

DOI: 10.15514/ISPRAS-2024-36(2)-9



Набор методических тестовых программ для численного исследования параметров высокопроизводительных вычислительных систем

А.О. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

А.В. Ершов, ORCID: 0009-0007-4439-1516 <a.v.ershov@vniitf.ru>

А.В. Карпеев, ORCID: 0009-0007-2203-9423 <a.v.karpeev@vniitf.ru>

Р.Ф. Мухамадиев, ORCID: 0009-0006-2059-2113 <r.f.mukhamadiev@vniitf.ru>

Е.М. Романова, ORCID: 0009-0006-1478-9059 <e.m.romanova@vniitf.ru>

Д.А. Ушаков, ORCID: 0009-0001-6458-2631 <d.a.ushakov@vniitf.ru>

В.О. Анисов, ORCID: 0009-0004-5620-099X <v.o.anisov@vniitf.ru>

Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина, 456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13.

Аннотация. Неотъемлемой частью процесса создания высокопроизводительных вычислительных систем, предназначенных для решения задач численного моделирования различных физических процессов, является проверка их соответствия на заявленные при их проектировании характеристики. При этом существует проблема оценки производительности вычислительных систем на синтетических тестах, значительно уступающих по математической сложности реальным прикладным задачам. В статье рассматривается разработанный авторами набор тестовых программ, позволяющий более точно оценивать реальную производительность вычислительных систем.

Ключевые слова: вычислительная система; высокопроизводительные вычисления; математическое моделирование; тестирование вычислительных систем.

Для цитирования: Игнатьев А.О., Мокшин С.Ю., Ершов А.В., Карпеев А.В., Мухамадиев Р.Ф., Романова Е.М., Ушаков Д.А., Анисов В.О. Набор методических тестовых программ для численного моделирования параметров высокопроизводительных вычислительных систем. Труды ИСП РАН, том 36, вып. 2, 2024 г., стр. 109–126. DOI: 10.15514/ISPRAS-2024-36(2)–9.

A Set of Methodological Test Programs for the Numerical Research of High-Performance Computing Systems Parameters

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

A.V. Ershov, ORCID: 0009-0007-4439-1516 <a.v.ershov@vniitf.ru>

A.V. Karpeev, ORCID: 0009-0007-2203-9423 <a.v.karpeev@vniitf.ru>

R.F. Mukhamadiev, ORCID: 0009-0006-2059-2113 <r.f.mukhamadiev@vniitf.ru>

E.M. Romanova, ORCID: 0009-0006-1478-9059 <e.m.romanova@vniitf.ru>

D.A. Ushakov, ORCID: 0009-0001-6458-2631 <d.a.ushakov@vniitf.ru>

V.O. Anisov, ORCID: 0009-0004-5620-099X <v.o.anisov@vniitf.ru>

Russian Federal Nuclear Center –

*Zababakhin All-Russian Research Institute of Technical Physics,
456770, Russia, Snezhinsk, Chelyabinsk region, Vasilieva street, 13.*

Abstract. An integral part of the process of creating high-performance computing systems designed to solve problems of numerical modeling of various physical processes is to check their compliance with the characteristics stated during their design. At the same time, there is a problem of evaluating the performance of computing systems on synthetic tests, which are significantly primitive in mathematical complexity to real applied problems. The article considers a set of test programs developed by the authors, which allows to more accurately assess the real performance of computing systems.

Keywords: computing system; high-performance computing simulation; mathematical modeling; high-performance computing system testing.

For citation: Ignatyev A.O., Mokshin S.Yu., Ershov A.V., Karpeev A.V., Mukhamadiev R.F., Romanova E.M., Ushakov D.A., Anisov V.O. A set of methodological test programs for the numerical research of high-performance computing systems parameters. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 2, 2024. pp. 109-126 (in Russian). DOI: 10.15514/ISPRAS-2024-36(2)-9.

1. Введение

При решении задач численного моделирования физических процессов требуется огромный объем вычислений, обеспечиваемых высокопроизводительными вычислительными системами (ВВС). Современные ВВС, предназначенные для решения задач численного моделирования, представляют собой множество вычислительных узлов, систему хранения с параллельным доступом и сервисные подсистемы, объединяемых высокопроизводительной коммуникационной средой [1]. При проектировании таких ВВС закладываются ожидаемые характеристики по производительности вычислений, основанные на физических показателях используемых элементов и используемой архитектуре ВВС. Однако, как показывает практика, переход от одной ВВС к другой с большей заявленной производительностью не гарантирует кратного сокращения сроков решаемых задач. Даже традиционно применяемый в рейтинге Супер-ЭВМ TOP-500 [2] тест High Performance Linpack (HPL) [3] уже не подтверждает заявленную производительность ВВС, так как его математический аппарат скорее типичен для вычислительных систем начала нашего века. Поэтому во всем мире, помимо предложенного Д. Донгарра нового теста HPCG [4], использующего для решения систем линейных алгебраических уравнений метод сопряженных градиентов, разрабатываются специализированные пакеты тестовых программ, реализующие различные численные алгоритмы и позволяющие оценить производительность ВВС применительно к данным алгоритмам.

Одним из первых подобных пакетов можно считать NAS Parallel Benchmark (NPB) [5], содержащий на момент его создания 8 параллельных программ, представляющих различные численные алгоритмы и использующих MPI [6] для распараллеливания. Этот пакет

продолжает развиваться, расширяется набор программ, добавляются новые алгоритмы и новые способы распараллеливания (OpenMP [7], CUDA [8]). Следующим шагом можно считать набор тестовых программ министерства энергетики США, используемых при создании Супер-ЭВМ ASC Purple и следующих за ней Супер-ЭВМ [9], содержащий упрощенные версии производственных программ. Существуют и более современные версии подобных тестовых программ, например, HPC (High Performance Computing) benchmarks [10]. В России, на предприятиях государственной корпорации «Росатом», также ведутся работы по созданию пакетов тестовых программ, максимально приближенных к используемым на этих предприятиях программам численного моделирования различных физических процессов. В результате разработан комплекс тестов, применяемых при создании и проведении приемочных испытаниях различных ВВС, включающих в себя наряду с рядом тестов из NPB (LU, SP, BT) и ASCI Purple (sPPM, IOR) методические тестовые программы, разработанные в РФЯЦ-ВНИИЭФ и РФЯЦ-ВНИИТФ [11].

