

# ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ НА ОСНОВЕ DVM-СИСТЕМЫ ПРОГРАММИРОВАНИЯ <sup>1</sup>

Л.П. Каменщиков

*Институт вычислительного моделирования СО РАН, г. Красноярск*

e-mail: lpk@icm.krasn.ru

## Аннотация

На ряде примеров показаны некоторые возможности DVM-системы программирования для параллельных вычислений на многопроцессорных кластерах. Рассмотрены следующие задачи: 1) решение уравнений переноса радиационной энергии в вакууме и определение светового давления на космический аппарат; 2) решение уравнений мелкой воды для моделирования распространения гравитационных поверхностных волн в морях и океанах; 3) решение решеточных уравнений Больцмана применительно к трехмерным нестационарным задачам гидродинамики.

## 1 О DVM-системе

Программирование в рамках модели передачи сообщения MPI является низкоуровневым, поэтому трудоемким, требующем многих «человеко-часов» для разработки и отладки параллельных программ. Однако для многих классов важных практических задач можно использовать альтернативный высокоуровневый подход к программированию на базе DVM-системы, разработанной в ИПМ им. М.В.Келдыша РАН. DVM-система предназначена для создания переносимых и эффективных вычислительных приложений на языках C-DVM и Fortran-DVM для многопроцессорных компьютеров с общей и распределенной памятью, включая и гибридные системы, в узлах которых вместе с универсальными многоядерными процессорами используются в качестве ускорителей и графические процессоры. В состав системы входят также средства отладки MPI-программ [1–8].

При использовании языков C-DVM и Fortran-DVM программист имеет только один вариант программы как для последовательного, так и для параллельного выполнения. Эта программа, помимо описания алгоритма обычными средствами языков C, Fortran-77 или Fortran-95, содержит правила параллельного выполнения этого алгоритма. Эти правила оформляются как комментарии для стандартных компиляторов с последовательных языков C и Fortran и не препятствуют выполнению и отладке DVM-программы на персональных компьютерах как обычной последовательной программы.

Программист имеет, например, такие возможности для распараллеливания программы: распределение элементов массива между процессорами; распределение витков цикла между процессорами; организация доступа к удаленным (расположенным на других процессорах) данным; организация выполнения редуцированных операций — глобальных операций с расположенными на различных процессорах данными и т.п. Компилятор переводит программу на языке C-DVM (Fortran-DVM) в программу на стандартном C (Fortran),

---

<sup>1</sup> Работа поддержана грантом РФФИ, проект № 11-01-00224

расширенную функциями системы поддержки выполнения DVM-программ, которая для организации межпроцессорного взаимодействия использует коммуникационные библиотеки MPI.

Разработанная сравнительно недавно гибридная модель (DVM/OpenMP-модель) параллельного программирования и язык Fortran-DVM/OpenMP позволяют во многом автоматизировать разработку программ для SMP-кластеров (кластеров, использующих в качестве узлов мультипроцессоры). Программист может управлять распределением вычислений не только между узлами кластера, но и между ядрами внутри каждого узла [5, 6]. DVM-система находится в постоянном развитии. На сайте [1] можно найти ее подробное описание, необходимую документацию, а также загрузить систему для установки на своем вычислительном кластере.

## 2 Примеры использования DVM-системы

Рассмотрим примеры численного решения ряда задач на вычислительных кластерах с помощью DVM-системы. Ниже представлены фрагменты в основном из уже опубликованных работ, выполнявшихся автором в последние годы. Все расчеты проводились на кластерах Межведомственного суперкомпьютерного центра РАН: МВС-1000М, МВС-15000ВМ и МВС-100К. Использовался язык Fortran-DVM, позволяющий в значительной степени автоматизировать процесс распараллеливания.

### 2.1 Расчет светового давления на космический аппарат

В настоящее время широкое развитие получили глобальные навигационные спутниковые системы (ГНСС), предназначенные для определения координат и скоростей различных объектов как на поверхности Земли, так и вне ее. Принцип работы ГНСС основан на том, что для нахождения своего местоположения объект-приемник сравнивает время отправки сигнала со спутника со временем его получения. Эта разница во времени позволяет найти расстояние до спутника. Если одновременно найдены расстояния до нескольких спутников, то путем соответствующих геометрических расчетов можно определить координаты приемника. Известны такие навигационные системы, как GPS, ГЛОНАСС, Galileo и др. [9].

Точность определения координат приемника напрямую связана с точностью координат самого навигационного спутника, которые должны быть известны для момента выполнения расчетов. Уравнения движения спутника или космического аппарата (КА) в геоцентрической системе координат можно представить в виде:

$$\frac{d^2 x_i}{dt^2} = -\frac{\mu x_i}{r^3} + \ddot{p}_i, \quad i = 1, 2, 3,$$

где  $x_i$  —  $i$ -я координата КА,  $r = (\sum x_i^2)^{1/2}$ . В константу  $\mu$  включены массы тел и гравитационная постоянная. Первое слагаемое в правой части представляет кеплеровское движение КА в центральном поле сил [10], а символом  $\ddot{p}_i$  обозначена сумма целого ряда

возмущающих воздействий, вызванных, в частности, неоднородностью гравитационного поля Земли, притяжением аппарата Солнцем, Луной и планетами, световым давлением, сопротивлением атмосферы, приливной деформацией Земли, электромагнитными силами, работой двигателей и т.п.

В данной работе рассматривается расчет только одного возмущающего фактора, а именно светового давления на КА в некоторый фиксированный момент времени, т.е. при численном решении уравнений движения подобные расчеты светового давления следует выполнять на каждом дискретном шаге по времени, поэтому распараллеливание алгоритма с целью сокращения общего времени счета становится весьма актуальным. Ускорение аппарата  $\vec{a} = \vec{F}/m$  при известной его массе  $m$  сводится к определению вектора силы  $\vec{F}$ , вычисление которого прямым методом Монте-Карло и представлено в работе.

Вычисление силы  $\vec{F}$  вполне вписывается в общую схему метода Монте-Карло для решения интегральных уравнений Фредгольма второго рода с оценкой линейных функционалов от решения, представленную в многочисленных публикациях [11]–[31]. В работах по приложениям метода Монте-Карло основное внимание уделяется традиционным для этого метода классам задач: перенос нейтронов, радиационной теплообмен, ряд краевых задач математической физики, задачи атмосферной оптики, ядерной геофизики и т.п. Примеры расчетов светового давления на космические аппараты в указанных выше публикациях отсутствуют.

Исходными данными для метода Монте-Карло служат и оптические свойства материала поверхности как функции, в общем случае, от частот, направлений излучения и температуры. Получение таких данных является самостоятельной задачей. Однако и при обычных упрощающих предположениях (серые поверхности, диффузно-зеркальное отражение и т.п.) метод Монте-Карло может рассматриваться в качестве основного при определении радиационного давления, особенно для аппаратов невыпуклых конфигураций, когда возможно облучение одних частей аппарата другими, а также с учетом простой и эффективной распараллеливаемости этого метода.

### 2.1.1 Математическая модель

**Основные уравнения и обозначения.** Наиболее близко к задаче расчета радиационного давления примыкают задачи теплообмена излучением между поверхностями в прозрачной среде. Особо отметим здесь работы [19, 20, 21].

Пусть OXYZ — связанная с КА система декартовых координат.  $\vec{e}_1, \vec{e}_2, \vec{e}_3$  — единичные векторы, направленные по соответствующим осям. Начало системы координат находится в центре тяжести аппарата, оси координат можно направить произвольно, но для определенности пусть  $\vec{e}_3$  направлен параллельно продольной оси аппарата (у спутников ГНСС такая ось имеется), вектор  $\vec{e}_1$  — вдоль солнечных панелей, а  $\vec{e}_2 = \vec{e}_3 \times \vec{e}_1$ . Для основных величин указываем размерность в единицах СИ.

Действующую на материальное тело, в частности, на КА силу радиационного дав-

ления ищем в виде линейного функционала:

$$\vec{F} = (\varphi, \vec{h}) = \int_X \varphi(x) \vec{h}(x) dx, \quad \text{H}, \quad (1)$$

где

$X = \{x = (\vec{r}, \vec{\omega}, \nu)\}$  — фазовое пространство;

$\vec{r}$  — радиус-вектор, м;

$\vec{\omega}$  — единичный вектор направления излучения;

$\nu \in (0, \infty)$  — частота излучения, Гц;

$\varphi(x)$  — удельная сила падающего излучения [Вт/(м<sup>2</sup> ср Гц)], определяемая из решения интегрального уравнения

$$\varphi(x) - \int_X k(x', x) \varphi(x') dx' = f(x) \quad \text{или} \quad \varphi - K\varphi = f; \quad (2)$$

$S$  — поверхность космического аппарата (в данной работе ограничиваемся непрозрачными поверхностями);

$S_0$  — сфера некоторого радиуса  $R_0$  с центром в начале координат, содержащая внутри себя КА, и принимаемая в качестве границы расчетной области;

$\mathbb{S} = S \cup S_0$ ;

$\Omega_-(\vec{r}) = \{\vec{\omega} : |\vec{\omega}| = 1, \vec{r} \in S, \vec{n}(\vec{r}) \cdot \vec{\omega} < 0\}$  — полусфера входящих направлений в точку  $\vec{r}$ ; аналогично:  $\Omega_+(\vec{r})$  — для выходящих направлений;

$\vec{n}(\vec{r})$  — единичная нормаль в точке  $\vec{r} \in \mathbb{S}$ , направленная в сторону вакуума;

$f(x) \equiv f(\vec{r}, \vec{\omega}, \nu) = \int_{\mathbb{S}} t(\vec{r}', \vec{r}, \vec{\omega}) Q_{own}(\vec{r}', \vec{\omega}, \nu) ds(\vec{r}')$ , Вт/(м<sup>2</sup> ср Гц);

$t(\vec{r}', \vec{r}, \vec{\omega}) = \chi(\vec{r}', \vec{r}) \delta(\vec{s} - \vec{\omega}) \frac{|\vec{n}(\vec{r}') \cdot \vec{\omega}|}{|\vec{r} - \vec{r}'|^2}$ , где  $\vec{s} = (\vec{r}' - \vec{r})/|\vec{r}' - \vec{r}|$ ;

$\chi(\vec{r}', \vec{r})$  — индикатор видимости двух точек между собой;

$T(\vec{r})$  — температура поверхности КА, К;

$\delta$  — функция Дирака;

$Q_{own}(\vec{r}', \vec{\omega}, \nu)$  — заданный источник излучения, Вт/(м<sup>2</sup> ср Гц);

$k(x', x) \equiv k(\vec{r}', \vec{\omega}', \nu', \vec{r}, \vec{\omega}, \nu) = t(\vec{r}', \vec{r}, \vec{\omega}) |\vec{n}(\vec{r}') \cdot \vec{\omega}| \frac{\rho''(\vec{r}', \vec{\omega}', \vec{\omega}, \nu)}{\pi}$ ;

$\rho''(\vec{r}', \vec{\omega}', \vec{\omega}, \nu) = [1 - \varepsilon(\vec{r}', -\vec{\omega}', \nu)] g(\vec{r}', \vec{\omega}', \vec{\omega}, \nu)$  — двунаправленная отражательная способность поверхности;

$\varepsilon(\vec{r}, \vec{\omega}, \nu)$  — степень черноты;

$g(\vec{r}, \vec{\omega}', \vec{\omega}, \nu)$  — индикатриса отражения с нормировкой

$$\int_{\Omega_+(\vec{r})} g(\vec{r}, \vec{\omega}', \vec{\omega}, \nu) \frac{\vec{n} \cdot \vec{\omega}}{\pi} d\vec{\omega} = 1, \quad \vec{\omega}' \in \Omega_-, \quad \vec{\omega} \in \Omega_+;$$

$\vec{h}(x)$  — «аппаратная» вектор-функция (Н · с/Дж = с/м), это величина порядка  $\vec{\omega}/c_0$ , где  $\vec{\omega} \in \Omega_-$  для падающих потоков и  $\vec{\omega} \in \Omega_+$  для отраженных и собственных потоков излучения;

$I_0(\nu, T) = 2h_0\nu^3 c_0^{-2} / (e^{h_0\nu/k_0T} - 1)$  — функция Планка, Дж/(с м<sup>2</sup> ср Гц);

$T_S = 5777$  — принятая температура Солнца, К;

$c_0 = 2.99792458 \cdot 10^8$  — скорость света, м/с;

$h_0 = 6.62606876 \cdot 10^{-34}$  — постоянная Планка, Дж·с;

$k_0 = 1.3806503 \cdot 10^{-23}$  — постоянная Больцмана, Дж/К;

$\sigma_0 = 5.6704 \cdot 10^{-8}$  — постоянная Стефана-Больцмана, Вт/(м<sup>2</sup>К<sup>4</sup>).

