном резервуаре высотой 1 м. Из-за действия сил гравитации более тяжелая фаза концентрируется

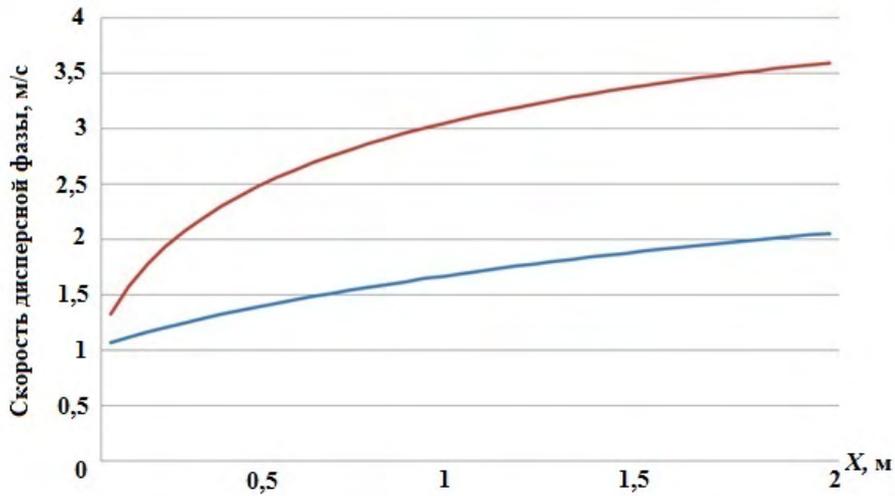


Рис. 1. Задача 1. Распределение скоростей двух дисперсных фаз вдоль оси X : — $d = 0,005$; — $d = 0,0005$

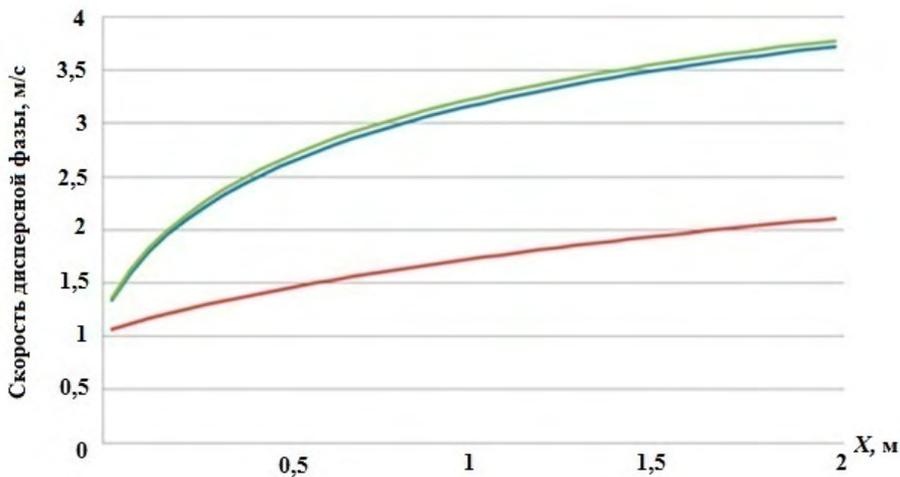


Рис. 2. Задача 1. Распределение скоростей трех дисперсных фаз вдоль оси X : — вода, $d = 0,0005$ м; — вода, $d = 0,005$ м; — нефть

у основания сосуда, и через некоторое время эта фаза полностью заполняет нижнюю часть резервуара, вытеснив легкую фазу наверх.

Смесь состоит из двух веществ — более тяжелой несущей фазы (воды) с плотностью 1000 кг/м^3 и вязкостью $0,001 \text{ кг/(м} \cdot \text{с)}$ и дисперсной фазы (воздуха) с плотностью 1 кг/м^3 и вязкостью $1,85 \times 10^{-5} \text{ кг/(м} \cdot \text{с)}$. Межфазное трение рассчитывается по модели Шиллера—Науманна.

Рассматривались три постановки задачи: 10 % воздуха и 90 % воды, 20 % воздуха и 80 % воды, 50 % воздуха и 50 % воды.

Результаты проведенных расчетов (рис. 3–5) показывают, что разделение фаз происходит в широком диапазоне начальных условий (концентрация легкой фазы от 10 до 50 %). Время разделения фаз возрастает с уменьшением содержания в исходной смеси тяжелой фазы. Положение границы раздела фаз точно соответствует их объемному соотношению, что в силу несжимаемости фаз свидетельствует о сохранении их масс.

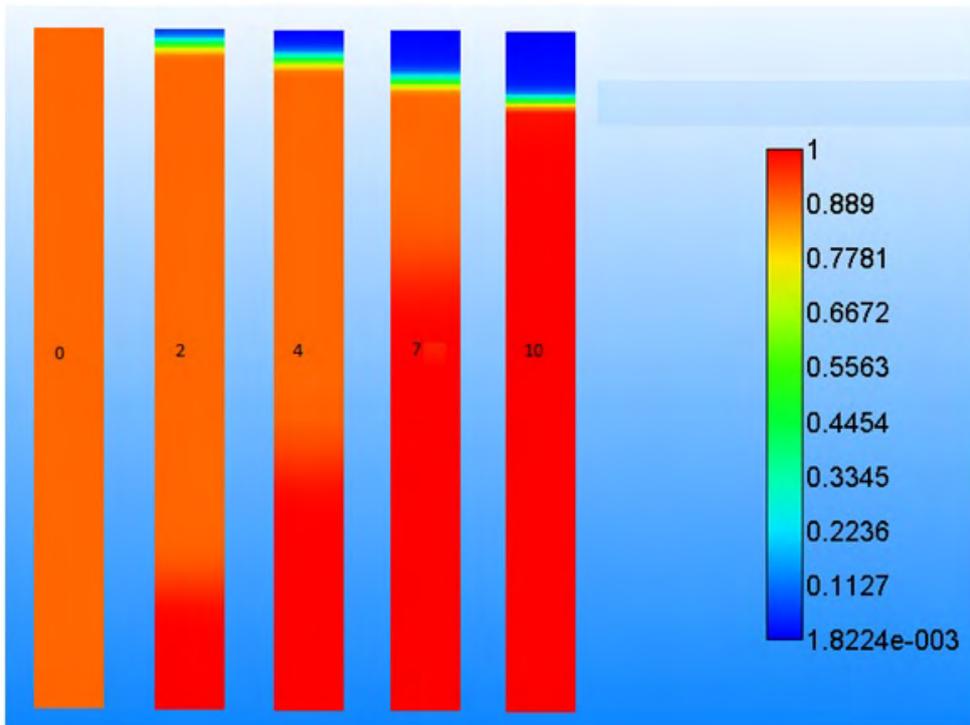


Рис. 3. Распределение объемной доли несущей фазы на разные моменты времени (0, 2, 4, 7 и 10 с) при расчете задачи 2 в первой постановке

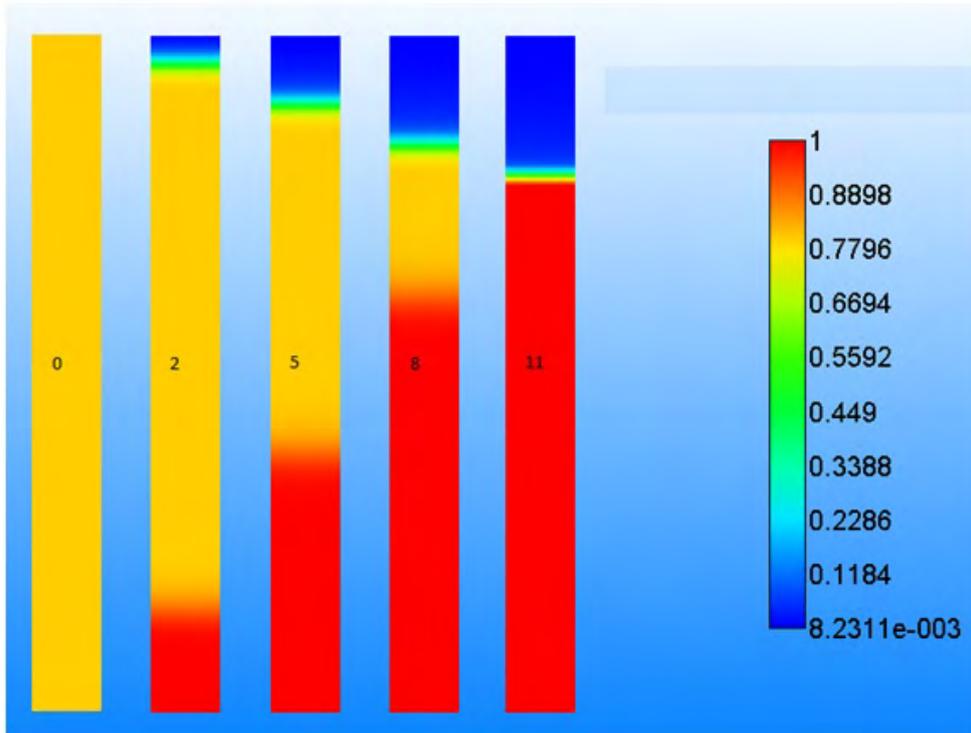


Рис. 4. Распределение объемной доли несущей фазы на разные моменты времени (0, 2, 5, 8 и 11 с) при расчете задачи 2 во второй постановке

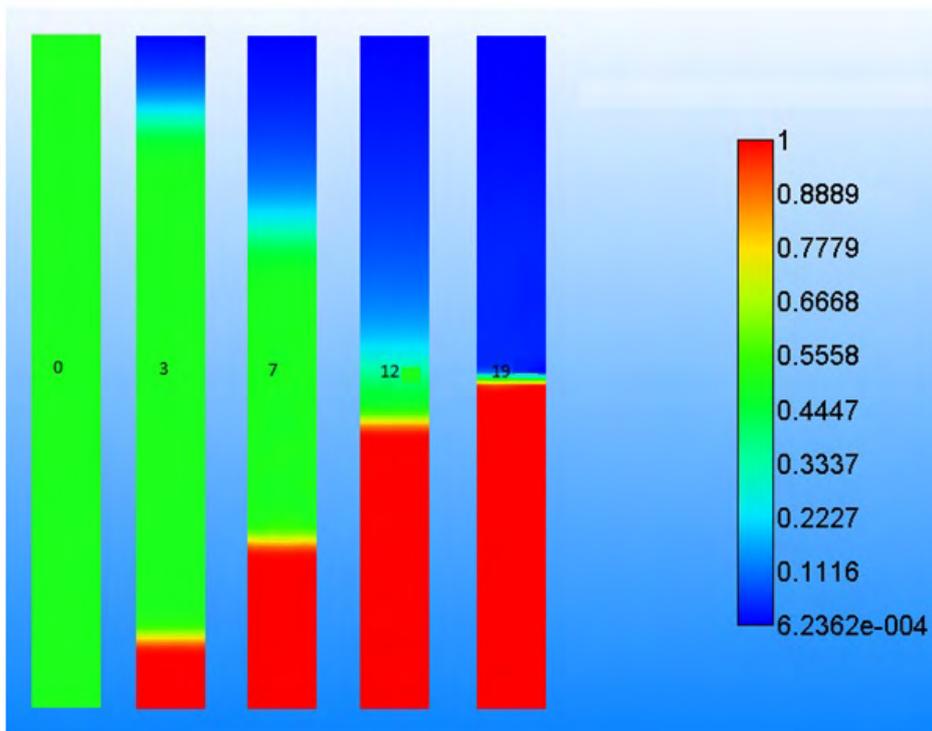


Рис. 5. Распределение объемной доли несущей фазы на разные моменты времени (0, 3, 7, 12 и 19 с) при расчете задачи 2 в третьей постановке

Заключение

В статье описана новая функциональная возможность пакета программ "Логос", предназначенная для решения сложных задач инженерного анализа, связанных с течениями многофазных сред с учетом межфазных взаимодействий. Лежащий в основе реализованной математической модели многожидкостный подход с описанием каждой фазы своими плотностями, скоростями и температурами ориентирован на моделирование интенсивных течений с динамично изменяющейся топологией межфазных границ и значительными межфазными обходами. Очевидно, что применимость соответствующего программного модуля должна быть подтверждена масштабными верификационными расчетами.

В данной работе рассмотрены несколько простых тестовых задач, описывающих изотермические многофазные течения. Полученные численные решения физически непротиворечивы, что можно считать первичным подтверждением применимости рассмотренного алгоритма. Дальнейшие работы должны быть направлены на поэтапное расширение верификационной базы многофазного модуля пакета программ "Логос", включающее усложнение геометрии задач, расширение диапазона свойств жидкости и газа, одновременный учет большего числа межфазных взаимодействий и пр. Это позволит перейти к задачам, описывающим процессы с известными экспериментальными данными и решению промышленно ориентированных задач.

Список литературы

1. *Нигматулин Р. И.* Динамика многофазных сред. Часть I. М.: Наука, 1987.
Nigmatulin R. I. Dinamika mnogofaznykh sred. Chast I. М.: Nauka, 1987.
2. *Бахрах С. М., Глаголева Ю. П., Самигулин М. С., Фролов В. Д., Яценко Н. Н., Янцлжкин Ю. В.* Расчет газодинамических течений на основе метода концентраций // Докл. АН СССР. 1981. Т. 257, № 3. С. 566–569.

- Bakhrakh S. M., Glagoleva Yu. P., Samigulin M. S., Frolov V. D., Yanenko N. N., Yanilkin Yu. V.* Raschet gazodinamicheskikh techeniy na osnove metoda kontsentratsiy. Dokl. AN SSSR. 1981. T. 257, № 3. S. 566–569.
3. *Hirt C. W., Nichols B. D.* Volume of fluid (VOF) method for the dynamics of free boundaries // J. Comp. Phys. 1981. Vol. 39. P. 201–226.
 4. *Podowski M. Z.* On the consistency of mechanistic multidimensional modeling of gas/liquid two-phase flows // Nucl. Eng. Design. 2009. Vol. 239, No 5. P. 933–940.
 5. *Patankar S. V., Spalding D. B.* A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows // Int. J. Heat Mass Transfer. 1972. Vol. 15. P. 1787–1806.
 6. *Козелков А. С., Курулин В. В., Тятюшкина Е. С., Пучкова О. Л., Лашкин С. В.* Реализация метода расчета вязкой несжимаемой жидкости с использованием многосеточного метода на основе алгоритма SIMPLE в пакете программ ЛОГОС // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 4. С. 44–56.
Kozelkov A. S., Kurulin V. V., Tyatyushkina E. S., Puchkova O. L., Lashkin S. V. Realizatsiya metoda rascheta вязкой neszhimaemoy zhidkosti s ispolzovaniem mnogosetochnoy metoda na osnove algoritma SIMPLE v pakete programm LOGOS // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoye modelirovaniye fizicheskikh protsessov. 2013. Vyp. 4. S. 44–56.
 7. *Betelin V. B., Shagaliev R. M., Aksenov S. V., Belyakov I. M., Deryugin Yu. N., Kozelkov A. S., Korchazhkin D. A., Nikitin V. F., Sarazov A. V., Zelenskiy D. K.* Mathematical simulation of hydrogen–oxygen combustion in rocket engines using LOGOS code // Acta Astronautica. 2014. Vol. 96. P. 53–64.
 8. *Podowski M. Z.* Toward Mechanistic Modeling of Boiling Heat Transfer // Nucl. Eng. Techn. 2012. Vol. 44, No 8. P. 889–896.
 9. *Самигулин М. С., Спиридонов В. Ф., Воронова О. А., Данилов Ю. Ф., Шкарубский В. В., Тарасова А. Н., Авдеев П. А., Артамонов М. В., Величко С. В.* Методика численного моделирования двумерных течений дисперсных сред на лагранжево-эйлеровых расчетных сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 4. С. 31–43.
Samigulin M. S., Spiridonov V. F., Voronova O. A., Danilov Yu. F., Shkarubskiy V. V., Tarasova A. N., Avdeev P. A., Artamonov M. V., Velichko S. V. Metodika chislennogo modelirovaniya dvumernykh techeniy dispersnykh sred na lagranzhevo-eylerovykh raschetnykh setkakh // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoye modelirovaniye fizicheskikh protsessov. 2013. Vyp. 4. S. 31–43.
 10. *Самигулин М. С., Кучерова П. А., Ёлкина А. Л., Тагирова И. Ю., Данилов Ю. Ф., Невмерзhitский Н. В.* Модель диспергирования среды на границе "жидкость-газ" // Там же. 2017. Вып. 1. С. 17–29.
Samigulin M. S., Kucherova P. A., Yelkina A. L., Tagirova I. Yu., Danilov Yu. F., Nevmerzhitskiy N. V. Model dispergirovaniya sredy na granitse "zhidkost-gaz" // Tam zhe. 2017. Vyp. 1. S. 17–29.
 11. *Sobieski W.* Drag coefficient in solid–fluid system modeling with the Eulerian multiphase model // Drying Technology: An Int. J. 2011. Vol. 29. P. 111–125.
 12. *Kolev N. I.* Multiphase Flow Dynamics 2: Thermal and Mechanical Interactions. 3rd ed. Berlin, Heidelberg: Springer, 2007.
 13. *Lahey R. T., Jr.* The virtual mass and lift force on a sphere in rotating and straining inviscid flow // Int. J. Multiphase Flow. 1987. Vol. 13, Issue 1. P. 113–121.
 14. *Lahey R. T., Jr.* The analysis of phase separation and phase distribution phenomena using two-fluid models // Nucl. Eng. Design. 1990. Vol. 122, Issue 1–3. P. 17–40.
 15. *Antal S. P., Lahey R. T., Jr., Flaherty J. E.* Analysis of phase distribution in fully developed laminar bubbly two-phase flow // Int. J. Multiphase Flow. 1991. Vol. 17, Issue 5. P. 635–652.

16. *Schiller L., Naumann A.* Uber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung // VDI Zeits. 1933. Vol. 77, No 12. P. 318—320.
17. *Патанкар С.* Численные методы решения задач теплообмена и динамики жидкости. М.: Энергоатомиздат, 1984.
Patankar S. Chislennyye metody resheniya zadach teploobmena i dinamiki zhidkosti. М.: Energoatomizdat, 1984.
18. *Kunz R. F., Siebert B. W., Cope W. K., Foster N. F., Antal S. P., Ettorre S. M.* A coupled phasic exchange algorithm for three-dimensional multy-field analysis of heated flows with mass transfer // Computers & Fluids. 1998. Vol. 27, No 7. P. 741—768.
19. *Ferziger J. H., Peric M.* Computational Methods for Fluid Dynamics. 3rd ed. Springer-Verlag, 1997.
20. *Rhie C. M., Chow W. L.* Numerical study of the turbulent flow past an airfoil with trailing edge separation // AIAA J. 1983. Vol. 21, No 11. P. 1525—1532.

Статья поступила в редакцию 16.01.20.

EULERIAN MULTI-PHASE FLOW SIMULATIONS BASED ON THE SIMPLE METHOD / O. G. Bliznyuk, O. E. Vlasova, A. V. Gichuk, A. S. Kozelkov, I. V. Lyalyushkina (FSUE "RFNC-VNIIEF", Sarov, Nizhniy Novgorod region).

Specific details of using the SIMPLE method for multi-phase flow simulations involving phase interactions are described. A multi-fluid model is used, in which each phase has its own density, velocity and temperature. Results of some test simulations are reported for the algorithm implemented in the software package "Logos"

Keywords: multi-phase flow, numerical simulations, SIMPLE method, "Logos" software package.

УДК 519.6

ПАКЕТ ПРОГРАММ "ЛОГОС". УЧЕТ КОНТАКТНОГО ВЗАИМОДЕЙСТВИЯ ПРИ РЕШЕНИИ ЗАДАЧ КВАЗИСТАТИЧЕСКОГО ДЕФОРМИРОВАНИЯ

А. Ю. Ерёменко, С. С. Косарим
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Рассмотрены особенности реализации учета контактного взаимодействия между различными телами или частями конструкции при решении задач квазистатического деформирования в пакете программ "Логос". Для реализации учета контактного взаимодействия в качестве основного выбран метод штрафа. Приводится описание алгоритма определения проникания контактирующих тел друг в друга. Выписаны основные формулы для получения контактных сил и их вклада в глобальную матрицу жесткости. Отдельно описывается учет трения между контактирующими телами. Точность реализованных методов подтверждается численным решением задачи Герца о контактном взаимодействии двух цилиндров бесконечной длины с учетом трения и его сравнением с аналитическим решением.

Ключевые слова: квазистатическое деформирование, учет контактного взаимодействия, метод штрафа, алгоритм определения проникания тел, учет трения, задача Герца о цилиндрах бесконечной длины, пакет программ "Логос".

Введение

Очень часто при решении задач квазистатического деформирования необходимо учитывать контактные явления, возникающие между различными телами или частями конструкции. При этом необходимо как обеспечивать условие непроникания тел друг в друга, так и учитывать возможные силы трения, возникающие между ними.

Контактное взаимодействие между телами в процессе решения задачи может происходить ввиду их касания исходя из начальной постановки задачи или образовываться в результате появления новых зон контакта у ранее не касавшихся тел, как показано на рис. 1. Также в процессе решения

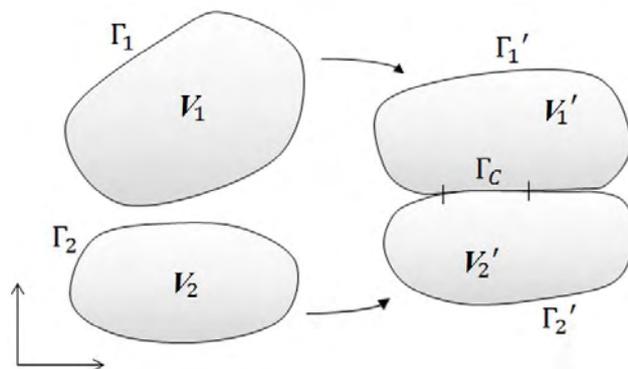


Рис. 1. Образование зоны контакта Γ_C между телами V_1 и V_2 , не имевшими касания в начальной конфигурации

задачи тела, изначально имевшие зоны контакта, могут расходиться, в результате чего контактные явления между ними будут исчезать. То есть при решении задач квазистатического деформирования возможно произвольное изменение зон контактного взаимодействия между телами.

Для недопущения проникания тел друг в друга в процессе решения задач квазистатического деформирования существуют различные методы [1–9]. Наиболее часто используемыми являются метод штрафа (метод штрафных функций, Penalty Method), метод множителей Лагранжа (Lagrange Multipliers Method) и пошаговый метод множителей Лагранжа (Augmented Lagrange Method) [1–9]. В качестве основного метода для реализации в пакете программ (ПП) "Логос" был выбран метод штрафа как наиболее универсальный и часто используемый.

Статья посвящена описанию особенностей реализации метода штрафа для учета контактного взаимодействия при решении задач квазистатического деформирования в ПП "Логос", разрабатываемом в РФЯЦ-ВНИИЭФ [10, 11].

Особенности реализации метода штрафа

Метод штрафа [1–9] является базовым для учета контактного взаимодействия в ПП "Логос" при решении задач квазистатического деформирования. Он получил свое название по аналогии с методом, используемым в разделе математической оптимизации, в котором ищется минимум некоторого функционала с наложенными ограничениями. При этом задача на минимизацию функционала с ограничениями преобразуется в задачу минимизации без ограничений, для чего к минимизируемому функционалу добавляется штрафная функция.

При решении задач квазистатического деформирования в ПП "Логос" решение ищется путем минимизации функционала, определяющего полную потенциальную энергию конструкции:

$$L = \Pi - A = \int_V \frac{1}{2} \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\sigma} dV - \int_V \mathbf{u}^T \cdot \mathbf{b} dV - \int_\Gamma \mathbf{u}^T \cdot \bar{\mathbf{t}} dS, \quad (1)$$

где Π — потенциальная энергия деформации; A — работа внешних сил; $\boldsymbol{\varepsilon}^T = (\varepsilon_x, \varepsilon_y, \varepsilon_z, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{xz})$, $\boldsymbol{\sigma}^T = (\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{yz}, \sigma_{xz})$ — тензоры деформации и напряжений, записанные в векторной форме; \mathbf{u} — вектор перемещений, \mathbf{b} и $\bar{\mathbf{t}}$ — векторы удельных объемных и поверхностных сил соответственно. Интегрирование ведется по объему V и поверхностям Γ всех тел в задаче.

Для учета контактного взаимодействия между телами в уравнение (1) добавляется штрафная функция w_C в виде удельной величины, определенной на поверхности тел Γ_C с потенциальным контактным взаимодействием и обеспечивающей условие геометрического непроникания тел:

$$w_C = \begin{cases} 0, & g_n > 0; \\ \frac{1}{2} k_n g_n^2, & g_n \leq 0, \end{cases} \quad (2)$$

где k_n — контактная жесткость; g_n — величина проникания тел через поверхность Γ_C . В результате при решении задач квазистатического деформирования ищется минимум выражения

$$L' = L + W_C, \quad (3)$$

где $W_C = \int_{\Gamma_C} w_C dS$ — полная штрафная функция.

При рассмотрении метода штрафа поверхности контактирующих тел разделяют на поверхности подчиненной и ведущей сторон и проверяют проникание поверхности подчиненной стороны через поверхность ведущей. Так, на рис. 2 показан фрагмент двух поверх-

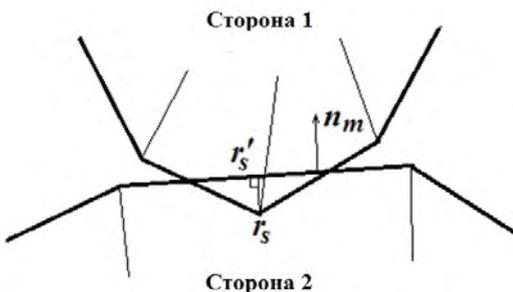


Рис. 2. Проникание узла одной из сторон через поверхность второй стороны

ностей (для простоты в двумерном случае), где узел \mathbf{r}_S (здесь и далее имеется в виду радиус-вектор) стороны 1 (подчиненной) проникает через поверхность противоположной стороны 2 (ведущей). Через \mathbf{r}'_S обозначена проекция узла \mathbf{r}_S подчиненной стороны на поверхность ведущей стороны (ближайшая точка поверхности стороны 2); \mathbf{n}_m — внешняя нормаль к ведущей стороне 2. Тогда g_n в точке \mathbf{r}_S определяется по формуле

$$g_n = (\mathbf{r}_S - \mathbf{r}'_S) \cdot \mathbf{n}_m. \quad (4)$$

Из определения (4) видно, что если проникание между поверхностями отсутствует, то $g_n > 0$, а исходя из формулы (2) можно сказать, что штрафная функция равна нулю и не оказывает никакого влияния на минимум функционала (3). Если же поверхность подчиненной стороны проникла через поверхность ведущей стороны, то $g_n \leq 0$, штрафная функция становится ненулевой и растет по мере увеличения проникания.

Выбор точек поверхности для приложения контактных сил. Возможны разные шаблоны приложения контактных сил к контактирующим поверхностям. Например, при рассмотрении контактного взаимодействия можно использовать шаблон *узел в узел* [3–5], при котором контактные силы прикладываются к узлам поверхностных сеток обеих сторон, или шаблон *узел в грань* [3–5] (на рис. 3, а для простоты показан двумерный случай), при котором контактные силы прикладываются к узлам поверхностной сетки подчиненной стороны и граням поверхностной сетки ведущей стороны. Также возможен вариант шаблона *грань в грань*, при котором контактные силы прикладываются в точках интегрирования поверхностных граней подчиненной стороны [2–9], как показано на рис. 3, б.

В ПП "Логос" для решения задач квазистатического деформирования реализованы шаблоны *узел в грань* и *грань в грань*. Однако на практике наиболее хорошо зарекомендовал себя шаблон *грань в грань*, и именно он используется по умолчанию.

На рис. 4 показаны фрагменты расчетных областей задачи, на которых контактные силы приложены в соответствии с разными шаблонами.

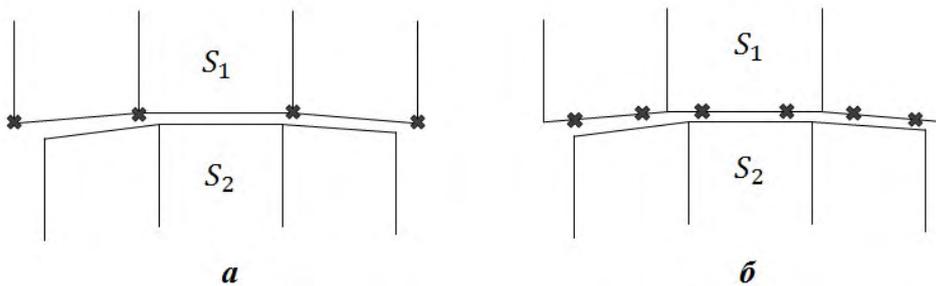


Рис. 3. Шаблоны приложения контактных сил: а — *узел в грань*; б — *грань в грань*

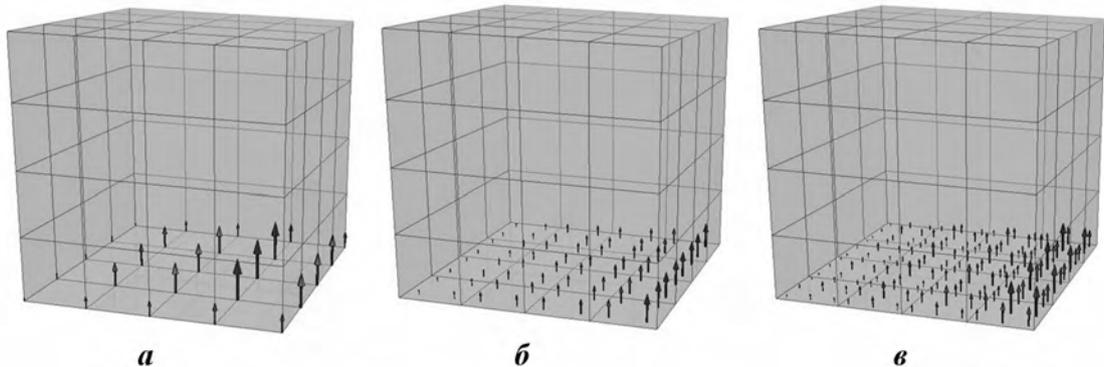


Рис. 4. Фрагменты расчетных областей задачи с разными шаблонами для приложения контактных сил: а — *узел в грань*; б — *грань в грань*; в — *грань в грань* с увеличенным порядком интегрирования

Алгоритм определения проникания одной контактирующей поверхности через другую. Одной из основных задач при реализации контактного взаимодействия является необходимость определения проникания одной контактирующей поверхности через другую. Для этого существует большое количество разных методов и алгоритмов [1–9]. В данной реализации используется достаточно простой и универсальный алгоритм, доказавший свою применимость на практике, особенностью которого является допустимость больших величин проникания через поверхности.

Пусть дана точка \mathbf{p}_j подчиненной стороны (точка интегрирования поверхностного элемента грани или узел сетки в зависимости от выбранного шаблона приложения контактных сил), как показано на рис. 5. Необходимо найти ближайшую грань s_m поверхности ведущей стороны, проекцию \mathbf{q}_m точки \mathbf{p}_j на грань s_m и определить, с какой стороны относительно грани s_m лежит точка \mathbf{p}_j .

Сначала для точки \mathbf{p}_j ищется ближайший узел \mathbf{r}_m сетки поверхности ведущей стороны. После этого среди всех граней поверхности ведущей стороны, сходящихся в найденном узле \mathbf{r}_m , ищется грань s_m , ближайшая к точке \mathbf{p}_j . Далее точка \mathbf{p}_j проецируется на поверхность грани s_m и, если точка проекции \mathbf{q}_m попала внутрь грани s_m , определяется, находится ли точка \mathbf{p}_j за поверхностью грани s_m . Если да, то считается, что точка \mathbf{p}_j проникла через контактную поверхность.

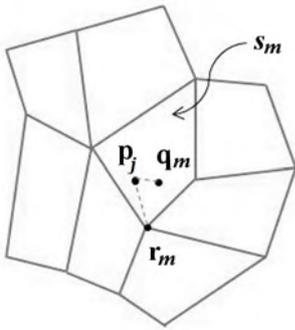


Рис. 5. Пример определения проникания точки \mathbf{p}_j подчиненной стороны через поверхность ведущей стороны

В ситуации, когда контактирующие поверхности построены поверх оболочечных конечных элементов (КЭ), необходимо однозначно задавать, какая из сторон контактирующих поверхностей является *наружной*, так как автоматически определить это из постановки задачи становится невозможно. Таким образом, наружные стороны контактирующих поверхностей задаются пользователем на этапе подготовки задачи. Например, на рис. 6, *а* показан выбор внешних нормалей для подчиненной стороны при задании контактного взаимодействия в задаче о контакте двух тонкостенных цилиндров, на рис. 6, *б* — для ведущей стороны контактирующей пары.

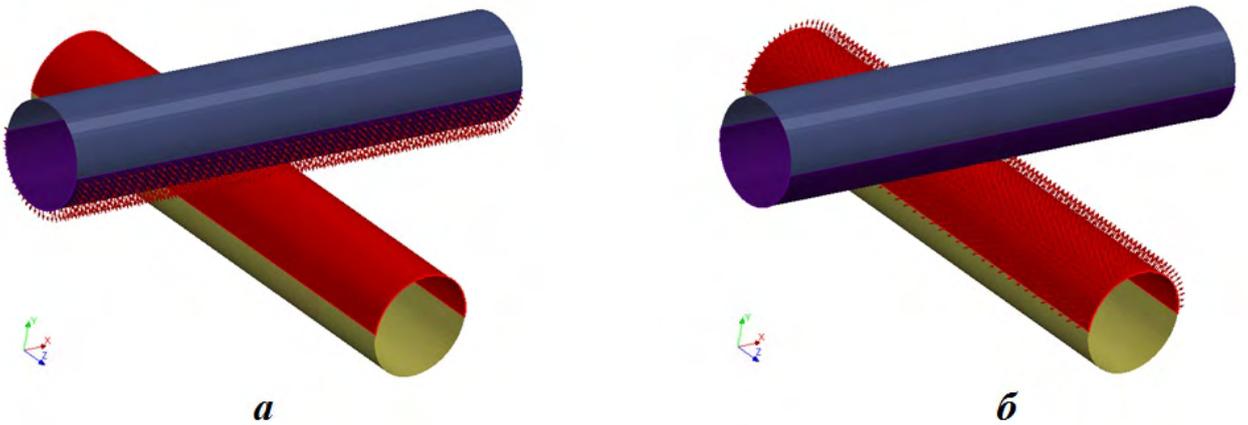


Рис. 6. Визуализация выбора внешних нормалей: *а* — для подчиненной стороны контактной пары; *б* — для ведущей стороны контактной пары

Определение контактной силы и ее жесткости. Далее приводятся формулы для получения контактных сил в случае использования шаблона *грань в грань*. При использовании шаблона *узел в грань* соотношения для контактных сил определяются аналогично.