В данной статье приводится описание набора методических тестовых программ, разработанных в РФЯЦ-ВНИИТФ и входящих в общий комплекс тестов государственной корпорации «Росатом».

2. Общая характеристика методических тестовых программ

Набор методических тестовых программ представляет широкий класс вычислительных алгоритмов, используемых при численном моделировании физических процессов и, в сочетании с другими тестовыми программами, позволяющих прогнозировать поведение реальных программ численного моделирования.

Требования к методическим тестовым программам узаконены в российском стандарте [12], принятом в 2019 году.

Методические тестовые программы удовлетворяют следующим условиям:

- используются языки программирования Фортран и Си (Си++);
- используется многоуровневое распараллеливание: OpenMP внутри вычислительного узла и MPI между узлами;
- программы готовят необходимые начальные данные для проведения расчета и выполняют расчет тестовой задачи;
- масштабирование задачи в программах производится как методом деления (единая вычислительная нагрузка на все потоки выполнения в пределах одного вычислительного узла), так и методом умножения (вычислительная нагрузка увеличивается кратно увеличению количества процессов при использовании множества вычислительных узлов);
- программы верифицируют полученные результаты;
- программы определяют полученную производительность в операциях или других единицах.

3. Описание методических тестовых программ

3.1 Тест SLON

Параллельная программа SLON решает двумерное стационарное уравнение переноса для сферических систем в односкоростном кинетическом приближении, записанное в цилиндрической системе координат и вычисляет критический параметр λ . Программа

предназначена для проведения исследований производительности и эффективности распараллеливания на многопроцессорных ЭВМ с распределенной и общей памятью.

3.1.1 Математическая модель

Математическая модель программы представляет собой решение однородной краевой задачи:

$$\operatorname{div}(\bar{\Omega} N_g) + (\alpha_g + \frac{\lambda}{v_g}) N_g = \frac{1}{2\pi} \sum_{g'=1}^G \int_{\Omega} \beta_{g'g} N_{g'} d\Omega$$

$$N_g = N_g(r, z, \mu, \psi) \Big|_r = 0, \text{ при } \bar{\Omega} \bar{n} < 0.$$

где:

r, z – цилиндрические координаты точки, в которой решается задача;

Γ – граница системы, в которой решается задача;

$N_g(r, z, \mu, \psi)$ – число нейтронов, летящих в направлении вектора единичной длины $\bar{\Omega}(\mu, \psi)$;

v_g – скорость нейтронов группы g относительно вещества;

g – индекс группы;

G – количество рассматриваемых групп, число групп задается как параметр распараллеливания при счете задачи, скорости в группах одинаковые;

$\alpha_g = \alpha_g(r, z)$ – коэффициент поглощения нейтронов;

$\beta_{g'g} = \beta_{g'g}(r, z)$ – коэффициент рассеяния нейтронов из группы g' в группу g ;

\bar{n} – нормаль к стороне ячейки;

ψ – угол между $\bar{\Omega}'$ и осью \bar{r} ;

$\bar{\Omega}' = \xi \bar{n}_r + \mu \bar{n}_z$ – проекция $\bar{\Omega}$ на плоскость (r, z) ;

$\mu = \cos \theta, \xi = \cos \psi \sqrt{1 - \mu^2}, \eta = \sin \psi \sqrt{1 - \mu^2}$;

θ – угол между $\bar{\Omega}$ и осью \bar{z} .

Для указанной задачи существует положительная собственная функция N^* , соответствующая характеристическому числу λ^* , которое по модулю больше других характеристических чисел λ_k . Решение однородной нестационарной задачи (без источников и с нулевыми граничными условиями) при $t \rightarrow \infty$ будет пропорционально $N_g^* e^{\lambda^* t}$, т. е. λ^* представляет собой временную постоянную размножения нейтронов [13]. Уравнение переноса решается Sn-методом по DDST-схеме [14] на четырехугольных сетках. На внутренних итерациях по интегралу столкновений используется метод Келлога [15]. На внешних итерациях используется метод Ньютона-Рафсона (модификация метода Ньютона [16]).

Программа написана на языке FORTRAN и содержит 4 уровня распараллеливания по параметрам фазового пространства задачи, реализованных на распределенной памяти средствами MPI и на общей памяти средствами OpenMP. Рассмотрим основные особенности каждого уровня распараллеливания и используемые при этом декомпозиции.

3.1.2 Декомпозиция по энергетическим группам

Многогрупповой режим моделируется с помощью размножения задачи по процессам MPI, без изменения скорости полета нейтронов. Такой подход сохраняет все арифметические операции, необходимые для реализации разностной схемы и позволяет оценить накладные расходы на межпроцессорные обмены данными. При этом естественным образом используется метод увеличения задачи, полностью сбалансированный по процессам MPI. Система нейтронно-ядерных констант в программе SLON не используется, решения на каждом процессе дублируются.

3.1.3 Геометрическая декомпозиция по областям системы

Уровень реализован средствами MPI, используется метод увеличения, реализованный с помощью увеличения числа точек в системе. В этом случае подобласть системы в одном процессе MPI будем называть блоком. Для совпадения решения такой задачи в параллельном режиме с решением в последовательном режиме требуется больше итераций. Метод обеспечивает примерно одинаковое число арифметических операций в каждом счетном блоке, что позволяет вычислять эффективность распараллеливания достаточно точно.

3.1.4 Распараллеливание по направлениям в пространстве полета нейтронов

Уровень реализован методом дробления, средствами OpenMP на общей памяти вычислительного узла ВВС. Позволяет ввести практически любую квадратуру для увеличения арифметической нагрузки на узел [17].

3.1.5 Мелкозернистое распараллеливание

Уровень реализован за счет дополнительной геометрической декомпозиции методом дробления блока средствами OpenMP для каждого MPI процесса. Блок системы на каждом MPI процессе разбивается на дополнительное число счетных областей с сохранением числа точек в блоке. Обмен граничными условиями между соседними областями производится во время счета итерации. Данный уровень распараллеливания предполагает решение на одном OpenMP потоке уравнения переноса для одной счетной области и части направлений полета нейтронов, что позволяет менять конфигурацию счетного сегмента фазового пространства и исследовать возможность наилучшего попадания задачи в быструю память узла.