В инженерных расчетах индикатрису отражения обычно представляют (если нет другой информации) в виде линейной комбинации диффузного и зеркального отражений

$$g(\vec{r}, \vec{\omega}', \vec{\omega}, \nu) = [1 - \rho_{sp}(\vec{r}, \nu)]g_d + \rho_{sp}(\vec{r}, \nu)g_{sp},$$

где  $0 \leq \rho_{sp}(\vec{r}, \nu) \leq 1$  — доля отражаемая зеркально; индикатриса диффузного отражения определяется равенством

$$g_d = g(\vec{r}, \vec{\omega}', \vec{\omega}, \nu) = 1,$$

а индикатриса зеркального отражения есть

$$g_{sp} = g_{sp}(\vec{r}, \vec{\omega}', \vec{\omega}, \nu) = \pi \delta(\vec{\omega}, \vec{\omega}_{sp}(\vec{\omega}')) / (\vec{n} \cdot \vec{\omega}),$$

здесь  $\vec{\omega}_{sp}(\vec{\omega}') = \vec{\omega}' - 2(\vec{n} \cdot \vec{\omega}')\vec{n}$  — направление луча после зеркального отражения в точке с нормалью  $\vec{n}$ .

**Источники излучения.** При расчете радиационного давления учитываем четыре вида источников.

1) Прямое излучение Солнца. Характеризуется параллельным потоком заданного направления  $\vec{\omega}_S$ , распределение по частоте имеет плотность вероятности

$$p_0(\nu) = \pi I_0(\nu, T_S) / \sigma_0 T_S^4, \quad (3)$$

а также известна  $E_S^0 = 1367$  Вт/м<sup>2</sup> — поверхностная плотность интегрального излучения Солнца на расстоянии 1 а.е. (=  $1.496 \cdot 10^{11}$  м).

2) Солнечное излучение, отраженное от Земли. Спектр его совпадает с солнечным, а плотность уходящего излучения  $E_{Ref} \approx 265$  Вт/м<sup>2</sup>, которая, в общем случае, зависит от принятого альбедо Земли и угла «Земля-КА-Солнце» [22].

3) Собственное излучение Земли. По данным ECSS [23] инфракрасное излучение Земли хорошо приближается функцией Планка при температуре  $T_E = 288$  К с плотностью потока вблизи поверхности  $E_{Inf} \approx 230$  Вт/м<sup>2</sup>. Известно также более общее соотношение для плотности потока собственного излучения на верхней границе атмосферы [24]

$$E_{Inf} = \frac{1 - \alpha_E}{4} E_S^0,$$

где  $\alpha_E$  — альбедо системы Земля–атмосфера ( $\approx 0.30$ – $0.35$ ).

4) Излучение поверхности КА при известной ее температуре  $T(\vec{r})$  и степени черноты  $\varepsilon$ . Плотность вероятности для частоты имеет вид

$$p_0(\nu) = \pi I_0(\nu, T(\vec{r})) / \sigma_0 T^4(\vec{r}),$$

распределение излучения по направлениям — диффузное, а плотность потока излучения  $E_{St}^0(\vec{r}) = \int d\nu \int_{\Omega_+} d\vec{\omega} \varepsilon(\vec{r}, \vec{\omega}, \nu, T(\vec{r})) I_0(\nu, T(\vec{r}))$  Вт/м<sup>2</sup>.

Таким образом, если форма аппарата не является выпуклой, то одни участки его поверхности могут оказывать силовое воздействие на другие участки как в результате собственных излучений, так и вследствие отражения фотонов от внешних источников.

## 2.1.2 Вычислительный алгоритм

**Общая схема алгоритма.** Применяем общую схему вычисления линейных функционалов от решения интегрального уравнения второго рода прямым методом Монте-Карло [13, 14, 25], являющегося частным случаем более общих весовых методов [26]. В фазовом пространстве формируется марковская цепь точек  $x_0, x_1, \dots, x_J$  с плотностью начального распределения  $f(x)/Q_0$ , где  $Q_0 = \int_X f(x) dx$  и переходной плотностью  $k(x', x)$ , причем величина  $q(x') = 1 - \int_X k(x', x) dx$  есть вероятность обрыва цепи в точке  $x'$ . Тогда стандартная оценка по столкновениям имеет вид  $\vec{\xi} = Q_0 \sum_{n=0}^J \vec{h}(x_n)$ , а  $\vec{F} = (\varphi, \vec{h}) = \mathbf{M}\vec{\xi}$ . Все условия для корректности математических выкладок считаются выполненными [25]:  $\varphi \in L_1(X)$ ,  $\vec{h} \in L_\infty(X)$ ,  $\exists n_0 : \|K^{n_0}\| < 1$ , кусочная непрерывность  $\varepsilon, T$  и т.п.

В силу линейности задачи (1)–(2) при нахождении итоговой силы, действующей на космический аппарат, находим последовательно силы отдельно для каждого из четырех источников, а затем берем их векторную сумму.

**Распараллеливание.** Распараллеливание расчетов выполнялось следующим образом: в силу независимости отдельных статистических испытаний подпрограмма, которая на однопроцессорной машине выполняет  $N$  испытаний, запускалась одновременно на  $N_{CPU}$  процессорах, проводя по  $N_t = N/N_{CPU}$  испытаний на каждом. В результате процессоры выдавали свои оценки вектора действующей силы  $\vec{F}_i$ , а затем брали среднее арифметическое:

$$\vec{F} = \frac{1}{N_{CPU}} \sum_{i=1}^{N_{CPU}} \vec{F}_i. \quad (4)$$

При этом обеспечивалось, чтобы каждый процессор работал со своим набором случайных чисел, т.к. иначе расчеты, проведенные одновременно на разных процессорах были бы тождественны.

## 2.1.3 Примеры расчетов

**Сравнение с точным решением.** Рассматривается аппарат в форме шара радиуса  $R$  и единственный источник — солнечное излучение, направленное по некоторому вектору  $\vec{\omega}_S$ . Считаем, что серая поверхность шара является полностью отражающей, притом диффузно-зеркальным образом, т.е. задается  $\rho_{df}$  — доля отражаемая диффузно и, соответственно,  $\rho_{sp} = 1 - \rho_{df}$  — доля отражаемая зеркально. Тогда сила светового давления, действующая на шар, имеет вид [27, 28]:

$$\vec{F}_{ex} = \frac{\pi(9 + 4\rho_{df})R^2 E_S^0}{9c_0} \vec{\omega}_S, \quad \text{Н.} \quad (5)$$

В численном примере полагали  $R = 3\text{ м}$ ,  $\vec{\omega}_S = (-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3})$ . Поверхность шара разбивалась на плоские зоны с помощью равномерной по  $\theta$  и  $\phi$  сетки в сферической системе координат  $(r, \theta, \phi)$ , где  $r \equiv 3$ ,  $0 \leq \phi \leq 2\pi$  и  $0 \leq \theta \leq \pi$ . В расчетах использовались три сетки с числом зон 156, 625 и 2400. Большинство зон четырехугольные и только прилегающие к двум полюсам — треугольные (рис. 1).

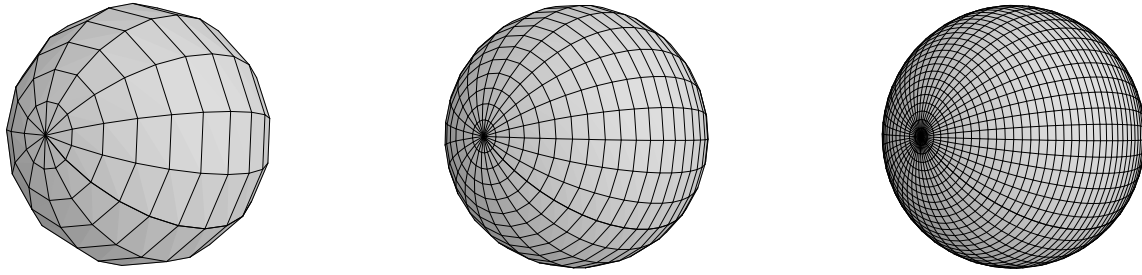


Рис. 1. Расчетные сетки: слева — 156 зон, в центре — 625 зон, справа — 2400 зон

В таблицах 1–6 приводятся некоторые результаты расчетов, выполнявшихся на кластере МВС-1000М, имевшем 768 процессоров DEC Alpha 21264. Обозначения в таблицах:  $N_Z$  — число зон в расчетной сетке;  $N_t$  — число испытаний в методе Монте-Карло;  $F$  — длина вектора силы, рассчитанная методом Монте-Карло;  $F_{ex}$  — точное значение силы для шара согласно формуле (5); Err — относительная погрешность расчета по сравнению с точным решением; Run — время счета; Eff — коэффициент эффективности распараллеливания, сообщаемый системой DVM, и определяемый как отношение Prod/(Prod+Lost), где Prod — время счета, умноженное на число процессоров; Lost — суммарное потерянное время, образуемое дублированием вычислений на разных процессорах (недостаточный параллелизм), общим временем коммуникаций, разбалансировкой, простоями отдельных процессоров и т.п.

Таблица 1. Результаты расчетов для шара на одном процессоре

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , Н	$F_{ex}$ Н	Err, %	Run, сек	Eff
0.0	1.0	156	1000000	1.24978E-04	1.28926E-04	3.06	18.42	1.00
0.7	0.3	156	1000000	1.63962E-04	1.69036E-04	3.00	18.80	1.00
1.0	0.0	156	1000000	1.80663E-04	1.86226E-04	2.99	18.82	1.00
0.0	1.0	625	1000000	1.27951E-04	1.28926E-04	0.75	68.46	1.00
0.7	0.3	625	1000000	1.67800E-04	1.69036E-04	0.73	69.10	1.00
1.0	0.0	625	1000000	1.84893E-04	1.86226E-04	0.72	68.31	1.00
0.0	1.0	2400	1000000	1.28663E-04	1.28926E-04	0.20	279.1	1.00
0.7	0.3	2400	1000000	1.68697E-04	1.69036E-04	0.20	273.2	1.00
1.0	0.0	2400	1000000	1.85882E-04	1.86226E-04	0.18	273.5	1.00

Из таблиц видно, что при увеличении числа процессоров сначала наблюдается линейное ускорение счета при практически неизменной погрешности, затем идет постепенный выход на «насыщение». Число испытаний на одном процессоре (1 млн.) взято здесь с «запасом». В реальных задачах оно подбирается на опыте, с тем, чтобы обеспечить необходимую точность, либо использовать известные способы определения числа испытаний из теории метода Монте-Карло. Если число процессоров еще увеличить (64, 128 и т.п.), то

Таблица 2. Результаты расчетов для шара на 2-х процессорах

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , H	$F_{ex}$ H	Err, %	Run, сек	Eff
0.0	1.0	156	500000	1.25071E-04	1.28926E-04	2.99	9.34	0.99
0.7	0.3	156	500000	1.63952E-04	1.69036E-04	3.00	9.56	0.99
1.0	0.0	156	500000	1.80653E-04	1.86226E-04	2.99	9.44	0.99
0.0	1.0	625	500000	1.28073E-04	1.28926E-04	0.66	34.50	0.99
0.7	0.3	625	500000	1.67808E-04	1.69036E-04	0.72	37.05	0.97
1.0	0.0	625	500000	1.84926E-04	1.86226E-04	0.70	34.90	0.99
0.0	1.0	2400	500000	1.28774E-04	1.28926E-04	0.12	147.5	0.97
0.7	0.3	2400	500000	1.68702E-04	1.69036E-04	0.19	148.8	0.96
1.0	0.0	2400	500000	1.85901E-04	1.86226E-04	0.17	147.7	0.97

