



Головченко Е.Н.,  
Дорофеева Е.Ю., Гасилов В.А.,  
Якововский М.В.

Вычислительный  
эксперимент по оценке  
качества алгоритмов  
параллельной декомпозиции  
больших сеток

**Рекомендуемая форма библиографической ссылки:** Вычислительный эксперимент по оценке качества алгоритмов параллельной декомпозиции больших сеток / Е.Н.Головченко [и др.] // Препринты ИПМ им. М.В.Келдыша. 2013. № 7. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2013-7>

**Ордена Ленина**  
**ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ**  
**имени М.В.Келдыша**  
**Российской академии наук**

**Е.Н. Головченко, Е.Ю. Дорофеева,**  
**В.А. Гасилов, М.В. Якобовский**

**Вычислительный эксперимент по оценке  
качества алгоритмов параллельной  
декомпозиции больших сеток**

**Москва — 2013**

**Е.Н. Головченко, Е.Ю. Дорофеева, В.А. Гасилов, М.В. Якововский**

Вычислительный эксперимент по оценке качества алгоритмов параллельной декомпозиции больших сеток

При численном решении задач механики сплошных сред, электродинамики и т.д. методом сеток на распределенных вычислительных системах возникает задача сбалансированного распределения сетки по процессорам. Для решения этой задачи был создан комплекс программ параллельной декомпозиции сеток. В комплекс вошло два алгоритма: параллельный алгоритм геометрической декомпозиции сеток и параллельный инкрементный алгоритм декомпозиции графов. Целью данной работы является проведение вычислительного эксперимента по оценке эффективности разработанных алгоритмов на примере решения ряда задач магнитогидродинамики.

**Ключевые слова:** численный эксперимент, метод сеток, декомпозиция сеток, параллельные вычисления, газовая динамика, механика сплошных сред

**E.N. Golovchenko, E.Yu. Dorofeeva, V.A. Gasilov, M.V. Yakobovskiy**

Computational experiment to qualitative assessment of the algorithms for parallel decomposition of the large grids

In the mesh-based numerical solution of problems of continuum mechanics, electrodynamics etc. on distributed computing systems the problem of a balanced grid distribution over the processors is arising. To this end, the program of parallel decomposition of meshes was created. The complex includes two algorithms: a parallel algorithm for the geometric decomposition of meshes and parallel incremental algorithm for decomposition of graphs. The aim of this work is to carry out computer simulation to evaluate the effectiveness of the developed algorithms for the solution of some problems of magnetohydrodynamics.

**Key words:** numerical simulation, mesh-based method, grids decomposition, parallel computing, gasdynamics, continuum mechanics

Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант №. 11-07-93939-ДЖИ8-а) и Минобрнауки России по государственному контракту № 07.524.12.4019 от 17.05.2012 г. Расчеты выполнены на суперкомпьютерах ЛОМОНОСОВ (НИВЦ МГУ), Helios (IFERC, EU(F4E) - Japan Broader Approach collaboration) и К-100 (ИПМ им. М.В. Келдыша РАН).

## Введение

Задача декомпозиции сетки возникает во многих приложениях метода сеток, таких как теория упругости, гидрогазодинамика, электродинамика и др. Развитие высокопроизводительных технологий сделало возможным решение больших задач, на расчет которых на одном процессоре не хватает памяти или вычислительных ресурсов. В рассматриваемых параллельных приложениях зачастую используется метод геометрического параллелизма, при котором сетка, аппроксимирующая расчетную область, распределяется между процессорами по геометрическому признаку. В дальнейшем каждый процессор обрабатывает свою часть сетки. Эффективность работы процессоров определяется тем, насколько равномерно распределена сетка по процессорам и насколько минимизированы затраты на передачу данных между процессорами. Объем передач данных между процессорами зависит от числа связей между доменами (частями сеток), распределенными по процессорам. Декомпозиция регулярных сеток намного проще декомпозиции нерегулярных сеток, однако нерегулярные сетки, в частности, треугольные и тетраэдральные, лучше аппроксимируют области сложной геометрической формы и легче сгущаются.

Задача сбалансированного разбиения сетки на домены сводится к более общей задаче разбиения графа на домены. В этом случае рассматривается граф, который соответствует коммуникационным нагрузкам сетки. Существует несколько моделей декомпозиции графов [1], отличающихся видом графа и критериями сбалансированного разбиения. В случае разбиения сеток хорошо себя зарекомендовал наиболее распространенный подход, использующий стандартную модель графа. В нем сетка аппроксимируется неориентированным графом  $G = (V, E)$ , где  $V$  – множество вершин,  $E$  – множество ребер. И вершины, и ребра имеют вес. Оптимальным считается разбиение на домены, при котором выровнен суммарный вес вершин в доменах и минимизирован суммарный вес разрезанных ребер между доменами (разрезанное ребро – ребро, соединяющее вершины из разных доменов). В данной модели суммарный вес вершин в доменах отвечает за равномерность разбиения сетки на домены (а в дальнейшем равномерность распределения по процессорам, которые будут обрабатывать эти домены), а суммарный вес разрезанных ребер – за коммуникационную нагрузку между процессорами. Как известно, данная задача декомпозиции графа является NP-полной, поэтому для ее решения используются различные эвристические методы: алгоритмы координатной, инерциальной и спектральной рекурсивной бисекции, алгоритм Kernighan-Lin (KL) и Fiduccia-Mattheyses (FM), жадные алгоритмы (greedy methods), иерархические алгоритмы. Эти алгоритмы реализованы в следующих последовательных пакетах декомпозиции графов: METIS, JOSTLE, SCOTCH, CHACO и PARTY. К параллельным пакетам относятся PARMETIS (параллельная версия пакета METIS), JOSTLE, PT-SCOTCH (параллельная версия пакета SCOTCH) и ZOLTAN.

Целью данной работы является получение сбалансированных разбиений гексаэдральных и тетраэдральных сеток, содержащих до  $10^9$  вершин, на большое число связанных микро-доменов.

Число процессоров, на которых будет считаться вычислительная задача, зачастую заранее неизвестно. Поэтому имеет смысл разбивать сетку на большое число микро-доменов, а потом формировать из них домены. Количество микро-доменов на несколько порядков меньше числа вершин, поэтому многократное разбиение микро-доменов на домены быстрее многократного разбиения всей сетки.