3.1.6 Получение отчетных показателей

В качестве отчетных показателей в тестовой программе SLON используются следующие измеряемые параметры:

- эффективность распараллеливания;
- производительность задачи в вещественных и целых операциях в секунду.

При этом время исполнения программы делится на 2 этапа:

- 1-й этап – подготовка начальных данных и упорядочивание системы. Время исполнения этого этапа не учитывается при измерении эффективности, поскольку эффективность распараллеливания на этом этапе составляет 100% (обменов между процессами нет) и, при небольшом числе итераций на 2-м этапе счета, может неоправданно завысить эффективность распараллеливания задачи.
- 2-й этап – счет итераций. Времена выполнения 2-го этапа на одном вычислительном узле и на множестве узлов используются для вычисления эффективности распараллеливания. Обычно задается число итераций = 100, что обеспечивает сходимость четырех значащих цифр параметра λ . В случае необходимости число итераций можно увеличить до сходимости решения.

3.2 Тест GROM

Параллельная программа GROM численно решает двумерное нестационарное квазилинейное уравнение теплопроводности для плоского случая или случая с осевой симметрией для задачи [18-19], которая имеет аналитическое решение в виде бегущей волны. Программа GROM позволяет проводить расчеты на вычислительных системах, как в однопроцессорном, так и в параллельном режиме. Программа предназначена для проведения исследований

производительности и эффективности распараллеливания на ВВС с распределенной и общей памятью.

3.2.1 Математическая модель

В двумерной области $r \in [0; r_0], z \in [0; z_0], t \in [t^0; t^n]$ рассматривается краевая задача для двумерного уравнения теплопроводности без учета независимого источника:

$$\frac{\partial \varepsilon}{\partial t} = \frac{1}{\rho R} \left[\frac{\partial}{\partial r} \left(R \cdot \chi \frac{\partial T}{\partial r} \right) + \frac{\partial}{\partial z} \left(R \cdot \chi \frac{\partial T}{\partial z} \right) \right]$$

где:

r, z – декартовы координаты;

$R=1$ – плоский случай;

$R=r$ – осевая симметрия (z – ось симметрии);

$T(t, r, z)$ – температура;

ρ – плотность;

$\varepsilon(\rho, T)$ – внутренняя энергия;

$\chi(\rho, T)$ – коэффициент теплопроводности.

В рассматриваемой задаче $\rho = 1, \varepsilon = T, \chi = \chi_0 T^\alpha$, где $\chi_0 > 0, \alpha > 0$,

и рассматривается с начальными условиями:

$$T(t^0, r, z) = T^0(r, z)$$

и граничными условиями (теплоизолирующая стенка):

$$T(t, r, z = 0) = \left(\frac{\alpha c^2}{\chi_0} t \right)^{1/\alpha}$$

где c – скорость распространения тепловой волны.

Задача имеет автомодельное решение, представляющее собой бегущую волну [18]:

$$T(t, r, z) = \begin{cases} \left(\frac{\alpha c}{\chi_0} (ct - z) \right)^{1/\alpha}, & z < ct, \\ 0, & z \geq ct \end{cases}$$

Уравнение $\frac{\partial \varepsilon}{\partial t}$ аппроксимируется неявной, безусловно устойчивой конечно-разностной схемой «РОМБ» [18,19,20]. Для решения получаемой системы разностных уравнений применяется итерационный метод со стабилизирующей поправкой (ИМСП) [19], который сводит многомерную разностную задачу к совокупности квазиодномерных подзадач, решаемых прогонами по каналам. Итерации метода ИМСП совмещаются с итерациями по нелинейности χ . По нелинейности χ применяется метод простой итерации (μ -итерации). Этот метод позволяет вести счет с крупным шагом по времени, при этом схема обладает свойством полной аппроксимации. Счет на одной итерации μ разбивается на два этапа. На первом этапе счет ведется вдоль каналов $j + 1/2 = const$, на втором – вдоль каналов $i + 1/2 = const$. Причем вторая система может быть решена только после решения первой.

3.2.2 Программная реализация

В программе GROM счетная область разделяется на прямоугольные в топологическом плане подобласти, каждая подобласть обрабатывается отдельным MPI процессом. Поскольку шаблон методики «РОМБ» строится в рамках одной ячейки, используются непересекающиеся подобласти. Обмены данными происходят лишь между «соседними» MPI процессами. В управляющем файле программы GROM пользователь задает конфигурацию

разбиения $\text{Numr} \times \text{Numz}$ счетной области, где Numr и Numz – количество MPI процессов по оси r и z соответственно. При этом данные равномерно распределяются между MPI процессами. Схематично декомпозицию счетной области можно представить так, как показано на рис. 1.

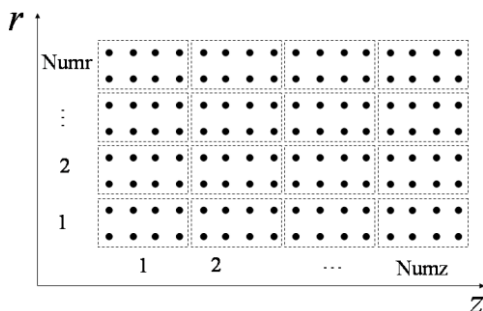


Рис. 1. Декомпозиция счетной области
Fig. 1. Calculation area decomposition

Программа GROM распараллелена по двум уровням:

- Первый уровень – геометрическая декомпозиция по пространственным переменным. Для этого система разбивается на подобласти так, как указано выше. Распараллеливание по подобластям производится на распределенной памяти средствами MPI. На каждом этапе итерации по нелинейности коэффициента теплопроводности происходит обмен массивами граничных условий между соседними подобластями. Число подобластей является задаваемым параметром. Количество подобластей должно быть кратно числу задействованных в расчете узлов BBC.
- Второй уровень распараллеливания – совокупность каналов $j + 1/2 = \text{const}$ или $i + 1/2 = \text{const}$ в зависимости от этапа. Данный уровень распараллеливания реализован на общей памяти при помощи библиотеки OpenMP. Число потоков является входным параметром и не должно превосходить числа ядер одного узла, тестируемого BBC.

Векторизация вычислений – неявная, средствами используемого компилятора.