Таблица 3. Результаты расчетов для шара на 4-х процессорах

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , H	$F_{ex}$ H	Err, %	Run, сек	Eff
0.0	1.0	156	250000	1.24985E-04	1.28926E-04	3.05	4.79	0.96
0.7	0.3	156	250000	1.64016E-04	1.69036E-04	2.97	4.86	0.97
1.0	0.0	156	250000	1.80689E-04	1.86226E-04	2.97	4.80	0.97
0.0	1.0	625	250000	1.27883E-04	1.28926E-04	0.81	17.38	0.98
0.7	0.3	625	250000	1.67865E-04	1.69036E-04	0.69	17.74	0.98
1.0	0.0	625	250000	1.84972E-04	1.86226E-04	0.67	17.63	0.98
0.0	1.0	2400	250000	1.28616E-04	1.28926E-04	0.24	82.4	0.90
0.7	0.3	2400	250000	1.68730E-04	1.69036E-04	0.18	73.0	0.96
1.0	0.0	2400	250000	1.85948E-04	1.86226E-04	0.15	73.7	0.96

Таблица 4. Результаты расчетов для шара на 8-ми процессорах

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , H	$F_{ex}$ H	Err, %	Run, сек	Eff
0.0	1.0	156	125000	1.24916E-04	1.28926E-04	3.11	4.83	0.53
0.7	0.3	156	125000	1.64159E-04	1.69036E-04	2.88	4.78	0.54
1.0	0.0	156	125000	1.80815E-04	1.86226E-04	2.91	4.95	0.53
0.0	1.0	625	125000	1.27778E-04	1.28926E-04	0.89	20.03	0.49
0.7	0.3	625	125000	1.68023E-04	1.69036E-04	0.60	17.82	0.53
1.0	0.0	625	125000	1.85059E-04	1.86226E-04	0.63	20.47	0.50
0.0	1.0	2400	125000	1.28528E-04	1.28926E-04	0.30	75.3	0.53
0.7	0.3	2400	125000	1.68938E-04	1.69036E-04	0.06	74.9	0.54
1.0	0.0	2400	125000	1.86065E-04	1.86226E-04	0.09	77.4	0.52



Таблица 5. Результаты расчетов для шара на 16-ти процессорах

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , H	$F_{ex}$ H	Err, %	Run, сек	Eff
0.0	1.0	156	62500	1.25080E-04	1.28926E-04	2.98	1.58	0.79
0.7	0.3	156	62500	1.64327E-04	1.69036E-04	2.79	1.51	0.80
1.0	0.0	156	62500	1.80914E-04	1.86226E-04	2.85	1.51	0.81
0.0	1.0	625	62500	1.27925E-04	1.28926E-04	0.78	5.32	0.81
0.7	0.3	625	62500	1.68184E-04	1.69036E-04	0.50	6.74	0.69
1.0	0.0	625	62500	1.85091E-04	1.86226E-04	0.61	5.31	0.82
0.0	1.0	2400	62500	1.28696E-04	1.28926E-04	0.18	23.4	0.80
0.7	0.3	2400	62500	1.69109E-04	1.69036E-04	0.04	24.7	0.74
1.0	0.0	2400	62500	1.86107E-04	1.86226E-04	0.06	22.9	0.80

Таблица 6. Результаты расчетов для шара на 32-х процессорах

$\rho_{df}$	$\rho_{sp}$	$N_Z$	$N_t$	$F$ , H	$F_{ex}$ H	Err, %	Run, сек	Eff
0.0	1.0	156	31250	1.25266E-04	1.28926E-04	2.84	1.14	0.53
0.7	0.3	156	31250	1.64608E-04	1.69036E-04	2.62	1.12	0.55
1.0	0.0	156	31250	1.81093E-04	1.86226E-04	2.76	1.19	0.51
0.0	1.0	625	31250	1.28127E-04	1.28926E-04	0.62	4.05	0.55
0.7	0.3	625	31250	1.68349E-04	1.69036E-04	0.41	4.22	0.54
1.0	0.0	625	31250	1.85240E-04	1.86226E-04	0.53	4.51	0.50
0.0	1.0	2400	31250	1.28865E-04	1.28926E-04	0.05	14.7	0.62
0.7	0.3	2400	31250	1.69317E-04	1.69036E-04	0.16	14.7	0.63
1.0	0.0	2400	31250	1.86244E-04	1.86226E-04	0.01	14.2	0.64

начинает расти относительная ошибка, поскольку число испытаний на каждом процессоре уменьшается и становится слишком малым для получения достаточно точного решения. Соответственно уменьшается и эффективность распараллеливания (до 0.1 и менее) в связи с ростом системных затрат времени на обслуживание коммуникаций при одновременном сокращении требуемого полезного времени счета.

**Схематическая модель спутника.** Рассмотрим аппарат (рис. 2), у которого имеются: 1) «центральный блок» с 8-ю боковыми гранями высотой 4 м, шириной 1 м (4 шт.) и 0.7 м (4 шт.), высота каждой пирамиды в нижнем и верхнем основании центрального блока составляет 0.5 м; 2) «солнечные панели» размером  $2.4 \times 5$  м каждая. Толщиной солнечных панелей для простоты пренебрегаем. Всего в расчетной сетке 40 зон: в центральном блоке каждый плоский участок образует отдельную зону (в сумме 24 зоны), а каждая сторона солнечной панели разбита на 4 зоны (линии деления проходят по диагоналям), т.е. в сумме на обеих солнечных панелях имеется 16 зон.

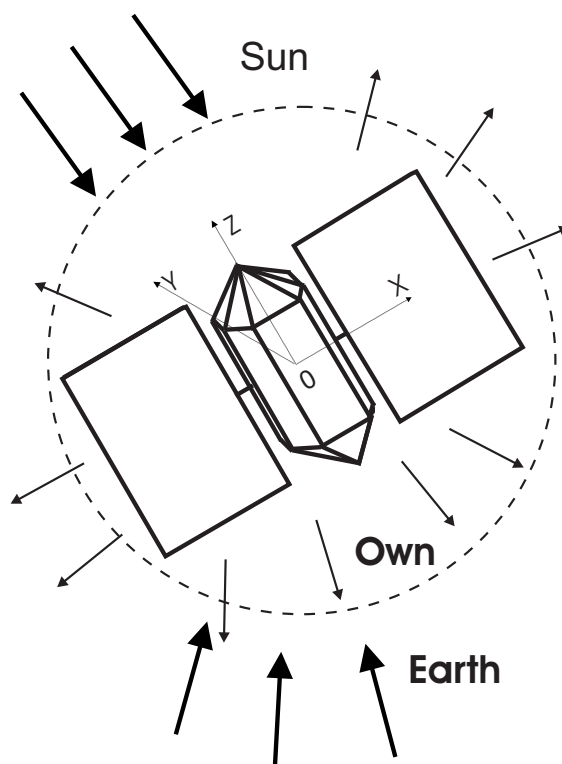


Рис. 2. Схематическая модель спутника (пунктир — воображаемая сферическая граница расчетной области)

Коэффициент поглощения в этих 16-ти зонах полагали  $\alpha = 0.76$ , доли излучения отражаемые зеркально и диффузно равно  $\rho_{sp} = 0.125$ ,  $\rho_{df} = 0.875$ , соответственно. Во всех зонах центрального блока было принято  $\alpha = 0.14$ ,  $\rho_{sp} = 0.0$ ,  $\rho_{df} = 1.0$ . Степень черноты определяется из отношения  $r = \alpha/\varepsilon$ , которое задается в исходных данных. Здесь полагали  $r = 1$ . Рассматриваемая конфигурация уже не является строго выпуклой, поэтому процесс отражения луча от стенок аппарата может быть многократным. В приведенных ниже расчетах максимально было отмечено 6 точек отражения у одного из лучей.

Направление излучения Солнца взято в данном примере по нормали к солнечным

панелям  $\vec{\omega}_S = (0, -1, 0)$ , т.е. ось  $OY$  направлена на Солнце. А направление от центра Земли было задано вектором  $\vec{\omega}_E = (-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3})$ .

Температуры поверхностей, от которых зависит, в частности, собственное излучение стенок, в расчетах полагали:  $120^\circ\text{C}$  — для 9-ти зон, нормали которых направлены строго на Солнце,  $80^\circ\text{C}$  — для 8-ми «скошенных» зон центрального блока, расположенных на солнечной стороне,  $0^\circ\text{C}$  — для 6-ти зон, нормали которых перпендикулярны  $\vec{\omega}_S$  и  $-100^\circ\text{C}$  — для 17-ти зон, находящихся строго в тени. Заметим, что температура поверхности аппарата, в один и тот же момент времени может меняться от  $-150$  до  $+150^\circ\text{C}$  [22], хотя этот диапазон для современных аппаратов будет, вероятно, меньше. Для «автоматического» определения температуры зон следовало бы решать уравнение теплопроводности для аппарата, рассматриваемого как твердое тело, в сопряжении с внешним радиационным теплопереносом, что не входило в задачу данной работы.

В таблицах 7–12 приведены вычисленные значения сил, действующих на спутник: столбиком указаны все три компоненты вектора, т.е.  $(F_x, F_y, F_z)$ . Число испытаний для каждого из 4-х источников излучения составляло  $N_t = 1000000$ . Обозначения в таблицах:  $E_{ref}$  — поверхностная плотность солнечного потока, отраженного от Земли, и наблюдаемую на высоте полета спутника,  $\text{Вт}/\text{м}^2$ ;  $E_{Inf}$  — аналогичная величина для собственного излучения Земли;  $\vec{F}_{Sun}$  — сила, вызванная прямым излучением Солнца;  $\vec{F}_{ref}$  — сила, вызванная излучением, отраженным от Земли;  $\vec{F}_{Inf}$  — сила, вызванная собственным излучением Земли;  $\vec{F}_{Satell}$  — сила, вызванная собственным излучением КА;  $\vec{F}_{tot}$  — результирующая сила; Run — время счета на МВС-1000М.

Поскольку величины  $E_{ref}$  и  $E_{Inf}$  на высоте орбиты точно не известны, то здесь приводятся демонстрационные сравнительные расчеты для трех разных значений этих величин. Для реальных приложений подобные константы (или функции времени) следует уточнить экспериментально, подобно тому как это сделано, например, в [24] для спутников, летавших на высотах  $\approx 600$  км.

Таблица 7. Результаты расчетов при  $E_{ref} = 265.0$ ,  $E_{Inf} = 230.0$ ,  $N_{CPU} = 1$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
(9.436E-9, -1.711E-4, -3.246E-8)	(-1.144E-5, -1.305E-5, -1.106E-5)	(-9.931E-6, -1.132E-5, -9.607E-6)	(-1.113E-8, -3.974E-5, 5.721E-8)	(-2.138E-5, -2.352E-4, -2.064E-5)	44.11

В таблицах 7–9 приведены результаты расчетов, выполнявшихся на одном процессоре. В таблицах 10–12 — результаты тех же самых расчетов, выполнявшихся на 20-ти процессорах, но при этом число статистических испытаний на каждом процессоре составляло  $N_t = 1000000/20 = 50000$  с усреднением результатов по формуле (4).