Еще одной областью использования разбиения сеток на микро-домены является хранение больших сеток. В микро-доменах функции достаточно гладкие, что позволяет отрезать биты в рамках одного микро-домена. В результате на хранение информации затрачивается меньше памяти. Экономит память также локальная нумерация вершин от нуля в рамках одного микро-домена, что позволяет на хранение одного номера вершины тратить меньше байтов. Существуют и другие способы сжатия информации о сетке в рамках одного микро-домена. При сжатом хранении сеток, разбитых на микро-домены, также важна сбалансированность разбиения.

Областью данного исследования являются нерегулярные сетки, содержащие до  $10^9$  вершин. В настоящее время такие сетки невозможно разместить в памяти одного процессора, следовательно, разбить их можно только параллельным алгоритмом. Методы разбиения графов параллельных пакетов PARMETIS, JOSTLE, PT-SCOTCH и ZOLTAN основываются на иерархических алгоритмах, состоящих из следующих частей: поэтапное огрубление графа, декомпозиция самого маленького из полученных графов и отображение разбиения на предыдущие графы с периодическим локальным уточнением границ доменов. Недостатком таких алгоритмов является образование доменов, границы которых состоят из неоптимальных наборов сегментов [2]. В частности, домены могут оказаться несвязными. Такое ухудшение качества доменов для некоторых задач является критичным. Например, на доменах с длинными границами или сложной конфигурацией алгоритмы решения систем линейных уравнений сходятся за большее число итераций. Связность доменов также важна при визуализации распределенных сеточных данных, и на связанных микро-доменах коэффициент сжатия информации о сетке будет больше.

Пакет JOSTLE разбивает на число доменов, равное числу процессоров, на которых производится разбиение. Поэтому для разбиения на микро-домены он не подходит. Кроме того, пакет JOSTLE не выложен в свободном доступе, и получение его проблематично. Пакет PARMETIS является основным пакетом, который используется в данный момент для разбиения сеток. Недостатком пакета PARMETIS является получение сильно несбалансированных разбиений.

В рамках данной работы был создан комплекс программ декомпозиции больших сеток (до  $10^9$  вершин) на большое число микро-доменов. В него вошли два алгоритма: параллельный алгоритм геометрической декомпозиции

сеточных данных и параллельный инкрементный алгоритм декомпозиции графов.

Проведено сравнение разбиений на микро-домены одной гексаэдральной и двух тетраэдральных сеток, полученных методами созданного комплекса программ, методами пакета PARMETIS, пакета ZOLTAN и пакетом PT-SCOTCH. Также проведено сравнение различных разбиений графов микро-доменов на домены и разбиений сразу на домены.

### **Комплекс программ параллельной декомпозиции сеток**

Комплекс программ параллельной декомпозиции сеток включает в себя два алгоритма: параллельный алгоритм геометрической декомпозиции сеточных данных и параллельный инкрементный алгоритм декомпозиции графов. Оба алгоритма реализованы на компьютерах с распределенной памятью с использованием библиотеки MPI.

Параллельный алгоритм геометрической декомпозиции основан на методе рекурсивной координатной бисекции. Параллельные алгоритмы геометрического разбиения реализованы только в пакетах PARMETIS и ZOLTAN. В пакете PARMETIS геометрический алгоритм основывается на разбиении кривой Гильберта. В пакете ZOLTAN реализованы три геометрических алгоритма: алгоритм рекурсивной координатной бисекции, алгоритм рекурсивной инерциальной бисекции и алгоритм, разрезающий кривую Гильберта. Отличие рекурсивной координатной бисекции созданного алгоритма от аналогичного алгоритма в пакете ZOLTAN состоит в том, что разбиение производится одновременно по нескольким координатам. Вершины сортируются сначала по одной координате, потом внутри нее по следующей координате в циклическом порядке и т.д., что позволяет обрабатывать ситуации наличия нескольких узлов с одним значением координаты. В результате медиана проводится точно, и в разбиении на равные домены числа вершин в доменах отличаются не больше, чем на единицу. В пакете ZOLTAN вершины из медианы распределяются по областям произвольным образом, что хотя и сохраняет разницу числа вершин в доменах не превышающей единицу, но увеличивает число разрезанных ребер. Достоинствами параллельного алгоритма геометрической декомпозиции, как и других геометрических методов, являются экономичное использование памяти и относительная быстрота работы. Недостатками является то, что алгоритм не учитывает связи между вершинами и что ему требуется геометрическая информация о сетке. Целью создания данного алгоритма являлось получение сбалансированных разбиений на микро-домены. В частности, при разбиениях на равные части разница числа вершин в доменах не превышает единицу.

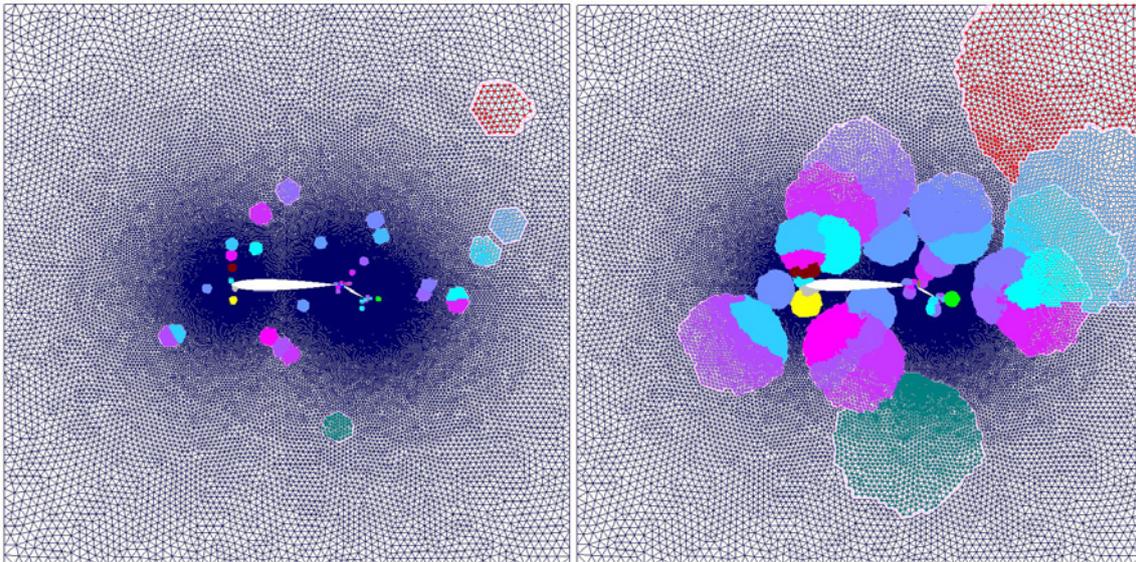
Параллельный инкрементный алгоритм основан на последовательном инкрементном алгоритме декомпозиции графов, созданном Якобовским М.В. в ИПМ им. М.В.Келдыша РАН [3]. Достоинством параллельного инкрементного алгоритма является формирование связных доменов. Сбалансированность

разбиений, получаемых параллельным инкрементным алгоритмом, оказывается несколько хуже, чем разбиений, получаемых алгоритмами геометрической декомпозиции, но заметно лучше, чем разбиений, получаемых пакетом PARMETIS, а также пакетом PT-SCOTCH. Наличие геометрической информации для алгоритма необязательно, поскольку предварительное распределение вершин по процессорам можно получить огрублением графа вместо геометрической декомпозиции.