3.2.3 Получение отчетных показателей

Программа GROM предназначена для тестирования различных BBC. При тестировании задаются параметры задачи, количество узлов, количество MPI процессов и количество потоков на вычислительном узле в управляющем файле конфигурации программы. При этом измеряемыми параметрами являются:

- эффективность распараллеливания;
- производительность ЭВМ в операциях в секунду над вещественными числами с двойной точностью.

3.3 Тест МНТ

Тестовая программа МНТ (Monte-Carlo High Performance Computing Test) предназначена для оценки производительности BBC. Программа МНТ использует двухуровневую схему распараллеливания: между узлами с помощью MPI, внутри узла – на общей памяти с помощью потоков стандартной библиотеки C++11. Геометрия системы представляет собой трехмерную сетку из произвольных шестигранников. Основным результатом работы теста

является условная производительность, представляющая собой число поколений в секунду (скорость счета).

3.3.1 Математическая модель

Программа МНТ имитирует расчет на произвольных гексагональных сетках с использованием метода максимального сечения. Тестовая задача: расчет K_{eff} -методом поколений с фиксированным количеством точек деления в поколении. Модель среды: приближение свободных покоящихся атомов. Константы: спектральные, на основе ENDF/B-5. Свободный пробег моделируется на прямоугольной сетке виртуальных областей 20x20x20.

Зависимость максимального сечения от энергии задается в виде таблиц с логарифмическим шагом по энергии. Число узлов 150. В точке соударения, производится поиск текущей ячейки, разыгрываются тип и параметры взаимодействия. В процессе поиска ячейки сетки используется ускоряющая структура – kD-дерево, равномерно разбивающая пространство (рис. 2, 3).

В ее узлах расположены по 8 ячеек, воспроизводящих нагрузку на операцию проверки принадлежности.

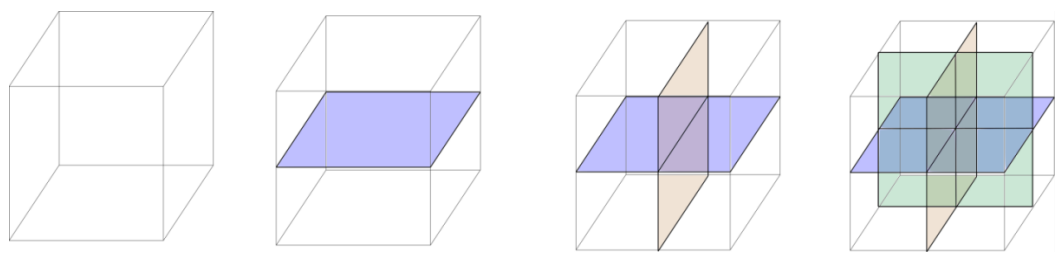


Рис. 2. Разбиение пространства
Fig. 2. Space partitioning

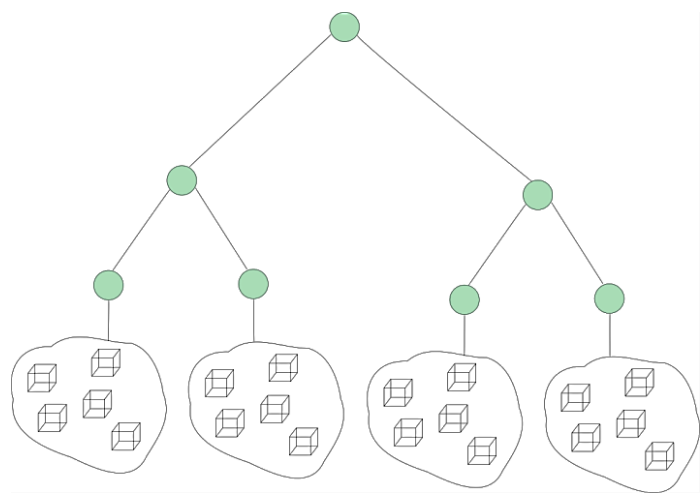


Рис. 3. kD-дерево
Fig. 3. kD-tree

Для неравномерных сеток (например, Лагранжево-Эйлеровых методик) к узлам дерева может быть привязано большое количество ячеек из-за того, что их невозможно разделить с помощью простых ограничивающих тел (рис. 4). В этом случае может существенно вырасти

время, приходящееся на работу геометрического модуля. В текущей версии программы количество ячеек, привязанных к узлу, фиксировано и равно 8. Имитация нагрузки, возникающей из-за неравномерности сеток, реализована путем искусственного увеличения количества выполняемых операций.

Система представляет собой критическую сборку типа Godiva [21], окруженную слоем воды (рис. 5). Задание системы производится с помощью указания материалов в ячейках сетки.

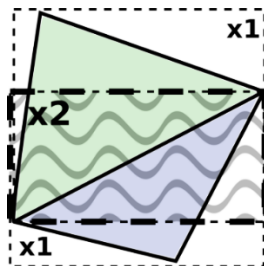


Рис. 4. Проблема разделимости ячеек
Fig. 4. Cell separation problem

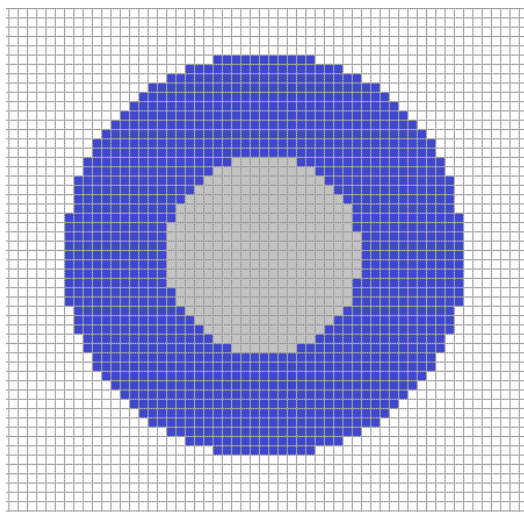


Рис. 5. Задание системы
Fig. 5. System definition

3.3.2 Программная реализация

Программа МНТ написана на языке программирования C++, при этом используется следующая схема распараллеливания:

- обмены между процессами средствами библиотеки MPI;
- разбиение алгоритмов на потоки средствами стандартной библиотеки языка.