Можно отметить, что поскольку в данных примерах направление излучения Солнца было взято строго вдоль оси  $OY$ , то в силах  $\vec{F}_{Sun}$  и  $\vec{F}_{Satell}$  преобладает проекция на эту ось. Проекция на оси  $OX$  и  $OZ$  меньше на несколько порядков и ими можно пренебречь. Направление от центра Земли было задано «по диагонали» декартовой системы координат,

Таблица 8. Результаты расчетов при  $E_{ref} = 153.5$ ,  $E_{Inf} = 133.2$ ,  $N_{CPU} = 1$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
(9.436E-9, -1.711E-4, -3.246E-8)	(-6.629E-6, -7.560E-6, -6.405E-6)	(-5.751E-6, -6.553E-6, -5.563E-6)	(-1.113E-8, -3.974E-5, 5.721E-8)	(-1.238E-5, -2.249E-4, -1.194E-5)	43.58

Таблица 9. Результаты расчетов при  $E_{ref} = 15.46$ ,  $E_{Inf} = 13.42$ ,  $N_{CPU} = 1$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
9.436E-9, 1.711E-4, -3.246E-8)	(-6.680E-7, -7.618E-7, -6.453E-7)	(-5.797E-7, -6.605E-7, -5.607E-7)	(-1.113E-8, -3.974E-5, 5.721E-8)	(-1.249E-6, -2.123E-4, -1.181E-6)	44.53

Таблица 10. Результаты расчетов при  $E_{ref} = 265.0$ ,  $E_{Inf} = 230.0$ ,  $N_{CPU} = 20$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
(3.477E-7, -1.724E-4, -3.215E-8)	(-1.158E-5, -1.317E-5, -1.106E-5)	(-9.995E-6, -1.136E-5, -9.560E-6)	(-5.830E-8, -3.955E-5, 1.174E-7)	(-2.129E-5, -2.364E-4, -2.047E-5)	2.83

Таблица 11. Результаты расчетов при  $E_{ref} = 153.5$ ,  $E_{Inf} = 133.2$ ,  $N_{CPU} = 20$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
(3.477E-7, -1.724E-4, -3.215E-8)	(-6.710E-6, -7.630E-6, -6.407E-6)	(-5.788E-6, -6.581E-6, -5.537E-6)	(-5.830E-8, -3.955E-5, 1.174E-7)	(-1.221E-5, -2.261E-4, -1.179E-5)	3.02

Таблица 12. Результаты расчетов при  $E_{ref} = 15.46$ ,  $E_{Inf} = 13.42$ ,  $N_{CPU} = 20$

$\vec{F}_{Sun}$ , Н	$\vec{F}_{ref}$ , Н	$\vec{F}_{Inf}$ , Н	$\vec{F}_{Satell}$ , Н	$\vec{F}_{tot}$ , Н	Run, сек
(3.477E-7, -1.724E-4, -3.215E-8)	(-6.758E-7, -7.685E-7, -6.423E-7)	(-5.832E-7, -6.630E-7, -5.578E-7)	(-5.830E-8, -3.955E-5, 1.174E-7)	(-9.695E-7, -2.133E-4, -1.053E-6)	2.71

поэтому проекции сил  $\vec{F}_{ref}$  и  $\vec{F}_{Inf}$  получились близки по величине. Коэффициент эффективности распараллеливания в данных примерах оказался 0.80–0.83. Время счета можно еще сократить, если одну и ту же траекторию луча использовать для учета как отраженного, так и собственного излучений Земли. В представленных примерах формировалось два разных набора траекторий. Модули векторов результирующих сил при расчетах на одном и 20-ти процессорах отличаются на 0.4–0.5 %, что вполне допустимо, учитывая использование метода Монте-Карло. В детерминированных алгоритмах результаты от числа процессоров не должны зависеть. Из таблиц видно, что если аппарат имеет массу  $\simeq 220$  кг, и при заданных выше остальных условиях, то он получит ускорение  $\simeq 10^{-6}$  м/с<sup>2</sup>.

В работе [35] приведены также примеры расчетов с учетом «селективности», т.е. для случая, когда оптические характеристики поверхности аппарата зависят от частоты падающего излучения.

## 2.2 Уравнения мелкой воды

В рамках моделей мелкой воды хорошо описывается большой круг природных явлений, таких как крупномасштабные гравитационные поверхностные волны, возникающие в морях и океанах, цунами, приливные течения и т.п. [36, 37].

### 2.2.1 Исходные уравнения

Пусть  $(r, \lambda, \theta)$  — стандартная сферическая система координат с началом в центре земного шара. Далее будем использовать вместо угла  $\theta$  географическую широту  $\varphi = \pi - \theta$ , так что  $0 \leq \varphi < \pi$ . Полагаем всюду  $r = R_3$ , где  $R_3$  — радиус Земли. В рамках модели учитывается ускорение Кориолиса, угловая скорость вращения Земли считается постоянной.

Уравнения мелкой воды получаются из уравнений Навье-Стокса в гидростатическом приближении в предположении, что горизонтальные масштабы процессов много больше вертикальных. Число Россби (отношение силы инерции к силе Кориолиса) считается малым, что позволяет пренебречь нелинейными слагаемыми.

В итоге рассматривается задача в следующей постановке. Пусть  $\Omega$  — заданная область на сфере с границей  $\Gamma = \Gamma_1 \cup \Gamma_2$ , где  $\Gamma_1$  — часть границы, проходящая вдоль берега, а  $\Gamma_2 = \Gamma \setminus \Gamma_1$  — часть границы, проходящая по морю. Обозначим через  $m_1$  и  $m_2$  — характеристические функции соответствующих участков границы. Для простоты считается, что точки  $\varphi = 0$  и  $\varphi = \pi$  (полюса) не входят в  $\Omega$ . Относительно неизвестных функций  $u = u(t, \lambda, \varphi)$ ,  $v = v(t, \lambda, \varphi)$  и  $\xi = \xi(t, \lambda, \varphi)$  запишем в  $\Omega \times (0, T)$  уравнения баланса импульсов и уравнение неразрывности [39]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= lv + mg \frac{\partial \xi}{\partial \lambda} - Ru + f_1, \\ \frac{\partial v}{\partial t} &= -lu + ng \frac{\partial \xi}{\partial \varphi} - Rv + f_2, \\ \frac{\partial \xi}{\partial t} &= m \left( \frac{\partial}{\partial \lambda} (Hu) + \frac{\partial}{\partial \varphi} \left( \frac{n}{m} Hv \right) \right) + f_3, \end{aligned} \tag{6}$$

где  $u, v$  — компоненты вектора скорости  $\mathbf{U}$  по осям  $\lambda$  и  $\varphi$  соответственно;  $\xi$  — отклонение

свободной поверхности от невозмущенного уровня;  $H(\lambda, \varphi)$  — глубина водоема в точке  $(\lambda, \varphi)$ ; функция  $R = r_*|\mathbf{U}|/H$  учитывает силу трения о дно,  $r_*$  — коэффициент трения;  $l = -2\omega \cos \varphi$  — параметр Кориолиса;  $m = 1/(R_3 \sin \varphi)$ ;  $n = 1/R_3$ ;  $g$  — ускорение силы тяжести;  $f_1 = f_1(t, \lambda, \varphi)$ ,  $f_2 = f_2(t, \lambda, \varphi)$  и  $f_3 = f_3(t, \lambda, \varphi)$  — заданные функции внешних воздействий.

Граничные условия берем в виде [39]:

$$HU_n + \beta m_2 \sqrt{gH} \xi = m_2 \sqrt{gH} d \quad \text{на } \Gamma \times (0, T), \quad (7)$$

где  $U_n = \mathbf{U} \cdot \mathbf{n}$ ,  $\mathbf{n} = (n_1, \frac{n}{m} n_2)$  — вектор единичной внешней нормали к границе;  $\beta \in [0, 1]$  — заданный параметр,  $d = d(t, \lambda, \varphi)$  — граничная функция, определенная на границе  $\Gamma_2$ . Зададим также начальные условия

$$u(0, \lambda, \varphi) = u_0(\lambda, \varphi), \quad v(0, \lambda, \varphi) = v_0(\lambda, \varphi), \quad \xi(0, \lambda, \varphi) = \xi_0(\lambda, \varphi).$$

Для дискретизации по времени разобьем временной отрезок  $[0, T]$  на  $K$  интервалов:  $0 = t_0 < t_1 < \dots < t_K = T$  с шагом  $\tau = T/K$ . Аппроксимируем производные по времени левыми разностями и рассмотрим систему (6)–(7) на временном интервале  $(t_k, t_{k+1})$ :

$$\begin{aligned} \left(\frac{1}{\tau} + R\right) u - lv - mg \frac{\partial \xi}{\partial \lambda} &= f_1 + \frac{1}{\tau} u^k & \text{в } \Omega, \\ \left(\frac{1}{\tau} + R\right) v + lu - ng \frac{\partial \xi}{\partial \varphi} &= f_2 + \frac{1}{\tau} v^k & \text{в } \Omega, \end{aligned} \quad (8)$$

$$\frac{1}{\tau} \xi - m \left( \frac{\partial}{\partial \lambda} (Hu) + \frac{\partial}{\partial \varphi} \left( \frac{n}{m} H v \right) \right) = f_3 + \frac{1}{\tau} \xi^k \quad \text{в } \Omega,$$

$$HU_n + \beta m_2 \sqrt{gH} \xi = m_2 \sqrt{gH} d \quad \text{на } \Gamma, \quad k = 0, 1, \dots, K-1, \quad (9)$$

где  $f(t_k, \lambda, \varphi) = f^k$ ,  $f(t_{k+1}, \lambda, \varphi) = f^{k+1} = f$ . Индекс  $k+1$  в разностных выражениях далее будем опускать там, где это не вызывает двойного толкования.

Для решения системы (8)–(9) применяем метод Галеркина на конечных элементах. Разобьем область  $\Omega$  на некоторое число  $N_{el}$  треугольных ячеек (элементов). Пусть  $N_{nd}$  — число узлов. Для  $t = t^{n+1}$  ищем решение в виде:

$$\Phi(\lambda, \varphi, t^{n+1}) = \sum_{j=1}^{N_{nd}} \alpha_j^\Phi(t^{n+1}) \Psi_j(\lambda, \varphi), \quad \Phi = (u, v, \xi)^\top \quad (10)$$

с условием ортогональности невязки базисным функциям:

$$\iint_{\Omega} \mathbb{R}_\Phi(\lambda, \varphi, t^{n+1}) \Psi_j(\lambda, \varphi) dS = 0, \quad j = 1, \dots, N_{nd}. \quad (11)$$

Условия (10)–(11) сводятся в итоге к системе линейных уравнений

$$A\Phi = \mathbf{B}$$

относительно  $3N_{nd}$  неизвестных  $\Phi = (\alpha_j^u, \alpha_j^v, \alpha_j^\xi)$ , которую решаем методом Гаусса-Зейделя

$$\Phi_j^{(\nu+1)} = \frac{1}{a_{jj}} \left( b_j - \sum_{\substack{i \in Nb(j) \\ i < j}} a_{ij} \Phi_j^{(\nu+1)} - \sum_{\substack{i \in Nb(j) \\ i > j}} a_{ij} \Phi_j^{(\nu)} \right), \quad j = 1, \dots, 3N_{nd}. \quad (12)$$

где  $Nb(j)$  — множество узлов, состоящее из узла  $j$  и всех соседних к нему, т.е. соединенных с  $j$ -м узлом стороной кого-либо треугольника,  $\nu$  — номер итерации.

Распараллеливание алгоритма состоит в том, что итерации в (12) проводятся одновременно на разных процессорах, но каждый из них имеет дело со своим множеством индексов  $j$ . При вычислении сумм вида  $\sum_{i \in Nb(j)}$ , входящих в (12), некоторые из соседних узлов могут оказаться на другом процессоре. Для доступа к таким узлам на каждом процессоре организуются вспомогательные массивы, которые и играют роль соседних узлов с другого процессора в момент нахождения упомянутых сумм.

Коммуникации между процессорами для обновления вспомогательных массивов проводятся перед очередной итерацией  $\nu$ , поэтому для тех узлов, у которых соседние узлы находятся на другом процессоре, частично используется метод Якоби, а не Гаусса-Зейделя.

### 2.2.2 Пример: распространение волн из центра квадрата

Рассмотрим решение уравнений (6)–(7) в квадрате размером  $2000 \times 2000$  км на неравномерной сетке  $201 \times 201$  узлов,  $N_{nd} = 40401$ ,  $N_{el} = 80000$ . Неравномерность состоит в том, что линейные размеры элементов максимальны в центре квадрата и уменьшаются по геометрической прогрессии по мере приближения к каждой из четырех границ. Фрагмент сетки (1/16 часть всей сетки) показан на рис. 3. В начальный момент задавался локальный подъем уровня в центре квадрата  $(\lambda_0, \varphi_0)$  вида:

$$\xi(0, \lambda, \varphi) = A \exp \left[ - \left( \frac{\lambda - \lambda_0}{2d} \right)^2 - \left( \frac{\varphi - \varphi_0}{2d} \right)^2 \right], \quad A = 2, d = 0.005.$$

Начальные скорости были нулевые. На границах квадрата задавались условия вида (7) при  $m_2 = 0$  на всей границе, т.е. условия непротекания через границу.

На рис. 4–6 видно как распространяется волна в среднем сечении квадрата при фиксированной географической широте  $\varphi = \varphi_0$ . Расчеты проводились на кластере МВС-15000ВМ. В табл. 13 представлены время счета, коэффициенты ускорения и эффективности для разного числа процессоров. Было сделано при этом 1000 шагов по времени.