### **Инкрементный алгоритм декомпозиции графов**

В начале выполнения инкрементного алгоритма декомпозиции графов [3] все вершины считаются свободными, то есть не распределенными по доменам. Затем для каждого домена случайным образом выбирается вершина, которая помещается в этот домен и становится его инициализирующей вершиной. Далее разбиение производится посредством итерационного процесса, на каждом шаге которого выполняются следующие действия:

1. Инкрементный рост доменов (захват вершин). На данном этапе происходит постепенное присоединение к доменам прилегающих к ним свободных вершин и диффузное перераспределение вершин на границах между доменами с целью выравнивания суммарного веса вершин в доменах. В конце данного этапа все вершины должны быть распределены по доменам. На рис. 1 отображены начальные этапы роста доменов. Домены, не соприкасающиеся с другими доменами, имеют шарообразную форму. Пространство между доменами состоит из свободных вершин.



*Рис. 1.* Инкрементный рост доменов

2. Локальное уточнение доменов. Выполняется KL/FM алгоритм локального уточнения [4]. Этот алгоритм предназначен для уменьшения веса разрезанных ребер между доменами. Результат выполнения локального

уточнения доменов представлен на рис. 3. Границы доменов стали более гладкими. Однако в разбиении все еще присутствуют несвязные домены.

На рис. 2 показан результат диффузного перераспределения вершин между доменами. Как видно из рисунка, границы доменов сильно изрезаны, что увеличивает вес разрезанных ребер между доменами. Перенос вершин между доменами приводит также к распаду доменов на отдельные части.

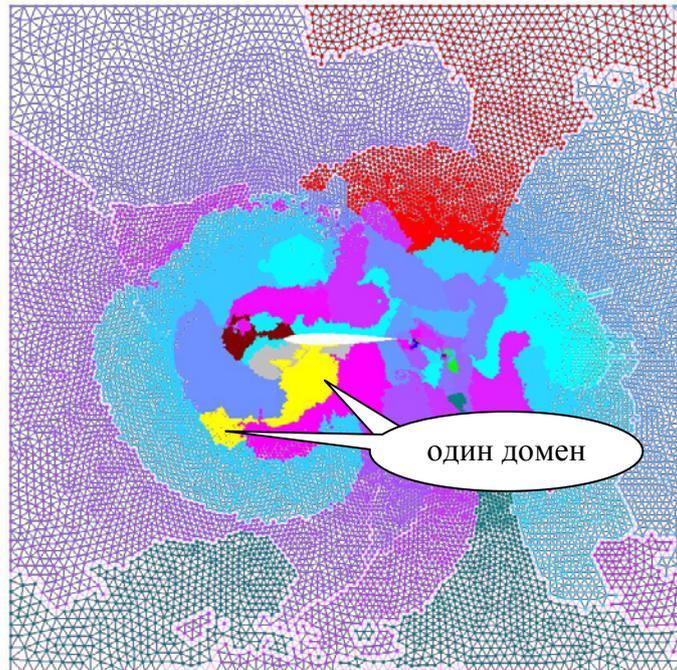


Рис. 2. Диффузное перераспределение вершин между доменами

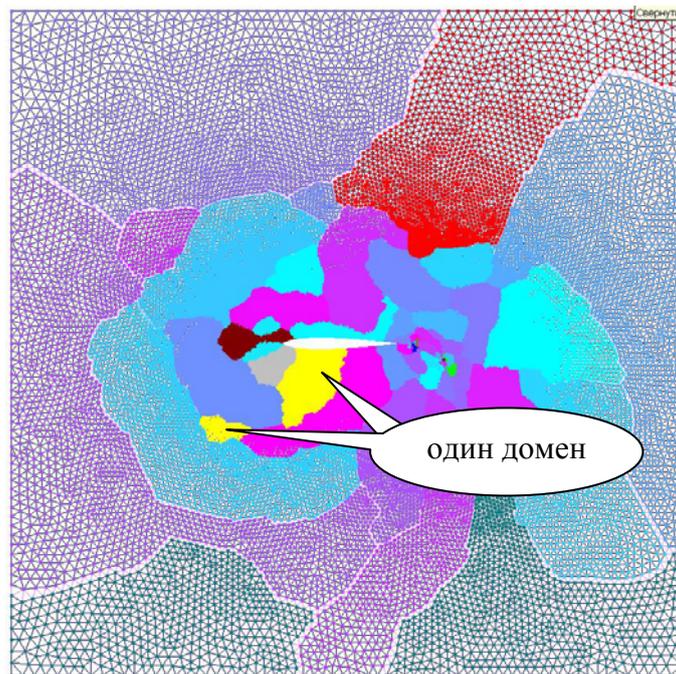
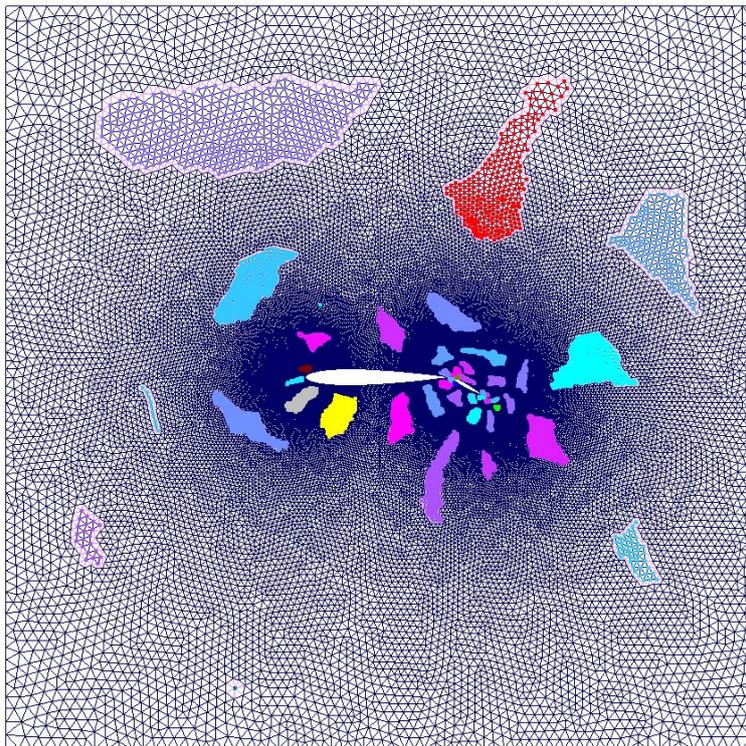