На рис. 7 показаны положение точки интегрирования \mathbf{p} некоторой грани подчиненной стороны, образованной узлами сетки \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 и \mathbf{p}_4 , и ближайшая грань ведущей стороны, образованная

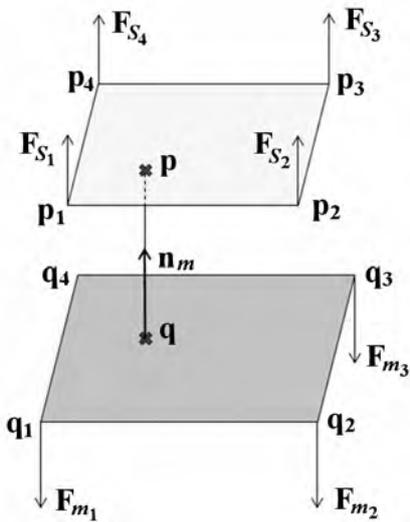


Рис. 7. Определение контактной силы между гранями контактирующих поверхностей

узлами сетки $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ и \mathbf{q}_4 (с внешней нормалью \mathbf{n}_m), на которую точка \mathbf{p} проецируется в точку \mathbf{q} . Для простоты показана ситуация, когда проникания грани подчиненной стороны через грань ведущей стороны нет, так как с точки зрения вывода формул это не влияет на результат.

Положение точек \mathbf{p} и \mathbf{q} определяется с применением функций формы КЭ $\varphi_S(\xi, \eta)$ и $\varphi_m(\xi, \eta)$ [12], используемых для построения аппроксимаций в рамках соответствующих граней, записывается по формулам

$$\mathbf{p} = \sum_i \varphi_{S_i}(\xi_S, \eta_S) \mathbf{p}_i; \quad \mathbf{q} = \sum_i \varphi_{m_i}(\xi_m, \eta_m) \mathbf{q}_i,$$

где ξ_S, η_S — локальные координаты точки \mathbf{p} в элементе поверхности подчиненной стороны; ξ_m, η_m — локальные координаты точки \mathbf{q} в элементе поверхности ведущей стороны.

Вектор внешней нормали \mathbf{n}_m к грани ведущей стороны определяется из соотношения

$$\mathbf{n}_m = \frac{\partial \mathbf{q}}{\partial \xi} \times \frac{\partial \mathbf{q}}{\partial \eta} / \left\| \frac{\partial \mathbf{q}}{\partial \xi} \times \frac{\partial \mathbf{q}}{\partial \eta} \right\|.$$

Вектор \mathbf{g} , соединяющий точки \mathbf{p} и \mathbf{q} , определяется по формуле

$$\mathbf{g} = \mathbf{p} - \mathbf{q}.$$

Тогда величина проникания g_n , определенная по формуле (4), для точки \mathbf{p} будет получаться из соотношения

$$g_n = \mathbf{g} \cdot \mathbf{n}_m. \quad (5)$$

С использованием введенных обозначений результат минимизации функционала (2) для удельной контактной силы в точке поверхности \mathbf{p} можно записать по формуле

$$\mathbf{f}_p = -k_n g_n \mathbf{n}_m.$$

Для получения полной контактной силы \mathbf{F}_p в точке \mathbf{p} удельная сила \mathbf{f}_p умножается на площадь S_p , отнесенную к точке интегрирования \mathbf{p} элемента поверхности подчиненной стороны:

$$\mathbf{F}_p = S_p \mathbf{f}_p.$$

В данной работе в качестве S_p берется площадь, которая вычисляется по формуле

$$S_p = w_p \left| \frac{\partial \mathbf{p}}{\partial \xi} \times \frac{\partial \mathbf{p}}{\partial \eta} \right|,$$

где w_p — весовой множитель гауссовой точки \mathbf{p} в КЭ [12], используемом для аппроксимации грани подчиненной стороны.

В случае использования шаблона *узел в грань* площадь, отнесенная к узлу подчиненной стороны, определяется по площадям поверхностных граней, сходящихся в нем.

В результате приложения контактной силы \mathbf{F}_p к грани подчиненной стороны в точке \mathbf{p} на грань ведущей стороны в точке \mathbf{q} начинает действовать равная по значению, но противоположная по направлению сила

$$\mathbf{F}_q = -\mathbf{F}_p.$$

Вклады сил \mathbf{F}_p и \mathbf{F}_q в силы, определенные в поверхностных узлах подчиненной и ведущей сторон, получаются с использованием КЭ-аппроксимаций соответствующих граней по формулам

$$\mathbf{F}_{S_i} = \varphi_{S_i}(\xi_S, \eta_S) \mathbf{F}_p; \quad (6)$$

$$\mathbf{F}_{m_i} = \varphi_{m_i}(\xi_m, \eta_m) \mathbf{F}_q = -\varphi_{m_i}(\xi_m, \eta_m) \mathbf{F}_p, \quad (7)$$

где \mathbf{F}_{S_i} — вклад в силу в узле \mathbf{p}_i грани подчиненной стороны; \mathbf{F}_{m_i} — вклад в силу в узле \mathbf{q}_i грани ведущей стороны.

В случае использования шаблона *узел в грань* сила \mathbf{F}_p прикладывается непосредственно к узлу поверхности подчиненной стороны.

Выбор контактной жесткости. Для успешного получения решения задачи необходимо правильно выбрать контактную жесткость k_n в (2). Относительно малые значения легче обеспечивают сходимость итераций по нелинейности, однако могут ухудшить точность получаемого решения и допустить избыточное проникание одной контактной поверхности через другую. Большие значения k_n сделают решение более точным, но могут привести к получению матриц с плохой обусловленностью, что приведет к расхождению нелинейного итерационного процесса. Таким образом, задача по выбору контактной жесткости формулируется как необходимость определить как можно большее значение k_n , которое обеспечит сходимость нелинейного итерационного процесса.

Автоматически выбрать контактную жесткость, которая бы применялась во всех случаях и на всех структурах, практически нельзя, поэтому инженеру предоставлена возможность управлять масштабирующим множителем для контактной жесткости, чтобы в зависимости от задачи получить итоговую сходимость с максимально возможной точностью.

Для получения значения контактной жесткости в ПП "Логос" при решении задач квазистатического деформирования реализованы два способа.

При первом способе для определения контактной жесткости используется формула

$$k_n = \frac{\min(E_S, E_m)}{L_{ave}},$$

где E_S, E_m — модули Юнга подчиненной и ведущей сторон соответственно; L_{ave} — средняя длина ребра элемента на поверхности подчиненной стороны. При таком способе определения жесткости получается единое значение k_n для всего контакта.

При использовании второго способа контактная жесткость становится локальной, определяется для каждого элемента контактирующей поверхности независимо и вычисляется по формуле

$$k_n = \min\left(\frac{E_S}{V_S} S_S, \frac{E_m}{V_m} S_m\right),$$

где V_S, V_m — объемы ячеек оригинальной сетки, прилегающих к элементам подчиненной и ведущей сторон соответственно; S_S, S_m — площади элементов подчиненной и ведущей сторон.

Практика показала, что второй способ более надежен, и он используется по умолчанию.

Вклад контактных сил в глобальную матрицу жесткости. Для получения вклада контактных сил в глобальную матрицу жесткости необходимо для каждой силы $\mathbf{F}_{S_i}, \mathbf{F}_{m_i}$ взять производные по всем перемещениям \mathbf{u}_{S_i} и \mathbf{u}_{m_i} . Для этого на первом этапе вычисляются производные от \mathbf{F}_p по всем перемещениям \mathbf{u}_{S_i} и \mathbf{u}_{m_i} по формуле

$$\frac{1}{S_p} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_i} = -k_n \left(\mathbf{n}_m \otimes \frac{\partial g_n}{\partial \mathbf{u}_i} \right) - k_n g_n \frac{\partial \mathbf{n}_m}{\partial \mathbf{u}_i}, \quad (8)$$

где \mathbf{u}_i — вектор перемещений узла либо подчиненной, либо ведущей стороны, а через \otimes обозначается тензорное произведение векторов [13]. Производные $\frac{\partial g_n}{\partial \mathbf{u}_i}$ и $\frac{\partial \mathbf{n}_m}{\partial \mathbf{u}_i}$ являются сложными функциями от координат узлов \mathbf{u}_i , требующими большого количества вычислений.

Для получения производных от вкладов в узловые силы \mathbf{F}_{S_j} подчиненной стороны из формулы (6) получается соотношение

$$\frac{\partial \mathbf{F}_{S_j}}{\partial \mathbf{u}_i} = \varphi_{S_j}(\xi_S, \eta_S) \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_i}. \quad (9)$$

Для вкладов в узловые силы \mathbf{F}_{m_i} ведущей стороны из формулы (7) получается соотношение

$$\frac{\partial \mathbf{F}_{m_j}}{\partial \mathbf{u}_i} = -\mathbf{F}_p \otimes \left(\frac{\partial \varphi_{m_j}(\xi_m, \eta_m)}{\partial \xi_m} \frac{\partial \xi_m}{\partial \mathbf{u}_i} + \frac{\partial \varphi_{m_j}(\xi_m, \eta_m)}{\partial \eta_m} \frac{\partial \eta_m}{\partial \mathbf{u}_i} \right) - \varphi_{m_j}(\xi_m, \eta_m) \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_i}. \quad (10)$$

В работе [14] было показано, что производные, полученные по формулам (8)–(10) для относительно больших значений g_n , могут приводить к появлению матриц жесткости, не являющихся положительно определенными, что катастрофически влияет на сходимость итераций по нелинейности ввиду контактного взаимодействия.

Для борьбы с этим явлением на начальных итерациях можно использовать упрощенные формулы, которые получаются в предположении, что в пределах одной итерации положение ξ_m, η_m меняется незначительно и этим изменением можно пренебречь. В результате выражение (8) сильно упрощается, и после подстановки выражения (5) получается соотношение

$$\frac{1}{S_p} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p} = -k_n (\mathbf{n}_m \otimes \mathbf{n}_m). \quad (11)$$

При этом выражения для производных от итоговых сил (9) и (10) упрощаются до формул

$$\frac{\partial \mathbf{F}_{S_j}}{\partial \mathbf{u}_{S_i}} = \varphi_{S_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_{S_i}} = \varphi_{S_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p} \frac{\partial \mathbf{u}_p}{\partial \mathbf{u}_{S_i}} = \varphi_{S_j} \varphi_{S_i} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p}; \quad (12)$$

$$\frac{\partial \mathbf{F}_{S_j}}{\partial \mathbf{u}_{m_i}} = \varphi_{S_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_{m_i}} = -\varphi_{S_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p} \frac{\partial \mathbf{u}_p}{\partial \mathbf{u}_{m_i}} = -\varphi_{S_j} \varphi_{m_i} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p}; \quad (13)$$

$$\frac{\partial \mathbf{F}_{m_j}}{\partial \mathbf{u}_{m_i}} = -\varphi_{m_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_{m_i}} = \varphi_{m_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p} \frac{\partial \mathbf{u}_p}{\partial \mathbf{u}_{m_i}} = \varphi_{m_j} \varphi_{m_i} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p}; \quad (14)$$

$$\frac{\partial \mathbf{F}_{m_j}}{\partial \mathbf{u}_{S_i}} = -\varphi_{m_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_{S_i}} = -\varphi_{m_j} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p} \frac{\partial \mathbf{u}_p}{\partial \mathbf{u}_{S_i}} = -\varphi_{m_j} \varphi_{S_i} \frac{\partial \mathbf{F}_p}{\partial \mathbf{u}_p}. \quad (15)$$

На текущий момент в ПП "Логос" пользователь сам выбирает, по каким формулам вести расчет. По умолчанию используются упрощенные формулы (11)–(15).

Учет трения на контактах. В ПП "Логос" при решении задач квазистатического деформирования учет сил трения реализован в соответствии с законом Кулона

$$\|\mathbf{F}_T\| \leq \mu \|\mathbf{F}_N\|, \quad (16)$$

где $\|\mathbf{F}_T\|$ — значение силы трения в плоскости контакта; $\|\mathbf{F}_N\|$ — значение нормальной контактной силы; μ — коэффициент трения. При этом используется подход, при котором диаграмма зависимости силы трения от перемещения в плоскости контакта выглядит, как показано на рис. 8.

В соответствии с рисунком в начале относительного перемещения контактирующих тел в плоскости контакта начинает нарастать сила трения, которая стремится вернуть тела в исходное положение. Чем большее относительное перемещение между телами возникает, тем большая возвращающая сила появляется. Таким образом, моделируется режим, соответствующий трению покоя.

Скорость возрастания силы трения покоя определяется при помощи тангенциальной контактной жесткости k_t , которая, как и нормальная контактная жесткость k_n , в идеальном случае должна стремиться к бесконечности. Однако слишком большие значения контактной жесткости k_t вызывают проблемы с достижением сходимости нелинейного итерационного процесса, поэтому на практике ее приходится ограничивать.

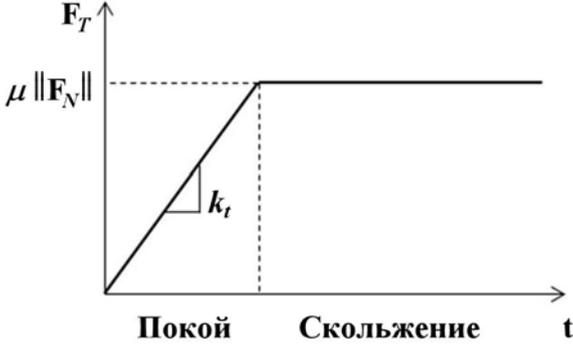


Рис. 8. Зависимость силы трения от тангенциального перемещения

\mathbf{q}^n — положение этих точек перед началом счета текущего шага по нагрузке. Тогда вектор $\Delta \mathbf{s}^{n+1}$, обозначающий перемещение точки \mathbf{p} относительно ведущей поверхности за $(n+1)$ -й шаг по нагрузке (инкремент), определяется по формуле

$$\Delta \mathbf{s}^{n+1} = (\mathbf{p}^{n+1} - \mathbf{q}^{n+1}) - (\mathbf{p}^n - \mathbf{q}^n).$$

При этом относительное перемещение точки \mathbf{p} в плоскости контакта (тангенциальное) за $(n+1)$ -й шаг вычисляется по формуле

$$\Delta \mathbf{t}^{n+1} = \Delta \mathbf{s}^{n+1} - \Delta s_n^{n+1} \mathbf{n}_m, \quad (17)$$

где $\Delta s_n^{n+1} = \Delta \mathbf{s}^{n+1} \cdot \mathbf{n}_m$ — изменение положения точки \mathbf{p} вдоль направления нормали \mathbf{n}_m .

Полное перемещение точки \mathbf{p} относительно ведущей стороны в плоскости контакта за $(n+1)$ -й шаг определяется по формуле

$$\mathbf{t}^{n+1} = \mathbf{t}^n + \Delta \mathbf{t}^{n+1}. \quad (18)$$

Перемещение \mathbf{t} точки \mathbf{p} в плоскости контакта раскладывается на перемещения в режиме, имитирующем состояние покоя, и в режиме скольжения тел относительно друг друга:

$$\mathbf{t} = \mathbf{t}_{(e)} + \mathbf{t}_{(p)},$$

где $\mathbf{t}_{(e)}$ — перемещение в режиме покоя; $\mathbf{t}_{(p)}$ — перемещение в режиме скольжения (индекс e используется по аналогии с упругостью (elastic), индекс p — по аналогии с пластичностью (plastic)).

Пусть известно, что к началу текущего $(n+1)$ -го шага по нагрузке точка \mathbf{p} "проскользила" расстояние, заданное вектором $\mathbf{t}_{(p)}^n$. Полное перемещение точки \mathbf{p} в плоскости контакта определяется вектором \mathbf{t}^{n+1} . Тогда для получения результирующей силы трения на шаге приложения нагрузки \mathbf{F}_T^{n+1} необходимо определить, что происходит с ней при перемещении точки \mathbf{p} , заданном вектором $\mathbf{t}^{n+1} - \mathbf{t}_{(p)}^n$. Для этого используется процедура, блок-схема которой показана на рис. 9.

Сначала вычисляется прогнозируемая сила трения по формуле

$$\mathbf{F}_T^{trial} = -k_t (\mathbf{t}^{n+1} - \mathbf{t}_{(p)}^n). \quad (19)$$

Если выполнено условие закона Кулона для силы трения (16), т. е. $\|\mathbf{F}_T^{trial}\| \leq \mu \|\mathbf{F}_N^{n+1}\|$, где \mathbf{F}_N^{n+1} — текущая нормальная контактная сила, то прогнозируемая сила \mathbf{F}_T^{trial} становится результирующей силой трения: $\mathbf{F}_T^{n+1} = \mathbf{F}_T^{trial}$.

Если условие (16) не выполнено, т. е. $\|\mathbf{F}_T^{trial}\| > \mu \|\mathbf{F}_N^{n+1}\|$, то сила трения выбирается как максимально возможная по формуле

$$\mathbf{F}_T^{n+1} = \mu \|\mathbf{F}_N^{n+1}\| \frac{\mathbf{F}_T^{trial}}{\|\mathbf{F}_T^{trial}\|}. \quad (20)$$

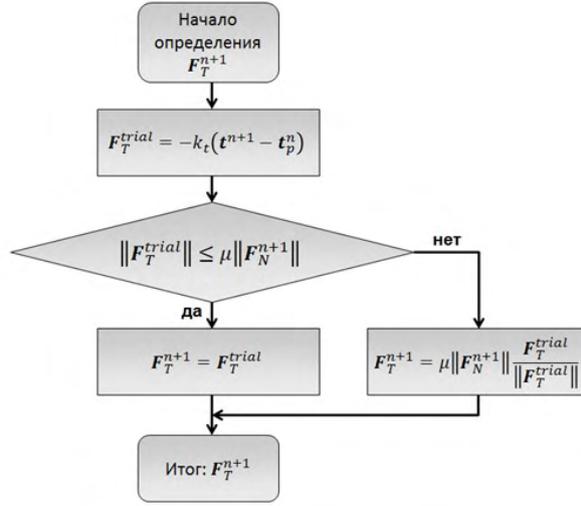


Рис. 9. Блок-схема вычисления контактной силы трения

Параллельно с вычислением результирующей силы трения вычисляется и накапливается расстояние скольжения точки \mathbf{p} :

$$\Delta \mathbf{t}_{(p)}^{n+1} = \frac{\|\mathbf{F}_T^{trial}\| - \mu\|\mathbf{F}_N^{n+1}\|}{k_t} \frac{\mathbf{F}_T^{trial}}{\|\mathbf{F}_T^{trial}\|}; \quad \mathbf{t}_{(p)}^{n+1} = \mathbf{t}_{(p)}^n + \Delta \mathbf{t}_{(p)}^{n+1}.$$

Чтобы учесть вклад от силы трения в глобальную матрицу жесткости, необходимо сначала вычислить производные от тангенциального перемещения \mathbf{t}^{n+1} по перемещению всех узлов, образующих грани подчиненной и ведущей сторон. Для этого после дифференцирования соотношения (18) с учетом формулы (17) получается выражение

$$\frac{\partial \mathbf{t}^{n+1}}{\partial \mathbf{u}_i} = \frac{\partial \Delta \mathbf{s}^{n+1}}{\partial \mathbf{u}_i} - \mathbf{n}_m \otimes \frac{\partial \Delta s_n^{n+1}}{\partial \mathbf{u}_i} - \Delta s_n^{n+1} \frac{\partial \mathbf{n}_m}{\partial \mathbf{u}_i}. \quad (21)$$

В случае использования упрощенных формул для нормальной контактной силы формула (21) также упрощается: остается только производная по перемещению \mathbf{u}_p узла \mathbf{p} , которая приобретает вид

$$\frac{\partial \mathbf{t}^{n+1}}{\partial \mathbf{u}_p} = \mathbf{I} - \mathbf{n}_m \otimes \mathbf{n}_m. \quad (22)$$

Затем, если для прогнозируемой силы трения выполняется условие (16), то после дифференцирования выражения (19) с учетом полученного соотношения (22) производная от силы трения в точке \mathbf{p} запишется по формуле

$$\frac{\partial \mathbf{F}_T^{n+1}}{\partial \mathbf{u}_p} = -k_t (\mathbf{I} - \mathbf{n}_m \otimes \mathbf{n}_m).$$

Если на текущем инкременте возникло скольжение, то необходимо дифференцировать уже соотношение (20), и тогда производная от силы трения получится в виде

$$\frac{\partial \mathbf{F}_T^{n+1}}{\partial \mathbf{u}_p} = \mu \mathbf{e}^{trial} \otimes \left(-\mathbf{n}_m \frac{\partial \mathbf{F}_N^{n+1}}{\partial \mathbf{u}_p} \right) + \mu \frac{\|\mathbf{F}_N^{n+1}\|}{\|\mathbf{F}_T^{trial}\|} \left(\mathbf{I} - \mathbf{e}^{trial} \otimes \mathbf{e}^{trial} \right) \frac{\partial \mathbf{F}_T^{trial}}{\partial \mathbf{u}_p},$$

где $\mathbf{e}^{trial} = \frac{\mathbf{F}_T^{trial}}{\|\mathbf{F}_T^{trial}\|}$ — направляющий единичный вектор действия силы трения.

Численное моделирование контактного взаимодействия двух цилиндров с параллельными осями

Постановка задачи. Для проверки применимости модуля учета контактного взаимодействия при решении задач квазистатического деформирования в ПП "Логос" была решена задача Герца о контактном взаимодействии двух бесконечно длинных цилиндров с параллельными осями с учетом трения между ними [15]. Полученные результаты сравниваются с аналитическим решением.

На рис. 10 показана исходная постановка задачи.

Два бесконечно длинных цилиндра с радиусами $R_1 = 10$ мм и $R_2 = 13$ мм и параллельными друг другу осями сдавливаются вместе нагрузкой, заданной в виде удельных сил \vec{F} , определенных на единицу длины цилиндров и распределенных вдоль их образующих. Кроме того, перпендикулярно действию удельных сил \vec{F} к цилиндрам прикладываются удельные силы \vec{Q} , также определенные на единицу длины цилиндров и вызывающие возникновение сил трения в зоне контактного взаимодействия между ними. Коэффициент трения между цилиндрами $\mu = 0,2$.

Значение сдавливающих цилиндры удельных сил составляет $|\vec{F}| = 800$ Н/мм, а значение сдвиговых удельных сил равно $|\vec{Q}| = 4,5$ Н/мм. Сдвиговые нагрузки \vec{Q} подобраны таким образом, чтобы цилиндры все еще оставались в контакте между собой, но в зоне контактного взаимодействия получались как области, соответствующие трению покоя, так и области скольжения.

В задаче используется линейный упругий материал с модулем Юнга $E = 30\,000$ Н/мм² и коэффициентом Пуассона $\nu = 0,3$.

Задача решалась в трехмерной постановке. Так как в результате решения задачи интересует форма зоны контактного взаимодействия между цилиндрами, то в силу малости возникающих деформаций во избежание проблем, связанных с приложением нагрузок в виде сил в точке, рассматриваются только нижняя часть верхнего цилиндра и верхняя часть нижнего цилиндра, а нагрузки заменяются эквивалентными и прикладываются к верхнему цилиндру. Длина цилиндров берется равной 0,1 мм. На рис. 11 показана итоговая постановка задачи, использованная при ее решении в ПП "Логос".

В постановке задачи были заданы следующие закрепления: ограничено перемещение узлов нижнего цилиндра, лежащих на горизонтальной плоскости среза вдоль оси Oy , а для узлов, лежащих на оси цилиндра, — еще и вдоль оси Ox ; ограничено перемещение всех узлов обоих цилиндров, лежащих в плоскостях $z = 0$ и $z = 0,1$ мм вдоль оси Oz .

На поверхности среза верхнего цилиндра задаются сжимающая удельная сила $2|\vec{F}| = 1\,600$ Н/мм, действующая вертикально вниз (против оси Oy), и сдвиговая удельная сила $2|\vec{Q}| = 9$ Н/мм, действующая вдоль оси Ox .

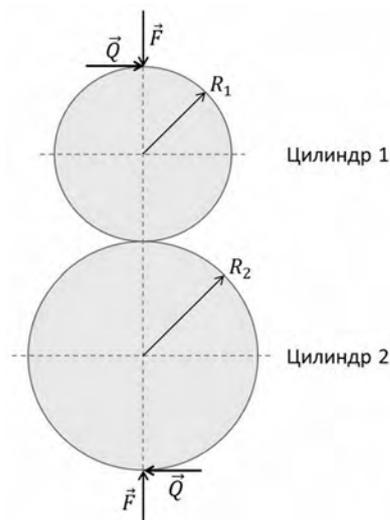


Рис. 10. Исходная постановка задачи о контакте двух цилиндров

Зона потенциального контактного взаимодействия между цилиндрами показана на рис. 12. Красным цветом обозначена ведущая сторона контактного взаимодействия (поверхность нижнего цилиндра), синим — подчиненная сторона (поверхность верхнего цилиндра).

Пространственная дискретизация расчетной области выполнена преимущественно на основе объемного восьмиугольного КЭ. В расчетах используются элементы сплошной среды с уточненными функциями формы. Для получения решения достаточной точности необходимо использовать подробную сетку с измельчением в области контакта цилиндров.

Для проведения расчетов использовались пять последовательно сгущающихся сеток в плоскости сечения цилиндров (сгущение вдоль оси цилиндров не делалось для экономии в силу существенной двумерности задачи). Размеры использованных КЭ-моделей составляли 21 080, 80 912, 220 544, 315 112 и 535 384 КЭ. Вид сетки для самой грубой КЭ-модели (21 080 КЭ) представлен на рис. 13.

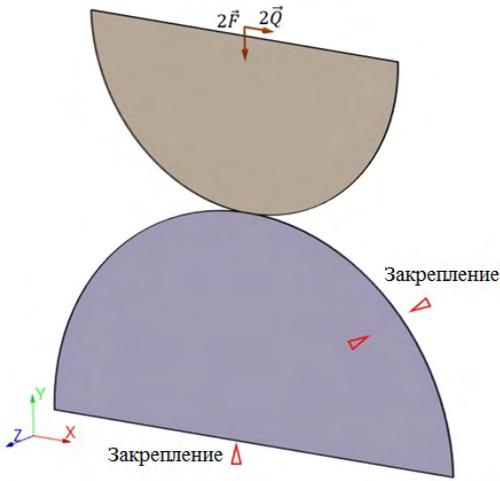


Рис. 11. Итоговая постановка и геометрия задачи о двух цилиндрах

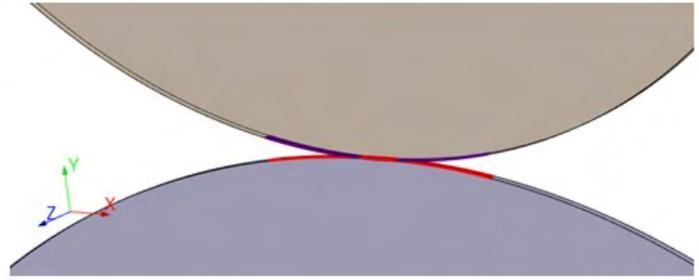


Рис. 12. Определение зоны потенциального контактного взаимодействия между цилиндрами

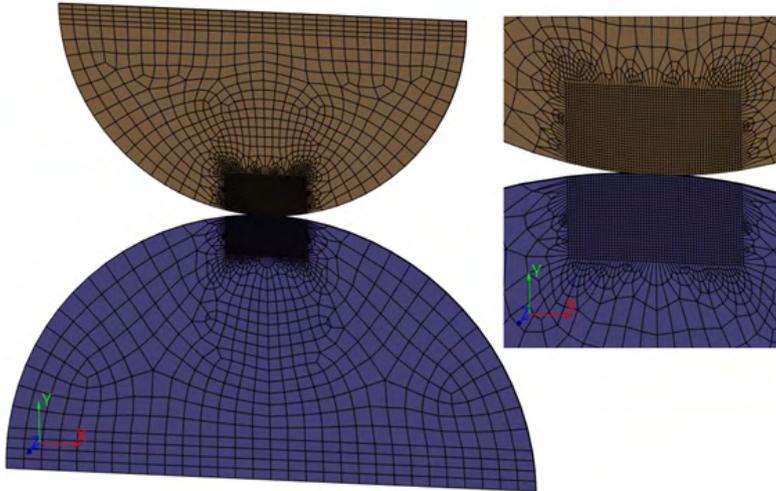


Рис. 13. КЭ-сетка для задачи о контактном взаимодействии между двумя цилиндрами

Аналитическое решение. Максимальное давление в зоне контакта p_{\max} определяется по формуле [15]

$$p_{\max} = \frac{2|\vec{F}|}{\pi a} = 1\,218,7 \text{ Па.}$$

Здесь a — половина ширины зоны контакта, которая рассчитывается по формуле

$$a = \sqrt{\frac{4|\vec{F}|(k_1 + k_2)R_1R_2}{R_1 + R_2}} = 0,8358 \text{ мм,}$$

где $k_i = \frac{1 - \nu_i^2}{\pi E_i}$. При этом распределение контактного давления вдоль оси Ox , лежащей в поперечном сечении цилиндров, определяется по формуле

$$p(x) = \frac{2|\vec{F}|}{\pi a^2} (a^2 - x^2)^{1/2}.$$

Начало зоны скольжения по отношению к началу зоны контакта определяется по формуле

$$c = a\sqrt{1 - \frac{|\vec{Q}|}{\mu|\vec{F}|}} = 0,7086 \text{ мм.}$$

Профиль тангенциального напряжения $q(x)$ поперек цилиндров получается по формуле

$$q(x) = \mu \frac{2|\vec{F}|}{\pi a^2} \left[(a^2 - x^2)^{1/2} - (c^2 - x^2)^{1/2} \right].$$

Результаты расчета. На рис. 14 приводятся поля перемещения вдоль оси Oy и интенсивности напряжения по Мизесу, полученные по ПП "Логос" без приложения сдвиговой нагрузки $2\vec{Q}$, а на рис. 15 — с ее приложением.

Видно, что в результате приложения сдвиговой нагрузки симметрия решения относительно вертикальной плоскости, проходящей через центры цилиндров, слегка нарушается. Но, как и предполагается в источнике [15], ввиду малости деформаций это искажение крайне мало влияет на распределение величин в зоне контактного взаимодействия между цилиндрами.

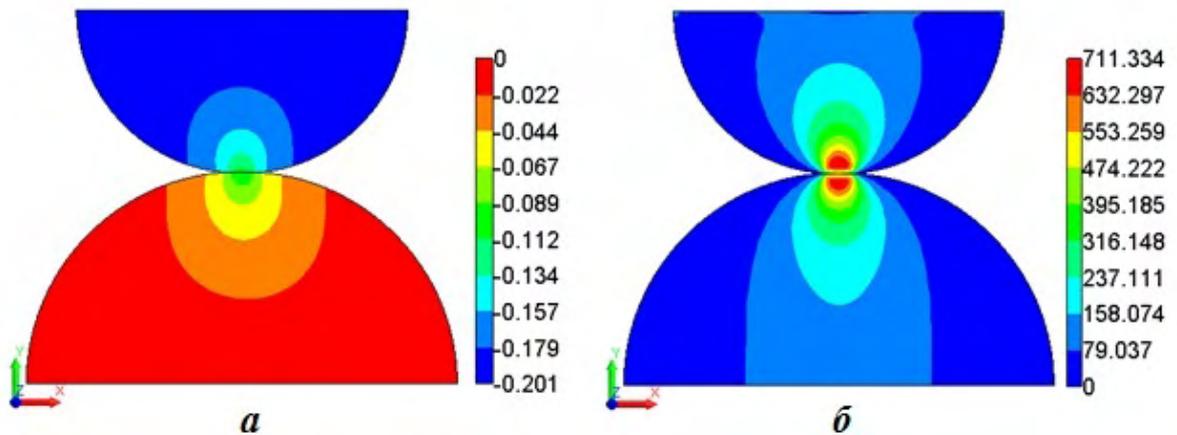


Рис. 14. Поля перемещения (а) и интенсивности напряжений по Мизесу (б), полученные с использованием ПП "Логос" без приложения сдвиговой нагрузки

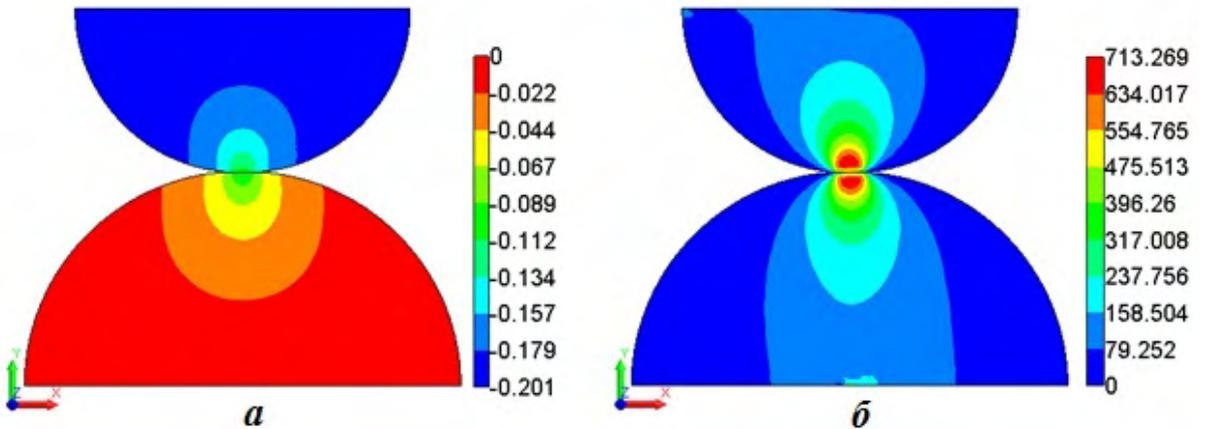


Рис. 15. Поля перемещения (а) и интенсивности напряжений по Мизесу (б), полученные с использованием ПП "Логос" с приложением сдвиговой нагрузки

На рис. 16 приводится график зависимости ошибки определения максимального контактного давления от густоты сетки (N — количество КЭ; N_0 — количество КЭ для самой грубой сетки). Видно, что по максимальному контактному давлению наблюдается по меньшей мере первый порядок сходимости, при этом на самой подробной сетке ошибка составляет около 0,1 %.