$P$	$t_P$ , сек	$S = t_1/t_P$	$E_1 = S/P$
1	1602.5	1.000	1.000
2	866.74	1.849	0.924
3	574.19	2.791	0.930
4	450.86	3.554	0.889
6	318.03	5.039	0.839
8	252.47	6.347	0.793
10	214.14	7.483	0.748
12	190.55	8.410	0.701
16	159.53	10.04	0.628

Таблица 13. Результаты счета при разном числе процессоров

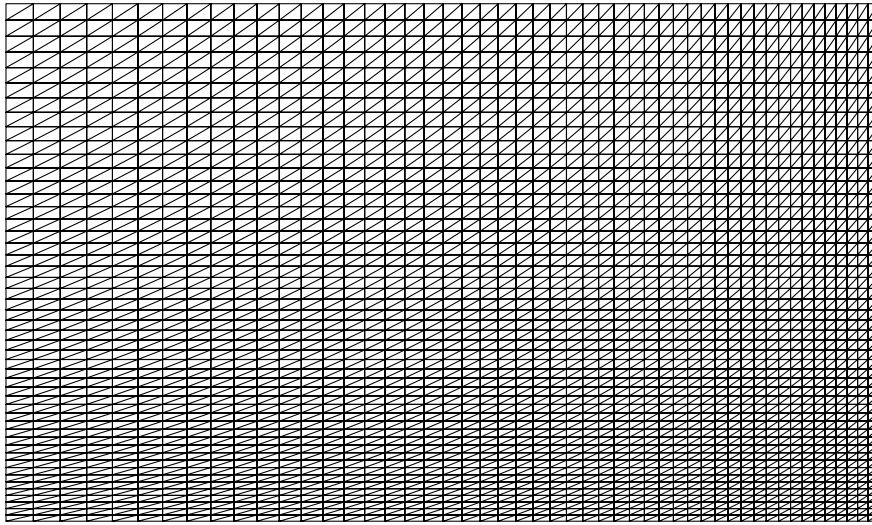


Рис. 3. Фрагмент сетки, правый нижний угол

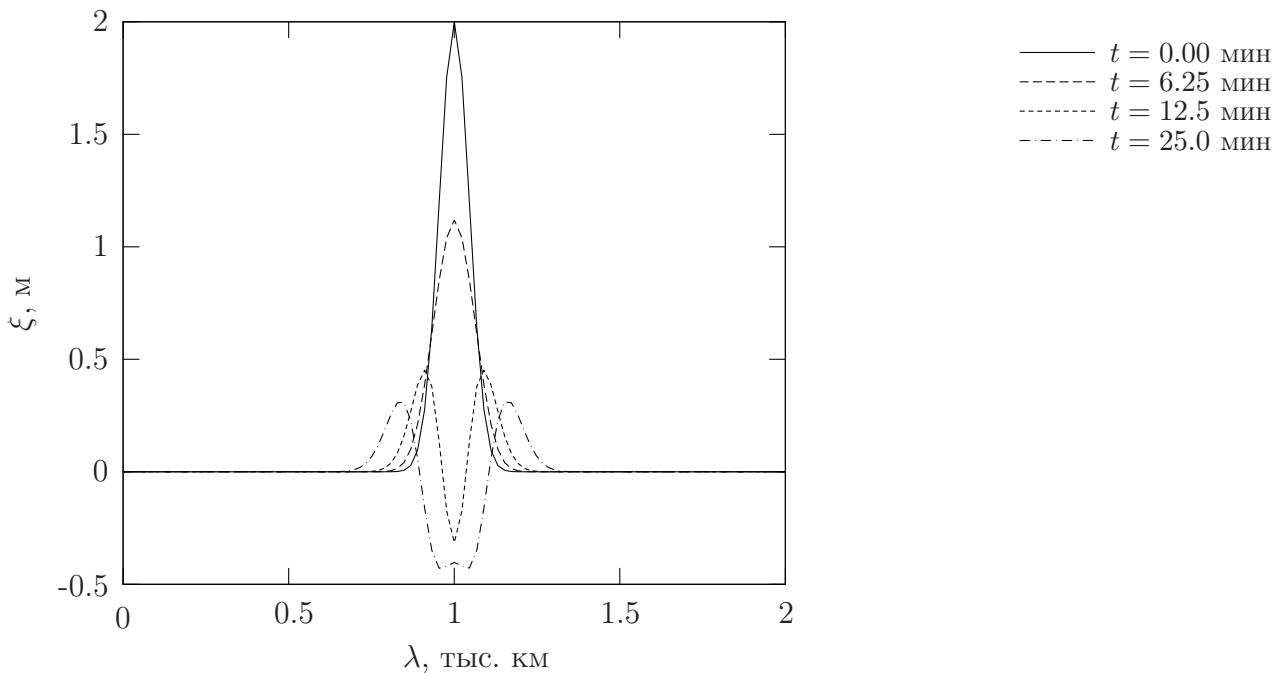


Рис. 4. Изменение поверхности со временем в среднем сечении квадрата



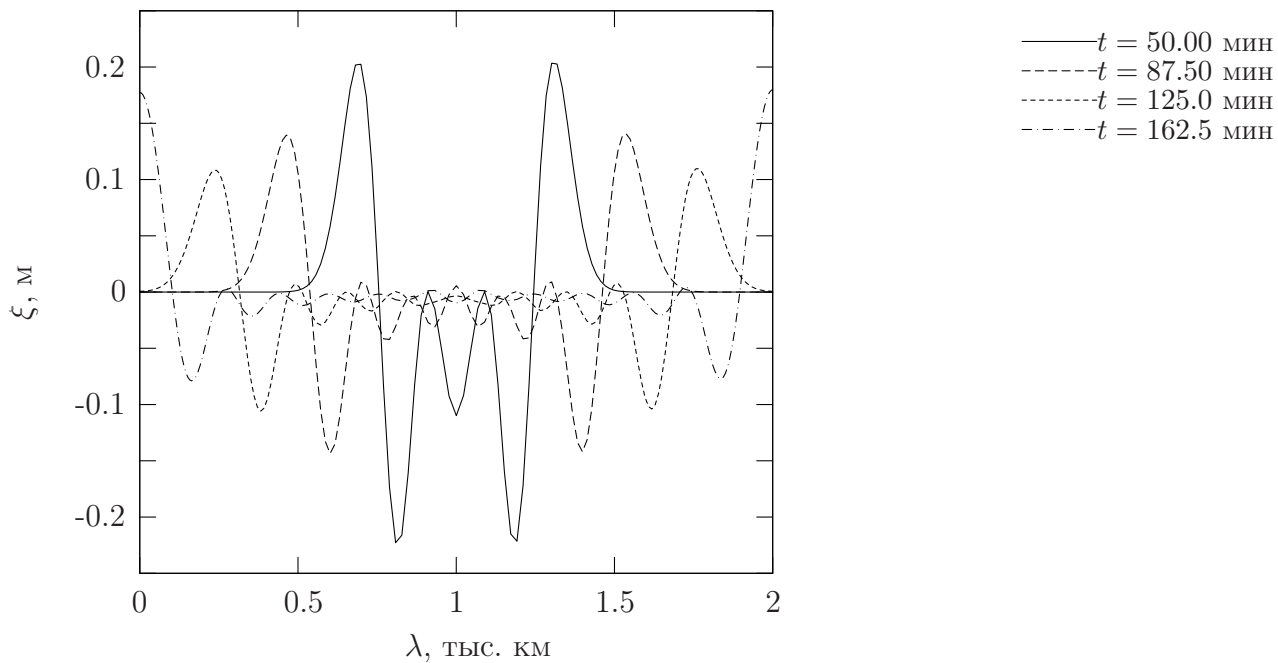


Рис. 5. Изменение поверхности со временем в среднем сечении квадрата

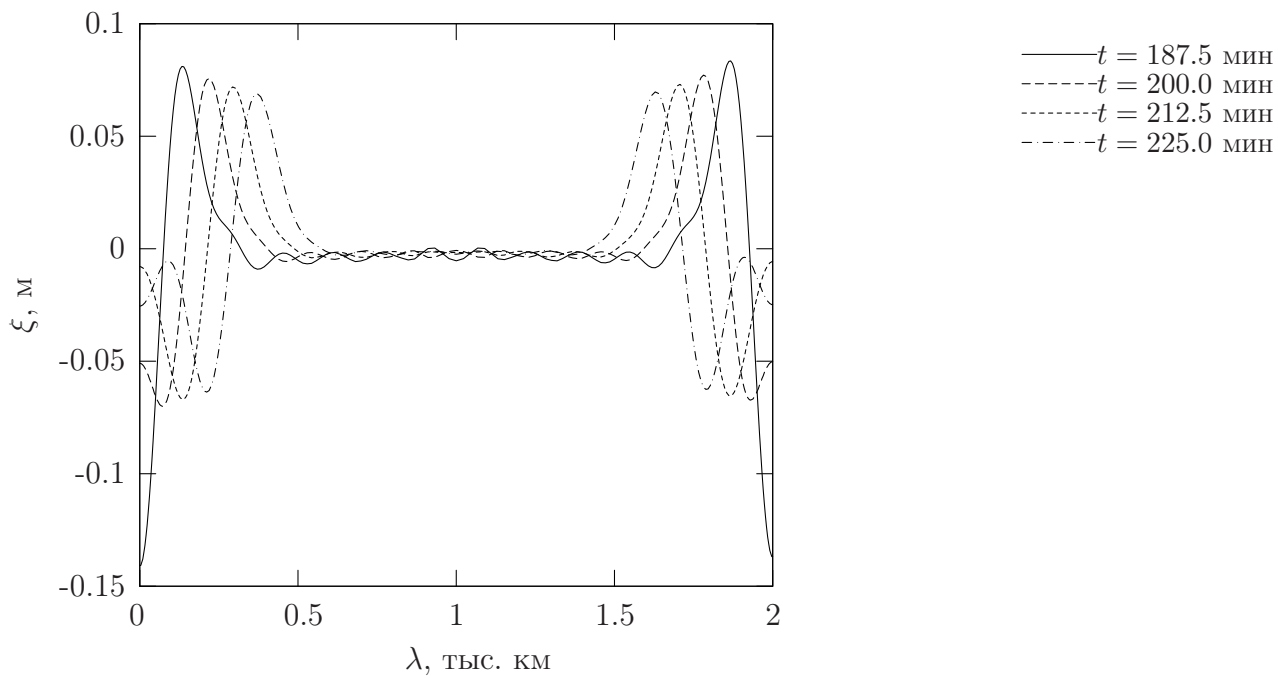


Рис. 6. Изменение поверхности со временем в среднем сечении квадрата

## 2.3 Метод решеточных уравнений Больцмана

Принято считать, что разработка метода решеточных уравнений Больцмана (LBM или LB-метода) началась с 1988 года, когда вышла статья Макнамары и Занетти [45]. Они предложили заменить логические переменные в давно известных дискретных моделях газов непрерывными функциями распределения. Статья положила начало огромному числу публикаций по LBM, например, [46]–[71]. LB-метод основан на статистической физике и описывает микроскопическое поведение частиц в упрощенной форме, но на макроскопическом уровне дает вполне корректные значения для скорости, плотности, давления и для других характеристик течения жидкости или газа.

Основные преимущества LB-метода: (1) исходные уравнения имеют простую форму, в них присутствуют производные только первого порядка, нелинейность имеется только в алгебраическом источнике; (2) ввиду локального характера вычислений метод легко реализуется на параллельных компьютерах; (3) процессы конвекции и диффузии учитываются посредством малого набора фиксированных векторов скоростей частиц, что сводит соответствующие вычисления к простому сдвигу вдоль этих векторов; (4) не нужно решать уравнение Пуассона на каждом шаге по времени во всей области для корректировки давления, как это делается во многих традиционных алгоритмах для решения уравнений Навье-Стокса; (5) алгоритм легко программируется для машинной реализации; (6) специалисты предсказывают этому методу большое перспективы для моделирования физических и химических процессов в геометрически сложных областях микро- и нано- размеров (пористые носители и микроструктуры).

### 2.3.1 Математическая модель

Следуя [49, 50, 51], рассмотрим основные уравнения и простейший численный алгоритм для моделирования изотермических несжимаемых потоков жидкости, как наиболее типичного случая.

LB-метод основан на классическом кинетическом уравнении Больцмана:

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \nabla_{\vec{r}} f + \vec{F} \cdot \nabla_{\vec{\xi}} f = Q(f), \quad (13)$$

где  $f \equiv f(\vec{r}, \vec{\xi}, t)$  — одночастичная функция распределения в фазовом пространстве  $(\vec{r}, \vec{\xi})$ ,  $\vec{r} = (x, y, z)$  — координата частицы,  $\vec{\xi}$  — ее скорость,  $t$  — время,  $\vec{F}$  — внешняя сила,  $Q(f)$  — интеграл столкновений.