3.3.3 Получение отчетных показателей

Измеряемыми параметрами являются:

- Generations/sec – количество рассчитываемых поколений за одну секунду;
- Generations processed – суммарное количество промоделированных поколений;

- Mean calculation time – среднее время счета активных поколений на узле;
- Skipping finished – время, затрачиваемое на отброс первых 30 поколений на каждом вычислительном потоке;
- RND time – время инициализации расчета;
- Calc time – время работы счетной части (включает время Skipping finished);
- Memory usage – объём памяти, занимаемый геометрией;
- Cells count – количество ячеек.

Основным оценочным рассчитываемым параметром является количество рассчитываемых поколений в секунду Generations/sec – условная производительность.

3.4 Тест Machete

Тестовая параллельная программа моделирует движение сплошной среды, решая для этого систему уравнений идеальной газовой динамики в Эйлеровых координатах на Декартовой кубической сетке. Программа позволяет проводить расчеты на BBS, как в последовательном режиме, так и с использованием произвольных комбинаций следующих подходов распараллеливания: векторизация, OpenMP, MPI. Программа предназначена для проведения исследований производительности и эффективности распараллеливания заложенных в нее численных алгоритмов на современных вычислительных системах.

3.4.1 Математическая модель

Для моделирования движения среды используется система уравнений идеальной газовой динамики в эйлеровых переменных:

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho U) &= 0, \\ \frac{\partial \rho U}{\partial t} + \operatorname{div}(\rho U \cdot U) + \operatorname{grad}(P) &= 0, \\ \frac{\partial \rho E}{\partial t} + \operatorname{div}(\rho EU) + \operatorname{grad}(PU) &= 0.\end{aligned}$$

где:

ρ – плотность;

$U=(u, v, w)$ – вектор скорость;

ε – удельная внутренняя энергия;

$E = \varepsilon + 0.5U^2$ – удельная полная энергия;

P – давление вещества, которое связано с плотностью и энергией соотношением $P = P(\rho, \varepsilon)$.

В качестве тестовой выбрана задача «о точечном взрыве». В данной задаче моделируется сферически-симметричное течение, возникающее в результате точечного мгновенного энерговыделения в изначально холодном однородном газе без противодействия. Аналитическое решение для данной задачи получено Л.И. Седовым [22].

Для аппроксимации дифференциальных уравнений область интегрирования задачи разбивается равномерной кубической сеткой на $N_x \times N_y \times N_z$ ячеек. Количество ячеек по направлению (линейный размер сетки), временной шаг и количество шагов в расчете являются параметрами задачи, которые варьируются в зависимости от исследуемой характеристики и определяются через входные параметры программы. В начальный момент времени распределение величин в ячейках задано. Разностная схема для перехода на следующий временной слой построена на основе FLIC-метода [22, 23].

3.4.2 Программная реализация

Программа MACHETE написана на языке программирования C++. В программе MACHETE реализована возможность использовать три уровня распараллеливания:

- на распределенной памяти средствами MPI;
- на общей памяти средствами OpenMP;
- векторизация.

При организации распараллеливания на распределенной памяти средствами MPI в программе MACHETE использован принцип пространственной декомпозиции счетной области на подобласти одинакового размера. Количество разбиений по каждому из направлений $N_p^{(x)}, N_p^{(y)}, N_p^{(z)}$ является параметром задачи и определяется в конфигурационном файле программы через размеры счетной области в одном MPI-процессе (параметр «box_size»). При этом общее число MPI-процессов в задаче $N_p = N_p^{(x)} \times N_p^{(y)} \times N_p^{(z)}$. Данный подход позволяет получать разбиения области и по одному, и по двум, и по трем направлениям. Такая гибкость дает возможность находить подходящее разбиение для различных комбинаций гибридного распараллеливания с участием MPI. В одном случае может быть оптимальным минимизировать объем пересылаемых сообщений, в другом – поддержать достаточное количество листов для OpenMP, в третьем – максимально использовать последовательность укладки данных для использования предвыборки.

Для согласований решений между процессами используется технология виртуальных ячеек. С целью минимизации количества MPI-вызовов внутри одного временного шага размер виртуальной области имеет ширину в три ячейки. Таким образом, если на процессе рассчитывается «полезная» область размером $n_x \times n_y \times n_z$ ячеек, то размер посылаемых и принимаемых на каждом шаге данных определяется выражением

$$V_{send} = V_{recv} = Nb_{cell} \cdot [(n_x + 6) + (n_y + 6) + (n_z + 6) - n_x \cdot n_y \cdot n_z]$$

где:

$Nb_{cell} = 56$ байт – объём данных, пересылаемых из одной ячейки.

Возможность распараллеливания на общей памяти в программе MACHETE реализована средствами библиотеки OpenMP. Используется независимый обход внешних циклов 3-го координатного направления. Количество используемых в расчете потоков является входным параметром и определяется через входной параметр задачи. Желательно, чтобы эта величина не превосходила количества ячеек в подобласти по z-направлению и допустимого числа потоков на одном вычислительном узле.

Все основные счетные циклы реализованного в программе MACHETE алгоритма допускают автоматическую векторизацию. Для эффективной реализации такой возможности ячейчные величины упакованы в структуры с последовательным хранением данных относительно внутреннего индекса. В программе MACHETE использованы одномерные N-компонентные массивы, которые в дополнение к автоматической векторизации позволяют минимизировать количество операций на определение адресов переменных для хранения данных рассматриваемой точки разностного шаблона.

3.4.3 Получение отчетных показателей

Тестовая программа MACHETE предназначена для тестирования различных ВВС. При тестировании варьируются параметры задачи, количество MPI-процессов и количество OpenMP потоков на них. Значения этих величин задаются в текстовом файле, имя которого подается в качестве аргумента основной программы.

Измеряемые параметры и методика их получения:

- производительность BBC (Флопс).
- эффективность распараллеливания.

Количество арифметических операций над вещественными числами, выполняемых программой за один временной шаг, является функцией от линейного размера задачи. Отсюда, с учетом полного времени выполнения программы и количества выполненных временных шагов, определяется производительность перемножением этих величин.