На рис. 17 на примере модели из 220 544 КЭ приводится график зависимости ошибки определения максимального контактного давления от контактной жесткости k_n . Видно, что увеличение контактной жесткости положительно повлияло на точность получаемого результата. Однако в данном случае увеличение контактной жесткости в 10 раз привело к затрудненной сходимости нелинейного итерационного процесса в ходе решения задачи, в результате чего двукратно возросло число выполненных итераций по нелинейности. Тем не менее даже с контактной жесткостью, которая в ПП "Логос" выбрана по умолчанию, погрешность по максимальному контактному давлению составила меньше 0,5 %.

На рис. 18 на примере модели из 220 544 КЭ показано сравнение профилей давления и тангенциального напряжения, полученных по ПП "Логос", с аналитическим решением. Видно, что расчетные профили обеих величин очень хорошо согласуются с аналитическим решением. Хорошо согласуется как зона зацепления между цилиндрами, так и зона скольжения.

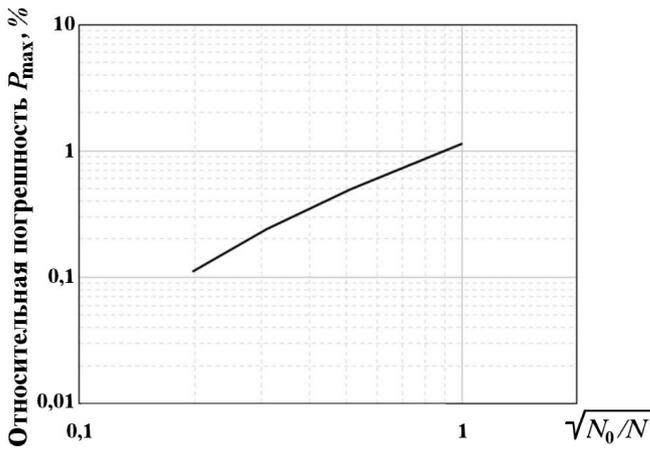


Рис. 16. Зависимость погрешности определения максимального контактного давления от густоты сетки

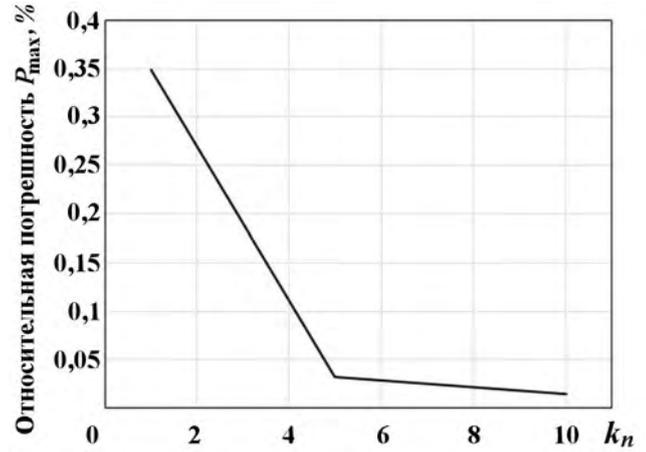


Рис. 17. Зависимость погрешности определения максимального контактного давления от контактной жесткости k_n ($k_n = 1$ — значение по умолчанию)

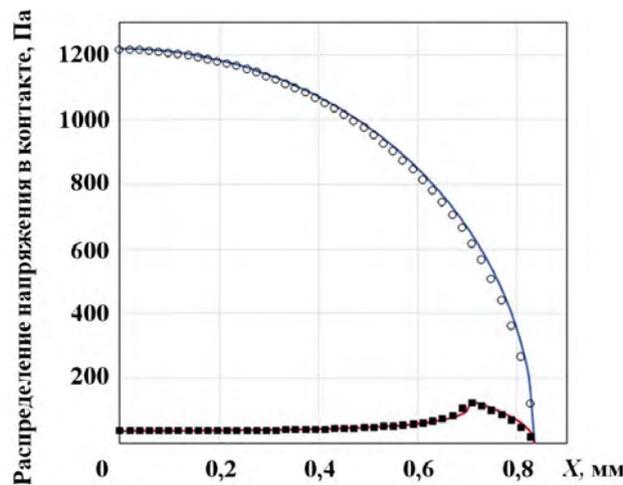


Рис. 18. Сравнение профилей давления и тангенциального напряжения, полученных по ПП "Логос", с аналитическим решением: — — давление, аналитическое решение; \circ — давление, "Логос"; — — тангенциальное напряжение, аналитическое решение; \blacksquare — тангенциальное напряжение, "Логос"

Заключение

В статье приведено описание реализации метода штрафа в ПП "Логос" для учета контактного взаимодействия при решении задач квазистатического деформирования. Данный метод, выбранный как наиболее универсальный и часто используемый, является базовым в ПП "Логос" для решения этого класса задач.

Изложены основные идеи данного метода и аспекты его реализации в ПП "Логос". Приведено описание алгоритма для определения проникания одной из контактирующих поверхностей через другую. Описаны реализованные шаблоны приложения контактных сил, приведены основные формулы для расчета нормальной контактной силы и силы трения.

В завершение приводится решение задачи Герца о контактном взаимодействии двух бесконечных цилиндров, находящихся под действием сжимающей и сдвиговой нагрузок, с учетом трения между ними. Решение, полученное с использованием ПП "Логос", сравнивается с аналитическим решением. Наблюдается очень хорошее согласие, что подтверждает корректность применения реализованных методов и алгоритмов.

Список литературы

1. *Taylor R. L., Cheng L. Y.* A consistent finite element formulation of nonlinear frictional contact problems // Numerical Techniques for Engineering Analysis and Design. 1987. Vol. 1. P. 37–49.
2. *Asano N.* A virtual work principle using penalty function method for impact contact problem of two bodies // Trans. JSME. 1985. A51, No 467. P. 1863–1898.
3. *Kloosterman G.* Contact Methods in Finite Element Simulations. Wageningen: Ponsen & Looijen, 2002.
4. *Rust W.* Non-Linear Finite Element Analysis in Structural Mechanics. Switzerland: Springer, 2015.
5. *Wriggers P.* Computational Contact Mechanics. Netherland: Springer, 2006.
6. *Bertsekas D. P.* Constrained Optimization and Lagrange Multiplier Methods. N.-Y.: Academic Press, 1984.
7. *Бураго Н. Г., Кукуджанов В. Н.* Обзор контактных алгоритмов // Известия РАН. Механика твердого тела. 2005. № 1. С. 44–85.
Burago N. G., Kukudzhyanov V. N. Obzor kontaktnykh algoritmov // Izvestiya RAN. Mekhanika tverdogo tela. 2005. № 1. S. 44–85.
8. *Yang B., Laursen T., Meng X.* Two dimensional mortar contact methods for large deformation frictional sliding // Int. J. Num. Meth. Eng. 2004. Vol. 62. P. 1183–1225.
9. *Puso M., Laursen T.* A mortar segment-to-segment frictional contact methods for large deformations // Comp. Meth. Appl. Mech. Eng. 2004. Vol. 193. P. 4891–4913.
10. *Дерюгин Ю. Н., Козелков А. С., Спиридонов В. Ф., Циберев К. В., Шагалиев Р. М.* Многофункциональный высокопараллельный пакет программ ЛОГОС для решения задач тепломассообмена и прочности // Сб. тез. докл. С.-Петербург. науч. форума "Наука и общество". С.-Пб.: Изд-во Политех. ун-та, 2012. С. 102.
Deryugin Yu. N., Kozelkov A. S., Spiridonov V. F., Tsiberev K. V., Shagaliev R. M. Mnogofunktsionalny vysokoparallelny paket programm LOGOS dlya resheniya zadach teplomassoperenosa i prochnosti // Sb. tez. dokl. S.-Peterburg. nauch. foruma "Nauka i obshchestvo". S.-Pb.: Izd-vo Politekh un-ta, 2012. S. 102.
11. *Речкин В. Н., Спиридонов В. Ф., Циберев К. В., Дьянов Д. Ю., Наумов А. О., Косарим С. С., Филимонкин Е. А., Бартенев Ю. Г., Щаникова Е. Б., Ерзунов В. А., Рябов А. А., Вяткин Ю. А.* Пакет программ ЛОГОС. Модуль решения квазистатических задач прочности и модального анализа // Тр. XIII межд. семинара "Супервычисления и математическое моделирование". Саров: РФЯЦ-ВНИИЭФ, 2011. С. 493–502.

Rechkin V. N., Spiridonov V. F., Tsiberev K. V., Dyanov D. Yu., Naumov A. O., Kosarim S. S., Filimonkin E. A., Bartenev Yu. G., Shchanikova E. B., Erzunov V. A., Ryabov A. A., Vyatkin Yu. A. Paket program LOGOS. Modul resheniya kvazistaticheskikh zadach prochnosti i modalnogo analiza // Tr. XIII mezhd. seminar "Supervychisleniya i matematicheskoe modelirovanie". Sarov: RFYaTs-VNIIEF, 2011. S. 493—502.

12. *Zienkiewicz O. C., Taylor R. L.* The Finite Element Method. Volume 1: The Basis. Oxford: Butterworth-Heinemann, 2000.
13. *Шаров Г. А.* Векторное, матричное и тензорное исчисление. Долгопрудный: Интеллект, 2014.
Sharov G. A. Vektornoe, matrichnoe i tenzornoe ischislenie. Dolgoprudny: Intellekt, 2014.
14. *Zavarise G., Lorenzis L., Taylor R.* A non-consistent start-up procedure for contact problems with large load-steps // Comp. Meth. Appl. Mech. Eng. 2012. Vol. 205. P. 91—109.
15. *Johnson K. L.* Contact Mechanics. Cambridge University Press, 1985.

Статья поступила в редакцию 20.11.19.

"LOGOS" SOFTWARE PACKAGE. INCORPORATION OF CONTACT INTERACTIONS IN QUASI-STATIC DEFORMATION SIMULATIONS / A. Yu. Eryemenko, S. S. Kosarim (FSUE "RFNC-VNIIEF", Sarov, Nizhniy Novgorod region).

Details of incorporating contact interactions between different bodies or structural components in quasi-static deformation simulations by the "Logos" software package are discussed. As a basic method for this, a penalty method has been chosen. An algorithm is described, which detects interpenetration of bodies in contact. Basic formulas for contact forces and their contributions to the global stiffness matrix are given. A separate description is provided for the way to incorporate friction between bodies in contact. The accuracy of the implemented methods is illustrated by numerical simulations of the Hertz contact problem with friction between two infinitely long cylinders and by comparison of these simulations with the analytical solution.

Keywords: quasi-static deformation, incorporation of contact interactions, penalty method, penetration detection algorithm, incorporation of friction, Hertz contact problem, "Logos" software package.

УДК 004.94:004.8

ТЕСТОВАЯ ПРОГРАММА "ПАУК" КАК ПОЛИГОН ДЛЯ АПРОБАЦИИ АЛГОРИТМОВ И ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

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

В тестовой программе ПАУК с помощью разностного S_n -метода численно решается трехмерное одногрупповое уравнение переноса нейтронов на ортогональной пространственной сетке. Представлены результаты исследований по адаптации тестовой программы к гетерогенной вычислительной системе, содержащей сопроцессоры Intel Xeon Phi поколения Knights Corner. При этом апробированы различные алгоритмы и технологии параллельного программирования: векторизация циклов по направлениям и элементам гиперплоскостей, автовекторизация и intrinsic-программирование, явная и неявная предвыборки данных, реализации КВА-алгоритма распараллеливания в трех моделях памяти (общая, распределенная, PGAS). Эффективность адаптации программы подтверждается результатами исследований производительности в различных режимах счета на гетерогенной вычислительной системе: на универсальных процессорах (CPU-only), сопроцессорах (native) и в симметричном режиме (symmetric).

Ключевые слова: S_n -метод, алгоритм бегущего счета, КВА-алгоритм, Intel Xeon Phi, векторизация, предвыборка данных, MPI-3 SHM.

Введение

Программа ПАУК предназначена для тестирования производительности суперЭВМ применительно к решению задач переноса частиц разностными методами [1]. В программе численно решается трехмерное стационарное одногрупповое кинетическое уравнение переноса частиц, записанное в декартовой системе координат. Используются характерные для данного типа приложений численные методы и алгоритмы: аппроксимация на основе схемы типа DS_n -метода, алгоритм бегущего счета и метод итераций по источнику. Язык программирования — Fortran-90. Первоначально программа ПАУК была ориентирована на тестирование производительности однородных суперЭВМ методом увеличения задачи, или в режиме слабой масштабируемости. Для этого в ней реализован с помощью стандарта MPI алгоритм распараллеливания, который основан на трехмерной пространственной декомпозиции, — модификация КВА-алгоритма [2]. Эффективность распараллеливания является основной метрикой

для оценки производительности теста на однородных суперЭВМ. Программа ПАУК входит в набор методических прикладных тестов РФЯЦ-ВНИИЭФ [3] и является миниприложением для методики САТУРН [4].

Появление гетерогенных, или гибридных, суперЭВМ, содержащих в одном узле вычислительные устройства различной архитектуры (например, multi- и many-core устройства), потребовало разработки новых алгоритмов распараллеливания и внедрения новых техник параллельного программирования в интересах эффективного отображения приложения на такие архитектуры. В данном случае имеются в виду смешанное (MPI+X)-распараллеливание и дисциплина вычислений SIMD (или SIMT в случае GPU-устройств).

Важной особенностью тестовой программы ПАУК является ее компактность. Суммарный объем кода составляет ~ 2000 строк, при этом основное счетное ядро (решение уравнения баланса) составляет ~ 50 строк кода. Такая компактность является существенным достоинством

при решении задачи отображения приложения на новые архитектуры. В работе рассматривается архитектура Intel Many Integrated Core (MIC) на примере устройств Intel Xeon Phi поколения Knights Corner. Методология отображения на гетерогенную систему с Intel Xeon Phi основана на выделении иерархических уровней: ядро, кристалл, узел/множество узлов. Подходы к параллельному программированию взяты из работы [5]. Для простоты и удобства выбрана симметричная модель MPI-программирования.

Исследования выполнены на вычислительном кластере ООО "Центр компетенций и обучения", г.Саров [6], который содержит гетерогенный сегмент с пятью узлами: 2 CPU Intel Xeon E5-2680 v2 (22 нм, Ivy Bridge, 10 ядер, 2,8 ГГц, 0,224 Тфлопс) + 2 Intel Xeon Phi Coprocessor 7120P (22 нм, Knights Corner, 61 ядро, 1,24 ГГц, 1,21 Тфлопс). Коммуникационная система — infiniband FDR, 56 Гбит/с. Пиковая производительность одного гетерогенного узла — 2,868 Тфлопс.

Основной результат данной работы заключается в сравнительных исследованиях производительности на Intel Xeon Phi поколения Knights Corner различных алгоритмов и техник параллельного программирования для программы решения трехмерного уравнения переноса частиц на ортогональной пространственной сетке.

Аналогичные работы

Результаты применения схожих методологии и приемов адаптации высокопроизводительных приложений из других предметных областей к архитектуре Intel MIC — различных подходов к векторизации циклов, предвыборки данных, (MPI+X)-распараллеливания, применения технологии MPI-3 SHM — изложены в сборниках [7].

В рассматриваемой предметной области наиболее близки к программе ПАУК тестовые приложения (miniapps или проху) трех лабораторий Министерства энергетики США: LANL (Sweep3D и SNAP), LLNL (KRIPKE), ORNL (miniSweep). Во всех этих приложениях решается трехмерное уравнение переноса нейтронов методом дискретных ординат (S_n) на ортогональных пространственных сетках, используются аналогичные методы и алгоритмы, в том числе бегущий счет и КВА-алгоритм распараллеливания.

Sweep3D — это компактное приложение предыдущего поколения (сейчас заменено на SNAP), разработанное в рамках программы ASCI для тестирования производительности различных однородных суперЭВМ [8]. Тест Sweep3D применялся для апробации элементов модели PGAS за счет использования `coarray` в Fortran [9], а также для отображения на различные гетерогенные архитектуры: Cell Broadband Engine (Cell BE) [10], GPU [11], Intel MIC [12].

SNAP (SN (Discrete Ordinates) Application Proxy) — это прокси-приложение, специально предназначенное для тестирования гибридных или гетерогенных суперЭВМ [13]. Тест SNAP не решает задачу переноса нейтронов, он имитирует вычислительную нагрузку и алгоритмы большого приложения PARTISN [14]. Разработчики приложения ориентировались на сопроцессоры Intel Xeon Phi поколения Knights Landing [15]. Тест активно используется для апробации новых инструментов параллельного программирования [16] и отображения на различные архитектуры [17].

Тест KRIPKE создан специально для апробаций технологий программирования и различных алгоритмов отображения на архитектуру. Он является прокси-приложением для комплекса ARDRA [18].

MiniSweep создан с целью отображения на различные архитектуры и апробации различных моделей программирования [19]. Например, в работе [20] приводятся результаты распараллеливания miniSweep с помощью технологий CUDA, OpenMP и OpenACC, полученные на различных гетерогенных платформах, в том числе Intel Xeon Phi поколения Knights Landing. Тест является прокси-приложением кода DENOVO, который эффективно адаптирован к суперЭВМ с GPU Nvidia [21].

Постановка задачи

В тестовой программе ПАУК решается стационарное трехмерное уравнение переноса в одногрупповом кинетическом приближении в декартовой системе координат на ортогональных пространственных сетках [3]:

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

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:

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

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.

Особенности используемых алгоритмов

Численное решение системы сеточных уравнений осуществляется с помощью алгоритма бегущего счета. Это алгоритм расчета ячеек пространственной сетки в определенной последовательности, которая зависит от освещенности граней ячеек. Алгоритм бегущего счета можно представить 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 $e \in \{E\}$ — поток частиц через грань смежных ячеек. Направление дуги определяется освещенностью грани, по которой в ячейке определяется свойство потока на грани — входящий он или выходящий. Полный телесный угол (4π) можно разделить на 8 угловых октантов по вариантам освещенности граней ячеек. Орграф алгоритма бегущего счета будет одинаковым для всех направлений полета частиц из одного октанта и разным для любых направлений из разных октантов.

На рис. 1 схематично представлен орграф бегущего счета в одном двумерном слое ячеек. Программная реализация алгоритма бегущего счета в циклах по сеточным направлениям (на

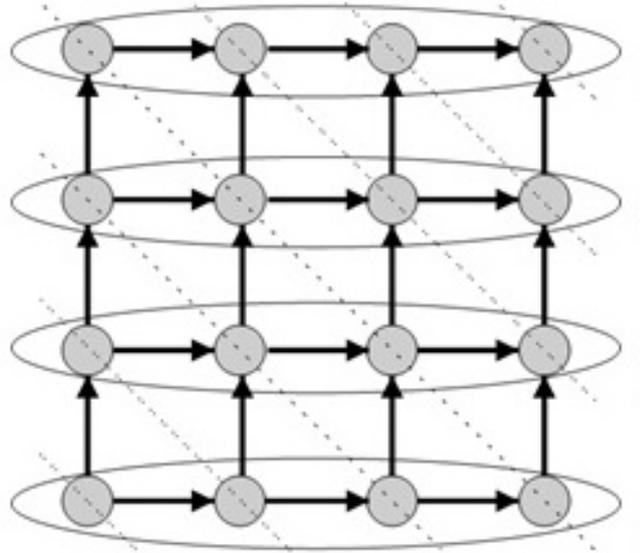


Рис. 1. Орграф $G = (V, E)$ алгоритма бегущего счета в двумерном слое

рис. 1 вершины, соответствующие одной строке расчетной сетки, выделены овалами) приводит к появлению зависимостей по данным, что не позволяет векторизовать такие циклы. С другой стороны, вершины графа можно объединить по признаку одинаковой максимальной длины пути от источника. На рис. 1 такие вершины выделены наклонными штриховыми линиями. Множество вершин с одинаковым расстоянием от источника образуют *гиперплоскость* [22], или *k-ломаную* [23]. Для вершин в одной гиперплоскости зависимостей по данным нет, что активно используется при MPI-распараллеливании алгоритма бегущего счета.

Первоначальный вариант распараллеливания тестовой программы ПАУК выполнен в модели распределенной памяти с помощью средств стандарта MPI. Алгоритм распараллеливания представляет собой модификацию КВА-алгоритма [2]. Это безытерационный алгоритм распараллеливания конвейерного типа, который основан на принципе геометрической декомпозиции. В случае регулярной пространственной декомпозиции алгоритм параллельного бегущего счета также имеет представление в виде орграфа (см. рис. 1). Независимость по данным у вершин в одной гиперплоскости приводит к свойству волнового параллелизма алгоритма. Зависимость по данным у вершин разных гиперплоскостей компенсируется за счет параллельных конвейеров.

Адаптация к ядру сопроцессора

Эффективная адаптация тестовой программы ПАУК к ядру сопроцессора Intel Xeon Phi поколения Knights Corner достигнута с помощью векторизации вычислений и программной предвыборки данных.

Векторизация вычислений. Дисциплина SIMD внедрена за счет согласованной модификации структуры данных и изменения последовательности вложенных циклов. Основным требованием векторизации цикла является отсутствие зависимостей по данным. Исходный вариант теста ПАУК такому требованию не удовлетворяет.

В работе рассмотрена векторизация двух циклов — по направлениям полета частиц в октанте и по элементам гиперплоскостей.

Векторизация по направлениям полета частиц. На ортогональной сетке бегущий счет для всех направлений в октанте n_w^{oct} представляется одним орграфом $G = (V, E)$. Это свойство позволяет сделать внутренним цикл по направлениям в октанте, тогда каждой вершине $v \in \{V\}$ соответствует решение уравнения баланса в n_w^{oct} ячейках фазового пространства. Единственной зависимостью по данным является операция редукции при вычислении интеграла по направлениям в центральной точке разностного шаблона: $\bar{n}^{(0)} = \sum_w N_0^w d\Omega_w$. Однако данная операция довольно эффективно векторизуется компилятором. Кроме того, часть данных, используемых при решении уравнения баланса, не зависит от направления, что уменьшает суммарную нагрузку на подсистему памяти.

В отличие от других схожих приложений, где используется векторизация по направлениям [13, 14, 24], в тесте ПАУК внутренний цикл организуется для порции из m направлений, а не для всех направлений в октанте. Размер порции определяется по формуле $m = \langle \text{число VPU-модулей в ядре} \rangle \times \langle \text{ширина векторного регистра в числах двойной точности} \rangle$. Для обоих рассматриваемых вычислителей (Ivy Bridge и Xeon Phi) этот размер равен 8. Такой подход позволяет использовать множество направлений полета частиц в октанте не только для векторизации вычислений, но и для различных алгоритмов распараллеливания: OpenMP и MPI.

Модификация структуры данных требуется только для массивов с зависимостью от направлений. Первое измерение таких массивов определяется числом направлений в порции, что приводит к их форме $(m, \dots, n_w^{oct}/m)$.

В аналогичных приложениях применяются различные подходы для векторизации цикла по направлениям: автовекторизация средствами компилятора [13], intrinsic-программирование [12], векторные типы данных библиотеки Eigen [24]. На тесте ПАУК были проведены сравнительные исследования производительности автовекторизации и intrinsic-программирования. Первоначальный, или *формальный*, вариант автовекторизации получен простым переносом цикла по m направлениям без каких-либо дополнительных модификаций в уравнении баланса. На Intel Xeon Phi коэффициент ускорения от векторизации составил

$Sp_{vec} = t(-no - vec)^*/t = 2,3$ при идеальном значении 8. Причины такого низкого результата заключаются в следующем:

- отсутствие информации у компилятора о выровненности данных и общем числе итераций векторизуемого цикла;
- вычислительная неоднородность итераций векторизуемого цикла.

При решении уравнения переноса используются немонотонные схемы второго порядка (например, DD-схема по пространству), которые могут приводить к появлению нефизических отрицательных решений на разностном шаблоне (например, выходящие потоки на гранях ячейки). Для борьбы с этим явлением в тесте ПАУК применяется алгоритм балансного зануления [25], который, однако, приводит к вычислительной неоднородности при решении уравнения баланса.

Возникающая проблема решается следующим образом. Коэффициент балансного зануления k_b рассчитывается безусловно, но применяется ко всем неизвестным в ячейке по маске

$$N_i = kN_i, \forall i \in \{0, \{i^{BYX}\}\}.$$

Здесь i^{BYX} — номера граней ячейки с выходящим потоком; $k = \begin{cases} k_b, & \text{если } \min\{N_i^{BYX}\} < 0; \\ 1, & \text{если } \min\{N_i^{BYX}\} \geq 0. \end{cases}$

После модификации алгоритма балансного зануления были реализованы еще два варианта векторизации цикла по восьми направлениям. *Предопределенный* вариант автовекторизации с точным указанием компилятору числа итераций в цикле ($m = 8$) и информации о выровненности данных обеспечил ускорение $Sp_{vec} = 6,1$. *Intrinsic*-вариант, реализованный с помощью intrinsic-функций, обеспечил значение $Sp_{vec} = 5,6$. Полученные результаты показывают, что программирование с использованием intrinsic-функций не всегда обеспечивает наилучшую производительность.

В табл. 1 приведены результаты ускорения выполнения процедуры бегущего счета в первом октанте для предопределенного варианта автовекторизации на CPU и сопроцессоре в последовательном режиме. Здесь Sp_{vec}^{ideal} — идеальный коэффициент ускорения от векторизации, определяемый шириной векторных регистров. В по-

Коэффициенты ускорения счета для векторизации по восьми направлениям

Архитектура	Sp_{vec}^{ideal}	Sp_{vec}	Sp_{opt}
Ivy Bridge	4	2,4	4,9
Knights Corner	8	6,1	8,2

следнем столбце приведены ускорения от оптимизации кода (Sp_{opt}), т. е. отношение времени счета двух версий теста ПАУК: исходной и векторизованной.

Более высокое значение $Sp_{opt}/Sp_{vec}^{ideal}$ на CPU, чем на сопроцессоре, вероятнее всего, объясняется улучшением параллелизма на уровне инструкций (ILP) за счет дополнительной возможности разворачивания (unrolling) векторизуемого цикла.

Векторизация по элементам гиперплоскостей. Переход от циклов по сеточным направлениям к циклам по гиперплоскостям и элементам в них не является алгоритмически сложным. Данный прием подробно описан в [22]. Для эффективной векторизации на Intel Xeon Phi сложность представляют плавающее число итераций векторизуемого цикла (от 1 до $\min(n_x, n_y)$, где n_x, n_y — соответственно число столбцов и строк) и формат представления данных, не обеспечивающий компактного размещения в памяти элементов одной гиперплоскости. Для решения перечисленных проблем в тестовой программе ПАУК были реализованы алгоритм выделения гиперплоскостей фиксированного размера и запись данных в специальном формате.

Идея заключается в переходе от представления по столбцам и строкам к представлению по столбцам и гиперплоскостям, т. е. от формы массивов (n_x, n_y, n_z) к форме (n_x, n_{hyp}, n_z) , где $n_{hyp} = n_x + n_y - 1$. Для случая на рис. 1 алгоритм бегущего счета преобразуется к виду, представленному на рис. 2 слева. В новом орграфе первое сеточное направление содержит элементы одной гиперплоскости и зависимостей по данным не имеет. Множество вершин нового графа состоит из объединения двух множеств: счетного $\{V\}$ (на рис. 2 вершины серого цвета) и фиктивного $\{V_f\}$ (вершины белого цвета). Счетные вершины соответствуют ячейкам сетки в пространственной области L , фиктивные — нет. Решение уравнения баланса выполняется либо в вершинах обоих типов $\{V\} \cup \{V_f\}$, либо

* Здесь и далее запись $t(\dots)$ означает время счета программы, скомпилированной с дополнительными ключами, указанными в скобках.

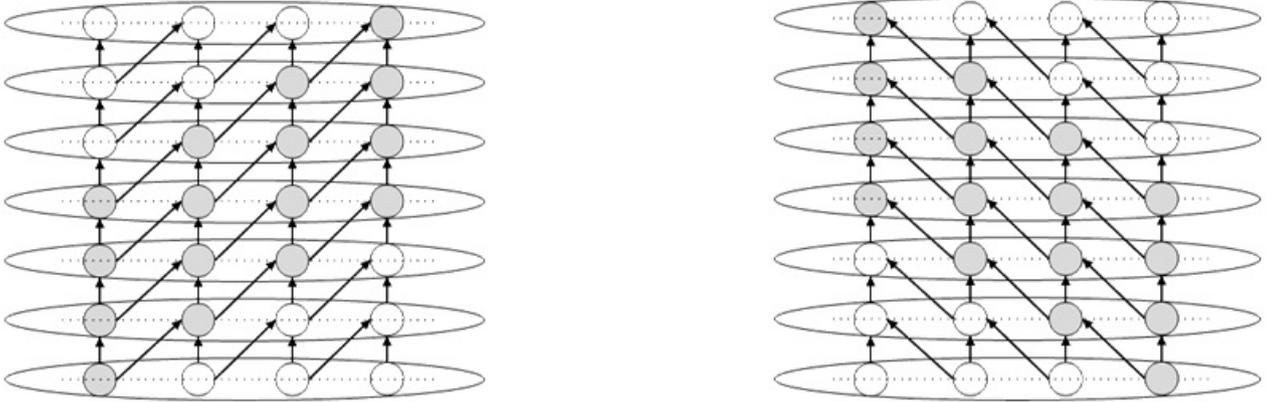


Рис. 2. Орграфы для двух вариантов бегущего счета на новых форматах данных

только в счетных $\{V\}$. В первом случае 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 $v \in \begin{cases} \{V\}, & \text{если } i_y \in [1, n_y]; \\ \{V_f\}, & \text{если } i_y \notin [1, n_y]. \end{cases}$ Взаимосвязь

между индексами строк и гиперплоскостей для орграфа на рис. 2, слева определяется выражением $i_y = i_{hyp} - i_x + 1$; для орграфа на рис. 2, справа $i_y = i_{hyp} - n_x + i_x$, где i_x, i_y, i_{hyp} — индексы столбцов, строк и гиперплоскостей соответственно. Данный подход позволяет решить проблему плавающего числа итераций векторизуемого цикла по элементам гиперплоскости. При выполнении требования кратности 8 для числа столбцов сетки данный цикл эффективно векторизуется.

Требование компактного размещения данных для каждой гиперплоскости приводит к необходимости дублирования всех сеточных данных, которые не зависят от направлений. Для двумерного слоя ячеек возможны четыре варианта алгоритма бегущего счета и четыре орграфа. Для каждого из этих орграфов есть симметричная пара, которая получается сменой направления во всех дугах орграфа. При такой смене направления множества счетных вершин в гиперплоскостях не меняются, поэтому для каждой пары симметричных орграфов достаточно одного формата записи данных.

Новый формат предполагает переход от массива A с формой (n_x, n_y, n_z) к двум массивам A_1 и A_2 с формой (n_x, n_{hyp}, n_z) по следующему правилу:

$$A_1(i_x, i_{hyp}) = \begin{cases} A(i_x, i_{hyp} - i_x + 1), & v \in \{V\}; \\ A(1, 1), & v \in \{V_f\}; \end{cases}$$

$$A_2(i_x, i_{hyp}) = \begin{cases} A(i_x, i_{hyp} - n_x + i_x), & v \in \{V\}; \\ A(1, 1), & v \in \{V_f\}. \end{cases}$$

При выполнении требования кратности 8 для числа столбцов сетки новый формат данных обеспечивает выровненный доступ к памяти.

Очевидным недостатком предлагаемого подхода является увеличение размеров всех массивов, особенно массивов без зависимости от направлений полета частиц. Данный недостаток можно компенсировать различными вариантами усложнения формата по гиперплоскостям для минимизации $|V_f|$. Другим недостатком является увеличение суммарной вычислительной работы за счет решения уравнения баланса в фиктивных ячейках.

Главным достоинством векторизации по элементам гиперплоскостей является возможность полного сохранения ресурса направлений для различных алгоритмов распараллеливания на уровне кристалла, узла и множества узлов. Последнее может быть ценным для задач с небольшим числом направлений.

Результаты исследования ускорения вычислений от векторизации по элементам гиперплоскостей представлены в табл. 2. Здесь приведены среднеарифметические значения ускорений, полученные на сетках с разным числом столбцов (8, 16, 32, 64) при числе строк и слоев, равном соответственно 128 и 4. Для более точной оценки ускорения от оптимизации в таблице приводится метрика, учитывающая разное число счетных

Значения коэффициентов ускорения счета для двух вариантов векторизации по элементам гиперплоскостей

Архитектура	$\{V\} \cup \{V_f\}$			$\{V\}$		
	Sp_{vec}	Sp_{opt}	Sp_{opt}^*	Sp_{vec}	Sp_{opt}	Sp_{opt}^*
Ivy Bridge	2,3	2,9	3,5	1,8	3,1	3,8
Knights Corner	5,6	5,0	6,1	5,3	5,2	6,4

вершин при счете по исходной и модифицированной версиям кода: $Sp_{opt}^* = \frac{t_{old} n_{hyp}}{t_{new} n_y}$.

Из результатов табл. 1, 2 следует, что с учетом различного числа вершин в графах вариант векторизации по восьми направлениям производительнее варианта векторизации по элементам гиперплоскостей в 1,39 раза на Ivy Bridge и в 1,34 раза на Knights Corner.

В итоговый вариант тестовой программы ПАУК вошел алгоритм векторизации цикла по направлениям с предопределенным числом итераций ($m = 8$) и автовекторизацией средствами компилятора языка Fortran.