Макроскопические переменные (плотность, скорости, давление) вычисляются как моменты от  $f$ :

$$\rho(\vec{r}, t) = \int f d^3 \xi, \quad \vec{u}(\vec{r}, t) = \frac{1}{\rho} \int \vec{\xi} f d^3 \xi, \quad p(\vec{r}, t) = \frac{1}{3} \int (\vec{\xi} - \vec{u})^2 f d^3 \xi.$$

Нелинейное интегро-дифференциальное уравнение (13) является слишком сложным для численного решения. Поэтому успех данного метода стал возможен только после изобретения упрощенных форм этого уравнения.

Во-первых, нелинейный многомерный интеграл столкновений  $Q(f)$  заменяется алгебраическим выражением, например, в форме Bhatnagar-Gross-Krook [72]:

$$Q(f) = -\frac{1}{\tau} (f - f^{eq}),$$

где

$$f^{eq} = \frac{\rho}{(2\pi RT)^{3/2}} \exp\left(-\frac{(\vec{u} - \vec{\xi})^2}{2RT}\right)$$

— функция равновесного распределения Максвелла-Больцмана,  $\tau$  — время релаксации при столкновениях,  $R$  — газовая постоянная,  $T$  — температура.

Во-вторых, фиксируется конечное множество (решетка) скоростей частиц в пространстве всех возможных скоростей:

$$\mathcal{L} = \{\vec{\xi}_m, m = 0, \dots, M-1\}.$$

Пример такой решетки для 3-D ( $M = 19$ ) пространства показан на рис. 7. Нулевой вектор всегда вводит в набор  $\mathcal{L}$ .

Обозначая  $f_m(\vec{r}, t) = f(\vec{r}, \vec{\xi}_m, t)$  и предполагая для простоты  $\vec{F} = \mathbf{0}$ , получаем систему дискретных уравнений Больцмана, которые и следует решать численно:

$$\partial_t f_m + \vec{\xi}_m \cdot \nabla_{\vec{r}} f_m = \frac{1}{\tau} (f_m^{eq} - f_m), \quad m = 0, \dots, M-1. \quad (14)$$

Предполагается также, что в пространстве задана равномерная сетка с шагом  $\delta_x$  по всем трем осям координат:

$$\mathcal{P} = \{\vec{r}_{ijk} = \vec{r}_0 + [(i-1)\delta_x, (j-1)\delta_x, (k-1)\delta_x], \\ i = 1, \dots, n_i, \quad j = 1, \dots, n_j, \quad k = 1, \dots, n_k\}.$$

Интегрируя (14) на малом интервале  $(0, \delta_t)$  вдоль прямой  $\vec{\xi}_m$  и пренебрегая переменными порядка  $O(\delta_t^2)$ , получаем решеточные уравнения Больцмана (ЛВ-уравнения):

$$f_m(\vec{r} + \vec{\xi}_m \delta_t, t + \delta_t) = f_m(\vec{r}, t) + \frac{\delta_t}{\tau} (f_m^{eq} - f_m), \quad m = 0, \dots, M-1,$$

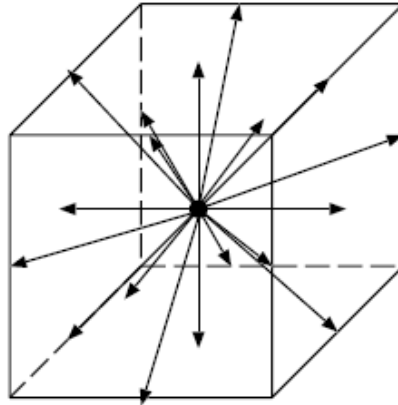


Рис. 7. Кубическая решетка скоростей  $\mathcal{L}$  для 3-D моделей ( $M = 19$ )

или, обозначая,  $\omega = \delta_t/\tau$ :

$$f_m(\vec{r} + \vec{\xi}_m \delta_t, t + \delta_t) = \omega f_m^{eq}(\vec{r}, t) + (1 - \omega) f_m(\vec{r}, t), \quad m = 0, \dots, M-1. \quad (15)$$

Уравнения (15) обычно решаются расщеплением на два шага [51]. На первом шаге учитываются только столкновения:

$$\widetilde{f}_m(\vec{r}, t) = \omega f_m^{eq}(\vec{r}, t) + (1 - \omega) f_m(\vec{r}, t), \quad m = 0, \dots, M-1, \quad \vec{r} \in \mathcal{P}.$$

На втором шаге происходят процессы переноса (или учет «свободного полета» [53]):

$$f_m(\vec{r} + \vec{\xi}_m \delta_t, t + \delta_t) = \widetilde{f}_m(\vec{r}, t), \quad m = 0, \dots, M-1.$$

Вычисляется плотность и скорость потока:

$$\rho(\vec{r}, t) = \sum f_m(\vec{r}, t), \quad \vec{u}(\vec{r}, t) = \frac{1}{\rho} \sum \vec{\xi}_m f_m(\vec{r}, t).$$

Было показано [50], что эти функции  $\rho(\vec{r}, t)$  и  $\vec{u}(\vec{r}, t)$  являются также решениями следующих уравнений Навье-Стокса:

$$\nabla \cdot \vec{u} = 0, \quad (16)$$

$$\rho \partial_t \vec{u} + \rho \vec{u} \cdot \nabla \vec{u} = -\nabla p + \nu \rho \nabla^2 \vec{u}, \quad (17)$$

где

$$p = c_s^2 \rho, \quad c_s = c/\sqrt{3}, \quad c = \delta_x/\delta_t, \quad \nu = \frac{(\tau^* - 1/2) \delta_x^2}{3 \delta_t}, \quad \tau^* = \tau/\delta_t.$$

При выводе уравнений (16)–(17) все выражения порядков  $O(\delta_x^2)$ ,  $O(Ma^2)$  и выше были отброшены. Здесь  $Ma = |\vec{u}|/c_s$  — число Маха, которое должно быть достаточно малым, чтобы обеспечить несжимаемость жидкости. Для этого случая и для постоянной температуры функции равновесного распределения могут быть записаны в более простом виде [49]:

$$f_m^{eq} = \rho w_m \left[ 1 + \frac{\vec{\xi}_m \cdot \vec{u}}{c_s^2} + \frac{1}{2c_s^4} \left( (\vec{\xi}_m \cdot \vec{u})^2 - c_s^2 |\vec{u}|^2 \right) \right], \quad m = 0, \dots, M-1,$$

где

$$w_0 = 1/3; \quad w_m = 1/9, \quad m = 1, \dots, 6; \quad w_m = 1/36, \quad m = 7, \dots, 18.$$

### 2.3.2 Пример расчета: течение в кубе с подвижной крышкой

Расчет ламинарного течения вязкой несжимаемой в 2-D квадратной области с подвижной верхней границей является классическим тестом, который хорошо изучен экспериментально и численно. Гораздо меньше работ по течениям в 3-D областях (рис. 8). Применим LB-метод для моделирования такого течения [71]. Пусть число Рейнольдса  $Re = U_L H/\nu = 1000$ , ребро куба  $H = 1$  м и кинематическая вязкость жидкости  $\nu = 1.5 \cdot 10^{-5}$  м<sup>2</sup>/с. Тогда скорость движения крышки составит  $U_L = 1.5 \cdot 10^{-2}$  м/с. На остальных стенках заданы условия прилипания.

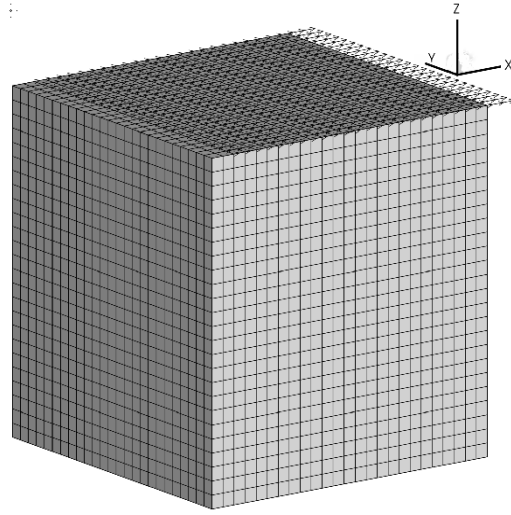


Рис. 8. Общий вид 3-D полости с подвижной крышкой.

На рис. 9–11 показано сравнение наших расчетов с расчетами из работы [73], в которой применялся смешанный метод граничных и конечных элементов. Картины течений в трех (по каждой из осей) средних сечениях куба оказались практически идентичными.

Для параллельных вычислений исходный куб разбиваем на прямоугольные блоки таким образом, чтобы в каждом блоке было бы по возможности равное количество узлов пространственной сетки. Уравнения в каждом блоке решаются своим отдельным процессором. Далее, для задания таких блоков используем запись  $[N_x, N_y, N_z]$ , где  $N_x, N_y, N_z$  — количество интервалов разбиения ребра куба вдоль осей X, Y, Z, соответственно. Т.е. общее число процессоров при этом составит  $N_{cpu} = N_x \cdot N_y \cdot N_z$ . В частности, запись  $[1,1,1]$  означает, что в таком варианте расчетов используется только один процессор.

Вычисления выполнялись для пяти равномерных сеток, начиная от  $64 \times 64 \times 64$  и до  $1024 \times 1024 \times 1024$ . В табл. 14 представлено время, необходимое для выполнения 1000 шагов по времени на разных сетках и при различном числе  $N_{cpu}$ . Из таблицы видно, как сокращается время счета с ростом числа процессоров.

Можно также сравнить время выполнения и коэффициент эффективности для вариантов с одинаковым числом процессоров, но с различной структурой блоков. Например, возьмем 128 процессоров, но с разным видом блоков:  $[8,4,4]$ ,  $[4,8,4]$  и  $[4,4,8]$ . Как видно из таблицы, блок вида  $[8,4,4]$  дают лучшие время и эффективность, чем блоки  $[4,8,4]$  и  $[4,4,8]$ . Последняя структура блоков дает худшие результаты. Этот факт можно объяснить принятым расположением массивов фортрана в памяти (первый индекс изменяется быстрее). Для блоков  $[8,4,4]$  время для коммуникаций между процессорами оказывается меньше чем для блоков  $[4,8,4]$  и еще меньше чем для  $[4,4,8]$ . Это подтверждается также статистикой, выдаваемой DVM-системой.

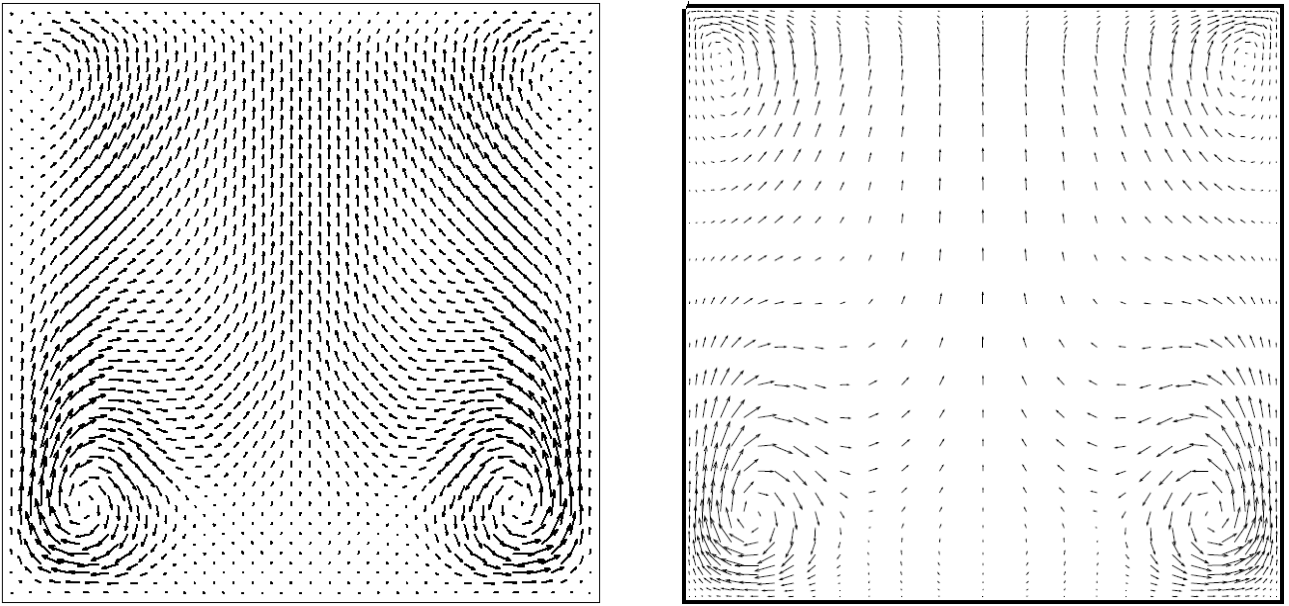


Рис. 9. Сравнение скоростных полей в вертикальном центрально-плоском  $x = \text{const}$ . Слева: наши расчеты; справа: из [73].