3.5 Тест HCTest

Тестовая параллельная программа HCTEST моделирует процесс нелинейной (лучистой) теплопроводности в цилиндрической системе координат на регулярной прямоугольной или нерегулярной треугольной сетке. Программа позволяет проводить расчеты на BBC, как в последовательном режиме, так и с использованием произвольных комбинаций следующих подходов распараллеливания: векторизация (средствами компилятора), OpenMP, MPI, CUDA. При этом используется собственная равномерная раскладка по NUMA узлам в рамках вычислительного узла.

3.5.1 Математическая модель

Для моделирования нелинейной теплопроводности используется система уравнений теплопроводности газа в цилиндрических координатах:

$$\begin{aligned}\rho \frac{d\varepsilon}{dt} + \operatorname{div} \Phi &= \rho Q, \\ \Phi + D \operatorname{grad} T &= 0, \\ \alpha(r, t)T - \beta(r, t)\Phi n &= \mu(r, t), r \in \Sigma.\end{aligned}$$

где:

$\varepsilon(\rho, T)$ – удельная внутренняя энергия;

$\rho(r, T)$ – плотность;

$T(r, t)$ – температура;

$\Phi(T)$ – тепловой поток;

$D(T)$ – коэффициент теплопроводности;

$\alpha(r, t), \beta(r, t), \mu(r, t)$ – тройка величин, задающих граничные условия;

Σ – граница области;

Φn – тепловой поток, умноженный на внешнюю нормаль к границе.

В качестве тестовой задачи выбрана автомоделная задача «о мгновенном точечном источнике». В данной задаче моделируется течение тепла, возникающее от точечного мгновенного теплового источника [24]. Расчетная область от 0 до 3 по оси X и от 0 до 1.5 по оси Y, точечный источник помещен в точку (1.5, 0). Коэффициент теплопроводности рассчитывается по формуле $D(T) = T^2$. В качестве начального профиля используется профиль в момент времени $t = 0.05$, который изображен на рис. 6.

Для аппроксимации дифференциальных уравнений область интегрирования задачи (расчетная область), покрывается регулярной прямоугольной сеткой или нерегулярной треугольной сеткой (определяется входным параметром). Множество всех ячеек обозначим за Ω , а величины ячейки $i_n \in \Omega$ будем записывать в виде T_{i_n} . Заметим, что нерегулярная сетка больших размеров, более двух миллионов ячеек, строится значительное время, поэтому необходимо её строить особым образом. Нерегулярная сетка получается в результате укладки необходимого числа одного и того же масштабированного квадратного блока (рис. 7), на котором заранее построена треугольная сетка специального вида в области $[0,0] \times [1,1]$. Специальный вид заключается в том, что узлы, лежащие на границе, расположены регулярно,

то есть с одинаковым шагом вдоль. Это делается для того, чтобы во время укладки блоков узлы, лежащие на границе, совпадали.

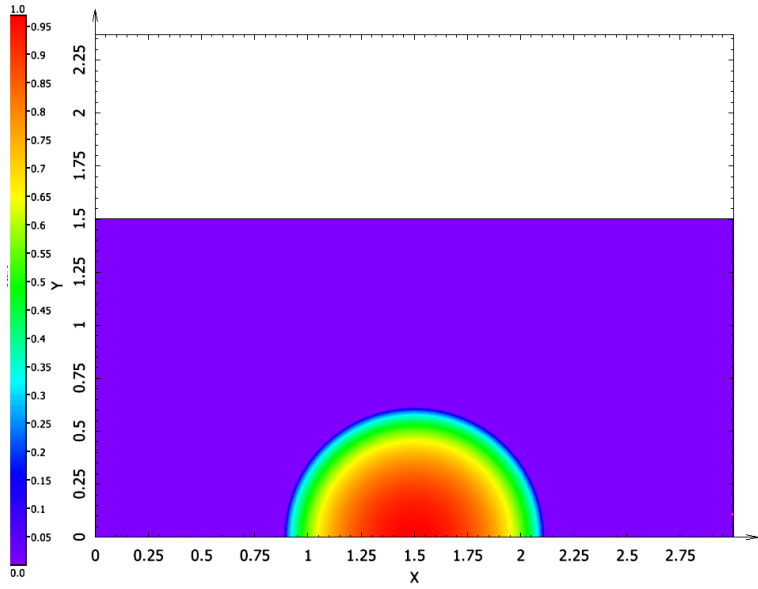


Рис. 6. Начальный профиль температуры
Fig. 6. Initial temperature profile

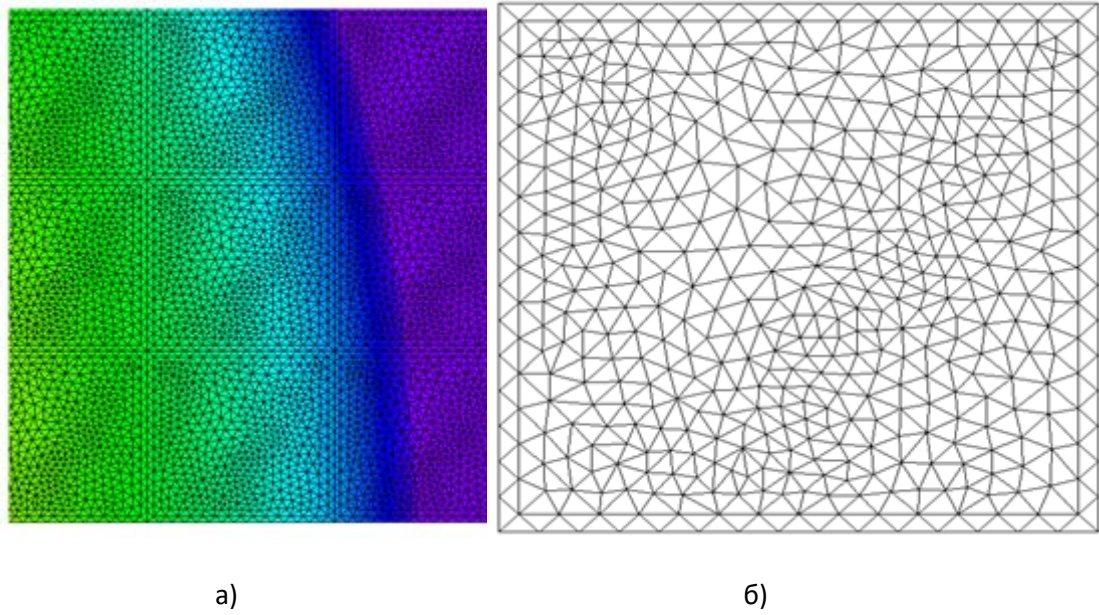


Рис. 7. Пример построения нерегулярной треугольной сетки (а), полученной в результате укладки набора одинаковых блоков (б)
Fig. 7. An example of irregular triangular mesh building (a), laying set of identical blocks (б)

Метод решения можно записать в виде следующего алгоритма.