Предвыборка данных. Из-за определенных аппаратных ограничений на Intel Xeon Phi эффективную предвыборку данных в кэш-память первого и второго уровня можно реализовать только программным путем [5]. Компилятор достаточно эффективно справляется с задачей программной предвыборки за счет вставки специальных инструкций на этапе компиляции, особенно при обращении к данным с единичным смещением в памяти. В алгоритме бегущего счета такой шаблон доступа (с единичным смещением) всегда реализуется только для двух октантов из восьми. Для остальных шести октантов в бегущем счете иногда требуются смещения в $\sim n_x$ или $\sim n_x n_y$ ячеек. Частота таких неединичных смещений, или "прыжков" по памяти — $1/n_x$ и $1/(n_x n_y)$. Влияние этих прыжков на общую производительность приложения оценено по метрике $\max_{i=1,8} \{t_i\} / \min_{i=1,8} \{t_i\}$, где t_i — время бегущего счета по всем направлениям в i -м октанте. На сопроцессоре Intel Xeon Phi данная метрика имеет значение 1,47 для варианта векторизации по восьми направлениям.

Для решения указанной проблемы были реализованы два варианта предвыборки: неявный и явный. Неявная предвыборка основана на такой модификации формата представления данных,

при которой для всех вариантов бегущего счета обеспечивается шаблон доступа к памяти с единичным смещением. Последовательность размещения данных полностью соответствует последовательности расчета циклов. Это потребовало четырехкратного увеличения всех массивов, инвариантных от направлений полета частиц. Явный вариант предвыборки основан на использовании единственной intrinsic-функции языка Fortran — `mm_prefetch`. В этом случае компилятору запрещается предвыборка данных, а все данные явно загружаются в L1/L2-кэши.

Из данных, приведенных в табл. 3, видно, что для обоих вариантов предвыборки время выполнения процедуры бегущего счета выравнивалось для всех октантов, а коэффициент ускорения от программной предвыборки составил $\sim 1,38$.

В итоговый вариант тестовой программы ПАУК вошел неявный вариант предвыборки данных, так как он обеспечил на 7% лучшую производительность по сравнению с явным вариантом.

Таблица 3

Результаты исследований производительности вариантов предвыборки данных в последовательном режиме на Knights Corner

Предвыборка	$\max\{t_i\} / \min\{t_i\}$	$t(-no - prefetch) / t$
Неявная	1,03	1,38
Явная	1,00	1,39

Адаптация к сопроцессору

Эффективная адаптация тестовой программы ПАУК к одному сопроцессору Intel Xeon Phi поколения Knights Corner достигнута за счет внедрения (MPI+X)-распараллеливания: по направлениям полета частиц и по геометрическим фрагментам (КВА).

Распараллеливание по направлениям.

Алгоритм распараллеливания по направлениям полета частиц одного октанта реализован в модели общей памяти с помощью средств стандарта OpenMP. Это внутренний алгоритм распараллеливания относительно КВА. Параллельный бегущий счет организуется не для всех, а только для $8N_{OMP}$ направлений одного октанта, где N_{OMP} — число OpenMP-потоков в одном MPI-процессе. Расчет интеграла $n^{(0)}$ по всем направлениям осуществляется через локальный промежуточный массив, что позволяет избежать обмена данными между потоками непосредственно в ходе параллельного бегущего счета. MPI-обмены выполняются только одним OpenMP-потоком. Зернистость, или *гранулярность*, такая же, как у алгоритма распараллеливания по геометрическим фрагментам (см. далее).

Распараллеливание по геометрическим фрагментам. В исходном варианте тестовой программы ПАУК КВА-алгоритм распараллеливания реализован в модели распределенной памяти с помощью средств стандарта MPI. Исследования эффективности распараллеливания на одном сопроцессоре показали значительные накладные расходы на MPI-обмены. Поэтому в рамках теста ПАУК были выполнены еще две реализации КВА-алгоритма — в моделях общей памяти и разделенного глобального адресного пространства (PGAS). Все реализации выполнены при условии, что единица вычислительного ресурса рассчитывает один геометрический фрагмент.

КВА-алгоритм в модели общей памяти реализован с помощью стандарта OpenMP. Для передачи потоков между фрагментами используются глобальные массивы, размер которых зависит от числа дуг в орграфе параллельного бегущего счета. Для улучшения локальности данных используется принцип "*запись — в свою память, чтение — из чужой памяти*".

Для организации параллельного бегущего счета задействован механизм OpenMP-замков. Комбинация с OpenMP-распараллеливанием по направлениям выполнена по вложенной схеме: перед чтением данных выполняется синхронизация между головными OpenMP-потоками соседних фрагментов, затем синхронизация между всеми OpenMP-потоками для текущего фрагмента; после записи данных последовательность синхронизаций обратная.

КВА-алгоритм в модели разделенного глобального адресного пространства (PGAS) реализован с помощью MPI-3 SHM [26]. Коммуникатор, *окно, эпоха* создаются для каждой дуги орграфа параллельного бегущего счета в отдельности. Для каждой дуги орграфа создаются окно для передачи данных и окно для передачи управления. Проблема синхронизации данных, размещенных в общем окне, решена с помощью дополнительного атрибута `asynchronous` стандарта Fortran-2003. Этот атрибут использован при описании всех массивов-указателей, выделенных или назначенных с помощью средств MPI-3 SHM.

В табл. 4 приводятся результаты исследований эффективности распараллеливания для трех реализаций КВА-алгоритма на одном сопроцессоре Knights Corner. Режим запусков: двумерная декомпозиция, 30 геометрических фрагментов, 4 OpenMP-потока по направлениям, 120 параллельных процессов суммарно, по 2 процесса на одно ядро сопроцессора. Эффективность распараллеливания получена методом слабой масштабируемости, или увеличения задачи. В случае одного фрагмента задача посчитана с 4 OpenMP-потоками по направлениям и явной привязкой по 2 процесса на одно ядро.

Результаты табл. 4 демонстрируют значительные накладные расходы библиотеки MPI в случае двухточечных обменов внутри одного сопроцессора. Эффективность распараллелива-

Таблица 4

Эффективность распараллеливания (в %) в зависимости от гранулярности

Тип распараллеливания	Гранулярность			
	960	1920	3968	7168
MPI+OpenMP	62	68	77	81
OpenMP+OpenMP	87	88	88	90
MPI-3 SHM+OpenMP	84	88	90	90

ния для реализаций в моделях общей памяти и PGAS имеет близкие значения и слабо зависит от объема вычислительной работы между обменами данных.

В итоговый вариант тестовой программы ПАУК вошла реализация КВА-алгоритма, выполненная в PGAS-модели. Реализация в модели общей памяти обладает существенным недостатком в части производительности MPI-обменов между различными устройствами/узлами, так как в стандарт MPI пока не входит функционал multi-EndPoints [27].

Адаптация к гетерогенному узлу

Для упрощения задачи отображения теста ПАУК на гетерогенную архитектуру был изменен исходный алгоритм MPI-распараллеливания. Пространственная 3D-декомпозиция заменена на 2D-декомпозицию (по столбцам и строкам), что упростило коммуникационный шаблон и снизило общую нагрузку на коммуникационную подсистему. Исключен дополнительный MPI-конвейер по слоям пространственной сетки, однако его работу можно симитировать увеличением числа направлений. Режим старта MPI-конвейеров заменен с последовательного на одновременный, что повысило значение теоретической эффективности MPI-распараллеливания. Стратегия обработки множества одновременных конвейеров — самоупорядочение.

Для учета особенностей гетерогенной архитектуры в тест ПАУК внедрены возможность балансировки вычислительной нагрузки и процедура **mapping** отображения графа задачи на архитектуру с учетом некоторых особенностей последней.

Балансировка вычислительной нагрузки реализована статическим образом за счет задания разного числа строк в геометрических фрагментах на вычислителях двух типов.

Процедура **mapping** выполняется в два этапа.

На первом этапе множество всех MPI-процессов переупорядочивается по типу вычислителя: сначала следуют все MPI на процессорах, а затем — на сопроцессорах. Общее число геометрических фрагментов (MPI-процессов) в запуске определяется по формуле

$$P = P_x(P_y^{CPU} + P_y^{MIC}),$$

где P_x — число разбиений по столбцам; P_y^{CPU} , P_y^{MIC} — число разбиений по строкам соответственно на процессорах и сопроцессорах.

На втором этапе множество всех MPI-процессов переупорядочивается по принадлежности устройств одному гетерогенному узлу, чтобы минимизировать число внешних дуг в орграфе параллельного бегущего счета. Внешней дуге в данном случае соответствует передача данных, выполняемая через коммуникационную подсистему гетерогенной суперЭВМ. Общее число дуг в орграфе бегущего счета рассчитывается по формуле

$$|E| = P_x(P_y - 1) + P_y(P_x - 1),$$

а число внешних дуг для симметричного режима — по формуле

$$|E_{ext}| = P_x + \sum_{dev} \left[P_x \left(\frac{P_y^{dev}}{m_y^{dev}} - 1 \right) + P_y^{dev} \left(\frac{P_x}{m_x} - 1 \right) \right].$$

Здесь m_x — новый параметр теста, позволяющий определить правило локального переупорядочения; $m_y^{dev} = MPI_{node}^{dev}/m_x$, где MPI_{node}^{dev} — число MPI-процессов одного типа на одном узле гетерогенной суперЭВМ; индекс *dev* обозначает тип вычислителя и принимает значение CPU или MIC.

Минимизация числа внешних дуг в орграфе позволяет снизить нагрузку на коммуникационную подсистему за счет выполнения большей части обменов через эффективный механизм MPI-3 SHM, а не через коммуникационное оборудование.

Результаты исследований производительности

При тестировании использовались следующие параметры дискретизации задачи: на один вычислительный узел — 300 столбцов, 480 строк, 4 слоя пространственной сетки; на четыре узла — 600 столбцов, 960 строк, 4 слоя. Число направлений в задаче $n_w = 18\,432$ позволяет имитировать MPI-конвейеры по другим переменным фазового пространства.

Процедура тестирования заключалась в запуске задачи на одинаковом числе вычислительных узлов в трех режимах: на процессорах (CPU-only), на сопроцессорах (native) и в симметричном режиме (symmetric).

Производительность на процессорах. Основной метрикой в режиме счета на процессорах является эффективность распараллеливания $E_n = t_1/t_n \cdot 100\%$. В расчетах в данном режиме использовалось только MPI-распараллеливание. Для варианта КВА-алгоритма, реализованного в новой версии программы ПАУК, значение теоретической эффективности распараллеливания (для идеальной суперЭВМ без шумов и с мгновенными обменами) оценивается по формуле

$$E_{theor} = \frac{n_w / (8N_{OMP}) \cdot 100\%}{n_w / (8N_{OMP}) + (P_x - 1) + (P_y - 1)}$$

Значение E_{theor} выступает в качестве критерия качества. В табл. 5 приведены значения двух эффективностей распараллеливания в запусках на одном и четырех вычислительных узлах.

По метрике E_n получены высокие показатели на одном узле — 97%, что объясняется эффективностью обменов через MPI-3 SHM. Некоторое снижение эффективности на четырех узлах обусловлено влиянием коммуникационного оборудования, так как значение E_{theor} слабо изменилось при переходе от одного к четырем узлам.

Таблица 5

Эффективность распараллеливания на процессорах

Число узлов	$E_{theor}, \%$	$E_n, \%$
1	99,7	97,2
4	99,3	91,3

Производительность на сопроцессорах.

Основной метрикой в режиме счета на сопроцессорах является ускорение счета: $Sp_{native} = t_{CPU}/t_{MIC}$. Критерий качества рассчитывается по пиковой производительности: $Sp_{native}^{ideal} = R_{peak}^{MIC}/R_{peak}^{CPU} = 5,4$.

В табл. 6 представлены результаты фактически достигнутых ускорений на одном и четырех вычислительных узлах при варьировании как числа MPI-процессов на каждом сопроцессоре (M_{MPI}), так и числа OpenMP-потоков на одном ядре (M_{OMP}). Число потоков в MPI-процессе (N_{OMP}) выбиралось так, чтобы обеспечить нагрузку в 2, 3 и 4 процесса на одно ядро сопроцессора.

Наилучшее ускорение достигает всего 3 раз при идеальном значении в 5,4 раза. Причина

такого отставания заключается в FMA (Fused Multiply-Add) инструкциях, относительно которых рассчитан пик сопроцессоров. Исследуемый CPU Intel Ivy Bridge такие инструкции выполнять не может. FMA-инструкции оказывают достаточно небольшое влияние на производительность теста ПАУК: $Sp_{fma} = t(-no - fma)/t = 1,2$, что позволяет скорректировать критерий качества: $\bar{Sp}_{native}^{ideal} = 0,5Sp_{fma}Sp_{native}^{ideal} = 3,2$. По новому критерию результаты табл. 6 позволяют утверждать об эффективной адаптации теста к сопроцессорам.

Таблица 6

Коэффициент ускорения счета на сопроцессорах

Число узлов	M_{OMP}	M_{MPI}		
		20	30	60
1	2	2,7	2,7	2,6
	3	3,0	2,9	3,0
	4	3,0	3,0	2,9
4	2	2,6	2,6	2,4
	3	2,8	2,8	2,6
	4	2,9	2,8	2,6

Производительность в симметричном режиме.

Основной метрикой в симметричном режиме является ускорение счета $Sp_{sym} = t_{CPU}/t_{CPU+MIC}$. Критерий качества $Sp_{sym}^{ideal} = B + 1$, где $B = \frac{P_y^{MIC} n_y^{MIC}}{P_y^{CPU} n_y^{CPU}}$ — коэффициент статической балансировки вычислительной нагрузки. Здесь n_y^{CPU} , n_y^{MIC} — число строк в одном геометрическом фрагменте на процессорах и сопроцессорах соответственно. Коэффициент балансировки определен по лучшему результату ускорения в режиме на сопроцессорах $Sp_{native} \equiv B = 3$, что означает $Sp_{sym}^{ideal} = 4$.

В табл. 7 представлены результаты фактически достигнутых ускорений на одном и четырех вычислительных узлах при варьировании как числа MPI-процессов на каждом сопроцессоре (M_{MPI}), так и числа OpenMP-потоков на одном ядре (M_{OMP}). Число потоков в MPI-процессе (N_{OMP}) выбиралось так, чтобы обеспечить нагрузку в 2, 3 и 4 процесса на одно ядро сопроцессора. На процессорах $N_{OMP} = 1$.

Результаты на одном узле близки к идеальному значению, на четырех узлах наблюдается некоторое уменьшение коэффициента уско-

Таблица 7

Коэффициент ускорения счета в симметричном режиме

Число узлов	M_{OMP}	M_{MPI}		
		20	30	60
1	2	3,6	3,5	3,3
	3	3,8	3,8	3,8
	4	3,9	3,7	3,8
4	2	3,1	3,0	3,0
	3	3,6	3,2	3,5
	4	3,6	3,1	3,5

рения. Данный эффект объясняется снижением теоретической эффективности распараллеливания в расчетах с использованием сопроцессоров. Число счетных тактов в MPI-конвейерах обратно пропорционально максимальному по устройствам числу OpenMP-потоков, а в режиме CPU-only $N_{OMP} = 1$.

Заключение

Тестовая программа ПАУК, изначально разработанная для тестирования производительности однородных суперЭВМ, за счет своей компактности является удобным полигоном для апробации различных алгоритмов и технологий параллельного программирования в интересах освоения гетерогенных архитектур.

На примере адаптации теста ПАУК к сопроцессорам Intel Xeon Phi поколения Knights Corner в работе представлены сравнения производительностей:

- двух алгоритмов векторизации циклов (по направлениям и элементам гиперплоскости);
- двух способов векторизации циклов (автовекторизация и intrinsic-программирование);
- двух способов предвыборки данных (возможности компилятора и intrinsic-функция);
- трех реализаций КВА-алгоритма на одном сопроцессоре в различных моделях памяти (общей, распределенной, PGAS).

Исходная версия программы ПАУК с алгоритмом только MPI-распараллеливания и неавтовекторизованным счетным ядром при адаптации к архитектуре Intel MIC была трансформирована в итоговый вариант с алгоритмом (MPI + MPI-3

SHM + OpenMP)-распараллеливания. Векторизация выполнена для цикла по восьми направлениям одного октанта за счет автовекторизации. Эффективность предвыборки данных средствами компилятора достигнута с помощью специального формата данных, обеспечившего шаблон доступа к памяти с единичным смещением для всех вариантов алгоритма бегущего счета.

Результаты тестирования на процессорах, сопроцессорах и в симметричном режиме продемонстрировали эффективность адаптации программы ПАУК к гетерогенной архитектуре с Intel Xeon Phi.

Список литературы

1. Бочков А. И., Нуждин А. А. Параллельный алгоритм решения трехмерного кинетического уравнения переноса. Программа ПАУК для тестирования многопроцессорных вычислительных систем // Параллельные вычислительные технологии (ПАВТ'2008): Тр. межд. науч. конф. (С.-Пб., 28 января — 1 февраля 2008 г.) Челябинск: Изд-во ЮУрГУ, 2008.
Bochkov A. I., Nuzhdin A. A. Parallelny algoritm resheniya trekhmernogo kineticheskogo uravneniya perenosa. Programma PAUK dlya testirovaniya mnogoprotsessornykh vychislitelnykh sistem // Parallelnye vychislitelnye tekhnologii (PAVT'2008): Tr. mezhd. nauch. konf. (S.-Pb., 28 yanvarya — 1 fevralya 2008). Chelyabinsk: Izd-vo YuUrGU, 2008.
2. 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.
3. Алексеев А. В., Беляев С. П., Бочков А. И., Быков А. Н., Ветчинников М. В., Залялов А. Н., Нуждин А. А., Огнев С. П., Самсонова Н. С., Сапронов И. С., Чистякова И. Н., Шемякина Т. В., Шагалиев Р. М., Ямилкин 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. S., Chistyakova I. N., Shemyakina T. V., Shagaliev R. M., Yanilkin 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.
4. *Alekseev A. V., Belyakov I. M., Bochkov A. I., Evdokimov V. V., Irinichev E. A., Morozov V. Yu., Moskvин A. H., Nuzhdin A. A., Pepelyaev M. P., Rezchikov V. Yu., Suchkova V. V., Shagaliev R. M., Sharifulin E. Sh., Shemyakina T. V., Shumilin V. A.* Методика SATURN-2005. Математические модели, алгоритмы и программы решения многомерных задач переноса частиц и энергии // *Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов.* 2013. Вып. 4. С. 17–30.
Alekseev A. V., Belyakov I. M., Bochkov A. I., Evdokimov V. V., Irinichev E. A., Morozov V. Yu., Moskvин A. N., Nuzhdin A. A., Pepelyaev M. P., Rezchikov V. Yu., Suchkova V. V., Shagaliev R. M., Sharifulin E. Sh., Shemyakina T. V., Shumilin V. A. Metodika SATURN-2005. Matematicheskie modeli, algoritmy i programmy resheniya mnogomernykh zadach perenosa chastits i energii // *Tam zhe.* 2013. Vyp. 4. S. 3–16.
 5. *Jeffers J., Reinders J.* Intel Xeon Phi Coprocessor High Performance Programming. USA: Morgan Kaufmann, 2013.
 6. Информация о вычислительном кластере ООО "Центр компетенций и обучения". <https://compcenter.org/predostavlenie-vychislitelnyh-resur>.
Informatsiya o vychislitelnom klasterе ООО "Tsentr kompetentsiy i obucheniya". <https://compcenter.org/predostavlenie-vychislitelnyh-resur>.
 7. *Reinders J., Jeffers J.* High-Performance Parallelism Pearls. Volume 1 and Volume 2: Multicore and Many-core Programming Approaches. USA: Morgan Kaufmann, 2015.
 8. *Hoisie A., Johnson G., Kerbyson D. J., Lang M., Pakin S.* A Performance comparison through benchmarking and modeling of three leading supercomputers: Blue Gene/L, Red Storm, and Purple // *SC'06: Proc. 2006 ACM/IEEE Conf. on Supercomputing.* Tampa, FL, 2006. P. 3-3.
DOI: 10.1109/SC.2006.4.
 9. *Coarfa C., Dotsenko Y., Mellor-Crummey J.* Experiences with Sweep3D implementations in co-array Fortran // *J. Supercomp.* 2006. Vol. 36 (2). P. 101–121.
DOI: 10.1007/s11227-006-7952-7.
 10. *Petrini F., Fossum G., Fernandez J., Varbanescu A. L., Kistler M., Perrone M.* Multicore surprises: Lessons learned from optimizing Sweep3D on the cell broadband engine // *2007 IEEE Int. Parallel and Distributed Processing Symposium.* Rome, Italy. March 26–30, 2007. P. 1–10.
DOI: 10.1109/IPDPS.2007.370252.
 11. *Gong C., Liu J., Gong Z., Qin J., Xie J.* Optimizing Sweep3D for Graphic Processor Unit. Algorithms and Architectures for Parallel Processing. ICA3PP 2010. Lecture Notes in Computer Science / Ed. by C. H. Hsu, L. T. Yang, J. H. Park, S. S. Yeo. Vol. 6081. Berlin: Heidelberg Springer, 2010.
 12. *Wang Q., Xing Z., Liu J., Qiang X., Gong C., Jiang J.* Parallel 3D deterministic particle transport on Intel MIC architecture // *2014 Int. Conf. on High Performance Computing & Simulation (HPCS).* Bologna, 2014. P. 186–192. DOI: 10.1109/HPCSim.2014.6903685.
 13. Информация о SNAP. <https://github.com/lanl/SNAP>.
Informatsiya o SNAP. <https://github.com/lanl/SNAP>.
 14. *Baker, R.* An SN Algorithm for modern architectures // *Nucl. Sci. Eng.* 2017. Vol. 185, No 1. P. 107–116.
DOI: 10.13182/NSE15-124.
 15. *Rajan M., Doerfler D., Hammond S.* Trinity Benchmarks on Intel Xeon Phi (Knights Corner). SAND2015-0454. Report. Sandia National Laboratories, 2015.
 16. *Amer A., Iwasaki S., Raffenetti K., Shiryaev M., Si M., Taura K., Thapaliya S., Balaji P., Archer C., Blocksome M., Cao C., Chuvelev M., Fujita H., Garzaran M., Guo Y., Hammond J.* Software combining to mitigate multithreaded MPI contention // *ICS'19:*

- Proc. ACM Int. Conf. Supercomputing. NY, USA: Association for Computing Machinery, 2019. P. 367–379.
DOI: 10.1145/3330345.3330378.
17. *Deakin T., McIntosh-Smith S., Gaudin W.* Many-core acceleration of a discrete ordinates transport mini-app at extreme scale // High Performance Computing: Proc. 31st Int. Conf. Frankfurt, Germany: Springer International Publishing, Cham, 2016. P. 429–448.
 18. *Kunen A. J., Bailey T. S., Brown P. N.* Kripke — A massively parallel transport mini-app // ANS MC2015 — Joint Int. Conf. on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method. Nashville, Tennessee. April 19–23, 2015.
 19. *Messer B., D’Azevedo E., Hill J., Joubert W., Berrill M., Zimmer C.* Miniapps derived from production hpc applications using multiple programming models // Int. J. High Performance Comp. Appl. 2018. Vol. 32, Issue 4. P. 582–593.
DOI:10.1177/1094342016668241.
 20. *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.
 21. *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.
 22. *Lampert L.* The parallel execution of DO loops // Communications of the ACM. 1974. Vol. 17, No 2. P. 83–93.
 23. *Трошчійев В. Е.* О классах сеток, допускающих консервативные аппроксимации двумерного оператора переноса треугольным разностным оператором // Журнал вычисл. мат. и мат. физ. 1976. Т. 16, № 3. С. 793–797.
Troshchiyev V. E. O klassakh setok, dopuskayushchikh konservativnye approksimatsii dvumernogo operatora perenosa treugolnym raznostnym operatorom // Zhurnal vychisl. mat. i mat. fiz. 1976. T. 16, № 3. S. 793–797.
 24. *Moustafa S., Fevotte F., Faverge M., Plagne L., Ramet P.* Efficient parallel solution of the 3D stationary Boltzmann transport equation for diffusive problems // J. Comp. Phys. 2019. Vol. 388. P. 335–349.
 25. *Елесин В. А., Трошчійев В. Е., Федянин В. И., Юдинцев В. Ф.* Численная методика и организация программы для решения многогруппового нестационарного кинетического уравнения // Комплексы программ математической физики. Новосибирск: ВЦ СО АН СССР, 1972. С. 18–23.
Elesin V. A., Troshchiyev V. E., Fedyanin V. I., Yudinsev V. F. Chislennaya metodika i organizatsiya programmy dlya resheniya mnogogruppovogo nestatsionarnogo kineticheskogo uravneniya // Kompleksy programm matematicheskoy fiziki. Novosibirsk: VTs SO AN SSSR, 1972. S. 18–23.
 26. *Hoefler T., Dinan J., Buntinas D., Balaji P., Barrett B., Brightwell R., Gropp W., Kale V., Thakur R.* MPI + MPI: A new hybrid approach to parallel programming with MPI plus shared memory // Computing. 2013. Vol. 95 (12). P. 1121–1136.
 27. *Dinan J., Balaji P., Goodell D., Miller D., Snir M., Thakur R.* Enabling MPI Interoperability Through Flexible Communication Endpoints. EuroMPI, 2013.

Статья поступила в редакцию 10.06.20.

TEST PROGRAM "PAUK" AS A TESTING RANGE FOR PARALLEL PROGRAMMING ALGORITHMS AND TECHNIQUES / A. A. Nuzhdin (FSUE "RFNC-VNIIEF", Sarov, Nizhniy Novgorod region).

PAUK is a test program that numerically solves the three-dimensional one-group neutron transport equation on orthogonal spatial grids by the difference S_n -method. The paper presents the results of its adaptation for a heterogeneous computing system with Intel's Knights Corner Xeon Phi coprocessors. The adaptation included testing of various parallel programming algorithms and techniques, namely loop vectorization in directions and hyperplane elements, automatic vectorization and intrinsic programming, explicit and implicit data prefetching, implementation of the KBA algorithm in three memory models (shared, distributed, PGAS). Adaptation efficiency was verified by program performance studies in various execution modes at the heterogeneous computing system: CPU-only, native and symmetric.

Keywords: S_n -method, sweep algorithm, KBA algorithm, Intel Xeon Phi, vectorization, data prefetching, MPI-3 SHM.

УДК 004.422.63

ЭФФЕКТИВНЫЙ АЛГОРИТМ ОБЪЕДИНЕНИЯ СОГЛАСОВАННЫХ ПО УЗЛАМ ФРАГМЕНТОВ ПОВЕРХНОСТНЫХ СЕТОК

В. В. Лазарев

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

Предложен новый алгоритм объединения согласованных по узлам фрагментов поверхностных сеток, построенных на геометрической модели в B-REP представлении. По сравнению с классическими алгоритмами объединения он имеет следующие отличия: 1) вместо рассмотрения всех узлов по отдельности рассматриваются их группы; 2) для нахождения совпадающих узлов используются не функция расстояния и ее минимальное значение, а связи между сеточными фрагментами, которые хранятся в структурах представления геометрической модели; 3) результат объединения не хранится в массивах, а представлен в функциональном виде — выражениями, которые по номеру узла вычисляют адрес этого узла в сеточном фрагменте, а по номеру ячейки — номера узлов этой ячейки.

Время выполнения разработанного алгоритма и объем потребляемой памяти почти не заметны. Эти параметры зависят только от количества сеточных фрагментов, но не зависят от количества узлов и ячеек в этих фрагментах.

Ключевые слова: составная сетка, алгоритм объединения поверхностных сеток, глобальная нумерация, функциональное представление множества, граничное представление геометрической модели (B-REP).

Введение

При решении большого класса задач используются так называемые составные сетки. Составная сетка получается разбиением области на подобласти и построением в каждой такой подобласти своей сетки (сеточного фрагмента). К составным можно отнести блочно-структурированные [1] и гибридные сетки [2], сетки с перекрытиями [3]. В блочно-структурированных сетках в каждой подобласти строится структурированная сетка, которая согласована по узлам с остальными. В гибридных сетках и сетках с перекрытиями используются комбинации структурированных и неструктурированных сеток. В общем случае составные сетки добавляют гибкость в построение сеток на сложных геометрических моделях и увеличивают степень структурированности для улучшения качества расчетов.

При реализации методов построения составных сеток перед разработчиком встает вопрос о выборе представления сетки в оперативной памяти. Существуют два основных представления — фрагментное и объединенное. У каждого из них свои преимущества и недостатки.

В ходе построения составной сетки происходят частые модификации сеточных фрагментов — добавления, удаления и редактирования, в связи с чем удобно использовать фрагментное представление сетки. В случае же ее представления в объединенном виде возникает фрагментация памяти из-за частого изменения размера массива, что приводит к невозможности выделения непрерывной области памяти большого размера при имеющихся для этого ресурсах.

Преимущество составной сетки, представленной в объединенном виде, состоит в том, что большинство алгоритмов — сглаживания, оптимизации, деформации — выполняются именно на таких

сетках. При фрагментном же представлении, прежде чем воспользоваться этими алгоритмами, сеточные фрагменты необходимо объединить в одно целое. Частые преобразования данных из одного представления в другое увеличивают задержку в получении конечного результата.

Предлагаемый в статье алгоритм объединения позволяет разработчику воспользоваться преимуществами фрагментного представления сетки и в любой момент переключаться на объединенную сетку быстро и без затрат по памяти. Для этого в алгоритме объединения используется предположение, что два смежных сеточных фрагмента были построены от одной и той же общей границы. Информация о смежности сеточных фрагментов и их общей границе хранится в структурах граничного представления геометрической модели [4, 5], с которой связана составная сетка. Это позволяет отказаться от функции расстояния между сеточными узлами, которая используется в классическом алгоритме объединения. Другая особенность алгоритма в том, что он представляет множество узлов и ячеек объединенной (результатирующей) сетки в функциональном виде.

Исходными данными для алгоритма являются структура геометрической модели в граничном представлении (B-REP) и сетки, которые описаны в разд. 1 и 3 соответственно. В разд. 2 вводятся используемые множества, операции над ними и их функциональные представления. Подробное описание предложенного алгоритма приведено в разд. 4. В разд. 5 выполнена оценка производительности реализации алгоритма на языке программирования C++.

1. Представление геометрической модели

В современных CAD-системах твердотельного и поверхностного моделирования геометрическая модель задана в граничном представлении (рис. 1), которое включает в себе набор поверхностей и информацию об их взаимных связях [4].

Для отслеживания связей поверхностей в рассмотрение вводятся топологические объекты — грани, циклы, ребра и вершины.

Вершина указывает на точку в пространстве — место углового стыка двух и более поверхностей, место излома кривой или стыка нескольких кривых.

Ребро указывает на участок кривой пересечения двух поверхностей. Оно задается двумя вершинами и направлено от первой ко второй. Ребро называется замкнутым, если оно начинается и заканчивается в одной и той же вершине. Одно и то же ребро может входить в несколько граней, но с разными направлениями.

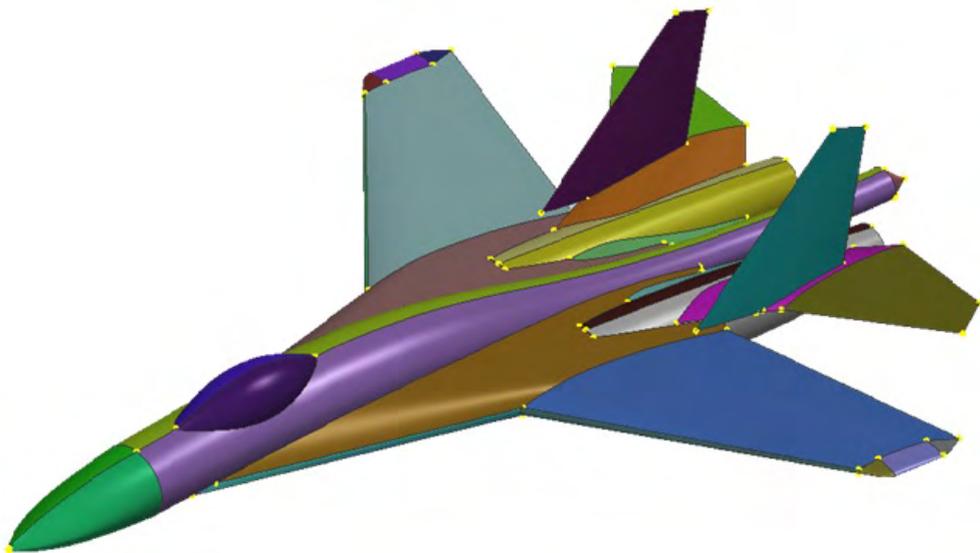


Рис. 1. Геометрическая модель самолета в граничном представлении из [6]

Цикл указывает на одну границу поверхности геометрической модели. Он задается набором стыкующихся в вершинах ребер. С ребром в цикле связан признак инверсного направления. Признак отрицателен, если направление ребра в цикле совпадает с его исходным направлением. В противном случае признак положителен. Одно и то же ребро не может входить в цикл более одного раза. Если в цикле больше одного ребра, то ни одно из этих ребер не может быть замкнутым. Данные ограничения на цикл налагаются алгоритмом объединения сеточных фрагментов.

Грань указывает на поверхность геометрической модели. Она задается одним или несколькими непересекающимися циклами. Первый цикл описывает внешнюю границу грани, а остальные, если они есть, — ее внутренние отверстия.

Множество всех вершин, ребер и граней обозначим соответственно через V , E , F . Через T обозначим объединенное множество вершин, ребер и граней, т. е. $T = V \cup E \cup F$. Введем понятие порядка топологического объекта из T : это целое число, приписанное топологическому объекту. Для вершин его значение равно нулю, для ребер — единице, для граней — двум.

2. Множества, операции над ними и их функциональное представление

Для описания алгоритма объединения сеток будут использованы множества и операции над ними. Все рассматриваемые множества конечные, а их элементы пронумерованы, начиная с нуля. Выражение $M(i)$ обозначает элемент с порядковым номером i множества M .

Используемые далее множества являются подмножествами следующих множеств: \mathbb{Z}_0 — натуральных чисел, дополненных нулем; \mathbb{A} — адресов ячеек памяти; T — объектов топологии; произведений $T \times \mathbb{Z}_0$ или \mathbb{Z}_0^n ($n \geq 3$). Конечные множества могут быть представлены двумя способами — массивом элементов или функцией. Массив элементов может быть заменен таблично заданной функцией. Таким образом, используемые далее множества имеют функциональное представление. Функциональное представление множества дает правило вычисления элемента множества по его номеру.

Функция f , задающая множество, — это функция одного целочисленного аргумента с областью определения $[0, N - 1] \in \mathbb{Z}_0$, где N — мощность этого множества. Мощность обозначается как $|f|$.