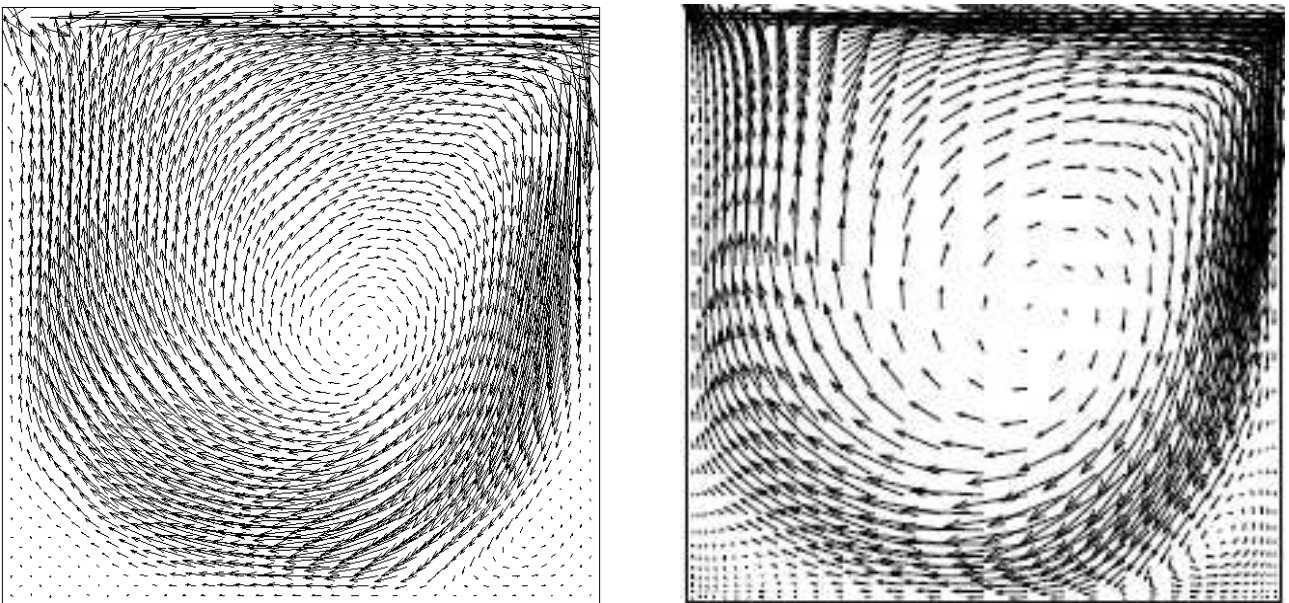


Рис. 10. Сравнение скоростных полей в вертикальном центрально-плоском  $y = \text{const}$ . Слева: наши расчеты; справа: из [73].

Таблица 14. Время выполнения и эффективность по сравнению со многими процессорами и структура блочного  $[N_x, N_y, N_z]$  для различного размера сетки.

Нет.	Number processors	$[N_x, N_y, N_z]$	Время выполнения (минуты) для 1000 time steps	Эффективность. Coef.
1) размер Сетки $64 \times 64 \times 64$				
1	1	[1,1,1]	1.437	1.000
2	2	[2,1,1]	0.900	0.960
3	2	[1,2,1]	0.974	0.947
4	2	[1,1,2]	1.021	0.936
5	4	[2,2,1]	0.807	0.913
6	8	[2,2,2]	0.846	0.686
2) размер Сетки $128 \times 128 \times 128$				
7	1	[1,1,1]	15.02	1.000
8	2	[2,1,1]	8.606	0.980
9	2	[1,2,1]	8.783	0.980
10	2	[1,1,2]	8.992	0.981
11	4	[2,2,1]	6.701	0.939
12	8	[2,2,2]	7.857	0.944
3) размер Сетки $256 \times 256 \times 256$				
13	32	[4,4,2]	15.48	0.850
14	64	[4,4,4]	7.132	0.648
15	128	[8,4,4]	3.601	0.613
16	128	[4,8,4]	4.047	0.490
17	128	[4,4,8]	4.167	0.451
18	256	[8,8,4]	1.859	0.496
4) размер Сетки $512 \times 512 \times 512$				
19	64	[4,4,4]	61.16	0.748
20	128	[8,4,4]	31.67	0.917
21	128	[4,8,4]	31.92	0.639
22	128	[4,4,8]	34.16	0.578
5) размер Сетки $1024 \times 1024 \times 1024$				
23	512	[8,8,8]	63.45	0.813
24	1024	[16,8,8]	32.83	0.902
25	1024	[8,16,8]	32.96	0.886
26	1024	[8,8,16]	47.38	0.494

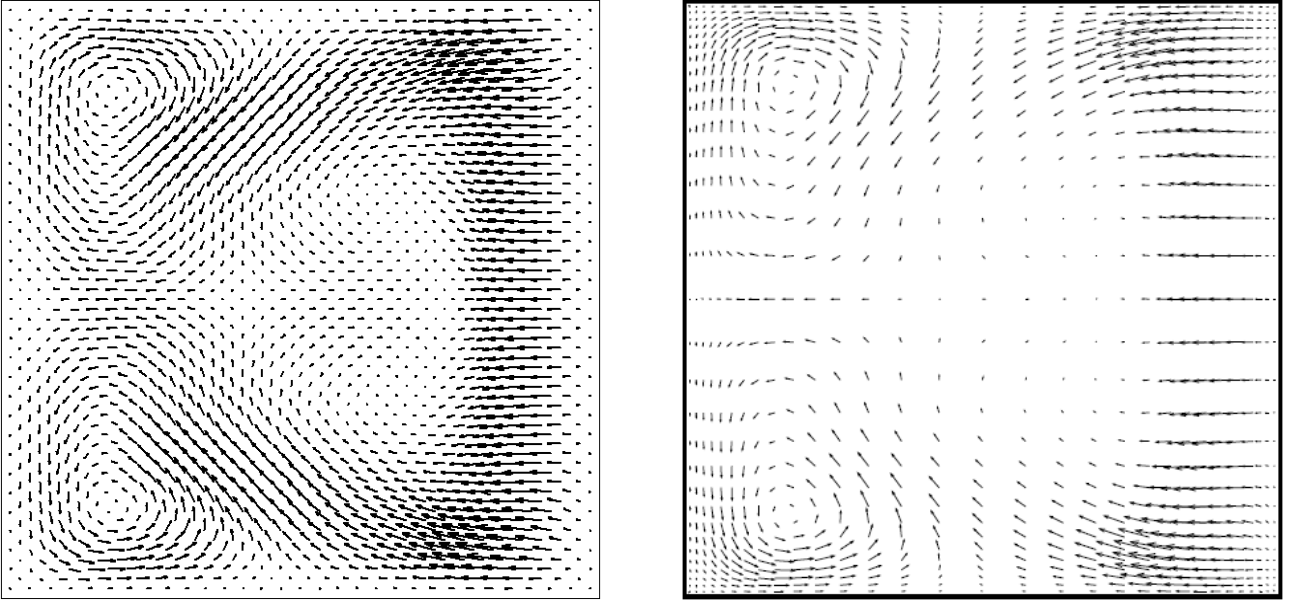


Рис. 11. Сравнение скоростных полей в горизонтальном  $z = \text{константа}$ . Слева: наши расчеты; справа: из [73].

## 2.4 Заключение

Применение DVM-системы программирования позволяет существенно упрощать и ускорять разработку и отладку параллельных программ для многих классов задач из различных прикладных направлений науки.

Так, в уравнениях мелкой воды граничная функция  $d(t, \lambda, \varphi)$  в уравнении (7), как правило, не задана, зато можно считать известными данные наблюдений за уровнем поверхности на некотором отрезке  $\Gamma_0$  вдоль морской границы  $\Gamma_2$ :

$$\xi = \xi_{obs} \quad \text{на} \quad \Gamma_0 \subset \Gamma_2.$$

Для решения таких задач привлекаются методы обратных задач, теории оптимального управления и сопряженных уравнений. Соответствующие теоретические обоснования, численные алгоритмы и программы разрабатываются в коллективах, руководимых В.И. Агошковым и В.В. Шайдуровым [38, 39, 40, 41, 42]. В частности, DVM-подход для параллельных расчетов реализован не только для решения «прямой» системы уравнений, но и сопряженной [43].

Кроме примеров приведенных выше, DVM-система распараллеливания успешно применялась нами для решения систем обыкновенных дифференциальных уравнений (включая жесткие системы) [74, 75, 76], для задач химической кинетики [77], а также молекулярной динамики.



## ЛИТЕРАТУРА

- [1] DVM-Система, <http://www.keldysh.ru/dvm>.
- [2] Konovalov N.A., Krukov V.A., Mihailov S.N., Pogrebtsov A.A. Fortran DVM — a Language for Portable Parallel Program Development. Proc. of Software For Multiprocessors & Supercomputers: Theory, Practice, Experience, Institute for System Programming RAS, Moscow. 124–133 (1994)
- [3] Коновалов Н.А., Крюков В.А., Михайлов С.Н., Погребцов А.А. Fortran-DVM — язык для разработки мобильных параллельных программ // Программирование. — 1995. — № 1. — С. 49–54.
- [4] Крюков В.А. Разработка параллельных программ для вычислительных кластеров и сетей // Информационные технологии и вычислительные системы. — 2003. — № 1–2. — С. 42–61.
- [5] Бахтин В.А., Коновалов Н.А., Крюков В.А. Расширение языка OpenMP Fortran для распределенных систем // Вопросы атомной науки и техники. Серия математическое моделирование физических процессов. 2002. Вып.4. С. 65–70.
- [6] Бахтин В.А., Воронков А.В., Голубев А.С., Крюков В.А., Поддерюгина Н.В., Сычугова Е.П., Устюгов С.Д. Использование языка Fortran-DVM/OpenMP для разработки больших задач // Труды Всерос. научной конф. «Научный сервис в сети Интернет: решение больших задач», сентябрь 2008 г., г. Новороссийск. —Ц М.: Изд-во МГУ, 2008, С. 185–191.
- [7] Клинов М.С., Крюков В.А. Автоматическое распараллеливание Фортран-программ. Отображение на кластер // Вестник Нижегородского университета им. Н.И. Лобачевского, 2009, № 2, С. 128–134.
- [8] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. Спб.: БХВ–Петербург, 2002, 608 с.
- [9] Координационный Научно-Информационный Центр системы ГЛОНАСС, <http://www.glonass-center.ru>.
- [10] Дубошин Г.Н. Справочное руководство по небесной механике и астродинамике. М.: Наука, 1976. 864 с.
- [11] Золотухин В.Г., Ермаков С.М. Применение метода Монте-Карло для расчета защиты от ядерных излучений // Вопросы физики защиты реакторов, М.: Госатомиздат, 1963. С. 171–181.
- [12] Марчук Г.И., Ермаков С.М. Метод Монте-Карло и методы вычислительной математики // Метод Монте-Карло в проблеме переноса излучений, М.: Атомиздат, 1967. С. 6–24.