Рис. 3. Локальное уточнение доменов

3. Проверка качества доменов. Если качество доменов соответствует заданному, разбиение считается найденным и происходит выход из цикла, иначе – переход к следующему этапу. Проверка качества доменов описана ниже.

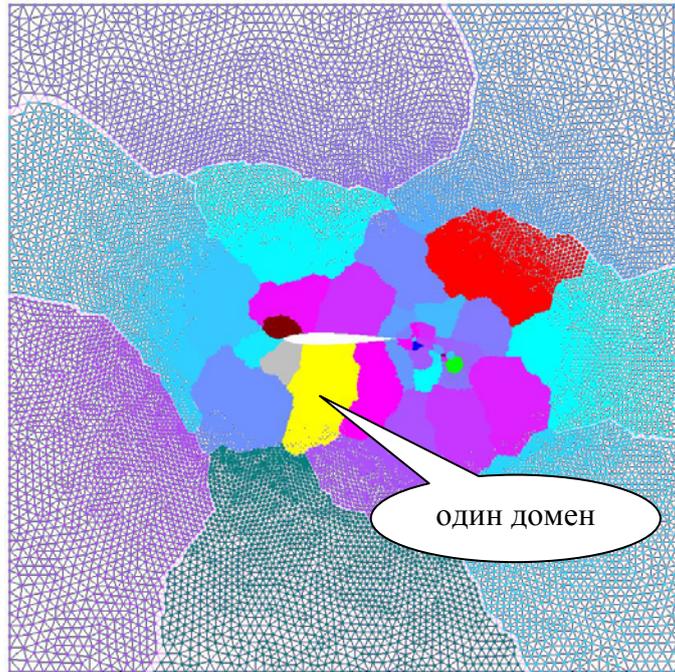
4. Освобождение части вершин плохих доменов и переход к первому этапу. Плохими доменами считаются домены, качество которых не соответствует заданному, и их соседи. В плохих доменах часть вершин освобождается, то есть вновь считается нераспределенной (см. рис. 4). При этом если домен состоит из нескольких отдельных частей, то оставляется только наибольшая часть.



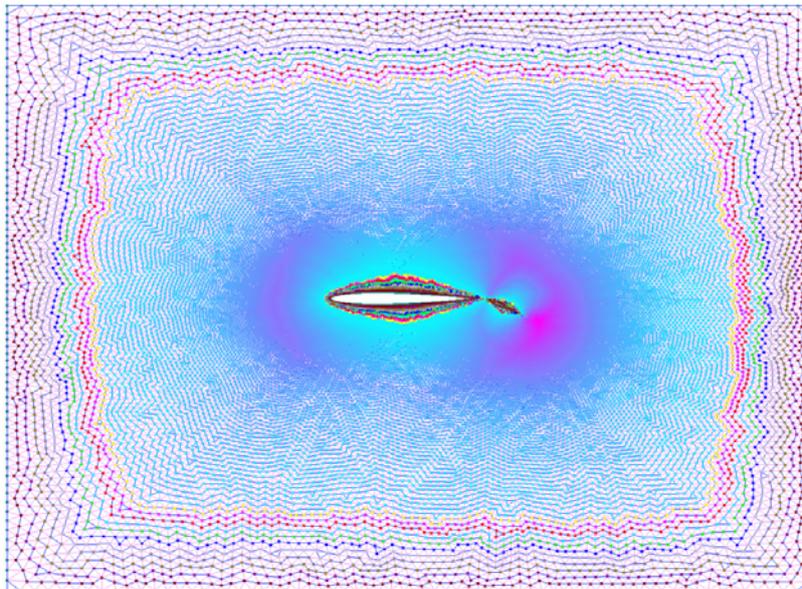
*Рис. 4. Освобождение части вершин*

Результатирующее разбиение представлено на рис. 5. Как видно из рисунка, домен, который раньше состоял из двух частей, теперь является связным.

Качество доменов проверяется следующим образом. Все вершины, расположенные на геометрической границе графа и на границах между доменами, считаются принадлежащими первой оболочке своего домена. В каждом домене вершинами второй оболочки считаются соседи вершин из первой оболочки данного домена, которые также принадлежат этому домену и не попали в первую оболочку. Остальные оболочки вычисляются аналогично. На рис. 6 представлены первые несколько оболочек при условии, что вся сетка принадлежит одному домену. Проверяется связность оболочек каждого домена и вычисляется номер наименьшей несвязной оболочки в каждом домене. Хорошими считаются те домены, номер наименьшей несвязной оболочки в которых больше, либо равен пороговому значению.



*Рис. 5. Результирующее разбиение*



*Рис. 6. Оболочки*

### **Параллельный инкрементный алгоритм декомпозиции графов**

Прямое распараллеливание всех этапов инкрементного алгоритма декомпозиции графов невозможно, поскольку этапы инкрементного роста и локального уточнения доменов последовательны по своей природе. В них требуется периодическое обновление информации, которое в параллельном варианте приведет к большим коммуникационным издержкам. Поэтому выбрана следующая стратегия распараллеливания. На каждом процессоре свой блок вершин разбивается локально инкрементным алгоритмом декомпозиции

графов, что позволяет избежать передач данных между процессорами. Недостатком локального разбиения является то, что группы не могут расширяться за пределы процессора, что ограничивает число просматриваемых хороших вариантов разбиения. В результате на границах между процессорами могут образоваться плохие группы доменов. Для устранения этого недостатка в последствии плохие группы доменов перераспределяются между процессорами так, чтобы каждая группа плохих доменов оказалась собранной на одном процессоре. Затем происходит локальное повторное разбиение плохих групп доменов.