Благодаря функциональному способу задания множества более экономично используется оперативная память. Такой способ применяется, например, в определении массива ячеек матрично-структурированной сетки. Для вычисления узлов ячейки по ее номеру необходимы только размеры этой сетки по направлениям I, J .

Будем использовать четыре операции над конечными множествами. Результатом выполнения этих операций являются множества, которые представляются в функциональном виде:

1. $f_1 = Reverse(f)$ — изменение последовательности элементов множества на обратный. Результирующее множество f_1 имеет следующее функциональное представление:

$$f_1(i) = f(|f| - i - 1).$$

При этом $|f_1| = |f|$.

Например, $Reverse(\{a, b, c, d, e\}) = \{e, d, c, b, a\}$.

2. $f_1 = ExcludeBounds(f)$ — исключение из множества первого и последнего элементов. Результирующее множество f_1 может быть представлено в виде

$$f_1(i) = f(i + 1), \text{ причем } |f_1| = |f| - 2.$$

Таким образом, $ExcludeBounds(\{a, b, c, d, e\}) = \{b, c, d\}$.

3. $f = f_1 \cup f_2$ — объединение множеств в предположении, что все их элементы уникальны. Результирующее множество f имеет следующее функциональное представление:

$$f(i) = \begin{cases} f_1(i), & \text{если } i < |f_1|; \\ f_2(i - |f_1|), & \text{если } i \geq |f_1|; \end{cases} \quad |f| = |f_1| + |f_2|.$$

Например, $\{a, b, c\} \cup \{d, e, f\} = \{a, b, c, d, e, f\}$.

4. $f = f_1 \circ f_2$ — выборка подмножества элементов. Результирующее множество f имеет следующее функциональное представление:

$$f(i) = f_1(f_2(i)),$$

причем $|f| = |f_2|$, $0 \leq \max(f_2) < |f_1|$. Например, $\{a, b, c, d, e, f\} \circ \{2, 4, 0\} = \{c, e, a\}$.

3. Представление сеточных фрагментов

Составной сеткой назовем совокупность сеточных фрагментов для топологических объектов из T . Сеточный фрагмент — это самодостаточная сетка, связанная с объектом топологии из T — вершиной, ребром или гранью.

Сеточные фрагменты для вершины и ребра задаются одномерными массивами узлов. Для случая с вершиной в массиве содержится только один элемент.

Особый интерес будет представлять *сеточный фрагмент для грани*. Это поверхностная сетка, которая в ячеечно-узловом представлении задается массивами узлов, ячеек и четырьмя вспомогательными множествами.

Элементом массива узлов в простом случае может быть пара (для двумерного пространства) или тройка (для трехмерного пространства) вещественных чисел — декартовы координаты узла. Массив узлов далее рассматривается как множество $nodes \in \mathbb{A}$.*

Элементами массива ячеек являются списки номеров узлов этих ячеек. Массив ячеек далее рассматривается как множество $cells \in \mathbb{Z}_0^n$, где $n \geq 3$.

Под граничными узлами сеточного фрагмента для грани понимаются узлы, лежащие на цикле грани. Внутренние узлы — это узлы, не являющиеся граничными. Информация о граничных и внутренних узлах задается в виде четырех множеств:

1. $BN \in \mathbb{Z}_0$. Содержит номера во множестве $nodes$ граничных узлов, упорядоченных по направлению обхода цикла и заданной последовательности циклов в грани. $|BN|$ дает количество граничных узлов.
2. $IN \in \mathbb{Z}_0$. Перечисляет номера внутренних узлов во множестве $nodes$ в порядке их возрастания.
3. $BNN \in \mathbb{Z}_0$. Множество имеет следующий функциональный вид:

$$BNN(i) = \begin{cases} x, & \text{если } BN(x) = i; \\ -1, & \text{если } \nexists x : BN(x) = i. \end{cases}$$

4. $INN \in \mathbb{Z}_0$. Множество имеет следующий функциональный вид:

$$INN(i) = \begin{cases} x, & \text{если } IN(x) = i; \\ -1, & \text{если } \nexists x : IN(x) = i. \end{cases}$$

Как $|BNN|$, так и $|INN|$ дает количество узлов в сеточном фрагменте для грани.

На рис. 2 представлен пример сеточного фрагмента с номерами узлов во множестве $nodes$. Стрелкой показаны начало и направление цикла. Для данной сетки $|BN| = 7$, $|IN| = 3$, $|BNN| = |INN| = 10$. Множества содержат следующие элементы:

$$BN = \{4, 0, 1, 9, 7, 5, 3\}; \quad IN = \{2, 6, 8\};$$

$$BNN = \{1, 2, -1, 6, 0, 5, -1, 4, -1, 3\}; \quad INN = \{-1, -1, 0, -1, -1, -1, 1, -1, 2, -1\}.$$

В общем случае нет необходимости в задании всех четырех множеств. Множества IN , BNN , INN могут быть вычислены по BN , но на практике для большинства случаев существует простой

*Далее для простоты будем отождествлять адрес узла с самим узлом.

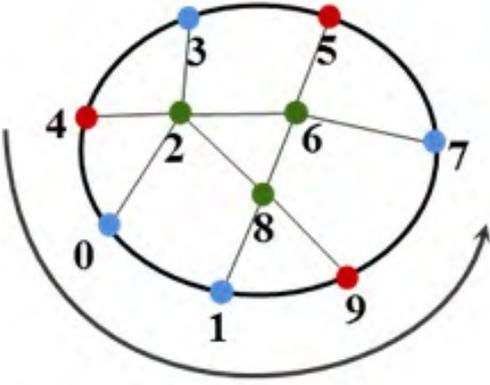


Рис. 2. Пример сеточного фрагмента с номерами узлов во множестве $nodes$

функциональный способ их задания. Для этого нужно заметить, что большинство методов построения неструктурированных сеток от заданной границы добавляют новые узлы (внутренние) в конец массива узлов. Таким образом, в массиве узлов сначала располагаются граничные узлы в порядке, заданном пользователем, а затем — внутренние узлы. Следовательно, если обозначить через n_b , n_i соответственно количество граничных и внутренних узлов в сеточном фрагменте, то

$$\begin{aligned}
 BN(i) &= i; & |BN| &= n_b; & IN(i) &= n_b + i; & |IN| &= n_i; \\
 BNN(i) &= \begin{cases} i, & \text{если } i < n_b; \\ -1, & \text{если } i \geq n_b; \end{cases} & |BNN| &= n_b + n_i; \\
 INN(i) &= \begin{cases} -1, & \text{если } i < n_b; \\ i - n_b, & \text{если } i > n_b; \end{cases} & |INN| &= n_b + n_i.
 \end{aligned}$$

Для каждого сеточного фрагмента задаются свои множества $nodes$, $cells$, BN , IN , BNN , INN . Далее запись $f|_t$ будет означать, что множество f относится к сеточному фрагменту для топологического объекта t .

4. Вывод функционального представления результата объединения сеточных фрагментов

Назовем сетку, полученную объединением сеточных фрагментов, результирующей. Она так же, как и сеточные фрагменты, состоит из множества узлов и ячеек. Но в отличие от сеточных фрагментов для результирующей сетки эти множества представлены в функциональном виде. В данном разделе показан вывод этих представлений.

4.1. Функциональное представление множества узлов. Множество узлов $nodes \in \mathbb{A}$ результирующей сетки содержит адреса этих узлов в памяти. Для узлов результирующей сетки память не выделяется. Узлы, или их адреса, берутся из исходных множеств $nodes|_{t \in T}$.

Граничные узлы у смежных сеточных фрагментов дублируются — один и тот же узел входит в несколько сеточных фрагментов. Задачей алгоритма объединения является выбор из нескольких одинаковых узлов *правильного*. Под правильным понимается узел, принадлежащий сеточному фрагменту для топологического объекта наименьшего порядка. Чтобы реализовать это условие, введем множество внутренних узлов $innerNodes|_{t \in T} \in \mathbb{A}$ сеточного фрагмента:

$$innerNodes|_{t \in T} = \begin{cases} nodes|_t, & \text{если } t \in V; \\ ExcludeBounds(nodes|_t), & \text{если } t \in E; \\ nodes|_t \circ IN|_t, & \text{если } t \in F. \end{cases}$$

Таким образом, внутренним узлом для вершины будет сам узел-сетка, для ребра — все узлы, включая крайние, для грани — все внутренние узлы, определенные множеством IN и выделенные операцией выборки.

Узлы результирующей сетки — это объединение внутренних узлов всех сеточных фрагментов. С использованием операции объединения, введенного в разд. 2, данный факт можно выразить в функциональном виде как

$$nodes = \bigcup_{t \in T} innerNodes|_t.$$

Заметим, что это выражение исключает дублируемые узлы.

4.2. Глобальная нумерация узлов. Глобальная нумерация для узла сеточного фрагмента определяет его номер в результирующей сетке. Зададим для грани f такую нумерацию множеством $l2g|_{f \in F} \in \mathbb{Z}_0$. Прежде чем определить это множество, определим $GN|_{t \in T} \in \mathbb{Z}_0$ — глобальную нумерацию внутренних узлов сеточного фрагмента для топологического объекта и $bounds|_{f \in F} \in \mathbb{Z}_0$ — глобальную нумерацию узлов на границе сеточного фрагмента грани.

Последовательность узлов в результирующей сетке определяется последовательностью топологических объектов во множестве T и неявно задает глобальную нумерацию внутренних узлов $GN|_{t \in T}$ сеточных фрагментов как

$$GN|_{T(j)}(i) = i + \sum_{k=0}^{k < j} |innerNodes|_{T(k)}|.$$

Множество $bounds|_f$ определяется как объединение множеств $l2g|_c$, заданных для циклов грани:

$$bounds|_{f \in F} = \bigcup_{c \in loops(f)} l2g|_c.$$

Здесь $loops(f)$ определяет множество циклов грани f . В свою очередь, $l2g|_c$ для цикла можно рассматривать как последовательное (для всех его ребер) объединение множеств GN , определяемых для одной из вершин ребра и для этого ребра с учетом признака направления:

$$l2g|_c = \bigcup_{(e, inv) \in Edges(c)} \left(GN|_{BegVertex(e, inv)} \cup GN|_{(e, inv)} \right).$$

Здесь $Edges(c)$ определяет множество пар — ребер e и признаков inv их направления в цикле c ; $BegVertex(e, inv)$ возвращает первую вершину ребра e при отрицательном inv и вторую вершину — при положительном. Множество GN для вершины введено выше, а для ребра с учетом признака направления определяется как

$$GN|_{(e, inv)} = \begin{cases} GN|_e, & \text{если } inv \text{ ложно;} \\ Reverse(GN|_e), & \text{если } inv \text{ истинно.} \end{cases}$$

На основе введенных выше вспомогательных множеств множество $l2g|_{f \in F}$ определяется как

$$l2g|_{f \in F}(i) = \begin{cases} bounds|_f(BNN|_f(i)), & \text{если } BNN|_f(i) \neq -1; \\ GN|_f(INN|_f(i)), & \text{если } BNN|_f(i) = -1. \end{cases}$$

Если узел внутренний, то его номер в результирующей сетке определяется значением из $GN|_f$, в противном случае — значением из $bounds|_f$. Тип узла, а также его порядковый номер в списке граничных или внутренних узлов определяются через $BNN|_f$ и $INN|_f$.

4.3. Локальная нумерация ячеек. Локальная нумерация для номера ячейки результирующей сетки определяет пару: сеточный фрагмент для грани, которому ячейка принадлежит, и ее номер в этом сеточном фрагменте. Зададим такую нумерацию множеством $c2c \in F \times \mathbb{Z}_0$. Заметим, что $|c2c| = N$ — количество ячеек в результирующей сетке.

Введем вспомогательное множество $s \in \mathbb{Z}_0$ номеров ячеек результирующей сетки, которые являются первыми ячейками множеств $cells|_{F(j)}$ сеточных фрагментов для граней:

$$s(i) = \begin{cases} 0, & \text{для } i = 0; \\ \left(\sum_{j=0}^{i-1} |cells|_{F(j)} \right) - 1, & \text{для } 0 < i \leq |T|. \end{cases}$$

Элементы множества строго возрастают в силу того, что $|cells|_{F(j)} > 0$ — количество ячеек в сеточном фрагменте всегда больше нуля.

Тогда множество $c2c$ определяется как

$$c2c(j) = \{F(i), j - s(i)\},$$

где i удовлетворяет условию $s(i) < j < s(i+1)$; $|c2c| = \sum_{f \in F} |cells|_f$.

4.4. Функциональное представление множества ячеек. Множество ячеек $cells \in \mathbb{Z}_0^n$ результирующей сетки содержит ячейки как списки номеров узлов этих ячеек во множестве $nodes$ (см. подразд. 4.1). Способ вычисления номеров узлов результирующей сетки по номеру ячейки выглядит следующим образом:

1. Номер ячейки результирующей сетки i преобразуется в номер ячейки i' в сеточном фрагменте для грани f . Это преобразование задается локальной нумерацией ячеек, а ее результат содержится в $c2c$.
2. Из множества $cells|_f$ берутся номера узлов (i_0, i_1, \dots) ячейки i' сеточного фрагмента f .
3. Номера узлов сеточного фрагмента f преобразуются в номера узлов результирующей сетки. Это преобразование задается глобальной нумерацией узлов, а ее результат содержится в $l2g|_f$.

В функциональном виде это представляется как

$$cells(i) = (l2g|_f(i_0), l2g|_f(i_1), \dots).$$

Здесь $(i_0, i_1, \dots) = cells|_f(i')$, где $(f, i') = c2c(i)$, $|cells| = |c2c|$.

5. Оценка производительности алгоритма

Алгоритм на основе введенных в предыдущих разделах множеств и операций над ними реализован на языке программирования C++ с использованием объектно-ориентированного подхода, классов и полиморфизма подтипов. Программа собрана компилятором Microsoft Visual Studio 2013 под конфигурацией Release со стандартными настройками для 64-разрядной платформы. Для запуска использован персональный компьютер с процессором Intel Core i5-2400, 3.10 ГГц и 16 Гбайт оперативной памяти.

Оценка производительности проведена на трех геометрических моделях, состоящих из 103, 188 и 94 граней. Для каждой модели построены две различные сетки, отличающиеся количеством сеточных элементов. Выполнены измерения трех параметров — времени объединения и скорости доступа к узлам и ячейкам. Доступ выполняется последовательно по всему массиву и повторяется 100 раз.

Полученные измерения представлены в табл. 1, из которой можно сделать следующие выводы. Время объединения сетки зависит от количества сеточных фрагментов и не зависит от количества сеточных элементов в них. Средняя скорость доступа составляет 30 млн узлов в секунду и 7 млн ячеек в секунду. Последовательный доступ непосредственно к элементам массива в ОЗУ дает скорости 350—400 млн узлов и 180—230 млн ячеек в секунду. Вычисление адреса узла и описания ячеек по предложенным в статье выражениям медленнее соответственно в 12—14 и 25—33 раза по сравнению с доступом в память по индексу массива. Скорость доступа может быть ускорена вычислением выражений параллельно, так как предложенные структуры не имеют побочных эффектов.

Оценка объема потребляемой памяти выполнена на геометрической модели с 250 гранями. Объединяемая фрагментная сетка на этой модели состояла из разного количества сеточных элементов. Результаты выполнения алгоритма для этих входных данных представлены в табл. 2. Заметим, что количество узлов/ячеек и количество дублируемых узлов в первом и втором столбцах таблицы меньше действительных значений в 1 000 раз.

Таблица 1

Время объединения и доступа к элементам сетки в зависимости от задачи

Количество граней	Количество узлов/ячеек	Время объединения сетки, с	Время стократного доступа ко всем узлам/ячейкам, с
103	39 603/51 364	0,002	0,133/0,821
103	308 129/393 254	0,002	1,07/6,312
188	248 72/36 480	0,003	0,091/0,553
188	43 184/67 968	0,003	0,147/0,906
94	25 913/40 404	0,002	0,083/0,528
94	171 103/245 733	0,001	0,552/3,332

Таблица 2

Измеренные параметры функции объединения в зависимости от количества сеточных элементов в результирующей сетке

Количество узлов/ячеек	Количество дублируемых узлов	Время объединения сетки, с	Потребляемая память до объединения, Мбайт	Память после объединения, Мбайт	Приращение памяти, Кбайт
8,3/17,5	3,7	0,005	51,141	51,441	308
15,9/32,6	8,7	0,005	51,578	51,848	276
26,2/53,4	12,3	0,005	52,840	52,984	148
57,3/115,5	18,6	0,005	52,687	53,176	504
97,3/195,6	24,8	0,005	54,816	55,227	420
168,6/338,2	34,7	0,006	58,328	58,691	372
253,1/507,1	44,6	0,005	63,176	63,660	496
377,2/755,4	56,7	0,005	71,066	71,336	276
501,4/100,4	66,6	0,005	75,508	75,715	212
1 240,3/2 481,5	107,0	0,005	112,969	113,402	443

Количество ячеек в сетке росло от 17 тыс. до 2,5 млн. При этом количество дублируемых узлов составило от 4 до 107 тыс. Как видно из табл. 2, время объединения не зависит от размеров сетки и занимает около 5 мс. Память, потребляемая за счет функции объединения, не коррелирует с размерами сетки и меняется случайным образом — от 150 до 500 Кбайт. По мнению автора, это связано с малостью измеряемого значения и логики действий менеджера памяти C++. В ходе выполнения функции объединения память запрашивается малыми порциями и может быть выделена в уже существующей странице памяти — в областях, возникших из-за ее фрагментации. Также память может быть выделена в новой странице, но страница будет использована частично. Отсутствие корреляции между размерностью сетки и потребляемой памятью показывает для алгоритма объединения одинаковую эффективность по памяти при увеличении количества сеточных элементов.

Заключение

Предложен новый алгоритм объединения поверхностных сеток, который вместо функции расстояния использует предположение, что смежные сеточные фрагменты построены от одной и той же границы. При этом объединенная сетка представляется в функциональном виде — функциями узлов и ячеек. Такие особенности алгоритма позволяют объединять фрагментную сетку эффективно как по времени, так и по памяти. Время выполнения алгоритма и потребляемая им память не зависят от количества сеточных элементов, а зависят только от количества сеточных фрагментов. Недостатком функционального представления является необходимость вычислений при доступе к элементам сетки. Для преодоления этого недостатка можно воспользоваться двумя способами — па-

параллельным доступом к элементам сетки или кэшированием сетки в оперативной памяти на время использования.

Предложенный алгоритм позволяет разработчику хранить составную сетку в оперативной памяти как набор сеточных фрагментов и в любой момент переходить к объединенной сетке. Такой подход уменьшает фрагментацию памяти при частых изменениях составной сетки, что свойственно при ее построении и редактировании. При этом алгоритмы, разработанные для единой сетки, можно использовать для составной без дублирования оперативной памяти.

Предложенный алгоритм используется для объединения блочных сеток в препроцессоре пакета программ "Логос" [8]. По сравнению с описанным в статье он расширен на случай объединения объемных структурированных сеток. Также объединение используется для подготовки сеточных данных к построению объемной сетки протягиванием [7].

Список литературы

1. *Weatherill N. P., Forsey C. R.* Grid generation and flow calculations for complex aircraft geometries using a multi-block scheme // AIAA 17th Fluid Dynamics, Plasma Dynamics, and Laser Conference. Snowmass, CO. June 25–27, 1984.
2. *Lohner R., Luo H., Spiegel S.* Hybrid grid generation method for complex geometries // AIAA. 2010. Vol. 48, No 11. P. 2639–2646.
3. *Benek J. A., Donegan T. L., Suhs N. E.* Extended chimera grid embedding scheme with application to viscous flow // AIAA Paper. 1987. № 87-1126.
4. *Голованов Н. Н.* Геометрическое моделирование. М.: Физматлит, 2002.
Golovanov N. N. Geometricheskoe modelirovanie. M.: Fizmatlit, 2002.
5. Industrial Automation Systems and Integration. Product Data Representation and Exchange. Part 42: Integrated Generic Resource: Geometric and Topological Representation. <https://www.iso.org/standard/78579.html>.
6. Sukhoi Su-27 | 3D CAD Model Library | GrabCAD. <https://grabcad.com/library/su-27-2>.
7. *Гиниятуллина А. Г., Лазарев В. В., Мартенс Р. В.* Построение сетки протягиванием поверхностных ячеек в отдельном объеме // XV науч.-тех. конф. "Молодежь в науке": сб. тез. Саров, 25–27 октября, 2016. С. 9.
Giniyatullina A. G., Lazarev V. V., Martens R. V. Postroenie setki protyagivaniem poverkhnostnykh yacheek v otdelnom obyeme // XV nauch.-tekh. konf. "Molodezh v nauke": sb. tez. Sarov, 25-27 oktyabrya, 2016. S. 9.
8. *Лазарев В. В.* Распараллеливание и оптимизация построения блочных расчетных сеток в препроцессоре пакета программ ЛОГОС // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 1. С. 54–63.
Lazarev V. V. Rasparallelvanie i optimizatsiya postroeniya blochnykh raschetnykh setok v preprotessore paketa programm LOGOS // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2018. Вып. 1. S. 54–63.

AN EFFICIENT ALGORITHM FOR MERGING NODE-MATCHED FRAGMENTS OF SURFACE GRIDS / V. V. Lazarev (FSUE "RFNC-VNIIEF", Sarov, Nizhniy Novgorod region).

A new algorithm is proposed for merging node-matched fragments of surface grids constructed based on a geometric model in the B-REP representation. As opposed to classical merging algorithms, this algorithm: 1) considers groups of nodes rather than each individual node; 2) identifies matching nodes based on the connections between grid fragments, which are stored in geometric model representation structures, rather than based on the distance function and its minimum value; 3) instead of storing the result of merging in arrays, represents it in the form of functions, which calculate the node address in the grid fragment based on the node index and the node indexes in a cell, based on the cell index.

The processor time and the memory consumed by the algorithm are negligibly small. They depend only on the number of grid fragments and do not depend on the number of nodes or cells in the fragments

Keywords: composite grid, surface grid merging algorithm, global indexing, functional representation of a set, boundary representation of a geometric model (B-REP).

УДК 519.6

ГЕНЕРАТОР ПОВЕРХНОСТНЫХ НЕСТРУКТУРИРОВАННЫХ ЧЕТЫРЕХУГОЛЬНЫХ СЕТОК В ПРЕПРОЦЕССОРЕ ПАКЕТА ПРОГРАММ "ЛОГОС"

Е. Ю. Арапова, В. Г. Куделькин, Е. А. Павлов, С. Ю. Полякова, А. В. Тихонов
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Приводится описание автоматического генератора поверхностных неструктурированных четырехугольных сеток для геометрических моделей в параметрическом представлении. Генератор предназначен для построения сеточных моделей при решении задач прочности в рамках пакета программ "Логос". Рассматриваются этапы генерации поверхностной сетки. Подробно рассмотрен один из этапов, посвященный генерации четырехугольных сеток в параметрической плоскости. Приведены примеры тестовых задач, в которых используются различные геометрические модели.

Ключевые слова: пакет программ "Логос", генератор поверхностных четырехугольных сеток, подвижный фронт, формирование четырехугольных ячеек, локальное сглаживание, топологическая оптимизация.

Введение

В современной научно-исследовательской деятельности большое внимание уделяется развитию методов численного моделирования физических процессов в областях сложной геометрии. При этом наиболее подходящим для численного решения задач прочности является *метод конечных элементов* (МКЭ) [1]. Обусловленность систем линейных алгебраических уравнений, возникающих при реализации МКЭ, тем лучше, чем больше минимальные углы между гранями (ребрами) конечных элементов. В частности, точность расчетов на сетках из треугольных элементов (ячеек) существенно ниже, чем на четырехугольных сетках.

Данная работа преследует две цели: 1) представить генератор неструктурированных четырехугольных сеток, входящий в состав пакета программ (ПП) "Логос" [2] и предназначенный для построения поверхностных сеток при подготовке задач в модуле "Логос Прочность"; 2) дать описание модифицированных алгоритмов разработанного генератора, которые позволили повысить качество формируемой сетки и надежность алгоритма.

Методы построения сеток делятся на два класса: прямые и не прямые (итерационные) [3]. Из-

за своей универсальности, в частности способности строить сетку в областях достаточно произвольной формы, не прямые методы получили наибольшее распространение.

Одним из наиболее эффективных по ряду критериев является не прямой метод подвижного фронта Q-Morph [3], предложенный Стивенем Оуэном в 1998 г. Сетка в нем строится столь же качественная, как и в лучших прямых методах. Используемые трансформации треугольных элементов позволяют контролировать размеры генерируемых элементов, анализировать границы области, а также делают его надежным и быстрым.

Существуют варианты метода подвижного фронта, которые напрямую строят сетки из конечных элементов непосредственно на поверхности. Как утверждают авторы статьи [4], к недостаткам этих методов можно отнести сложность реализации, низкую скорость построения и неустойчивость. Ими же в [4] был предложен метод построения поверхностной треугольной сетки с использованием генерации сетки на плоскости с учетом римановой метрики поверхности для контроля размера и формы треугольников в параметрическом пространстве. Получены удовлетворительные результаты, и данный генера-

тор включен в состав коммерческого программного продукта ANSYS [5]. Авторами настоящей работы было принято решение повторить опыт использования плоского генератора для построения поверхностной сетки [4], но с его обобщением на четырехугольные элементы.

Программной реализации метода Q-Morph посвящено большое количество публикаций. Авторами известных российских программ являются А. В. Сковпень (РФЯЦ-ВНИИТФ, г. Снежинск) [6] и В. И. Лысухин (ОАО "Туполев", г. Москва) [7]. В процессе разработки ими были модифицированы некоторые имеющиеся процедуры метода и добавлены новые. Кроме того, ими (каждым отдельно) была разработана технология топологической оптимизации отстраиваемой сетки. В настоящей работе в качестве базового алгоритма был взят алгоритм, предложенный В. И. Лысухиным в [7].

На основе анализа данных, полученных при построении четырехугольных сеток на достаточном количестве геометрических моделей (произвольной формы и с различным разбиением на граничных контурах), авторы сделали вывод о необходимости доработки реализуемого алгоритма, так как многие моменты, связанные с обработкой различных геометрических и топологических особенностей, в описании либо отсутствовали, либо были преподнесены достаточно поверхностно. Основным изменениям были подвергнуты процедуры обработки фронтов и построения нового четырехугольного элемента.

В процессе формирования четырехугольников стартовая треугольная сетка подвергается сильным трансформациям. Чем дальше от границы области, тем более неравномерными становятся размеры треугольников. Также могут появиться так называемые вывернутые треугольники и треугольники с нулевой площадью. Поэтому после успешного построения каждого четырехугольного элемента выполняется процедура локального сглаживания. За основу взят метод, описанный в работе Т. Д. Блэклера и М. Б. Стефенсона [8]. Тестовые расчеты выявили ряд проблем и недостатков в используемом алгоритме многоэтапного локального сглаживания, которые, в свою очередь, потребовали изменения некоторых этапов алгоритма. Дополнительно используется сглаживание либо по Лапласу, либо по Лапласу с ограничениями [9].

Наконец, было принято решение отказаться от разработанного и предложенного В. И. Лысухиным [7] метода топологической оптимизации.

Топологическая оптимизация — это совокупность операций, предназначенных для улучшения качества сетки. Для четырехугольной сетки считается, что внутренний узел обладает оптимальной топологией, если он принадлежит четырем ячейкам (условие регулярности). В базовой версии решение о проведении той или иной процедуры принимается на основании анализа соседних ячеек. Практическое использование показало низкую скорость выполнения программы, обусловленную многократно повторяющейся процедурой определения соседей для каждого четырехугольника и дублированием операций определения условий нерегулярности вокруг узла, потому что каждый узел принадлежит сразу нескольким четырехугольникам. Реализованная версия программы топологической оптимизации использует результаты работ [10–12].

1. Технология построения неструктурированных четырехугольных поверхностных сеток в препроцессоре ПП "Логос"

Построение сетки на геометрической модели в параметрическом представлении состоит из нескольких этапов:

1. Для построения сетки на поверхности пользователю необходимо задать разбиение ее границы на набор ребер с желаемым размером. Для замкнутых поверхностей, не имеющих границ, типа сферы или поверхностей сложной формы, например с изломами, стандартной считается практика, когда пользователь перед построением сетки упрощает геометрию, разбивая деталь на части с более простой топологией. Программный продукт "Логос" обладает большим набором средств для выполнения подобных операций [13].
2. Геометрическая модель отображается в параметрическую плоскость R^2 . В препроцессоре ПП "Логос" используется функция отображения, реализованная в коде библиотеки геометрического ядра C3D [14].
3. В плоскости R^2 строится треугольная сетка. Построение осуществляется генератором треугольных сеток, реализованным в библиотеке CM2 MeshTools [15].
4. В плоскости R^2 строится четырехугольная сетка методом подвижного фронта с учетом внутренней геометрии поверхности при вы-

числении размера элемента сетки и углов между ребрами.

- Обратным преобразованием построенная в R^2 сетка отображается в пространство R^3 . В результате исходная геометрическая модель покрывается поверхностной неструктурированной сеткой четырехугольных ячеек.

Этапы 2–5 выполняются автоматически, без вмешательства пользователя. На этапах 1–3 и 5 использовались уже реализованные в ПП "Логос" программные средства. Далее рассмотрим более детально алгоритмы, реализованные авторами данной статьи для этапа 4.

2. Общая схема алгоритма построения неструктурированных четырехугольных сеток

Как было сказано во Введении, в качестве базового используется алгоритм, предложенный в работе [7]. В целом он повторяет алгоритм Q-Morph и состоит из следующих основных этапов:

- Определение подвижного фронта.** Фронтом называется замкнутая совокупность ребер, на которых будут формироваться новые четырехугольные элементы. Если область построения многосвязная, имеем несколько независимых фронтов. Естественно, в самом начале к фронтам относятся ребра, принадлежащие границам области. Фронты обрабатываются последовательно. Формирование четырехугольников начинается с фронта, описывающего внешнюю границу области. На всех ребрах выбранного фронта выполняется процедура построения четырехугольной ячейки. Обход ребер фронта осуществляется против часовой стрелки (область, подлежащая разбиению, всегда слева

от фронта). После этого выполняется переход к обработке следующего фронта. По мере построения четырехугольников фронт меняется (движется) и представляет собой границу, отделяющую уже построенные четырехугольные ячейки от области, содержащей треугольные ячейки. Фронты могут делиться и объединяться в процессе построения сетки. В конце, после смыкания всех фронтов, поверхность оказывается полностью заполненной четырехугольными элементами. Заметим, что получаемая сетка гарантированно будет полностью четырехугольной, если фронт содержит четное число ребер (условие четности фронта).

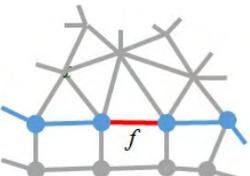
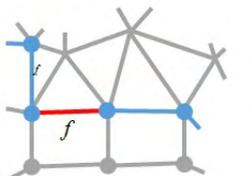
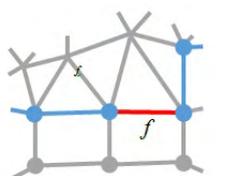
- Классификация ребер фронта.** Будем считать, что общий узел двух соседних ребер фронта находится в состоянии 0 (состоянии 1), если угол между этими ребрами больше $3\pi/4$ (меньше $3\pi/4$). В свою очередь, в процессе создания четырехугольных элементов каждое ребро f фронта принимает одно из трех состояний (табл. 1) в зависимости от значений состояний своих узлов.

На начальном этапе построения четырехугольной сетки в процессе обработки фронта, описывающего внешнюю границу области, определяется *стартовое* ребро. Для его определения используются два признака: минимальная длина ребра и минимальный угол между ребрами.

- Построение новой ячейки на ребре фронта.** Ребро фронта, на котором строится новая четырехугольная ячейка, называется *активным*. Для построения используются различные процедуры, основанные на классификации ребер фронта. Если новая ячейка сформирована, выполняется локальное сглаживание. После построения ячейки

Таблица 1

Характерные состояния ребер фронта

0 – 0	1 – 0 (0 – 1)	1 – 1
		

активное ребро фронта меняется — им становится следующее ребро текущего фронта.

4. Проверка наличия ребер подвижного фронта. Если существуют ребра подвижного фронта, выполняется переход на этап 2.
5. Топологическая оптимизация (регуляризация сетки). Данная процедура направлена на оптимизацию связности сетки.
6. Финальное сглаживание. Выполняется сглаживание всех ячеек области построения.

Далее рассмотрим подробнее изменения, внесенные авторами в базовый алгоритм из [7]. При этом обращаем внимание на тот факт, что в работе не преследуется цель описать все изменения. Остановимся только на тех процедурах, модификация которых позволила получить сетку лучшего качества при более стабильном выполнении алгоритмов.

3. Модификации базового алгоритма

Выбор направления вектора нормали. В процессе построения ячейки на активном ребре фронта из начальной треугольной сетки требуется выбрать ребра, которые будут являться боковыми сторонами формируемого четырехугольника. Подходящим считается ребро, инцидентное узлу активного ребра фронта (от которого строится боковая сторона) и имеющее наименьшее отклонение от нормали V_n (диапазон отклонения $\pm\pi/6$). Классически (см. [3, 6, 7]) нормаль в узле определяется как биссектриса угла, образованного соседними ребрами фронта. На рис. 1 показано определение направления нормального вектора, исходящего из узла B , где $\varphi = \angle ABC = \angle(AB, BC)$; AB — активное ребро фронта.

На рис. 2 приведен пример, когда применение классического подхода к определению вектора

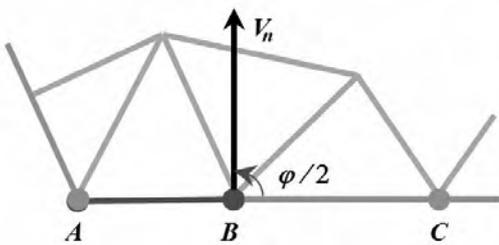


Рис. 1. Классическое определение направления нормали V_n между активным ребром AB и ребром BC

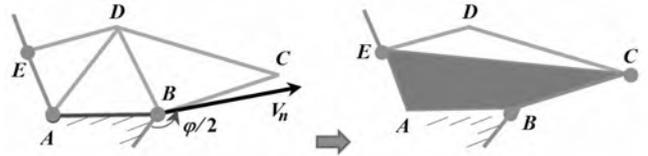


Рис. 2. Выбор ребра BC в качестве бокового ребра четырехугольника $ABCE$ с использованием классического подхода к определению вектора нормали V_n

нормали приводит к образованию четырехугольной ячейки низкого качества.