- [13] Ермаков С.М. Метод Монте-Карло и смежные вопросы. М.: Наука, 1971. 328 с.
- [14] Михайлов Г.А. Некоторые вопросы теории методов Монте-Карло. Новосибирск: Наука, 1974. 144 с.
- [15] Спанье Дж., Гелбард Э. Метод Монте-Карло и задачи переноса нейтронов. М.: Атомиздат, 1972. 272 с.
- [16] Соболев И.М. Численные методы Монте-Карло. М.: Наука, 1973. 312 с.
- [17] Метод Монте-Карло в атмосферной оптике / Под ред. Г.И. Марчука. Новосибирск: Наука, 1976. 280 с.
- [18] Решение краевых задач методом Монте-Карло / Отв. ред. Б.А. Каргин. Новосибирск: Наука, 1980. 175 с.
- [19] Ухинов С.А. К решению задач теплообмена излучением методом Монте-Карло // Численные методы и статистическое моделирование в теории переноса. Новосибирск: ВЦ СО АН СССР, 1980. С. 155-162.
- [20] Ухинов С.А. Применение методов Монте-Карло и конечных элементов в расчетах теплообмена излучением // Теория и алгоритмы статистического моделирования. Новосибирск: ВЦ СО АН СССР, 1984. С. 112-121.
- [21] Махоткин О.А. Математические основы радиационного теплообмена. Новосибирск: ВЦ СО АН СССР, 1988. 156 с.
- [22] Залетаев В.М., Капинос Ю.В., Сургучев О.В. Расчет теплообмена космического аппарата. М.: Машиностроение, 1979. 208 с.
- [23] European Cooperation for Space Standardization (ECSS Organization).  
<http://www.spennis.oma.be/spennis/ecss/ecss.html>.
- [24] Внешний теплообмен космических объектов / Г.П. Кобранов, А.П. Цветков и др. М.: Машиностроение, 1977. 104 с.
- [25] Ермаков С.М., Михайлов Г.А. Статистическое моделирование. М.: Наука, 1982. 296 с.
- [26] Михайлов Г.А. Оптимизация весовых методов Монте-Карло. М.: Наука, 1987. 240 с.
- [27] Toshihiro Kubo-oka, Arata Sengoku. Solar radiation pressure model for the relay satellite of SELENE // Earth Planets Space. 1999. Vol. 51, No 9. P. 979–986.
- [28] Аксенов Е.П. Теория движения искусственных спутников Земли. М.: Наука, 1977. 360 с.
- [29] Каргин Б.А. Статистическое моделирование поля солнечной радиации в атмосфере. Новосибирск: ВЦ СО АН СССР, 1984. 208 с.

- [30] Хисамутдинов А.И., Стариков В.Н., Морозов А.А. Алгоритмы Монте-Карло в ядерной геофизике. М.: Наука, 1985. 158 с.
- [31] Сабельфельд К.К. Статистическое моделирование в задачах математической физики. Новосибирск: НГУ, 1992. 232 с.
- [32] Ziebart M., Cross P., Adhya S. Modelling photon pressure — the key to high precision GPS satellite orbits // GPS World. 2002. Vol. 13, No 1. P. 43–50.
- [33] Ziebart M., Adhya S., Sibthorpe A., and Cross P.A. GPS Block IIR Non-conservative Force Modelling: Computation and Implications // Proceedings of ION GPS/GNSS 2003. Portland, Oregon, USA. 2003. P. 2671–2678.
- [34] Michael Modest. Radiative Heat Transfer. NY: Academic Press, 2003. 860 p.
- [35] Каменщиков Л.П. Оценка величины ускорения космического аппарата, вызванного световым давлением (с распараллеливанием расчетов) // Труды Второй Межд. конф. «Параллельные вычисления и задачи управления» 2–4 окт. 2004 г., Москва, ИПУ РАН. ISBN 5-201-14974-X. С. 110–125.
- [36] Марчук Г.И., Каган Б.А. Динамика океанских приливов. — Л.: Гидрометиздат, 1983.
- [37] Gill A.E. Atmosphere-Ocean Dynamics. — Academic Press, 1982.
- [38] Агошков В.И. Методы оптимального управления и сопряженных уравнений в задачах математической физики. — М.: ИВМ РАН, 2003. — 256 с.
- [39] Agoshkov V.I. Inverse problems of the mathematical theory of tides: boundary-function problem // Russ. J. Numer. Anal. Math. Modelling. — 2005. — vol. 20, № 1. — P. 1–18.
- [40] Agoshkov V.I., Kamenshchikov L.P., Karepova E.D., Shaidurov V. V. Numerical solution of some direct and inverse mathematical problems for tidal flows // Notes on Numerical Fluid Mechanics and Multidisciplinary Design. - Berlin: Springer-Verlag. - 2008. Vol. 101, P. 41–54.
- [41] Каменщиков Л.П., Карепова Е.Д., Шайдуров В.В. Численное моделирование крупномасштабных поверхностных волн в акваториях с усвоением данных наблюдений на базе использования многопроцессорных ЭВМ // Сб. трудов VIII Всероссийской конференции «Современные методы математического моделирования природных и антропогенных катастроф», ИУУ СО РАН, Кемерово, 2005, С. 36–45.
- [42] Kamenshchikov L.P., Karepova E.D., Shaidurov V.V. Simulation of surface waves in basins by the finite element method // Russian J. On Numerical Analysis and Mathematical Modelling — 2006. — Vol. 21. — № 4. — P. 305–320.

- [43] Каменщиков Л.П., Кареева Е.Д., Шайдуров В.В. Моделирование распространения крупномасштабных волн в морях и океанах с параллельной реализацией // Труды Четвертой Сибирской школы-семинара по параллельным и высокопроизводительным вычислениям / Под ред. проф. А.В.Старченко. Томск: ТГУ, ООО «Дельтаплан», 2008. С. 108–115.
- [44] National Geophysical Data Center. <http://www.ngdc.noaa.gov/ngdc.html>
- [45] McNamara G.R., Zanetti G. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata // Phys. Rev. Lett. 1988. 61, P. 2332–2335.
- [46] Wolf-Gladrow D.A. Lattice-Gas Cellular Automata and Lattice Boltzmann Models. Lecture Notes in Mathematics, No. 1725. Springer, 2000.
- [47] Succi S. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, New York, 2001.
- [48] Sukop M.C., Thorne D.T., Jr. Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. Springer, Berlin, 2007.
- [49] Qian Y. H., D’Humières D., Lallemand P. Lattice BGK Models for Navier-Stokes Equation // Europhys. Lett. 1992. 17, P. 479–484.
- [50] He Xiaoyi, Luo Li-Shi. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation // Phys. Rev. E. 1997. 56, P. 6333–6336.
- [51] Chen S., Doolen G.D. Lattice Boltzmann methods for fluid flows // Annu. Rev. Fluid Mech. 1998. 30, P. 329–364.
- [52] Zou Q, He X. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model // Phys Fluids. 1997. 9, P. 1591–1598.
- [53] Brownlee R. A., Gorban A. N., Levesley J. Stability and stabilization of the lattice Boltzmann method // Phys. Rev. E. 2007. 75, P. 1–17.
- [54] Encyclopedia of Microfluidics and Nanofluidics. Li Dongqing (Ed.). Springer, 2008. 2226 p. in 3 volumes.
- [55] Chen Y S, Shan X W, Chen H D. New direction of computational fluid dynamics and its applications in industry // Sci China Ser E-Tech Sci. 2007. 50, P. 521–533.
- [56] Shouxin Hu, Guangwu Yah, Weiping Shi: A lattice Boltzmann model for compressible perfect gas // Acta Mechanica Sinica. 1997. 13, P. 218–226.
- [57] He Bing, Feng Wei-Bing, Zhang Wu, Cheng Yu-Ming: Parallel Simulation of Compressible Fluid Dynamics Using Lattice Boltzmann Method. In: The First International Symposium on Optimization and Systems Biology (OSB’07) Beijing, China. 2007. P. 451–458.

- [58] Li Baoming, Kwok Daniel Y. A Lattice Boltzmann model with high Reynolds number in the presence of external forces to describe microfluidics // *Heat and Mass Transfer*. 2004. 40, P. 843–851.
- [59] Zhou Y, Zhang R, Staroselsky I, Chen H. Numerical simulation of laminar and turbulent buoyancy-driven flows using a lattice Boltzmann based algorithm // *Int J Heat Mass Trans.* 2004. 47, P. 4869–4879.
- [60] Thürey N., Rüde U. Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids // *Computing and Visualization in Science*. 2008. DOI 10.1007/s00791-008-0090-4, P. 179–196.
- [61] Zhou J.G. A lattice Boltzmann model for the shallow water equations with turbulence modeling // *Int. J. of Modern Physics C*. 2002. 13, P. 1135–1150.
- [62] Zhang X. Bengough A.G. Crawford J.W. Young I.M. A lattice BGK model for advection and anisotropic dispersion equation // *Advances in water Resources*. 2002. 25, P. 1–8.
- [63] Chen Shiyi, Chen Hudong, Martinez Daniel, Matthaeus William: Lattice Boltzmann model for simulation of magnetohydrodynamics // *Phys. Rev. Lett.* 1991. 67, P. 3776–3779.
- [64] Derksen J.J. The Lattice-Boltzmann Method for Multiphase Fluid Flow Simulations and Euler-Lagrange Large-Eddy Simulations // In: Daniele L. Marchisio, Rodney O. Fox. (eds.) *Multiphase Reacting Flows: Modelling and Simulation*. Springer Vienna. 2007. P. 181–228.
- [65] Swift Michael R., Osborn W. R., Yeomans J. M. Lattice Boltzmann Simulation of Nonideal Fluids // *Physical Review Lett.* 1995. 75, P. 830–833.
- [66] Xu You-sheng, Liu Ci-qun, Yu Hui-dan. New studying of lattice Boltzmann method for two-phase driven in porous media // *Appl. Math. and Mech.* 2002. 23, P. 387–393.
- [67] C. Shu, X. D. Niu, and Y. T. Chew. A Lattice Boltzmann Kinetic Model for Microflow and Heat Transfer // *J. Stat. Phys.* 2005. 121, P. 239–255.
- [68] Куперштох А. Л. Учет действия объемных сил в решеточных уравнениях Больцмана // *Вестник НГУ: Серия «Математика, механика и информатика»*, 2004. Т. 4, № 2, С. 75–96.
- [69] Kupershtokh A.L., Medvedev D.A. Lattice Boltzmann equation method in electrohydrodynamic problems // *Journal of Electrostatics*. 2006. 64, P. 581–585.
- [70] Kupershtokh A.L. A lattice Boltzmann equation method for real fluids with the equation of state known in tabular form only in regions of liquid and vapor phases // *Computers & Mathematics with Applications*, 2011, vol. 61, No. 12, P. 3537–3548.
- [71] Kamenshchikov L.P. Parallel Implementation of Lattice Boltzmann Flow Simulation in Fortran-DVM Language // *Proceedings of 10th International Conference "Parallel Computing Technologies" (PaCT-2009)*, LNCS 5698, Springer-Verlag Berlin Heidelberg. Novosibirsk, Russia, August/September 2009. P. 216–225.

- [72] Bhatnagar P., Gross E.P., Krook M.K. A model for collision processes in gases: I. small amplitude processes in charged and neutral one-component system // *Phys. Rev.* 1954. 94, P. 511–525.
- [73] Žunič Zoran, Hriberšek Matjaž, Škerget Leopold, Ravnik Jure. 3D Lid Driven Cavity Flow By Mixed Boundary and Finite Element Method // In: P. Wesseling, E. Oñate and J. Périaux (eds.) *ECCOMAS CFD 2006*, TU Delft, The Netherlands, 2006. P. 1–12.
- [74] Новиков Е. А., Каменщиков Л. П. Параллельная реализация метода Рунге-Кутты-Фельберга переменного порядка и шага // *Вычислительные технологии.* 2003. Т. 8. — *Региональный вестник Востока.* 2003. № 3. (Совм. выпуск. Ч. 2.), С. 279–287.
- [75] Новиков Е. А., Каменщиков Л. П. Реализация полуявного (4,2)-метода решения жестких систем на параллельной ЭВМ // *Вычислительные технологии.* 2004. Т. 9. — *Вестник КазНУ им. аль-Фараби. Сер. Математика, механика, информатика.* № 3 (42). (Совм. выпуск. Ч. 1), С. 235–241.
- [76] Каменщиков Л.П., Новиков Е.А. Параллельная реализация (2,1)-метода решения жестких систем // *Тр. IV школы-семинара «Распределенные и кластерные вычисления».* 2005. Красноярск: ИВМ СО РАН, С. 125–135.
- [77] Каменщиков Л.П., Быков В.И. Моделирование нестационарных трехмерных систем типа «реакция + диффузия» (с распараллеливанием расчетов) // *Материалы Шестого Всеросс. сем-ра «Моделирование неравновесных систем — 2003»*, Красноярск, 24–26 окт. 2003. С. 213.