В данной модели предполагается, что после локального разбиения инкрементным алгоритмом декомпозиции графов плохими окажутся не все домены на процессорах, а только небольшая их часть на границах между процессорами. Это утверждение выполняется, когда на процессорах образуется достаточно большое количество доменов. На практике число процессоров, на которых будет считаться задача, заранее неизвестно, и выгоднее сразу разбить граф на большое количество микро-доменов, чтобы в дальнейшем распределять эти домены по нужному числу процессоров. Количество микро-доменов на несколько порядков меньше числа вершин, поэтому многократное распределение микро-доменов по нужному числу процессоров быстрее многократного разбиения всего графа. Таким образом, предположение о том, что количество доменов будет достаточно велико, является оправданным.

Существует несколько подходов к перемещению вершин между доменами. Именно в них состоит основное отличие между иерархическими алгоритмами PARMETIS и JOSTLE. В PARMETIS используется виртуальное перемещение вершин между доменами, при котором вершины остаются на том же процессоре, меняется только значение поля, отвечающего за домен, которому принадлежит вершина. В JOSTLE каждый процессор отвечает за один домен, и если вершина из этого домена переносится в другой домен, то она передается на другой процессор, отвечающий за новый домен. Поскольку реальная передача вершин между доменами предполагает коммуникационные издержки и не позволяет использовать алгоритм локально, в параллельном инкрементном алгоритме декомпозиции графов используется виртуальная передача вершин.

Как и в любом параллельном алгоритме декомпозиции графов, предполагается, что вершины изначально неким образом распределены по процессорам. Здесь возникает «задача в задаче», когда для нахождения разбиения нужно изначально разбить граф. В качестве начального разбиения обычно берут более простое разбиение, например, разбиение по номерам вершин, или геометрическое разбиение. Ограничение на расширение доменов за пределы процессоров позволило исключить синхронизацию между процессорами. Однако в результате наложения данного ограничения возникла зависимость локального алгоритма инкрементного роста доменов от начального распределения вершин по процессорам. Если при начальном распределении на процессор попадет множество мелких групп вершин, то после выполнения алгоритма инкрементного роста некоторые группы вершин

окажутся не распределенными по доменам. Поэтому для получения относительно компактного начального распределения вершин по процессорам целесообразно использовать геометрическое разбиение. Подробнее о геометрическом разбиении написано далее.

После геометрического разбиения на процессор должен попасть такой общий вес вершин, какой будет в образуемых на нем доменах, что позволит сформировать домены одинакового веса.

Однако геометрическое разбиение не гарантирует, что на процессор вообще не попадут мелкие группы вершин. Например, если граф описывает сетку, содержащую в себе отверстия, при геометрическом разбиении на процессор может попасть основной блок вершин с одной стороны отверстия и небольшая группа вершин с другой стороны (см. рис. 8а). Таким образом, даже после геометрической декомпозиции необходимо предварительное перераспределение малых блоков вершин между процессорами.

Учитывая все сказанное, параллельный инкрементный алгоритм декомпозиции графов выглядит следующим образом:

1. Разбиение вершин на число доменов, равное числу процессоров, на которых будет выполняться дальнейшее разбиение, параллельным алгоритмом геометрической декомпозиции сеток, описанным ниже.

2. Распределение вершин по процессорам в соответствии с результатами геометрической декомпозиции. Общий вес вершин на процессорах определяется количеством будущих доменов. На рис. 7а и 8а сетка разбита по пяти процессорам. Поскольку используемая геометрическая декомпозиция разбивает сетку перпендикулярно осям координат, области имеют прямоугольную форму. На рис. 8а видно, что у двух процессоров части сетки содержат малые блоки вершин, отделенные от основных блоков отверстием.

3. Присоединение малых блоков вершин к большим и восстановление исходного веса вершин на процессорах. Как показано на рис. 8б, малые блоки вершин отданы процессорам, с частями сеток которых они связаны. Взамен получены равноценные блоки вершин, связанные с частями сеток отдавших процессоров (см. рис. 7б и 8б).

4. Инициализация доменов. Поскольку на процессоре может оказаться несколько больших блоков вершин, сначала оценивается вес вершин в блоках; в соответствии с ним определяется, сколько доменов будет образовано в каждом из блоков, а затем случайным образом выбирается нужное количество инициализирующих вершин в каждом из блоков.

5. Локальное разбиение вершин на домены инкрементным алгоритмом декомпозиции графов. На рис. 9а и 10а четко выражены границы между процессорами, домены не пересекают эти границы.

6. Перераспределение плохих групп доменов. Сбор каждой группы плохих доменов на одном процессоре.

7. Локальное повторное разбиение плохих групп доменов инкрементным алгоритмом декомпозиции графов (см. рис. 9б и 10б). На рис. 10б видно, что

после перераспределения плохих групп доменов и их повторного разбиения домены вышли за границы между процессорами.

При считывании данных использовалась библиотека распределенного хранения и обработки тетраэдральных сеток (Суков С.А., ИПМ им. М.В.Келдыша РАН).