Для улучшения качества строящейся сетки был разработан и реализован следующий критерий выбора угла β между активным ребром AB и вектором нормали V_n :

$$\beta = \begin{cases} \frac{1}{4}\varphi, & \text{если } \varphi \geq 1,7\pi; \\ \frac{1}{3}\varphi, & \text{если } 1,2\pi \leq \varphi < 1,7\pi; \\ \frac{1}{2}\varphi, & \text{если } \varphi < 1,2\pi. \end{cases}$$

На рис. 3 схематично показан результат применения модифицированного алгоритма для определения направления вектора нормали V_n с последующим образованием ячейки $EABD$ удовлетворительного качества.

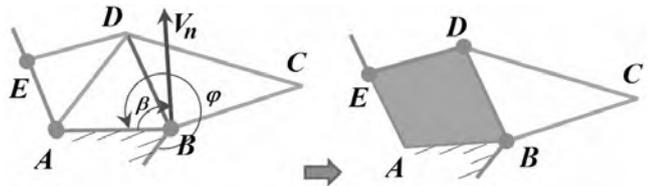


Рис. 3. Формирование четырехугольной ячейки по модифицированному алгоритму

Обработка узких фронтов. Будем называть фронт *узким*, если противолежащие ребра фронта разделены малым количеством треугольников (1–3 слоя). Признаком узкого фронта является то, что построенная из узла активного ребра нормаль V_n пересекает второе, третье или четвертое ребро фронта, считая от активного при обходе фронта по часовой стрелке. При этом из-за проведенных ранее трансформаций треугольники сильно деформированы по сравнению с первоначальной формой: площади треугольников существенно меньше первоначальных или отсутствуют треугольники, для

которых отношение площади к периметру стремится к нулю. Как правило, такая ситуация может возникнуть перед закрытием фронта, когда все входящие во фронт ребра находятся в состояниях 0-0 или 1-0 (0-1).

На рис. 4 схематично показано построение четырехугольника в узком фронте по алгоритму из работы [7]. В левой части рисунка видно, что образованный четырехугольник имеет малый размер (длина его ребер меньше характерной длины ребер фронта). Последующее применение операции сглаживания улучшит ситуацию, но треугольная сетка при этом сильно деформируется, что в дальнейшем скажется на качестве вновь формируемых ячеек. В правой части рисунка сформирован четырехугольник хорошего качества, но треугольная сетка сгущается. Также возможно заикливание по ребрам фронта (на каждом активном ребре фронта происходит отказ от построения по какой-либо причине).

Чтобы избежать подобных ситуаций, для обработки узкого фронта, когда активные ребра находятся в состояниях 0-0 или 1-0 (0-1), разработан другой алгоритм. Рассматривается узел активного ребра фронта, соответствующий состоянию 0 (для ребра фронта 0-0 берется левый узел). На рис. 5 это узел A . Далее определяется ребро фронта, которое пересекается нормалью, выходящей из рассматриваемого узла. Интересно его номер по отношению к активному ребру при прохождении фронта по часовой стрелке. Как правило, это третье или четвертое ребро фронта от активного (см. рис. 5). В других случаях четырехугольник либо образуется исполь-

зованием базового алгоритма [7], либо происходит отказ от его формирования на текущем ребре фронта.

На рис. 5 показано получение четырехугольных элементов по новому алгоритму, когда нормаль (обозначена стрелкой) из узла A пересекает третье (см. рис. 5, слева) или четвертое (см. рис. 5, справа) ребро фронта. Во втором случае формируемое ребро фронта AD заменяется на два ребра AP и PD . Тем самым сохраняется четность фронта, а вырожденность ячейки $APDE$ в дальнейшем устраняется операциями топологической оптимизации и сглаживания.

После формирования четырехугольников активным становится следующее ребро фронта BC . Процесс продолжается, пока весь узкий фронт не закроется.

Обработка шестигонных фронтов. В работе [7] каких-либо рекомендаций по обработке шестигонных фронтов предложено не было. Интуитивно понятно, что образуемый ребрами фронта шестиугольник можно трансформировать в два четырехугольника. Для этого требуется применить один из трех вариантов разбиения. В случае невыпуклых шестиугольников для выбора корректного разбиения (рис. 6, в) требуется каждый раз выполнять проверку получаемых четырехугольников на отсутствие пересекающихся ребер (рис. 6, а) и вывернутости (рис. 6, б).

На рис. 7, а показан пример, когда нельзя провести корректное разбиение на два четырехугольника. В этом случае шестиугольный фронт

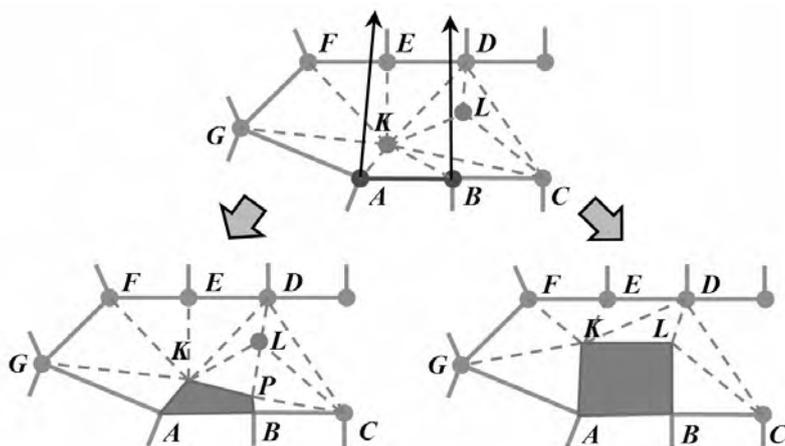


Рис. 4. Формирование четырехугольника в узком фронте по базовому алгоритму: добавлением точки P на пересекаемое ребро KC (слева); созданием ребра BL с помощью поворота ребра KC (справа)

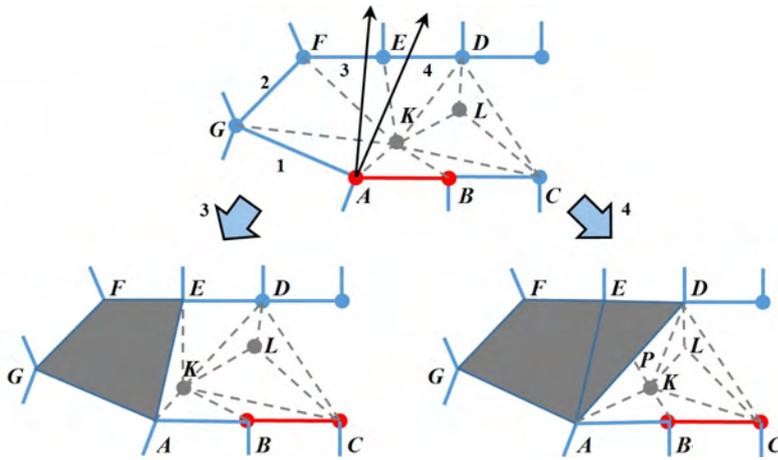


Рис. 5. Формирование из узла A активного ребра фронта AB одного четырехугольника (слева); двух четырехугольников с добавлением средней точки P ребра AD (справа)

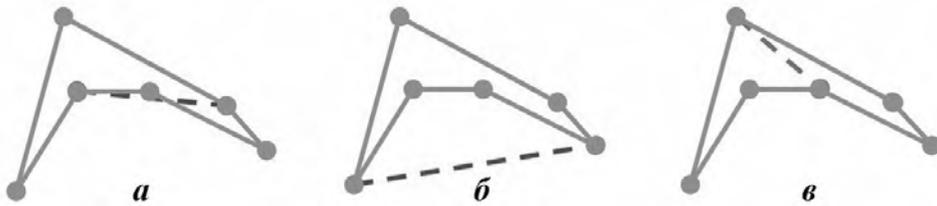


Рис. 6. Три варианта построения диагонали невыпуклого шестиугольника: a — с образованием четырехугольника с пересекающимися ребрами; $б$ — с образованием вывернутого четырехугольника; $в$ — с корректным разбиением

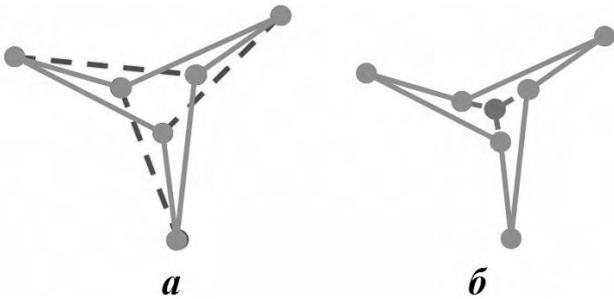


Рис. 7. Разбиение невыпуклого шестиугольника: a — на два четырехугольника (некорректное); $б$ — на три четырехугольника (корректное)

можно корректно разбить на три четырехугольника (рис. 7, б).

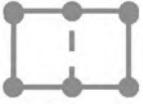
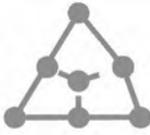
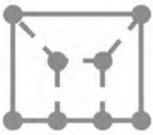
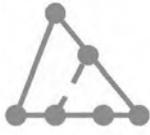
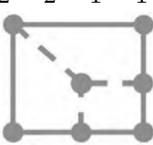
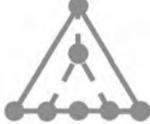
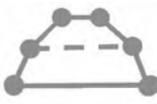
В итоге для разрешения шестиугольных фронтов был выбран алгоритм из работы [8], основанный на классификации шестиугольника по четырем шаблонам в зависимости от последовательности состояний фронтальных ребер. Предложенные в [8] шаблоны приведены в табл. 2. Числа в последовательности над каждым шаблоном

указывают количество исходных ребер, образующих одну сторону шаблонной фигуры. В зависимости от шаблона четырехугольник может быть разбит на 2, 3 или 4 четырехугольника.

Следует отметить, что алгоритм шеститочечных фронтов также потребовал доработки. Например, согласно базовому алгоритму для шестиугольника, изображенного на рис. 6, подходящим является шаблон 3-2-1 с результатом, показанным на рис 6, a . Но в этом случае образуется четырехугольник с пересекающимися ребрами. В связи с этим для нескольких шаблонов в алгоритм добавлены проверки на образование таких четырехугольников. Также был расширен список шаблонов: добавлен шаблон *пятиугольник*, показанный на рис. 8.

Восстановление ребра. Одним из этапов формирования ячейки является процедура восстановления ребра. Например, для восстановления ребра CD (рис. 9) необходимо определить набор ребер, пересекаемых CD , и затем путем последовательных поворотов совместить одно из ребер с точками C и D [7].

Шаблоны, предложенные в [8] для классификации шестиугольника

Прямоугольник	Треугольник	Сегмент круга	Круг
2-1-2-1	2-2-2	3-3	1-1-1-1-1-1
			
3-1-1-1-1	3-2-1 (1-2-3)	4-2	6
			
2-2-1-1-1	4-1-1	5-1	
			

2-1-1-1

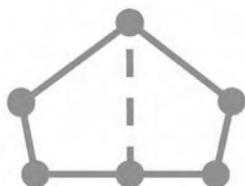


Рис. 8. Шаблон пятиугольник

Достаточно часто возникают ситуации, когда на пересекаемом ребре уже сформирована четырехугольная ячейка или оно принадлежит какому-либо контуру, ограничивающему область построения сетки. Поскольку в данных случаях восстановление ребра невозможно, выполняется переход к следующему шагу основного цикла,

где происходят смена активного ребра фронта и попытка сформировать на нем новую ячейку.

Поворот ребра разрешается, если содержащие его треугольники после поворота не являются вывернутыми. При нарушении этого условия осуществляется возврат сетки в состояние до вызова процедуры восстановления ребра.

Отметим еще один момент. Если при восстановлении ребра требуется повернуть более пяти ребер (данное значение получено апостериорно), то достигнуть хорошего качества результирующей четырехугольной сетки становится невозможным. В подобных ситуациях выполняется отказ от формирования ячейки на данном ребре фронта. Берется в рассмотрение следующее ребро, и делается новая попытка построения четырехугольника уже на нем. В случае закливания по фронту применяются специальные методы для его устранения.

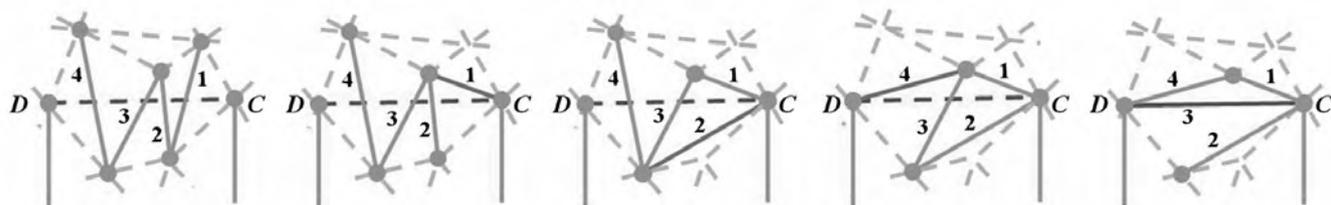


Рис. 9. Последовательность восстановления ребра CD

Объединение ребер фронта. Данная процедура применяется, если в процессе классификации ребер фронта обнаруживается, что угол между активным и предшествующим ему ребрами фронта меньше $\pi/6$, а соотношение их длин не больше 2. При этом объединение возможно только в случае, когда одно или оба рассматриваемых ребра фронта не принадлежат границе области (рис. 10, 11).

Теоретически объединение производится несложно, такая процедура описана в [7]. Однако практически всегда возникают ситуации, когда в область между объединяемыми ребрами попадают узлы и/или образуются перехлесты. Заранее предусмотреть и обработать все возможные ситуации не представляется возможным. Для устранения лишних узлов и

перехлестов был реализован следующий алгоритм (см. рис. 10, 11 с объединяемыми ребрами AB и BC):

- 1) восстанавливаем ребро AC (если оно отсутствует);
- 2) находим $\triangle ACD$ — смежный с $\triangle ABC$ по ребру AC ;
- 3) для четырехугольника $ABCD$ проводим процедуру *схлопывания* по ребру AC влево ($C \rightarrow A$, см. рис. 10, а), вправо ($A \rightarrow C$, см. рис. 10, б) или в середину отрезка AC (см. рис. 11) в зависимости от ситуации.

Такой подход гарантирует, что никакая лишняя точка не попадет в область схлопывания. Однако предварительно необходимо сделать проверку на возможное появление перехлестов. Для этого производится оценка треугольников, непосредственно примыкающих к области схлопывания. Если какой-нибудь треугольник оказывается вывернутым, то процедура объединения ребер фронта не выполняется и восстанавливается исходная треугольная сетка. Далее в основном цикле в качестве активного берется следующее ребро фронта.

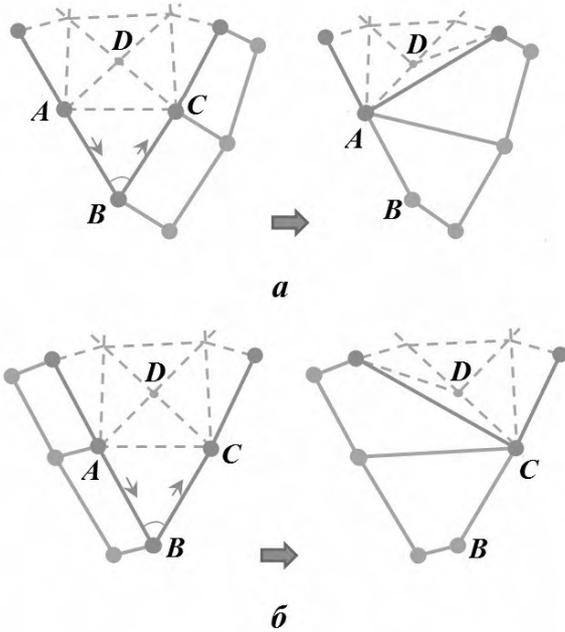


Рис. 10. Схема объединения ребер фронта, когда одно ребро принадлежит границе: а — со схлопыванием влево; б — со схлопыванием вправо

Локальное сглаживание. Сглаживание является одним из наиболее используемых инструментов в процессе построения сеток. Процедура сглаживания способствует улучшению качества элементов посредством придания ячейкам более правильной формы без изменения топологии сетки. Сглаживание осуществляется как в процессе построения ячеек, каждый раз после формирования нового четырехугольного элемента (локальное), так и уже на полностью построенной четырехугольной сетке (глобальное). В работе [7] при выполнении процедур локального и глобального сглаживания используется хорошо себя зарекомендовавший и простой в реализации метод Лапласа [9].

На практике выяснилось, что для областей сложной конфигурации локальный метод Лапласа не достаточен для получения сеток хорошего качества. Поэтому за основу процедуры локального сглаживания взят многоэтапный алгоритм, предложенный Т. Блэкером и М. Стивенсоном в [8]. Тестовые расчеты выявили ряд проблем и недостатков в используемом алгоритме многоэтапного локального сглаживания, которые, в свою очередь, потребовали изменения некоторых этапов алгоритма.

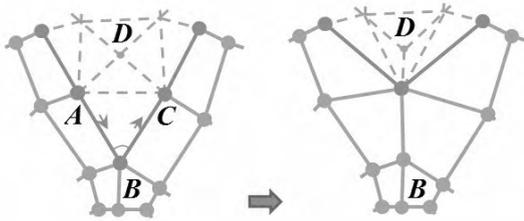


Рис. 11. Схема объединения не принадлежащих границе ребер фронта

Первым важным изменением является введение очередности при обработке узлов сглаживаемой ячейки. На рис. 12 красным цветом выделена вновь построенная ячейка, а зеленым обведены обрабатываемые в этой ячейке узлы. Рассматриваются узлы двух типов в зависимости от количества четырехугольников, в которые эти узлы входят, — одного или двух.

Наивысшим приоритетом при обработке обладают узлы, входящие только в один четырехугольник, т. е. принадлежащие только обрабатываемой четырехугольной ячейке (узел 1 на рис. 12). Затем обрабатываются узлы, входящие в две четырехугольные ячейки со смежным ребром (узел 2 на рис. 12). Положение всех остальных узлов ячейки в процессе сглаживания не корректируется. Данный подход способствует улучшению сохранения формы и размера как уже построенных четырехугольных ячеек, так и вновь образованных.

Помимо изменения подхода к порядку обработки узлов, модификации подверглись также и некоторые этапы их обработки.

На рис. 13 показана корректировка положения узла первого типа (входящего только в одну четырехугольную ячейку). Сначала определяются оптимальные положения ребер обрабатываемой ячейки, содержащих рассматриваемый узел i . Ребра P_{ij} и P_{ik} поворачиваются в зависимости от фронтальных углов α и β , также производится корректировка их длин [8]. Для узла i вычисляются приращения координат $\Delta\vec{v}_{ij}$ и $\Delta\vec{v}_{ik}$ от ребер P_{ij} и P_{ik} соответственно. Итоговое приращение, определяющее искомое положение узла, вычисляется по формуле

$$\Delta\vec{v} = \frac{\Delta\vec{v}_{ij} + \Delta\vec{v}_{ik}}{2}.$$

Также внесены изменения в процедуру обработки узлов второго типа. На рис. 14 показано положение узла i после поворота и начальной

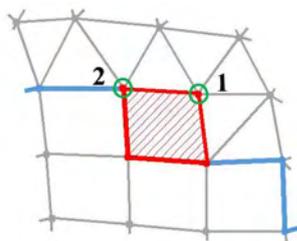


Рис. 12. Узлы новой ячейки, обрабатываемые в процедуре локального сглаживания

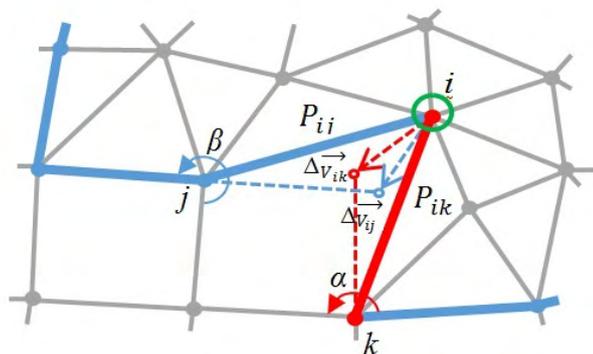


Рис. 13. Определение положения узла, принадлежащего только одной четырехугольной ячейке

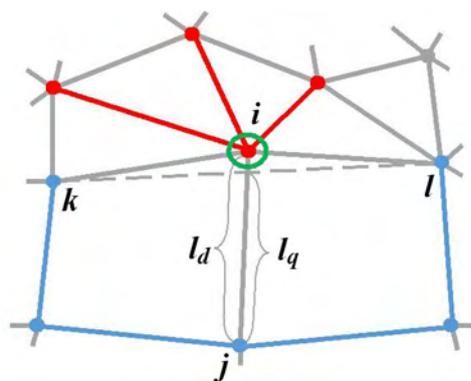


Рис. 14. Дополнительная корректировка длины ребра между узлами i и j

корректировки длины инцидентного ребра P_{ij} (между узлами i и j) [8]. Так называемая желаемая длина l_d вычисляется с помощью длин ребер четырехугольных и треугольных ячеек следующим образом:

$$l_d = \frac{l_{quad} + \sum_{i=1}^{N_{tr}} l_i}{4 + N_{tr}},$$

где l_{quad} — сумма длин ребер, не содержащих корректируемый узел и входящих в четырехугольные ячейки (выделены синим цветом); l_i и N_{tr} — соответственно длины ребер треугольных ячеек, содержащих узел i и не являющихся ребрами четырехугольников (выделены красным цветом), и их количество.

На конечном этапе обработки узла i происходит еще одна корректировка длины смежного ребра $|P_{ij}|$, которая выполняется по следующему правилу (см. рис. 14):

$$|P_{ij}| = \begin{cases} \frac{l_q + l_d}{2}, & \text{если } l_d > l_q; \\ l_d, & \text{если } l_d \leq l_q. \end{cases}$$

Здесь l_q — расстояние от узла j до точки пересечения отрезков, соединяющих узлы i и j , k и l . Данная корректировка способствует минимальному искажению фронта при построении новых элементов.

Топологическая оптимизация. Для реализации процедур топологической оптимизации были использованы результаты работ [7, 10–12]. Основная цель топологической оптимизации — уменьшить количество *нерегулярных* узлов и их концентрацию, особенно около границ области. Регулярность узла определяется его валентностью — количеством выходящих из узла ребер. Для четырехугольных сеток регулярным является четырехвалентный узел. Для граничных узлов валентность определяется с учетом внутреннего угла α между двумя инцидентными узлу граничными ребрами и числа исходящих ребер N_r .

Граничный узел считается регулярным, если его валентность V_r соответствует следующим значениям:

$$V_r = \begin{cases} 2, & \text{при } \alpha < \frac{2\pi}{3}; \\ 3, & \text{при } \frac{2\pi}{3} \leq \alpha < 1,206\pi; \\ 4, & \text{при } 1,206\pi \leq \alpha < 1,717\pi; \\ 5, & \text{при } 1,717\pi \leq \alpha. \end{cases}$$

Окончательно валентность граничного узла вычисляется по следующей формуле:

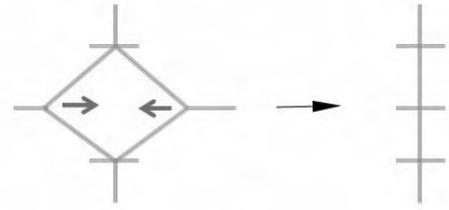
$$V = 4 + N_r - V_r.$$

Следуя [10], большинство операций топологической оптимизации сводится к последовательному применению базовых операций: схлопыванию/раскрытию четырехугольника, добавлению/удалению двухвалентного узла. Данные операции проиллюстрированы на рис. 15, 16.

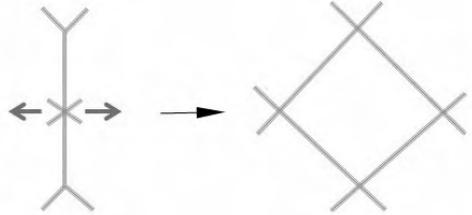
На рис. 17, 18 показаны примеры комбинаций операций схлопывания для удаления нерегулярных узлов.

Операции оптимизации выполняются в соответствии с уменьшением приоритета устранения нерегулярности узлов в следующем порядке:

- 1) удаление двухвалентных узлов;



a



б

Рис. 15. Операции схлопывания (a) и раскрытия (б) четырехугольника

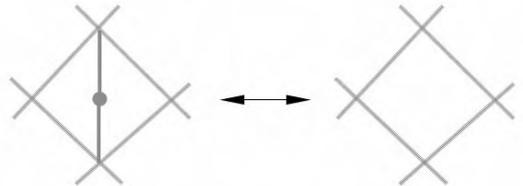


Рис. 16. Операции добавления/удаления двухвалентной вершины

- 2) удаление нерегулярных граничных узлов;
- 3) удаление трехвалентных узлов;
- 4) удаление шестивалентных узлов. На данном этапе выполняется операция раскрытия четырехугольника по ребрам, имеющим минимальную сумму валентностей узлов;
- 5) процедура релаксации, направленная на удаление комбинаций трех- и пятивалентных узлов [7].

После каждой операции на этапах 3–5 проверяется возникновение нерегулярных соседних узлов и при их наличии вызывается соответствующая процедура их удаления. Если в результате операции нарушается регулярность граничных узлов, то операция отменяется.

После ряда операций топологической оптимизации вызывается процедура сглаживания для затронутых узлов. По окончании всего цикла производится сглаживание всей полученной сетки.

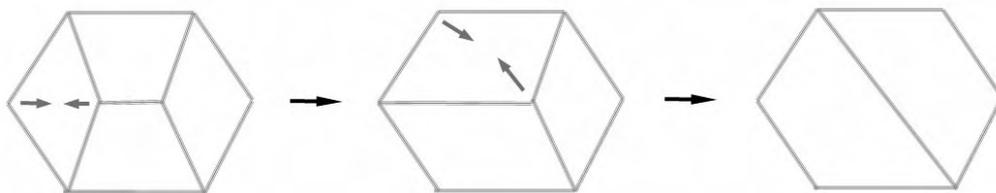


Рис. 17. Удаление двух соседних трехвалентных узлов [12]

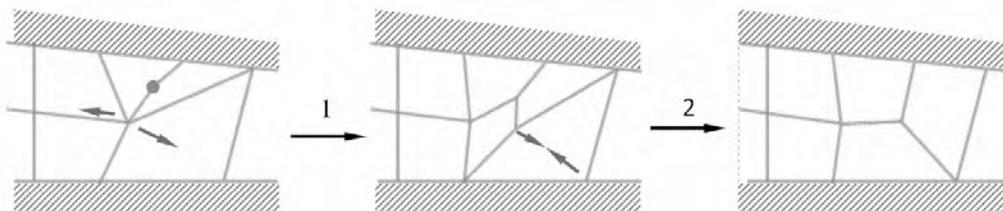


Рис. 18. Удаление двухвалентного узла около границы с помощью двойной операции раскрытия (1) и схлопывания (2) четырехугольника [10]

4. Примеры построения сеток

Рассмотрим примеры построенных сеток для нескольких геометрических моделей произвольной формы. На рис. 19 представлены результаты построения четырехугольных сеток с использованием генератора, реализующего базовые процедуры алгоритма [7], и модифицированного генератора. Видно, что внесенные в алгоритм изменения качественно улучшают получаемые

сетки. Наблюдается ортогональность ячеек с границей области, размер ячеек более равномерный, сетка имеет меньше измельчений и более регулярный характер.

На рис. 20 показаны результаты работы генератора поверхностных неструктурированных четырехугольных сеток в препроцессоре ПП "Логос", полученные для одной из граней пространственной модели на этапах построения 3–5 (см. разд. 1).

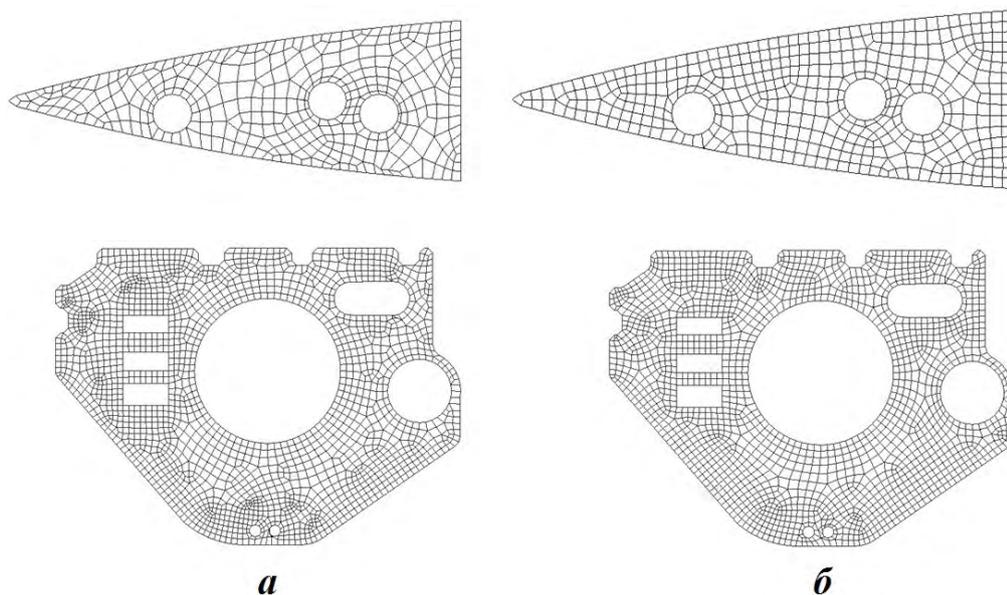


Рис. 19. Четырехугольные сетки, построенные по базовым процедурам алгоритма [7] (а) и по модифицированному алгоритму (б)

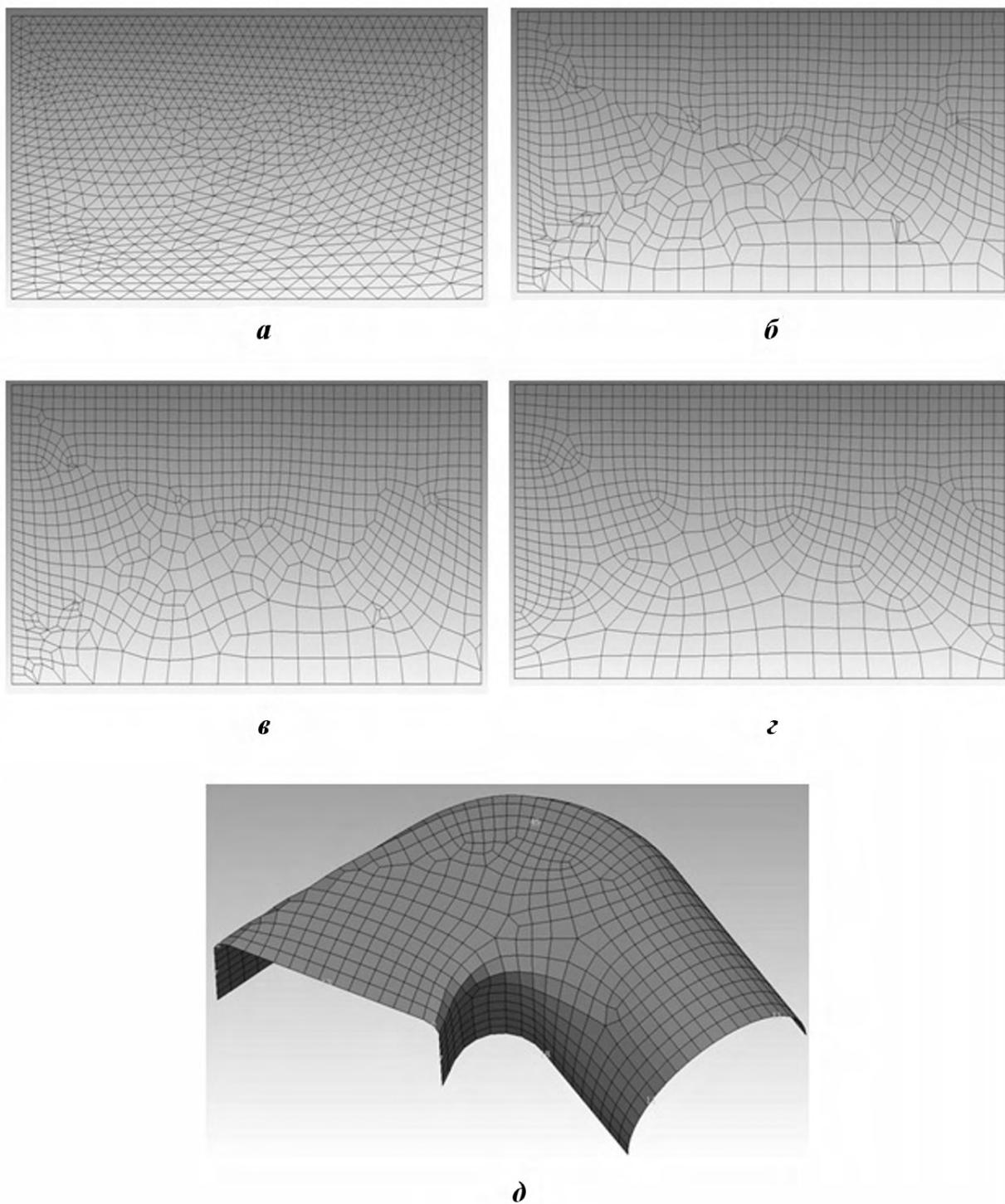


Рис. 20. Поэтапное построение поверхностной сетки на грани пространственной геометрической модели: *a* — треугольная сетка в параметрической плоскости на этапе 3; *б* — четырехугольная сетка в параметрической плоскости на этапе 4 после построения; *в* — четырехугольная сетка в параметрической плоскости после сглаживания на этапе 4; *г* — четырехугольная сетка в параметрической плоскости после топологической оптимизации на этапе 4; *д* — обратное отображение сетки из параметрической плоскости в пространство R^3 на этапе 5

Заключение

Приведено описание автоматического генератора поверхностных неструктурированных четырехугольных сеток в произвольных областях. Используется методика, позволяющая строить поверхностные сетки при помощи плоского генератора с учетом римановой метрики поверхности для контроля размера и формы четырехугольников в параметрическом пространстве. Данный генератор поверхностных сеток включен в состав препроцессора ПП "Логос".