В начальный момент времени распределение величин в ячейках задано. Для перехода на следующий временной слой используется итерационный процесс по нелинейности внутренней энергии и коэффициентов теплопроводности: положим $T_{i_n}^v = T_{i_n}^n$, найдем $T_{i_n}^{v+1}$, для этого запишем уравнение итоговой схемы для каждой ячейки, получив систему линейных алгебраических уравнений (СЛАУ), имеющую вид $Ax = b$. Матрица A и вектор B рассчитываются на основе выбранной схемы. Во время итерационного процесса неизвестный вектор x – это “промежуточный” профиль температуры между временными слоями, а по окончанию, x – это профиль следующего временного слоя. В программе на выбор используются два условия остановки решателя:

$$\begin{aligned} \|Ax_n - B\|_{L_2} &\leq tol \cdot \|B\|_{L_2} \\ \|Ax_n - B\|_{L_2} &\leq tol \cdot \|Ax_0 - B\|_{L_2} \end{aligned}$$

где tol – точность решателя СЛАУ.

Среднее время одной итерации решателя зависит от количества проделанных итераций. Число выполненных итераций ограничено снизу числом 40. Данное число было подобрано для тестовой задачи при разном размере сетки, временном шаге и других параметрах, а также при вычислениях на различном оборудовании.

В результате решения СЛАУ будут получены температуры в центрах ячеек на новой итерации $T_{i_n}^{v+1}$. Далее проверяется условие остановки итерационного процесса:

$$|T_{i_n}^{v+1} - T_{i_n}^v| - \varepsilon(1 + T_{i_n}^v) \leq 0, i_n \in \Omega, \varepsilon \rightarrow 0$$

Если указанное условие не выполнено, тогда повторяем процедуру нахождения температуры на следующей итерации, положив $T_{i_n}^v = T_{i_n}^{v+1}$. В противном случае, если это условие выполнено, тогда за температуру на следующем временном слое $T_{i_n}^{n+1}$ принимается $T_{i_n}^{v+1}$.

3.5.2 Программная реализация

Для реализации на универсальном процессоре был использован язык программирования C++ с применением технологий распараллеливания OpenMP, MPI и векторизацией средствами компилятора. Для реализации на графических ускорителях используется технология NVidia CUDA.

В программе HCTest используется двухуровневое распараллеливание. Нижний уровень OpenMP и CUDA для независимого обхода внутри циклов, средствами OpenMP и CUDA NVidia соответственно. Верхний уровень MPI – используется геометрическая декомпозиция расчетной сетки.

Возможность распараллеливания на общей памяти в программе HCTEST реализована средствами библиотеки OpenMP и библиотеки CUDA. Используется независимый обход элементов внутри циклов. Далее представлена табл. 1, которая наглядно показывает четыре режима счета и используемые средства распараллеливания на нижнем уровне.

Табл. 1. Нижний уровень распараллеливания алгоритмов для режимов счета
Table 1. The lower parallelization level of algorithms for calculation modes

Алгоритмы Режим счета	Построение сетки	Моделирование процесса теплопроводности, кроме решения СЛАУ	Решение СЛАУ
CPU	OpenMP	OpenMP	OpenMP
CPUGPU	OpenMP	OpenMP	CUDA
GPU	OpenMP	CUDA	CUDA
HYBRID	OpenMP	OpenMP / CUDA	OpenMP / CUDA

Большинство основных счетных циклов реализованного в программе HCTEST алгоритма допускают автоматическую векторизацию. Для эффективной реализации такой возможности ячеечные величины упакованы в структуры с последовательным хранением данных относительно внутреннего индекса.

3.5.3 Получение отчетных показателей

Тестовая программа HCTEST предназначена для тестирования различных BBC на основе, как универсальных узлов, так и гибридных узлов с графическими ускорителями (поддерживающими CUDA). В ходе тестирования могут быть получены следующие параметры:

- верификация полученного решения на основе известного точного решения;
- производительность BBC (Флопс);
- эффективность распараллеливания.

4. Использование методических тестовых программ при проведении испытаний BBC

При тестировании BBC с использованием методических прикладных тестов на первом этапе определяется эффективность распараллеливания внутри одного вычислительного узла методом деления (при этом запускается один процесс MPI и множество потоков выполнения OpenMP).

Ускорение при распараллеливании рассчитывается по формуле:

$$Ksp_n = \frac{T_1}{T_n}$$

где:

T_1 – время выполнения расчета тестовой задачи на одном потоке OpenMP;

T_n – время выполнения расчета тестовой задачи на n потоках OpenMP.

Эффективность распараллеливания рассчитывается по формуле (в процентах):

$$E_n = \frac{Ksp_n}{n} * 100$$

На втором этапе производится серия параллельных расчетов на множестве вычислительных узлов методом увеличения. Ускорение при увеличении числа задействованных узлов в случае, если программа умеет определять производительность, рассчитывается по формуле:

$$Ksp_n = \frac{V_n}{V_1}$$

Если программа не определяет производительность, то ускорение рассчитывается по формуле:

$$Ksp_n = \frac{T_1 \times n}{T_n}$$

где:

V_n – производительность при использовании n узлов;

V_1 – производительность при использовании одного узла.

T_1 – производительность при использовании одного узла;

T_n – время выполнения расчета тестовой задачи на n узлах.

5. Заключение

В статье рассмотрены методические тестовые программы, разработанные в РФЯЦ-ВНИИТФ. Вместе с тестовыми программами из состава тестов NPB, ASCI Purple, а также методическими тестовыми программами, разработанными в РФЯЦ-ВНИИЭФ, они составляют комплекс тестовых программ, используемый как в РФЯЦ-ВНИИТФ, так и на других предприятиях госкорпорации «Росатом». Этот комплекс позволяет оценить производительность и реальную эффективность использования ВВС, подобрать оптимальный режим выполнения расчетных программ (число вычислительных узлов и используемых на узлах процессов MPI и потоков OpenMP) в зависимости от размера решаемой задачи, а также выявить возникающие в ходе наладки и эксплуатации ВВС проблемы, влияющие на производительность отдельных компонент и подсистем ВВС.