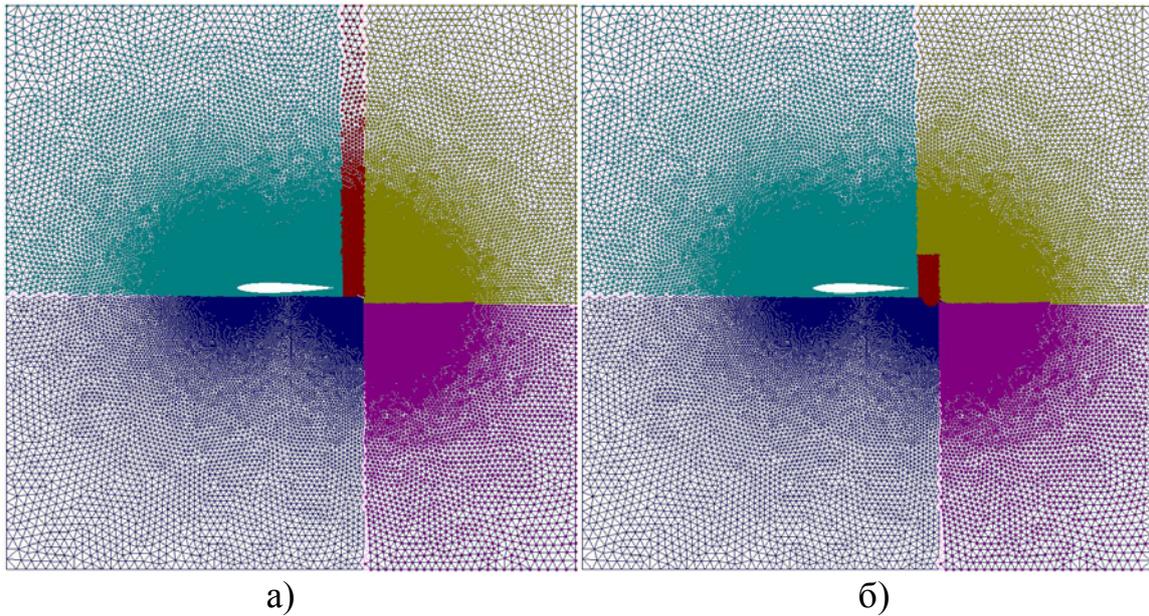


Рис. 7. Геометрическое разбиение сетки по пяти процессорам (а) и перераспределение малых блоков вершин (б)

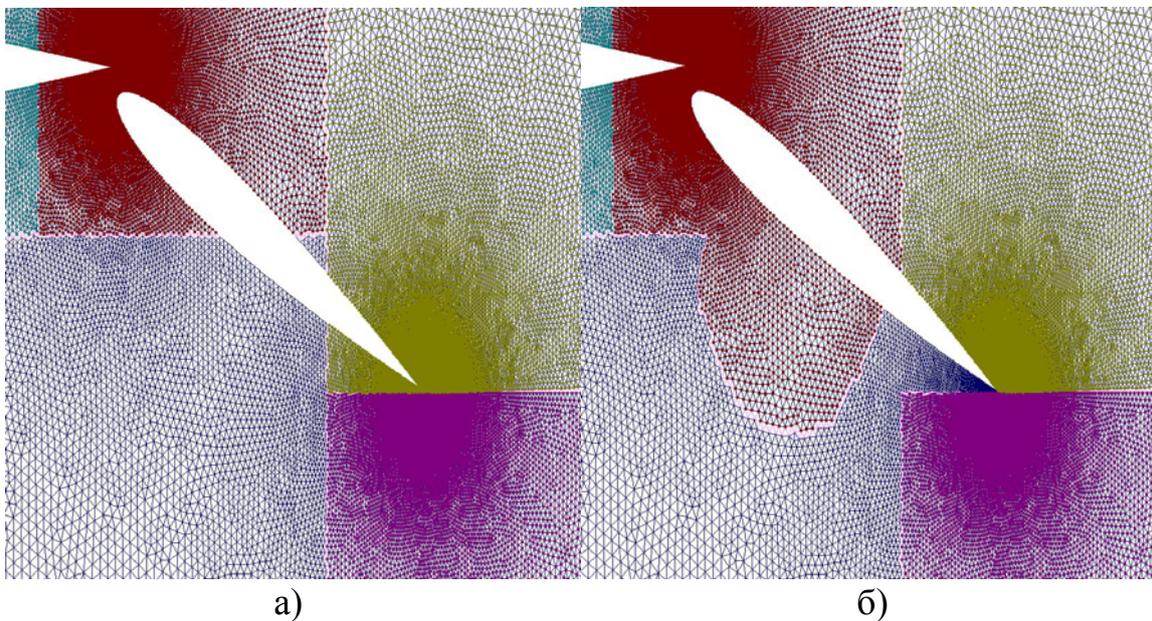
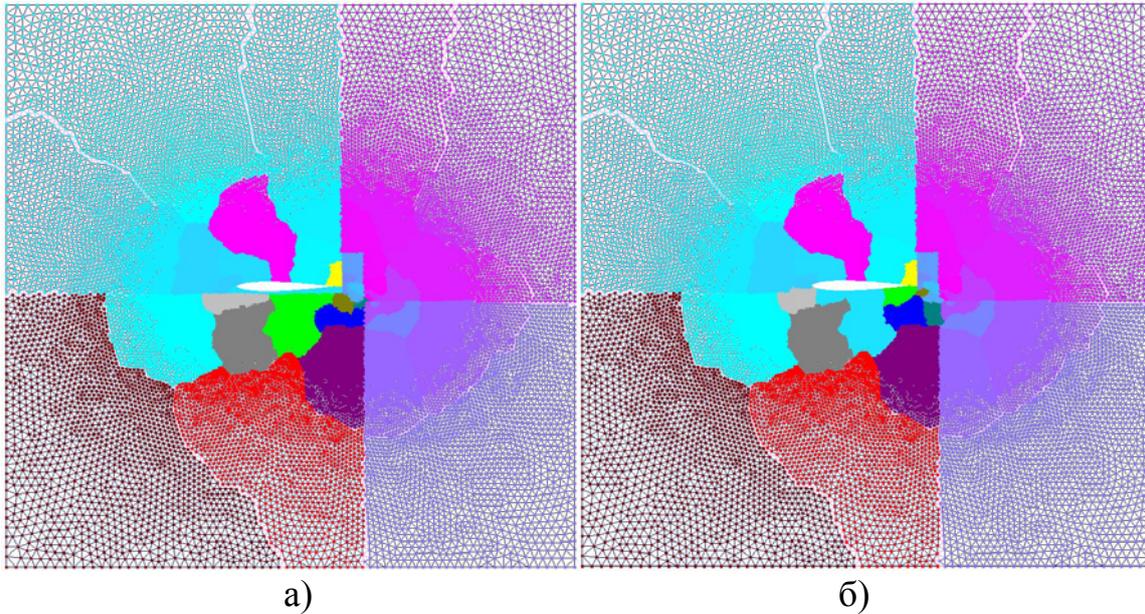
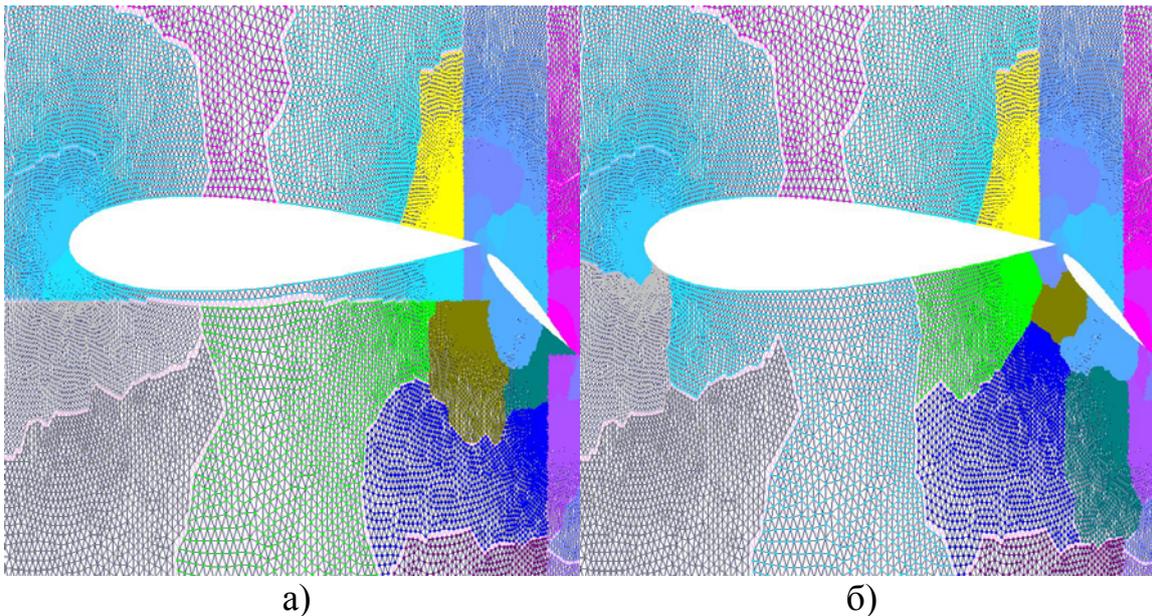