Представлены значимые изменения, коснувшиеся алгоритмов формирования ячеек и обработки фронтов и позволившие улучшить качество отстраиваемых сеток.

Разработаны и реализованы новые алгоритмы локального сглаживания и оптимизации.

Дальнейшее развитие созданного средства генерации поверхностной сетки планируется в направлении улучшения качества имеющихся и реализации новых алгоритмов генерации и оптимизации сеток.

Список литературы

1. *Bame K.-Ю.* Методы конечных элементов. М.: Физматлит, 2010.

Bate K.-Yu. Metody konechnykh elementov. М.: Fizmatlit, 2010.

2. Пакет программ инженерного анализа и суперкомпьютерного моделирования. <http://logos.vniief.ru>.

Paket programm inzhenernogo analiza i superkompyuternogo modelirovaniya. <http://logos.vniief.ru>.

3. *Owen S. J., Staten M. L., Cannan S. A., Saigal S.* Advancing front quadrilateral meshing using triangle transformations // Proc. 7th Int. Meshing Roundtable. Dearborn, Michigan, Sandia National Labs. October 26–28, 1998. P. 409–428.

4. *Tristano J. R., Owen S. J., Canann S. A.* Advancing Front Surface Mesh Generation in Parametric Space Using a Riemannian Surface Definition. ANSYS, Inc., 1999.

5. Официальный сайт ANSYS. <http://www.ansys.com>.

Ofitsialny sayt ANSYS. <http://www.ansys.com>.

6. *Сковпень А. В.* Реализация фронтального алгоритма построения нерегулярных четы-

рехугольных сеток // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 1. С. 9–30.

Skovpen A. V. Realizatsiya frontalnogo algoritma postroeniya neregulyarnykh chetyrekhugolnykh setok // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2005. Вып. 1. С. 9–30.

7. *Лысухин В. И.* Построение сетки четырехугольных элементов на произвольных областях // Прикладная геометрия (электронный журнал МАИ). 2010. Вып. 12, № 24. С. 1–29. <http://www.mai.ru>.

Lysukhin V. I. Postroenie setki chetyrekhugolnykh elementov na proizvolnykh oblastiakh // Prikladnaya geometriya (elektronny zhurnal MAI). 2010. Вып. 12, № 24. С. 1–29. <http://www.mai.ru>.

8. *Blackler T. D., Stephenson M. B.* Paving: a new approach to automated quadrilateral mesh generation // Int. J. Num. Meth. Eng. 1991. Vol. 32. P. 811–847.

9. *Field D. A.* Laplacian smoothing and delaunay triangulation // Communications in Applied Numerical Methods. 1988. Vol. 4. P. 709–712.

10. *Canann S. A., Muthukrishnan S. N., Phillips R. K.* Topological improvement procedures for quadrilateral finite element meshes // Engineering with Computers. 1998. Vol. 14. P. 168–177.

11. *Kinney P.* CleanUp: Improving Quadrilateral Finite Element Meshes // Proc. 6th Int. Meshing Roundtable. Ford Motor Company, 1997. P. 449–467.

12. *Lo S. H.* Finite Element Mesh Generation. CRC Press, 2015.

13. *Беспалов А. П., Логинов Д. В., Соловьёв А. Н.* Метод автоматического построения поверхностных сеток на гранях геометрических моделей в параметрическом представлении в препроцессоре пакета программ "Логос" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 3. С. 23–39.

Bespalov A. P., Loginov D. V., Solovyev A. N. Metod avtomaticheskogo postroeniya poverkhnostnykh setok na granyakh geometricheskikh modeley v parametricheskom predstavlenii v

- preprotsessore paketa programm "Logos" // Вопросы атомной науки и техники. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2020. Вып. 3. S. 23–39.
14. Геометрическое ядро C3D. <https://c3dlabs.com/ru/products/c3d-kernel/>.
Geometricheskoe yadro C3D. <https://c3dlabs.com/ru/products/c3d-kernel/>.
15. Генератор сеток CM2 MeshTools. <https://www.computing-objects.com/cm2-meshtools-suite/>.
Generator setok CM2 Mesh Tools. <https://www.computing-objects.com/cm2-meshtools-suite/>.
- Статья поступила в редакцию 17.03.20.

AN UNSTRUCTURED QUADRILATERAL SURFACE grid GENERATOR IN THE PREPROCESSOR OF THE "LOGOS" SOFTWARE PACKAGE / E. Yu. Arapova, V. G. Kudelkin, E. A. Pavlov, S. Yu. Polyakova, A. V. Tikhonov (FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region).

The paper describes an automatic generator of unstructured quadrilateral surface grids for geometric models in the parametric representation. The grid generator is intended to construct grid models for strength simulations in the "Logos" software package. Steps of the surface grid generation procedure are considered. One of the steps, namely generation of quadrilateral grids in the parametric plane, is discussed in detail. Examples of test problems with different geometric models are given.

Keywords: "Logos" software package, quadrilateral surface grids, moving front, generation of quadrilateral cells, local smoothing, topological optimization.

УДК 519.6

О РАЗВИТИИ АЛГОРИТМА ПОСТРОЕНИЯ СЕТОК В ДЕФОРМИРОВАННЫХ ТЕЛАХ ВРАЩЕНИЯ ДЛЯ СЛУЧАЯ ИХ ДЕФОРМАЦИИ ТЕЛАМИ ВРАЩЕНИЯ, ОБРАЗОВАННЫМИ НЕСКОЛЬКИМИ ПОВЕРХНОСТЯМИ

Н. А. Артёмова, О. В. Ушакова
(ИММ УрО РАН, г. Екатеринбург)

Рассматривается задача о построении сетки в телах вращения, деформированных телами вращения. Тело вращения может быть образовано цилиндрическими, коническими или сферическими поверхностями и формироваться вращением элементов двух видов — отрезков прямых и дуг окружностей. Ранее был предложен нестационарный алгоритм построения сеток в областях с подвижными деформируемыми границами для частных случаев деформации цилиндром, конусом и сферой. В данной работе рассматривается дальнейшее развитие нестационарного алгоритма. Суть алгоритма осталась прежней, усложнилась реализация в связи с усложнением конструкции деформирующей области. Приводятся примеры расчетов сеток.

Ключевые слова: структурированные сетки, область вращения, деформированная область вращения, оптимальные сетки, подвижные сетки.

Введение

В работе [1] предложен алгоритм построения структурированных сеток в телах (областях) вращения, деформированных другими телами вращения. Способ получения таких деформированных тел можно описать следующим образом. Деформация тела вращения осуществляется давлением на него другим телом вращения, в процессе которого деформирующее тело движется (давит) на основное в направлении заданного вектора деформации до нужной степени. В результате такого воздействия основное тело вращения деформируется: его граница в области деформации вдавлируется и принимает форму вспомогательного (деформирующего) тела, при этом форма остальной части основного тела (не подвергшейся воздействию) не изменяется. Под телом вращения понимается тело, которое получается поворотом плоской образующей кривой, состоящей из отрезков прямых и дуг окружностей, на 180° . Следует отметить отличие такого тела вращения от классического, получаемого поворотом плоской образующей кривой на 360° . Оно подробно описано в [2].

Предложенный в [1] алгоритм был реализован для частных случаев давления на тело вращения конусом, цилиндром и сферой. Это дало основу для разработки алгоритма в более общем случае. В данной работе описывается дальнейшее развитие алгоритма для случаев, когда давящая поверхность деформирующего тела вращения может состоять из кусков цилиндрических, конических и сферических поверхностей.

Сетки в деформированных областях применяются для математического моделирования задач многокомпонентной гидродинамики [3]. Один из подходов, используемый для построения сеток в деформированных телах вращения, — это метод отображений [4]. Согласно данному методу построение структурированных сеток в области геометрически сложной формы, называемой физической, осуществляется с помощью невырожденного отображения области более простой формы, называемой параметрической. В трехмерном случае параметрическая область — это прямоугольный параллелепипед. Соответственно физическая область представляется в виде криволинейного

шестигранника, а структурированная сетка в физической области является образом равномерной ортогональной сетки в прямоугольном параллелепипеде для этого отображения. Способ представления области в виде криволинейного шестигранника определяет конфигурацию области. Для построения сетки задается фиксированная (стационарная) геометрия деформированной области вращения и ее конфигурация, а отображение, определяющее сетку в заданной области, находится с помощью нестационарного алгоритма.

Используемый способ построения сеток в рассматриваемых областях относится к методам построения подвижных сеток, когда в процессе построения сетка движется, подстраиваясь под нестационарный процесс решаемой физической задачи: либо под ее решение, либо под меняющуюся геометрию области (см. [5–7]). В данном случае искусственно (без связи с физикой решаемой задачи) организуется специальный нестационарный процесс, в котором меняется геометрия области и осуществляется адаптация к ней сетки так, чтобы в конечном итоге были получены заданная (заранее зафиксированная) геометрия области и сетка для нее.

Процесс осуществляется итерационно. На каждой итерации незначительно меняется граница основного тела (физической области) так, чтобы в сетке не возникало самопересечений: деформирующее тело сдвигается, и узлы сетки основного тела, попавшие внутрь деформирующего (деформированные граничные узлы), проецируются на него. Другими словами, деформированные граничные узлы перемещаются на деформирующую поверхность. Затем выполняется оптимизация сетки вариационным методом [8]. Такой процесс осуществляется до тех пор, пока деформация основного тела не достигнет нужной степени и не будет получено деформированное тело, в котором требуется построить структурированную сетку.

Выбранный подход представляется наиболее естественным, так как технология построения сеток в телах вращения уже создана [9]. Для областей вращения в работах [2, 8, 10, 11] предложен метод построения структурированных сеток, в котором созданы средства описания геометрий и конфигураций областей вращения, алгоритмы построения начальных сеток и их оптимизации.

Отметим, что аналогичный нестационарный алгоритм может быть применен и для построения неструктурированных сеток в деформированных телах вращения. В этом случае этап деформации совпадает по своей сути с этапом деформации для структурированных сеток, а для этапа оптимизации должен применяться алгоритм построения сеток, удовлетворяющих требуемым критериям качества применительно к неструктурированным сеткам.

Разработка нестационарных алгоритмов построения подвижных сеток всегда являлась сложной задачей, так как в ней, помимо организации самого нестационарного процесса, в данном случае изменения геометрии области, необходимо на каждом его шаге обеспечить построение невырожденной сетки [12], удовлетворяющей заданным критериям качества (равномерность, ортогональность) [8].

В разд. 1 приводится постановка задачи о построении сетки в области вращения, деформированной другой областью вращения. В разд. 2 описывается нестационарный алгоритм построения сеток в деформированных областях. В разд. 3 приводятся примеры расчетов сеток.

1. Постановка задачи

Для удобства и полноты описания повторим постановку задачи из [1].

В декартовой системе координат X с осями x^1, x^2, x^3 задана односвязная область U_x с границей (образующей) ∂U_x , состоящей из отрезков прямых и дуг окружностей, называемых (см. [2, 10, 13, 14]) элементами образующей. При повороте образующей ∂U_x на угол $\varphi = \pi$ вокруг оси x^3 получается область G_x с границей ∂G_x . В декартовой системе координат Ξ с осями ξ^1, ξ^2, ξ^3 задана образующая ∂W_ξ , также состоящая из элементов (отрезков прямых и дуг окружностей), которая при повороте вокруг оси ξ^3 на угол $\varphi = \pi$ формирует поверхность вращения ∂Q_ξ . Двумерная область U_x определяет трехмерную область вращения G_x (рис. 1, а), которую далее будем называть основной, а двумерная область W_ξ — область вращения Q_ξ (рис. 1, б), которую далее будем называть деформирующей или вспомогательной.

Заданы также координаты точки $O_{\Xi X} = (x^1_0, x^2_0, x^3_0)$ начала координат системы Ξ в системе координат X и матрица C перехода от системы координат Ξ к системе координат X . Столбцы матрицы

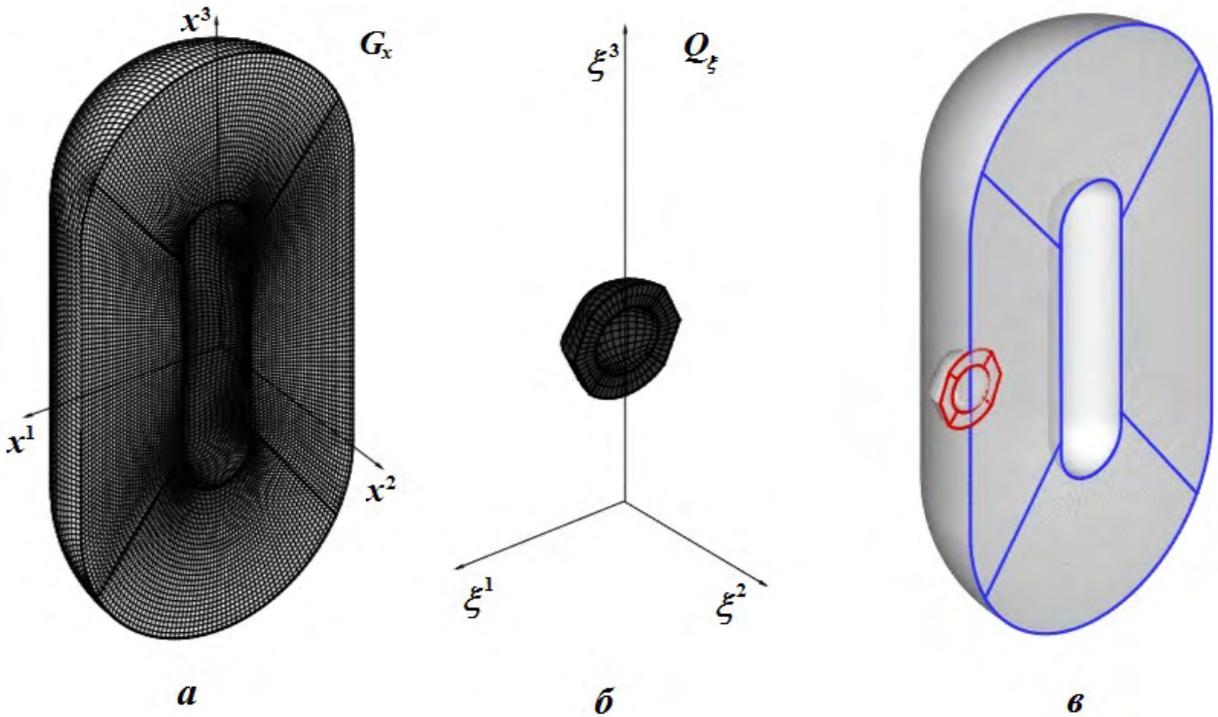


Рис. 1. Построение деформированной сетки: *a* — сетка в области G_x ; *б* — область Q_x ; *в* — области Q_x и G_x в системе координат X

C — координаты ортов ξ^1, ξ^2, ξ^3 в системе координат X . Задан вектор деформации \vec{V}_X в системе координат X , указывающий направление вдавливания деформирующего тела в основное. Точка $O_{\Xi X}$ и матрица C задаются таким образом, чтобы после преобразования координат деформирующее тело Q_x приняло окончательное положение внутри основного (рис. 1, в).

В основном теле G_x по алгоритмам [10, 14] построена структурированная трехмерная сетка $T \left\{ \left(x_{ijk}^1, x_{ijk}^2, x_{ijk}^3 \right) \right\}$, $i = 0, \dots, N - 1$, $j = 0, \dots, M - 1$, $k = 0, \dots, L - 1$. Задача состоит в построении структурированной сетки \bar{T} в деформированной области $D = G_x \setminus (G_x \cap Q_x)$ таким образом, чтобы узлы деформированной части границы сетки T переместились на деформирующую поверхность ∂Q_x .

2. Нестационарный алгоритм построения сеток в деформированных телах вращения

Выполнение нестационарного алгоритма, как и в [1], начинается с подготовительного этапа, на котором также определяются параметры деформирующего тела и осуществляется выдвигание основного тела из деформирующего. Но если в [1] параметром деформирующего тела был его вид (цилиндр, конус или сфера), то в новом алгоритме необходимо определить количество и вид элементов образующей, формирующих деформирующее тело (подразд. 2.1).

Заданная форма основного тела деформируется постепенно, итерационно, пока деформация не достигнет необходимой степени (деформирующее тело Q_x примет окончательное положение внутри основного тела (см. рис. 1, в)). Так же, как в [1], итерационный процесс состоит из двух этапов — деформации сетки в основной области (подразд. 2.2) и оптимизации деформированной сетки (подразд. 2.3). Снова на этапе деформации выбирается такая скорость (шаг) деформации, чтобы деформация не затронула внутренние узлы сетки. При этом граничные узлы, подвергшиеся деформации, проецируются на деформирующую поверхность. Но в отличие от [1], так как в новом алгоритме деформирующая поверхность может состоять из нескольких поверхностей вращения, нужно определить, на какую из поверхностей будет проецироваться каждый из узлов, подвергшихся деформации.

На этапе оптимизации сетка перестраивается в соответствии с заданными критериями качества, как и в [1].

2.1. Подготовительный этап. Все рассматриваемые тела вращения (основные и вспомогательные) можно условно разделить (см. [10, 14]) на три типа: *полное тело вращения*, *оболочка* и *срезанная оболочка*. Тип тела вращения определяется типом образующей (рис. 2). Образующая в зависимости от типа делится на определенное количество блоков (см. [10]) — участков, концы которых называются далее вершинами. Так, образующая полного тела вращения делится на 4 блока (на рис. 2, а это блоки между вершинами 1 и 2, 2 и 3, 3 и 4, 4 и 1), образующая оболочки — на 6 блоков (см. рис. 2, б). Каждый из блоков может состоять из одного или нескольких элементов — отрезков прямых или дуг окружностей. На подготовительном этапе сначала в зависимости от типа тела вращения определяются номера и количество блоков и элементов образующей, формирующих деформирующую (давящую) поверхность.

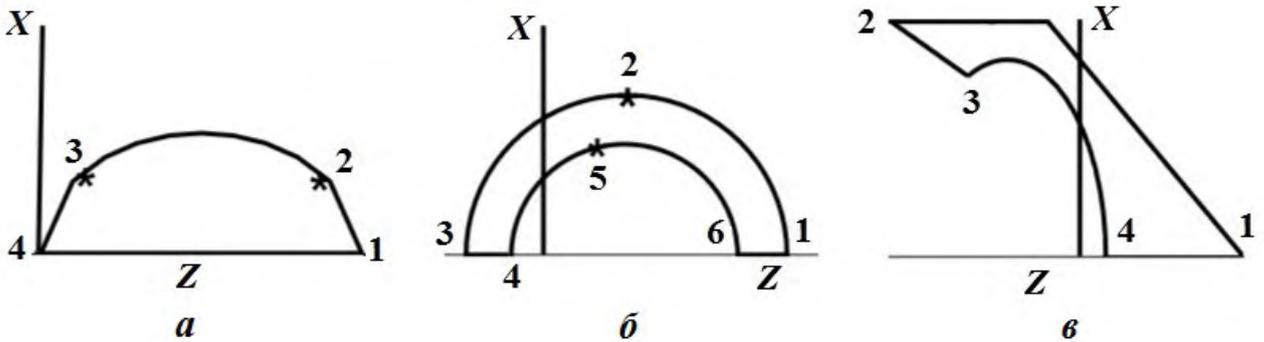


Рис. 2. Типы образующих: а — для полного тела вращения; б — для тела типа *оболочка*; в — для тела типа *срезанная оболочка*

В то же время от типа образующей зависит, какие блоки формируют давящую (деформирующую) поверхность, а какие — внутреннюю (не влияющую на деформацию). Для полного тела вращения деформирующую поверхность формируют три блока (на рис. 2, а это блоки между вершинами 1 и 2, 2 и 3, 3 и 4), для оболочки — два блока (на рис. 2, б — блоки между вершинами 1 и 2, 2 и 3), для срезанной оболочки — один блок (на рис. 2, в блок между вершинами 1 и 2).

На рис. 3 изображена образующая для тела вращения типа *оболочка*. Здесь деформирующую поверхность формируют два блока: DBA и AB_1D_1 . Каждый из этих блоков состоит из двух элементов (DB и BA ; AB_1 и B_1D_1).

Определение деформирующей поверхности в зависимости от образующей AB приведено в работе [1]. Имеют место формулы:

– для прямого кругового цилиндра

$$(\xi^1)^2 + (\xi^2)^2 \leq R^2, \quad \xi^3 \in [\xi_{\min}^3, \xi_{\max}^3], \quad R = \xi_A^1 = \xi_B^1 \quad (1)$$

(здесь и далее $\xi_{\min}^3 = \min(\xi_A^3, \xi_B^3)$; $\xi_{\max}^3 = \max(\xi_A^3, \xi_B^3)$);

– для сферы с центром в точке $(0, 0, 0)$

$$(\xi^1)^2 + (\xi^2)^2 + (\xi^3)^2 \leq R^2; \quad (2)$$

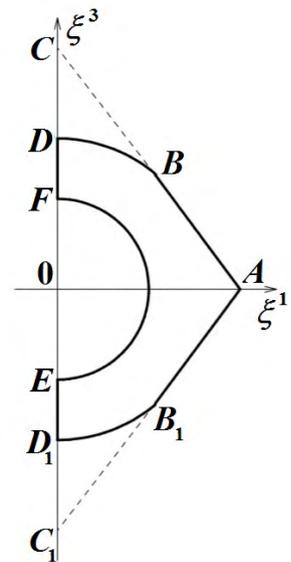


Рис. 3. Образующая для тела вращения

– для конуса

$$(\xi^1)^2 + (\xi^2)^2 \leq t^2 (\xi^3 - \xi_v^3)^2, \quad \xi^3 \in [\xi_{\min}^3, \xi_{\max}^3], \quad t = \frac{\xi_B^1 - \xi_A^1}{\xi_B^3 - \xi_A^3}, \quad (3)$$

где ξ_v^3 – координата вершины конуса.

Заметим, что в работе [1] для конуса одна из координат ξ_A^1 или ξ_B^1 должна равняться нулю. Другими словами, один из концов отрезка AB должен был являться вершиной конуса, т. е. усеченный конус не рассматривался. В данной работе это ограничение снято.

Таким образом, в работе [1] вершина конуса выбиралась из точек A и B в зависимости от угла наклона прямой, проходящей через эти точки, к оси ξ^3 :

$$\xi_v^3 = \begin{cases} \xi_{\min}^3, & \text{если } t > 0; \\ \xi_{\max}^3, & \text{если } t < 0. \end{cases}$$

На рис. 4, слева, $t > 0$, следовательно, $\xi_v^3 = \xi_{\min}^3 = \xi_A^3$.

В новой версии программы конус может быть усеченным, а координата вершины конуса вычисляется по формуле

$$\xi_v^3 = \xi_{\min}^3 - \frac{\xi_{\max}^3 - \xi_{\min}^3}{\xi_{\max}^1 - \xi_{\min}^1} \xi_{\min}^1. \quad (4)$$

Для случая, показанного на рис. 4, слева $\xi_{\min}^1 = \xi_A^1 = 0 \Rightarrow \xi_v^3 = \xi_{\min}^3 = \xi_A^3$ ($t > 0$). При $t < 0$ и $\xi_{\max}^1 = 0 \Rightarrow \xi_v^3 = \xi_{\max}^3$. Формула (4) получена из формулы прямой, проходящей через две точки $(\xi_{\min}^1, \xi_{\min}^3)$ и $(\xi_{\max}^1, \xi_{\max}^3)$:

$$\frac{\xi^1 - \xi_{\min}^1}{\xi_{\max}^1 - \xi_{\min}^1} = \frac{\xi^3 - \xi_{\min}^3}{\xi_{\max}^3 - \xi_{\min}^3}$$

при условии $\xi^1 = 0$.

На рис. 3 представлена образующая $DBAB_1D_1EFD$ для тела вращения, изображенного на рис. 1, б. При вращении участков BA и AB_1 получаются усеченные конусы. Вершины этих конусов находятся соответственно в точках C и C_1 .

В работе [1] рассмотрены три вида деформирующего тела – цилиндр, конус и сфера. Вид деформирующего тела определялся по первому элементу образующей. В данной работе рассматривается случай деформации телом, давящая поверхность которого состоит из нескольких поверхностей вращения. Она может состоять из кусков конических, цилиндрических и сферических поверхностей и формируется вращением плоской образующей, состоящей из отрезков прямых и дуг окружностей.

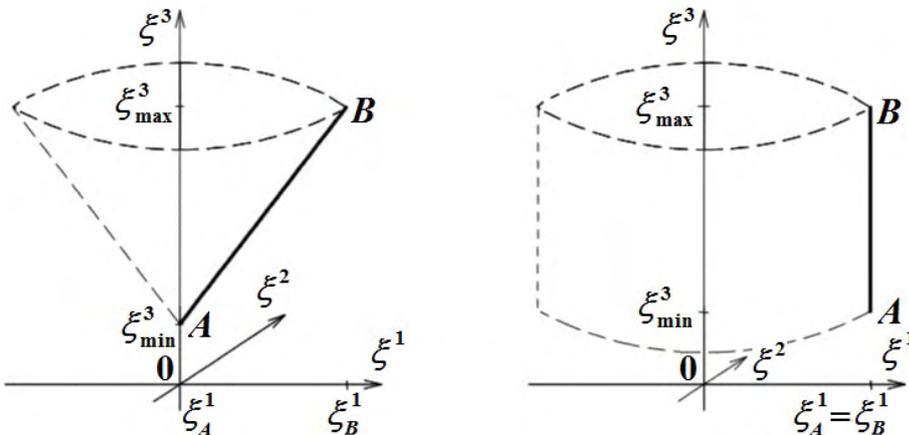


Рис. 4. Конус и цилиндр, полученные вращением отрезка AB

Таким образом, каждый элемент образующей формирует соответствующую поверхность вращения и принадлежность узлов сетки деформирующему телу будет проверяться с помощью условий (1)—(3).

Так как при осуществлении процесса деформации узлы основного тела попадают внутрь деформирующего, заданного в системе координат Ξ , и их принадлежность телу должна проверяться, для упрощения реализации алгоритма наиболее естественным представляется все расчеты проводить в системе координат Ξ , а не X , чтобы иметь возможность воспользоваться формулами (1)—(3). При такой реализации (проверке условий (1)—(3)) деформирующее тело должно оставаться неподвижным (координаты деформирующей поверхности ∂Q_ξ не должны меняться в процессе деформации), вместо этого должно сдвигаться основное тело. Для этого все величины, заданные в системе координат основного тела (начало координат, вектор деформации, а также координаты всех узлов основной области G_x) необходимо перевести в систему координат Ξ по формулам, приведенным в [1].

В результате этих преобразований основное тело примет свое конечное положение (деформирующее тело окажется внутри основного, см. рис. 1, в). Чтобы выдвинуть основное тело так, чтобы не было пересечения (касание допускается) с деформирующим телом перед началом деформации, используется алгоритм, подробно описанный в работе [1]. После этого можно начать деформировать основное тело.

2.2. Этап деформации. Так же, как и в [1], для выполнения алгоритма деформации требуется определить ее скорость (шаг деформации). Первоначально шаг полагается равным среднему расстоянию между узлами сетки T . Все узлы сетки сдвигаются в направлении вектора \vec{V}_Ξ на этот шаг.

Суть алгоритма осталась та же, что и в [1]. Нужно основное тело сдвинуть в направлении деформирующего так, чтобы при деформации в сетке основного тела не возникло самопересечений. Это можно реализовать следующим образом: сдвиг должен быть таким, чтобы основное тело пересекало деформирующее только своей границей, а поверхность, образованная приграничными узлами, деформирующее тело не пересекала. В [1] деформирующая поверхность формировалась вращением только одного элемента и проверка пересечения, а затем проецирование граничных узлов основного тела на деформирующую поверхность осуществлялись только для этого элемента. В более общем случае, так как деформирующая поверхность состоит из нескольких поверхностей вращения, реализация алгоритма усложнилась.

В новом алгоритме организуется процесс перебора элементов, причем он выполняется по крайней мере дважды. При переборе элементов проверяется одно из условий (1), (2) или (3) в зависимости от вида элемента. В первый раз этот процесс осуществляется для проверки пересечения каждой поверхности вращения деформирующей поверхности с приграничной поверхностью основного тела. Если приграничная поверхность основного тела G_ξ пересекла деформирующую поверхность ∂Q_ξ (для какого-либо узла выполнилось одно из условий (1), (2) или (3)), то шаг делится пополам. Узлы сетки \bar{T} сдвигаются с этим меньшим шагом, и снова проверяется наличие пересечения. Если поверхности не пересекаются, то приемлемый шаг выбран, координаты всех узлов сетки \bar{T} основного тела изменяются на значение выбранного шага и процесс перебора элементов осуществляется снова. Теперь поверхности вращения, составляющие деформирующую поверхность, проверяются на пересечение с границей основного тела. Узлы основного тела, для которых выполнилось условие (1), (2) или (3), проецируются на соответствующую поверхность вращения. Проецирование узлов осуществляется по алгоритму О. В. Ушаковой, подробное описание которого представлено в [2, 11, 13].

После проецирования всех узлов, подвергшихся деформации, на деформирующую поверхность, сетка оптимизируется. Для этого осуществляется обратное преобразование координат в систему координат X .

2.3. Этап оптимизации. Алгоритм оптимизации сетки подробно описан в [1]. Оптимизация сетки осуществляется вариационным методом [8, 13], где в качестве критериев оптимальности рассматриваются критерии близости сетки к равномерной и ортогональной. Оптимизация может выполняться как в деформированном фрагменте, так и во всей области.

3. Примеры расчетов сеток

Для тестирования предложенного нестационарного алгоритма использовались различные вспомогательные тела и направления деформации.

Расчет 1. Рассматривается деформация, представленная на рис. 1; образующая изображена на рис. 3. Деформация областью, составленной из сегментов сфер и конусов, вместе с построенной сеткой показана на рис. 5. Для расчета потребовалось 76 шагов деформации, по 100 итераций оптимизации на каждом шаге. Оптимизация осуществлялась во фрагменте.

Расчеты 2 и 3. Деформация осуществлялась тем же телом, что и в расчете 1, но было изменено ее направление. В расчете 2 тело деформировалось справа (рис. 6, *а*), при этом деформирующее тело повернуто на 90° , а в расчете 3 — сзади (рис. 6, *б*). В этих расчетах на каждом шаге деформации выполнялось по 100 итераций оптимизации, причем на 100-м шаге оптимизация проводилась во всей области, на остальных шагах — только в деформированном фрагменте. Для полной деформации сетки в расчете 2 потребовалось 103 шага, в расчете 3 — 156 шагов.

Расчет 4. Деформация выполнялась сзади телом, составленным из цилиндра и конуса (рис. 7). Для расчета потребовалось 97 шагов деформации, по 200 итераций оптимизации на каждом шаге. На 50-м шаге сетка оптимизировалась во всей области, на остальных шагах — во фрагменте.

Расчет 5. Деформация выполнялась сзади телом, составленным из цилиндра и двух конусов (рис. 8). Потребовалось 143 шага деформации. Параметры расчета следующие: 80 итераций оптимизации, оптимизация во всей области проводилась на 100-м шаге.