Авторы выражают надежду, что изложенная в статье информация о составе и функциональных возможностях методических тестовых программ окажется полезной другим специалистам, занимающимся разработкой ВВС.

Список литературы/References

- [1]. Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020 г., 21 с.
- [2]. TOP-500. Available at: <https://www.top500.org/> accessed 07.05.2024.
- [3]. HPC Linpack. Available at: <https://netlib.sandia.gov/benchmark/hpl/>, accessed 07.05.2024.
- [4]. HPCG. Available at: <https://www.hpcg-benchmark.org/>, accessed 07.05.2024.
- [5]. NAS Parallel Benchmarks. Available at: <https://www.nas.nasa.gov/software/npb.html>, accessed 07.05.2024.
- [6]. Gropp W., Lusk E., Skjellum A. Using MPI: Portable parallel programming with the message-passing interface. – Second edition. – The MIT Press, 1999
- [7]. OpenMP. Available at: <https://www.openmp.org/>, accessed 07.05.2024.
- [8]. CUDA Toolkit. Available at: <https://developer.nvidia.com/cuda-toolkit/>, accessed 07.05.2024.
- [9]. ASC Purple. Available at: <https://asc.llnl.gov/computers/historic-decommissioned-machines/purple>, accessed 07.05.2024.
- [10]. High Performance Computing benchmarks. Available at: <https://openbenchmarking.org/suite/pts/hpc>, accessed 07.05.2024.
- [11]. Алексеев А.В., Беляев С.П., Бочков А.И. и др. Методические прикладные тесты РФЯЦ-ВНИИЭФ для численного исследования параметров высокопроизводительных вычислительных систем. ВАНТ, сер. Математическое моделирование физических процессов. 2020. Вып. 2., с. 86-100.
- [12]. ГОСТ Р 57700.18-2019. Высокопроизводительные вычислительные системы. Требования к тестовым программам приемочных испытаний.
- [13]. Carlson B.G. A method of characteristics and other improvements in solutions methods for the transport equation. Nuclear Science and Engineering. 1976, №3, p. 408-425.
- [14]. Басс Л.П., Волощенко А.М., Гермогенова Т.А. Методы дискретных ординат в задачах о переносе излучения. Сборник, ИИП им. М.В. Келдыша АН СССР, 1986.
- [15]. Владимиров В.С. Математические задачи односкоростной теории переноса частиц. Труды МИАН СССР, 1961.
- [16]. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров – М.: Наука, 1970.
- [17]. Лебедев С.Н., Писарев В.Н., Романова Е.М. и др. Параллельные вычисления при решении стационарного уравнения переноса. Международный семинар “Супервычисления и математическое моделирование”, Тезисы, Саров, 2000.
- [18]. Самарский А.А. Теория разностных схем. – 3-е изд., испр. – М.: Наука, 1989. – 616 с.
- [19]. Гаджиев А.Д., Писарев В.Н., Рыкованов В.В., Шестаков А.А. Методика и программа для решения двумерного уравнения теплопроводности. ВАНТ. Методики и программы численного решения задач математической физики, 1985, вып. 1, с. 53-65.

- [20]. Писарев В.Н. О параметрическом семействе схем «РОМБ» для нелинейного уравнения теплопроводности. ВАНТ. Методики и программы численного решения задач математической физики, 1986, вып. 3, с. 44-52.
- [21]. Модестов Д.Г. Оценка временной постоянной при учете запаздывающих нейтронов. ВАНТ, 2022, вып.1, с 17-26.
- [22]. Седов Л.И. Методы подобия и размерности в механике. – 8-ое, переработанное изд. – М.: Наука, 1977.
- [23]. Gentry R.A., Martin R.E., Daly B.J. An eulerian differencing method for unsteady compressible flow problems // Journal of Computational Physics. – 1966. – vol. 1.
- [24]. Зельдович Я.Б., Райзер Ю.П. Физика ударных волн и высокотемпературных гидродинамических явлений. М. Наука, 1966г, 688стр.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 1998 года. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 1998. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Сергей Юрьевич МОКШИН – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2016 года. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2016. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.

Александр Викторович ЕРШОВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2020 года. Сфера научных интересов: разностные методы решения интегро-дифференциальных уравнений, разработка параллельных программ численного моделирования.

Alexander Victorovich ERSHOV — The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2020. Research interests: Difference methods of solution integrate-differential equations, parallel numerical simulation programs development.

Артем Владимирович КАРПЕЕВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2015 года. Сфера научных интересов: численные методы решения уравнения механики

сплошной среды, параллельные технологии для современных высокопроизводительных вычислительных систем.

Artem Vladimirovich KARPEEV – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2015. Research interests: numerical methods for solving the equation of continuum mechanics, parallel technologies for modern high-performance computing systems.

Рим Фанавиевич МУХАМАДИЕВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2012 г. Сфера научных интересов: разработка параллельных программ численного моделирования

Rim Fanavievich MUKHAMADIEV – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2012. Research interests: development of parallel numerical simulation programs.

Елена Михайловна РОМАНОВА – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2015 года. Сфера научных интересов: уравнения переноса частиц, разработка алгоритмов решения уравнения переноса частиц для современных вычислительных платформ.

Elena Mikhailovna ROMANOVA – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2015. Research interests: particle transport equations, development of algorithms for solving the particle transport equation for modern computing platforms.

Денис Александрович УШАКОВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2015 года. Сфера научных интересов: уравнения диффузии, разработка алгоритмов решения уравнения диффузии для современных вычислительных платформ.

Denis Alexandrovich USHAKOV – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2015. Research interests: diffusion equations, development of algorithms for solving the diffusion equation for modern computing platforms.

Вадим Олегович АНИСОВ – научный сотрудник Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2018 года. Сфера научных интересов: уравнения диффузии, разработка алгоритмов решения уравнения диффузии для современных вычислительных платформ.

Vadim Olegovich ANISOV – The scientific assistant of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2018. Research interests: diffusion equations, development of algorithms for solving the diffusion equation for modern computing platforms.