Рис. 8. Фрагменты геометрического разбиения сетки по пяти процессорам (а) и перераспределения малых блоков вершин (б)



*Рис. 9.* Результаты локального разбиения (а) и сбора плохих групп доменов, а также их повторного разбиения (б)



*Рис. 10.* Фрагменты результатов локального разбиения (а) и сбора плохих групп доменов, а также их повторного разбиения (б)

### **Параллельный алгоритм геометрической декомпозиции сеточных данных**

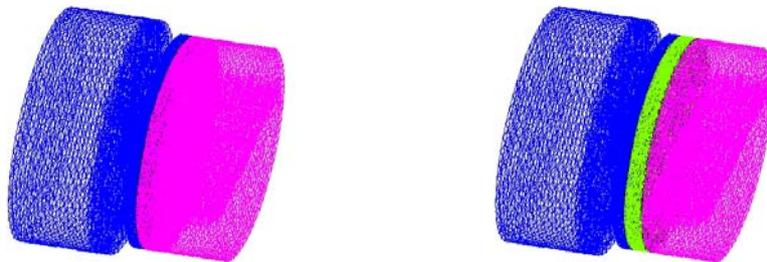
Алгоритм геометрической декомпозиции основывается на методе рекурсивной координатной бисекции [5, 6]. На каждом этапе рекурсивной бисекции область разбивается на две части. Соотношение размеров частей зависит от количества доменов, которые должны быть образованы в каждой из частей. Полученные подобласти разбиваются дальше аналогичным образом до тех пор, пока в подобластях не останется по одному домену. При рекурсивной

координатной бисекции на этапе разбиения выбирается координатная ось, вдоль которой область имеет наибольшую протяженность. Область разбивается перпендикулярно полученной оси. Алгоритм работает только с координатами вершин и не учитывает связи между ними, что делает его экономичным по памяти.

Основные этапы алгоритма следующие:

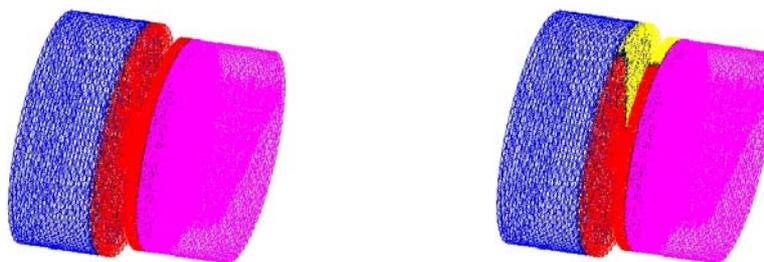
1. Начальное распределение узлов по процессорам. На процессор попадает столько узлов, сколько будет в образуемых на нем доменах. Начальное распределение может быть любым, например, в соответствии с порядковыми номерами узлов.

2. Рекурсивная координатная бисекция вершин по процессорам. На каждом этапе сначала вычисляется координатная ось, вдоль которой область имеет наибольшую протяженность. Затем блок вершин бьется на две части перпендикулярно данной оси (см. рис. 11). Группа процессоров делится на две, далее каждая из групп делит свой блок вершин аналогичным образом. Для разделения блока вершин используется параллельная сортировка по выбранной оси координат. В процессе сортировки, а также после нее число узлов на процессорах сохраняется. На рис. 11 отображена рекурсивная координатная бисекция вершин по трем процессорам. Сначала сетка была разбита на две части, затем вторая часть разделена еще на две.

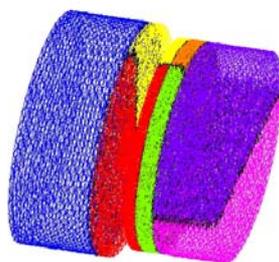


*Рис. 11.* Рекурсивная координатная бисекция вершин по трем процессорам

3. Локальная рекурсивная координатная бисекция вершин по доменам. Дальнейшее разбиение на домены производится локально на каждом процессоре. На рис. 12 показана локальная рекурсивная координатная бисекция на три домена на одном из процессоров. На рис. 13 представлен результат разбиения сетки на семь доменов на трех процессорах.