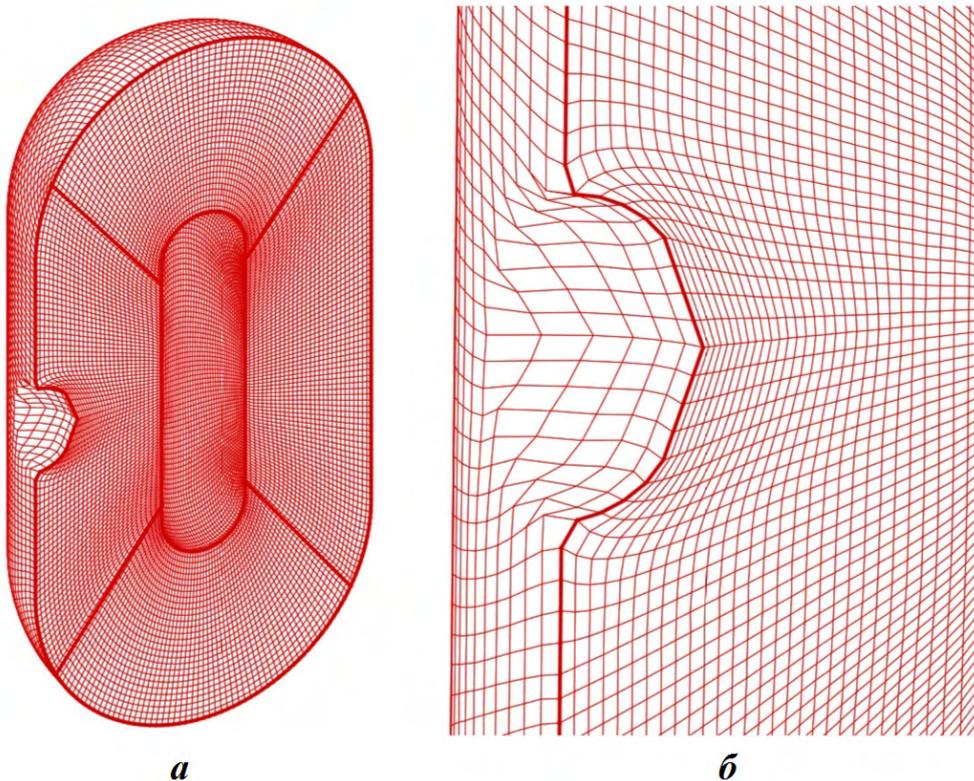


Рис. 5. Расчет 1: сетка на деформированном теле (*а*) и в деформированном фрагменте в увеличенном масштабе (*б*)

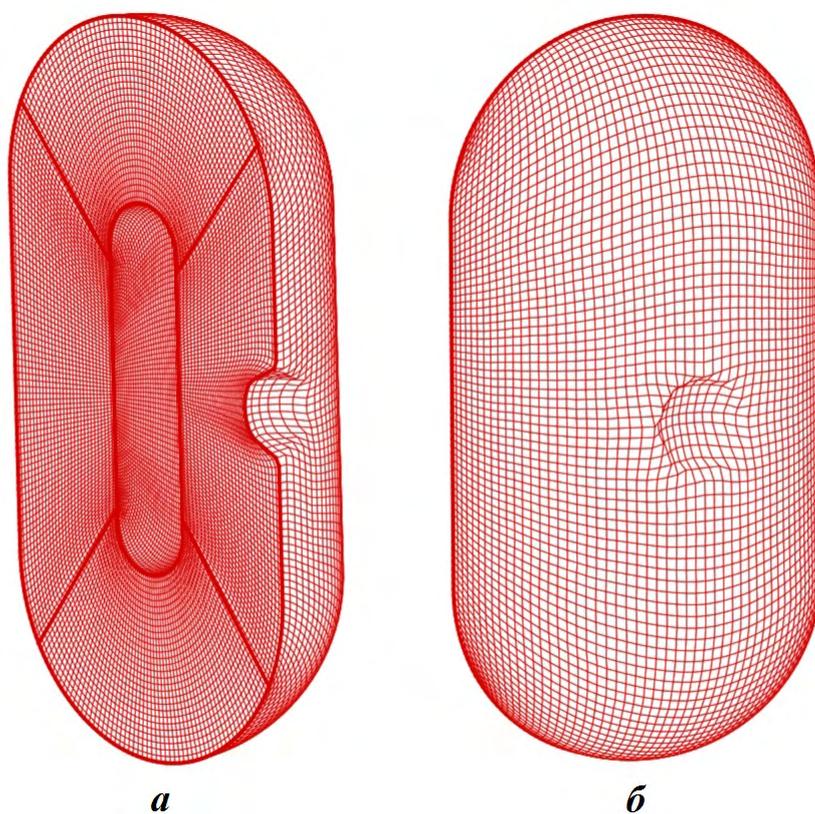


Рис. 6. Сетки на деформированном теле в расчетах 2 (а) и 3 (б)

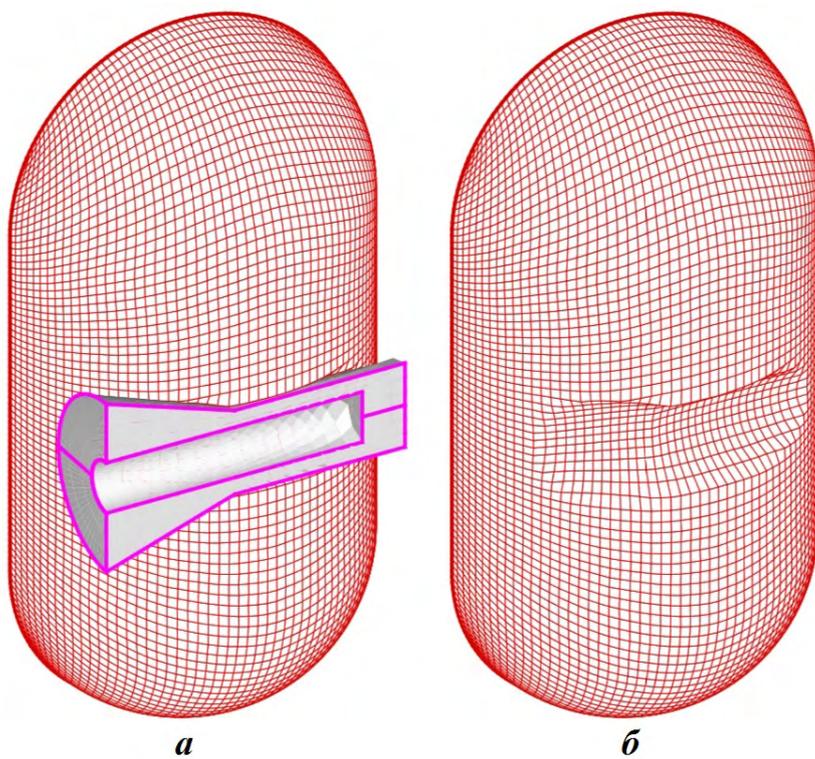


Рис. 7. Расчет 4: деформация телом вращения (а) и сетка на деформированном теле (б)

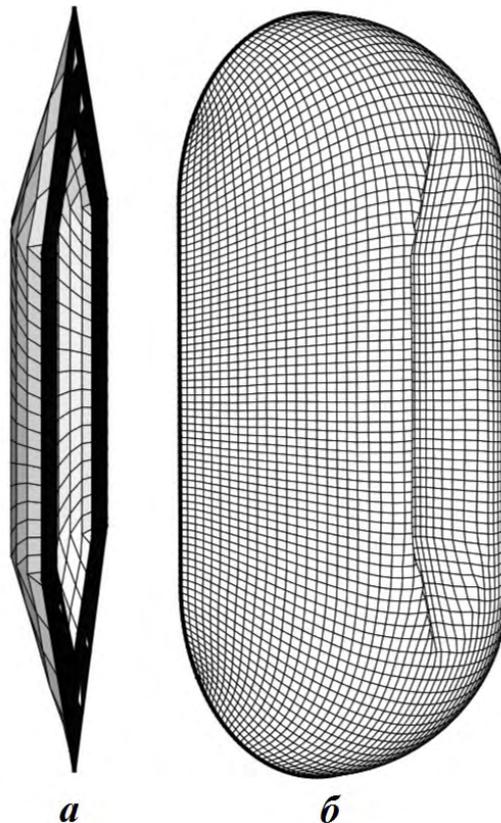


Рис. 8. Расчет 5: вспомогательное (а) и деформированное (б) тела

Заключение

В данной статье рассмотрена деформация телом вращения, образованным цилиндрическими, коническими и сферическими поверхностями, которые можно получить, вращая отрезки прямых линий и дуги окружностей. Проведенные тестовые расчеты по созданным алгоритмам показали работоспособность программ и целесообразность их дальнейшего развития с целью расширения класса деформирующих областей.

Далее планируется реализация алгоритма для класса деформирующих областей, ограниченных поверхностями вращения второго порядка с параллельными осями вращения [15]. Об алгоритмах построения начальных сеток в таких областях написано в работах [9, 16].

Список литературы

1. Артёмова Н. А. Нестационарный алгоритм построения структурированных сеток в деформированных областях // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2018. Вып. 4. С. 76–86.
Artyomova N. A. Nestatsionarny algoritm postroeniya strukturirovannykh setok v deformirovannykh oblastiakh // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2018. Vyp. 4. S. 76–86.
2. Ушакова О. В. Алгоритм коррекции сетки к области вращения // Там же. 2016. Вып. 1. С. 16–27.
Ushakova O. V. Algoritm korrektsii setki k oblasti vrashcheniya // Tam zhe. 2016. Vyp. 1. S. 16–27.

3. Anuchina N. N., Volkov V. I., Gordeychuk V. A., Es'kov N. S., Ilyutina O. S., Kozyrev O. M. Numerical simulation of 3D multi-component vortex flows by MAH-3 code // *Advances in Grid Generation* / Ed. by O. V. Ushakova. N.-Y.: Novascience Publishers, 2007. P. 337–380.
4. Годунов С. К., Забродин А. В., Иванов М. Я., Крайко А. Н., Прокопов Г. П. Численное решение многомерных задач газовой динамики. М.: Наука, 1976.
Godunov S. K., Zabrodin A. V., Ivanov M. Ya., Krayko A. N., Prokopov G. P. Chislennoe reshenie mnogomernykh zadach gazovoy dinamiki. M.: Nauka, 1976.
5. Zegelning P. A. Moving grid techniques // *Handbook of Grid Generation* / Ed. by J. F. Thompson, B. K. Soni, N. P. Weatherill. Boca Raton, FL: CRC Press, 1999. P. 37-1–37-22.
6. Staten M. L., Owen S. J., Shontz S. M., Salinger A. G., Coffey T. S. A Comparison of mesh morphing methods for 3D shape optimization // *Proc. 20th Int. Meshing Roundtable* / Ed. by W. R. Quadros. Berlin, Heidelberg: Springer, 2011. P. 293–311.
7. Прокопов Г. П. Вариационные методы расчета двумерных сеток при решении нестационарных задач: Препринт № 4. М.: ИПМ им. М. В. Келдыша, РАН. 2003.
Prokopov G. P. Variatsionnye metody rascheta dvumernykh setok pri reshenii nestatsionarnykh zadach: Preprint № 4. M.: IPM im. M. V. Keldysha, RAN. 2003.
8. Ушакова О. В. Алгоритмы оптимизации трехмерных сеток для областей вращения // *Труды института математики и механики*. 2008. Т. 14. № 1. С. 150–180.
Ushakova O. V. Algoritmy optimizatsii trekhmernykh setok dlya oblastey vrashcheniya // *Trudy instituta matematiki i mekhaniki*. 2008. T. 14. № 1. S. 150–180.
9. Anuchina A. I., Artyomova N. A., Gordeychuck V. A., Ushakova O. V. A Technology for grid generation in volumes bounded by the surfaces of revolutions // *Numerical Geometry, Grid Generation and Scientific Computing. Lecture Notes in Computational Science and Engineering*. 2019. Vol. 131. P. 281–292.
10. Бронина Т. Н. Алгоритмы построения начальных трехмерных структурированных сеток для областей вращения // *Труды института математики и механики*. 2008. Т. 14, № 1. С. 3–10.
Bronina T. N. Algoritmy postroeniya nachalnykh trekhmernykh strukturirovannykh setok dlya oblastey vrashcheniya // *Trudy instituta matematiki i mekhaniki*. 2008. T. 14, № 1. S. 3–10.
11. Ушакова О. В. Применение алгоритма коррекции сетки к области вращения // *Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов*. 2016. Вып. 2. С. 31–37.
Ushakova O. V. Primenenie algoritma korrektsii setki k oblasti vrashcheniya // *Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov*. 2016. Vyp. 2. S. 31–37.
12. Ушакова О. В. О невырожденности трехмерных сеток // *Труды института математики и механики*, 2004. Т. 11, № 1. С. 78–100.
Ushakova O. V. O nevyrozhdennosti trekhmernykh setok // *Trudy instituta matematiki i mekhaniki*, 2004. T. 11, № 1. S. 78–100.
13. Ушакова О. В. Алгоритм коррекции сетки к деформированной области вращения // *Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов*. 2017. Вып. 2. С. 53–65.
Ushakova O. V. Algoritm korrektsii setki k deformirovannoy oblasti vrashcheniya // *Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov*. 2017. Vyp. 2. S. 53–65.
14. Bronina T. N., Ushakova O. V. Application of optimal grid generation algorithms to the volumes of revolution // *Advances in Grid Generation* / Ed. by O. V. Ushakova. N.-Y.: Novascience Publishers, 2007. P. 283–320.
15. Ушакова О. В. Алгоритм коррекции сетки к области, образованной поверхностями вращения с параллельными осями вращения // *Вопросы атомной науки и техники. Сер. Математическое*

- моделирование физических процессов. 2018. Вып. 1. С. 30–41.
Ushakova O. V. Algoritm korrektsii setki k oblasti, obrazovannoy poverkhnostyami vrashcheniya s parallelnymi osyami vrashcheniya // *Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov.* 2018. Вып. 1. С. 30–41.
16. *Ушакова О. В., Артёмова Н. А., Бронина Т. Н., Анучина А. И., Гордейчук В. И.* Построение сеток в деформированных объемах вращения // *Актуальные проблемы вычислительной и прикладной математики — 2015: Межд. конф., посвящ. 90-летию со дня рожд. акад. Г. И. Марчука.* Новосибирск, 19–23 октября 2015 г. Новосибирск: Абвей, 2015. С. 782–788.
Ushakova O. V., Artyomova N. A., Bronina T. N., Anuchina A. I., Gordeychuk V. I. Postroenie setok v deformirovannykh obemakh vrashcheniya // *Aktualnye problemy vychislitelnoy i prikladnoy matematiki — 2015: Mezhd. konf., posvyashch. 90-letiyu so dnya rozhd. akad. G. I. Marchuka.* Novosibirsk, 19–23 oktyabrya 2015 g. Novosibirsk: Abvey, 2015. S. 782–788.

Статья поступила в редакцию 12.02.20.

ON THE DEVELOPMENT OF A GRID GENERATION ALGORITHM FOR DEFORMED SOLIDS OF REVOLUTION FORMED BY SEVERAL SURFACES / N. A. Artyomova, O. V. Ushakova (IMM UB RAS, Ekaterinburg).

A problem of grid generation in solids of revolution deformed by solids of revolution is under consideration. The solid of revolution can be generated by cylindrical, conical or spherical surfaces and formed by rotating two types of elements — segments of straight lines and arcs of circles. A nonstationary algorithm of grid generation in the domains with moving deforming boundaries has been offered earlier for special cases of deformation by a cylinder, a cone and a sphere. This work describes an extension of this nonstationary algorithm. The idea of the algorithm is the same, but its implementation has become more complicated as a result of the increased structural complexity of the deforming domain. Some examples of grids are provided.

Keywords: structured grids, domain of revolution, deformed domain of revolution, optimal grids, moving grids.

СВЕДЕНИЯ ОБ АВТОРАХ

Арапова Евгения Юрьевна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: EYuArapova@vniief.ru

Артёмова Наталья Александровна — ИММ УрО РАН, г. Екатеринбург, старший программист, *e-mail*: ana@imm.uran.ru

Близнюк Оксана Геннадьевна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: OGBliznyuk@vniief.ru

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

Власова Ольга Евгеньевна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: OEVlasova@vniief.ru

Гичук Александр Владимирович — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: AVGichuk@vniief.ru

Дёмина Марина Александровна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: MADemina@vniief.ru

Ерёменко Андрей Юрьевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, *e-mail*: AYuEremenko@vniief.ru

Козелков Андрей Сергеевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательского отдела, *e-mail*: ASKozelkov@vniief.ru

Косарим Снежана Сергеевна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: SSKosarim@vniief.ru

Куделькин Вадим Григорьевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательского отдела, *e-mail*: VGKudelkin@vniief.ru

Лазарев Владимир Валерьевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: VIVLazarev@vniief.ru

Лялюшкина Ирина Владимировна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, младший научный сотрудник, *e-mail*: IVLyalyushkina@vniief.ru

Нуждин Андрей Алексеевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, начальник научно-исследовательской лаборатории, *e-mail*: AANuzhdin@vniief.ru

Полякова Сабина Юрьевна — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, научный сотрудник, *e-mail*: SYuPolyakova@vniief.ru

Софронов Василий Николаевич — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, ведущий научный сотрудник, *e-mail*: VNSofronov@vniief.ru

Тихонов Александр Викторович — ФГУП "РФЯЦ–ВНИИЭФ", г. Саров Нижегородской области, старший научный сотрудник, *e-mail*: AVTikhonov@vniief.ru

Ушакова Ольга Васильевна — ИММ УрО РАН, г. Екатеринбург, ведущий научный сотрудник, *e-mail*: uov@imm.uran.ru

INFORMATION ABOUT AUTHORS

Arapova Evgeniya Yuryevna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist,
e-mail: EYuArapova@vniief.ru

Artyomova Natalya Aleksandrovna – IMM UB RAS, Ekaterinburg, senior programmer,
e-mail: ana@imm.uran.ru

Bliznyuk Oksana Gennadyevna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist,
e-mail: OGBliznyuk@vniief.ru

Vetchinnikov Maksim Vladimirovich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head
of research laboratory, *e-mail*: MVVetchinnikov@vniief.ru

Vlasova Olga Evgenyevna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist,
e-mail: OEVlasova@vniief.ru

Gichuk Aleksandr Vladimirovich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior
scientist, *e-mail*: AVGichuk@vniief.ru

Dyemina Marina Aleksandrovna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist,
e-mail: MADemina@vniief.ru

Eryemenko Andrey Yuryevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of research
laboratory, *e-mail*: AYuEremenko@vniief.ru

Kozelkov Andrey Sergeevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of research
department, *e-mail*: ASKozelkov@vniief.ru

Kosarim Snezhana Sergeevna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist,
e-mail: SSKosarim@vniief.ru

Kudelkin Vadim Grigoryevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of research
department, *e-mail*: VGKudelkin@vniief.ru

Lazarev Vladimir Valeryevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior scientist,
e-mail: VIVLazarev@vniief.ru

Lyalyushkina Irina Vladimirovna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, junior
scientist, *e-mail*: IVLyalyushkina@vniief.ru

Nuzhdin Andrey Alekseevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, head of research
laboratory, *e-mail*: AANuzhdin@vniief.ru

Polyakova Sabina Yuryevna – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, scientist,
e-mail: SYuPolyakova@vniief.ru

Sofronov Vasiliy Nikolaevich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, leading scientist,
e-mail: VNSofronov@vniief.ru

Tikhonov Aleksandr Viktorovich – FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region, senior
scientist, *e-mail*: AVTikhonov@vniief.ru

Ushakova Olga Vasilyevna – IMM UB RAS, Ekaterinburg, leading scientist, *e-mail*: uov@imm.uran.ru

ПЕРЕЧЕНЬ

статей, опубликованных в 2020 г. в научно-техническом сборнике "Вопросы атомной науки и техники". Сер. "Математическое моделирование физических процессов"

Николаева О. В., Казанцева А. С. Точность схем метода конечных элементов для решения уравнения переноса на неструктурированных тетраэдрических и призматических сетках. Вып. 1. С. 3–19.

Антонов С. Н., Русаков А. Д., Устинов А. Н., Фатеев М. В., Фролова М. В., Семенова Т. В., Волков Н. А. Расчет нейтронно-физических характеристик бенчмарка FullCore по программам TDMSS И САПФИР_95&RC_ВВЭР. Вып. 1. С. 20–33.

Аблесимов В. Е., Пашарина О. Ю. Особенности формирования сигнала времяпролетного детектора при измерении излучения плазменного фокуса в замкнутом пространстве. Вып. 1. С. 34–42.

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. Вып. 1. С. 43–57.

Коваленко О. В., Крючков И. А., Огородников А. В., Хочкин Н. И. Возможности пользовательского интерфейса визуализационно-интеграционной платформы ОптИМУС с учетом перспектив интеграции технологий искусственного интеллекта. Вып. 1. С. 58–65.

Аверина Н. С., Баканова Т. Ю., Лащманова М. Г., Савиных Е. А., Серова Т. Н. Методика тестирования пакета программ ЛОГОС. Вып. 1. С. 66–76.

Бочков А. И., Сучкова В. В. DS_n -метод решения трехмерного кинетического уравнения переноса, записанного в сферической системе координат. Вып. 2. С. 3–15.

Блажнов И. Д., Вишнякова М. Н., Полищук С. Н., Тихомиров Б. П. Полностью консервативные SPH-системы газодинамики и схема их численного интегрирования. Вып. 2. С. 16–32.

Ватулин В. В., Волкова К. А., Кибкало А. А., Певная П. И., Плетнёва Н. П., Шагалев Р. М. Численное исследование переноса рентгеновского излучения в вакуумной области. Вып. 2. С. 33–44.

Дьянов Д. Ю., Казанцев А. В., Стародубов С. В., Циберев К. В., Челаков А. А. Пакет программ "Логос". Учет контактного взаимодействия при решении нелинейных быстропротекающих задач механики деформируемого твердого тела. Вып. 2. С. 45–59.

Абдуллин М. Ф., Барабанов Р. А., Кварацхелия Д. Д., Филимонкин Е. А., Китаев М. В., Корнев А. В., Никитин М. В., Саганов Е. Б. Пакет программ "Логос". Функциональные возможности для моделирования статического деформирования слоистых композиционных материалов на основе оболочечного конечного элемента. Вып. 2. С. 60–70.

Ерёменко А. Ю., Косарим С. С. Пакет программ "Логос". Методика расчета отклика конструкции при действии широкополосной случайной вибрации. Вып. 2. С. 71–85.

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. С., Чистякова И. Н., Шемякина Т. В., Шагалмиев Р. М., Янилкин Ю. В. Методические прикладные тесты РФЯЦ-ВНИИЭФ для численного исследования параметров высокопроизводительных вычислительных систем. Вып. 2. С. 86–100.

Янилкин Ю. В., Гужова А. Р., Дегтяренко Л. И., Синькова О. Г. Определение "локальной" степени гомогенного смещения при прямом численном моделировании турбулентного перемешивания. Вып. 3. С. 3–10.

Бондаренко С. В., Митрофанов Е. И., Потапкина Л. Ф., Шаров О. О. Результаты трёхмерных расчетов конверсии лазерного излучения в рентгеновское в сферическом боксе для условий мегаджоульной лазерной установки. Вып. 3. С. 11–22.

Беспалов А. П., Логинов Д. В., Соловьёв А. Н. Метод автоматического построения поверхностных сеток на гранях геометрических моделей в параметрическом представлении в препроцессоре пакета программ "Логос". Вып. 3. С. 23–39.

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". Вып. 3. С. 40–52.

Лазарев В. В. Блочная структурированная сетка с контактом блоков по частям сторон. Вып. 3. С. 53–63.

Потехин А. Л., Журнов В. В. Параллельная система постобработки ScientificView. Вычисление "дополнительных" сеточных величин. Вып. 3. С. 64–72.

Киселёв А. Б., Киселёв С. Н. Новые возможности отладчика параллельных программ PD. Вып. 3. С. 73–80.

УДК 539.3

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ГАМИЛЬТОНОВОЙ ДИНАМИКИ В ЧИСЛЕННЫХ РАСЧЕТАХ ЗАДАЧ МЕХАНИКИ СПЛОШНОЙ СРЕДЫ / В. Н. Софронов, М. В. Ветчинников, М. А. Дёмина // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 5—21.

Представлен метод численного решения задач механики сплошной среды, основанный на дискретной гамильтоновой динамике. Для численного решения гамильтоновых уравнений используются симплектические разностные схемы. Приведены примеры моделирования бездиссипативных процессов (рис. — 6, табл. — 1, список лит. — 26).

Ключевые слова: дискретная гамильтонова динамика, фазовый объем, симплектические разностные схемы, динамические задачи упругости, пакет программ MoDyS.

УДК 519.6

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ МНОГОФАЗНЫХ ТЕЧЕНИЙ В ЭЙЛЕРОВОМ ПРИБЛИЖЕНИИ НА ОСНОВЕ МЕТОДА SIMPLE / О. Г. Близнюк, О. Е. Власова, А. В. Гичук, А. С. Козелков, И. В. Лялюшкина // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 22—32.

Описаны особенности применения метода SIMPLE для численного моделирования течений многофазных сред с учетом межфазных взаимодействий. Используется многожидкостная модель, в которой каждая фаза описывается своими плотностью, скоростью и температурой. Для алгоритма, реализованного в пакете программ "Логос", приведены результаты расчета некоторых тестовых задач (рис. — 5, список лит. — 20).

Ключевые слова: многофазные течения, численное моделирование, метод SIMPLE, пакет программ "Логос".

УДК 519.6

ПАКЕТ ПРОГРАММ "ЛОГОС". УЧЕТ КОНТАКТНОГО ВЗАИМОДЕЙСТВИЯ ПРИ РЕШЕНИИ ЗАДАЧ КВАЗИСТАТИЧЕСКОГО ДЕФОРМИРОВАНИЯ / А. Ю. Ерёменко, С. С. Косарим // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 33—47.

Рассмотрены особенности реализации учета контактного взаимодействия между различными телами или частями конструкции при решении задач квазистатического деформирования в пакете программ "Логос". Для реализации учета контактного взаимодействия в качестве основного выбран метод штрафа. Приводится описание алгоритма определения проникания контактирующих тел друг в друга. Выписаны основные формулы для получения контактных сил и их вклада в глобальную матрицу жесткости. Отдельно описывается учет трения между контактирующими телами. Точность реализованных методов подтверждается численным решением задачи Герца о контактном взаимодействии двух цилиндров бесконечной длины с учетом трения и его сравнением с аналитическим решением (рис. — 18, список лит. — 15).

Ключевые слова: квазистатическое деформирование, учет контактного взаимодействия, метод штрафа, алгоритм определения проникания тел, учет трения, задача Герца о цилиндрах бесконечной длины, пакет программ "Логос".

УДК 004.94:004.8

ТЕСТОВАЯ ПРОГРАММА "ПАУК" КАК ПОЛИГОН ДЛЯ АПРОБАЦИИ АЛГОРИТМОВ И ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ / А. А. Нуждин // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 48–61.

В тестовой программе ПАУК с помощью разностного S_n -метода численно решается трехмерное одногрупповое уравнение переноса нейтронов на ортогональной пространственной сетке. Представлены результаты исследований по адаптации тестовой программы к гетерогенной вычислительной системе, содержащей сопроцессоры Intel Xeon Phi поколения Knights Corner. При этом апробированы различные алгоритмы и технологии параллельного программирования: векторизация циклов по направлениям и элементам гиперплоскостей, автовекторизация и intrinsic-программирование, явная и неявная предвыборка данных, реализации КВА-алгоритма распараллеливания в трех моделях памяти (общая, распределенная, PGAS). Эффективность адаптации программы подтверждается результатами исследований производительности в различных режимах использования гетерогенной вычислительной системы: на универсальных процессорах (CPU-only), сопроцессорах (native) и в симметричном режиме (symmetric) (рис. — 2, табл. — 7, список лит. — 27).

Ключевые слова: S_n -метод, алгоритм бегущего счета, КВА-алгоритм, Intel Xeon Phi, векторизация, предвыборка данных, MPI-3 SHM.

УДК 004.422.63

ЭФФЕКТИВНЫЙ АЛГОРИТМ ОБЪЕДИНЕНИЯ СОГЛАСОВАННЫХ ПО УЗЛАМ ФРАГМЕНТОВ ПОВЕРХНОСТНЫХ СЕТОК / В. В. Лазарев // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 62–71.

Предложен новый алгоритм объединения согласованных по узлам фрагментов поверхностных сеток, построенных на геометрической модели в B-REP представлении. По сравнению с классическими алгоритмами объединения он имеет следующие отличия: 1) вместо рассмотрения всех узлов по отдельности рассматриваются их группы; 2) для нахождения совпадающих узлов используется не функция расстояния и ее минимальное значение, а связи между сеточными фрагментами, которые хранятся в структурах представления геометрической модели; 3) результат объединения не хранится в массивах, а представлен в функциональном виде — выражениями, которые по номеру узла вычисляют адрес этого узла в сеточном фрагменте, а по номеру ячейки — номера узлов этой ячейки.

Время выполнения разработанного алгоритма и объем потребляемой памяти почти не заметны. Эти параметры зависят только от количества сеточных фрагментов, но не зависят от количества узлов и ячеек в этих фрагментах (рис. — 2, табл. — 2, список лит. — 8).

Ключевые слова: составная сетка, алгоритм объединения поверхностных сеток, глобальная нумерация, функциональное представление множества, граничное представление геометрической модели (B-REP).

УДК 519.6

ГЕНЕРАТОР ПОВЕРХНОСТНЫХ НЕСТРУКТУРИРОВАННЫХ ЧЕТЫРЕХУГОЛЬНЫХ СЕТОК В ПРЕПРОЦЕССОРЕ ПАКЕТА ПРОГРАММ "ЛОГОС" / Е. Ю. Арапова, В. Г. Куделькин, Е. А. Павлов, С. Ю. Полякова, А. В. Тихонов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 72—85.

Приводится описание автоматического генератора поверхностных неструктурированных четырехугольных сеток для геометрических моделей в параметрическом представлении. Генератор предназначен для построения сеточных моделей при решении задач прочности в рамках пакета программ "Логос". Рассматриваются этапы генерации поверхностной сетки. Подробно рассмотрен один из этапов, посвященный генерации четырехугольных сеток в параметрической плоскости. Приведены примеры тестовых задач, в которых используются различные геометрические модели (рис. — 21, табл. — 2, список лит. — 15).

Ключевые слова: пакет программ "Логос", генератор поверхностных четырехугольных сеток, подвижный фронт, формирование четырехугольных ячеек, локальное сглаживание, топологическая оптимизация.

УДК 519.6

О РАЗВИТИИ АЛГОРИТМА ПОСТРОЕНИЯ СЕТОК В ДЕФОРМИРОВАННЫХ ТЕЛАХ ВРАЩЕНИЯ ДЛЯ СЛУЧАЯ ИХ ДЕФОРМАЦИИ ТЕЛАМИ ВРАЩЕНИЯ, ОБРАЗОВАННЫМИ НЕСКОЛЬКИМИ ПОВЕРХНОСТЯМИ / Н. А. Артёмова, О. В. Ушакова // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 4. С. 86—96.

Рассматривается задача о построении сетки в телах вращения, деформированных телами вращения. Тело вращения может быть образовано цилиндрическими, коническими или сферическими поверхностями и формироваться вращением элементов двух видов — отрезков прямых и дуг окружностей. Ранее был предложен нестационарный алгоритм построения сеток в областях с подвижными деформируемыми границами для частных случаев деформации цилиндром, конусом и сферой. В данной работе рассматривается дальнейшее развитие нестационарного алгоритма. Суть алгоритма осталась прежней, усложнилась реализация в связи с усложнением конструкции деформирующей области. Приводятся примеры расчетов сеток (рис. — 8, список лит. — 16).

Ключевые слова: структурированные сетки, область вращения, деформированная область вращения, оптимальные сетки, подвижные сетки.

USE OF HAMILTONIAN DYNAMICS METHODS IN COMPUTATIONAL CONTINUUM MECHANICS / V. N. Sofronov, M. V. Vetchinnikov, M. A. Dyemina // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 5–21.

The paper presents a computational continuum mechanics method based on discrete Hamiltonian dynamics. Hamilton's equations are numerically solved using symplectic difference schemes. Examples of dissipation-free process simulations are provided.

Key words: discrete Hamiltonian dynamics, phase volume, symplectic difference schemes, dynamic elasticity problems, MoDyS software package.

EULERIAN MULTI-PHASE FLOW SIMULATIONS BASED ON THE "SIMPLE" METHOD / O. G. Bliznyuk, O. E. Vlasova, A. V. Gichuk, A. S. Kozelkov, I. V. Lyalyushkina // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 22–32.

Specific details of using the SIMPLE method for multi-phase flow simulations involving phase interactions are described. A multi-fluid model is used, in which each phase has its own density, velocity and temperature. Results of some test simulations are reported for the algorithm implemented in the software package "Logos".

Key words: multi-phase flow, numerical simulations, SIMPLE method, "Logos" software package.

"LOGOS" SOFTWARE PACKAGE. INCORPORATION OF CONTACT INTERACTIONS IN QUASI-STATIC DEFORMATION SIMULATIONS / A. Yu. Eryemenko, S. S. Kosarim // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 33–47.

Details of incorporating contact interactions between different bodies or structural components in quasi-static deformation simulations by the "Logos" software package are discussed. As a basic method for this, a penalty method has been chosen. An algorithm is described, which detects interpenetration of bodies in contact. Basic formulas for contact forces and their contributions to the global stiffness matrix are given. A separate description is provided for the way to incorporate friction between bodies in contact. The accuracy of the implemented methods is illustrated by numerical simulations of the Hertz contact problem with friction between two infinitely long cylinders and by comparison of these simulations with the analytical solution.

Key words: quasi-static deformation, incorporation of contact interactions, penalty method, penetration detection algorithm, incorporation of friction, Hertz contact problem, "Logos" software package.

TEST PROGRAM "PAUK" AS A TESTING RANGE FOR PARALLEL PROGRAMMING ALGORITHMS AND TECHNIQUES / A. A. Nuzhdin // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 48–61.

PAUK is a test program that numerically solves the three-dimensional one-group neutron transport equation on orthogonal spatial grids by the difference S_n -method. The paper presents the results of its adaptation for a heterogeneous computing system with Intel's Knights Corner Xeon Phi coprocessors. The adaptation included testing of various parallel programming algorithms and techniques, namely loop vectorization in directions and hyperplane elements, automatic vectorization and intrinsic programming, explicit and implicit data prefetching, implementation of the KBA algorithm in three memory models (shared, distributed, PGAS). Adaptation efficiency was verified by program performance studies in various execution modes at the heterogeneous computing system: CPU-only, native and symmetric.

Key words: S_n -method, sweep algorithm, KBA algorithm, Intel Xeon Phi, vectorization, data prefetching, MPI-3 SHM.

AN EFFICIENT ALGORITHM FOR MERGING NODE-MATCHED FRAGMENTS OF SURFACE GRIDS / V. V. Lazarev // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 62–71.

A new algorithm is proposed for merging node-matched fragments of surface grids constructed based on a geometric model in the B-REP representation. As opposed to classical merging algorithms, this algorithm: 1) considers groups of nodes rather than each individual node; 2) identifies matching nodes based on the connections between grid fragments, which are stored in geometric model representation structures, rather than based on the distance function and its minimum value; 3) instead of storing the result of merging in arrays, represents it in the form of functions, which calculate the node address in the grid fragment based on the node index and the node indexes in a cell, based on the cell index.

The processor time and the memory consumed by the algorithm are negligibly small. They depend only on the number of grid fragments and do not depend on the number of nodes or cells in the fragments.

Key words: composite grid, surface grid merging algorithm, global indexing, functional representation of a set, boundary representation of a geometric model (B-REP).

AN UNSTRUCTURED QUADRILATERAL SURFACE grid GENERATOR IN THE PREPROCESSOR OF THE "LOGOS" SOFTWARE PACKAGE / E. Yu. Arapova, V. G. Kudelkin, E. A. Pavlov, S. Yu. Polyakova, A. V. Tikhonov // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 72–85.

The paper describes an automatic generator of unstructured quadrilateral surface grids for geometric models in the parametric representation. The grid generator is intended to construct grid models for strength simulations in the "Logos" software package. Steps of the surface grid generation procedure are considered. One of the steps, namely generation of quadrilateral grids in the parametric plane, is discussed in detail. Examples of test problems with different geometric models are given.

Key words: "Logos" software package, quadrilateral surface grids, moving front, generation of quadrilateral cells, local smoothing, topological optimization.

ON THE DEVELOPMENT OF A GRID GENERATION ALGORITHM FOR DEFORMED SOLIDS OF REVOLUTION FORMED BY SEVERAL SURFACES / N. A. Artyomova, O. V. Ushakova // VANT. Ser.: Mat. Mod. Fiz. Proc. 2020. No 4. P. 86–96.

A problem of grid generation in solids of revolution deformed by solids of revolution is under consideration. The solid of revolution can be generated by cylindrical, conical or spherical surfaces and formed by rotating two types of elements — segments of straight lines and arcs of circles. A nonstationary algorithm of grid generation in the domains with moving deforming boundaries has been offered earlier for special cases of deformation by a cylinder, a cone and a sphere. This work describes an extension of this nonstationary algorithm. The idea of the algorithm is the same, but its implementation has become more complicated as a result of the increased structural complexity of the deforming domain. Some examples of grids are provided.

Key words: structured grids, domain of revolution, deformed domain of revolution, optimal grids, moving grids.