*Рис. 12.* Локальная рекурсивная координатная бисекция вершин на одном из процессоров

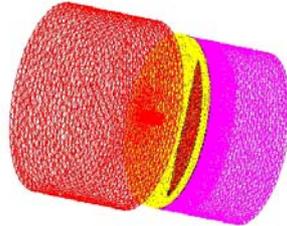


*Рис. 13.* Результат разбиения сетки на семь доменов на трех процессорах

В процессе выполнения рекурсивной координатной бисекции вершин по процессорам локально на каждом процессоре данные сортируются либо сортировкой слиянием, либо пирамидальной, либо оптимизированной сортировкой (гибрид сортировки слиянием и пирамидальной сортировки) [7,8]. В качестве параллельной сортировки была взята битонная сортировка Бэтчера [7, 8]. Для сортировки данных использовалась библиотека параллельной сортировки с сайта <http://lira.imamod.ru/FondProgramm/Sort/> (Якобовский М.В., ИПМ им. М.В.Келдыша РАН). Рекурсивная параллельная сортировка увеличивает время работы алгоритма, но улучшает качество распределения блоков вершин по процессорам, в сравнении с однократным разделением общего блока вершин между процессорами по одной из координат. Поскольку количество памяти, необходимое для работы данного алгоритма, невелико (сетка  $2 \cdot 10^8$  вершин разбивалась на 40 процессорах, 512 Мб на каждом), такой вариант приемлем.

Отличие локальной рекурсивной координатной бисекции от обычного метода рекурсивной координатной бисекции состоит в следующем. После каждого разбиения производится разделение данных по аналогии с обменной сортировкой с разделением Хоару (быстрая сортировка) [7], благодаря чему на каждом шаге просматриваются только узлы из разбиваемой области. Выполняется сортировка не всего разбиваемого блока вершин, а только заранее определенного интервала, в котором находится медиана. Разбиение производится одновременно по нескольким координатам, вершины

сортируются сначала по одной координате, потом внутри нее по следующей координате в циклическом порядке и т.д., что позволяет обрабатывать ситуации наличия нескольких узлов с одним значением координаты (см. рис. 14). В результате медиана проводится точно, и в разбиении на равные домены числа вершин в доменах отличаются не больше, чем на единицу.

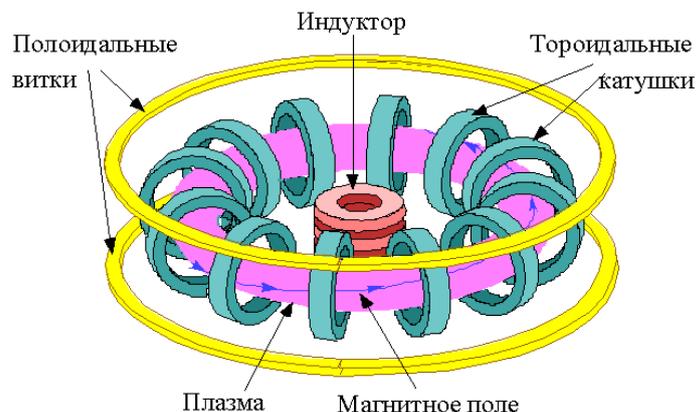


*Рис. 14.* Разбиение по нескольким координатам в локальной рекурсивной координатной бисекции

### Подготовка численного эксперимента

С целью проверки эффективности описанных выше алгоритмов был поставлен ряд численных экспериментов. Исследования выполнялись на примере трех задач, описанных далее.

**Моделирование газоплазменных потоков в диверторе токамака.** Токамак (тороидальная камера с магнитными катушками) - тороидальная установка для магнитного удержания плазмы с целью достижения условий, необходимых для протекания управляемого термоядерного синтеза. Плазма в токамаке удерживается специально создаваемым комбинированным магнитным полем – тороидальным внешним, и полоидальным полем тока, протекающего по плазменному шнуру. Ток в плазме обеспечивает разогрев плазмы и удержание равновесия плазменного шнура в вакуумной камере. Этим токамак, в частности, отличается от стелларатора, являющегося одной из альтернативных схем удержания, в котором и тороидальное, и полоидальное поля создаются с помощью внешних магнитных катушек.



*Рис. 15.* Схема принципиальных узлов токамака

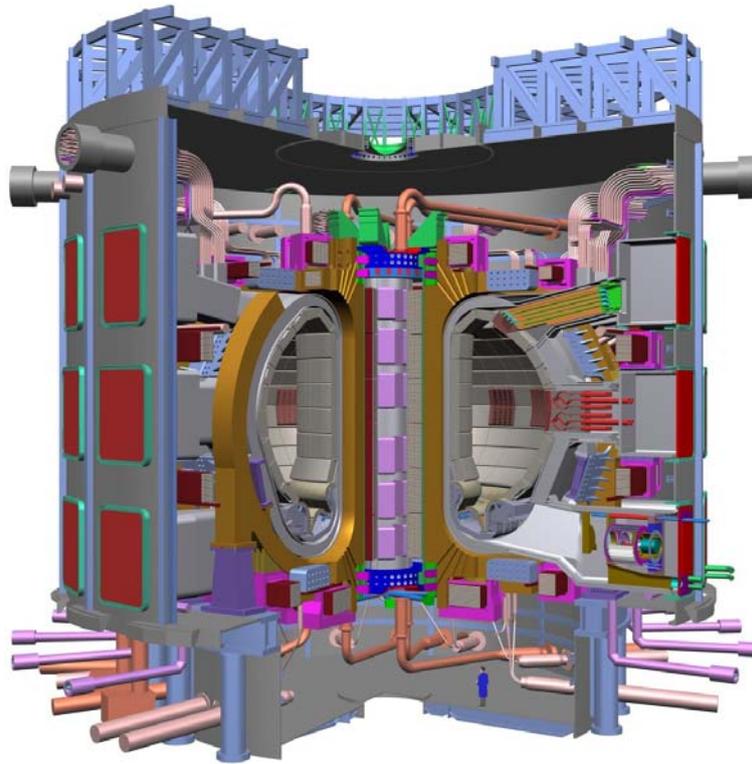


Рис. 16. Проектируемый токамак ITER - общий вид

Дивертор является одним из ключевых компонентов токамака ITER [9]. Он расположен вдоль нижней части вакуумной камеры и служит для приема потоков примесей и излучений из плазмы. В диверторе можно выделить два принципиальных блока: несущая конструкция, изготавливаемая из нержавеющей стали и компоненты, непосредственно контактирующие с плазмой. В машине ITER дивертор – это 54 дистанционно заменяемые кассеты, каждая из которых содержит три основных компонента, контактирующие с плазмой: диверторные пластины (внутренняя и внешняя) и купол (см. рис. 17). Диверторные пластины расположены на пересечении силовых линий магнитного поля. Здесь потоки высокоэнергетичной плазмы врезаются в пластины. Их кинетическая энергия преобразуется в тепло, и тепловой поток, приходящийся на пластины, чрезвычайно интенсивен. В диверторных камерах удаётся смягчить нагрузку от плазмы на диверторные пластины за счёт дополнительного охлаждения.

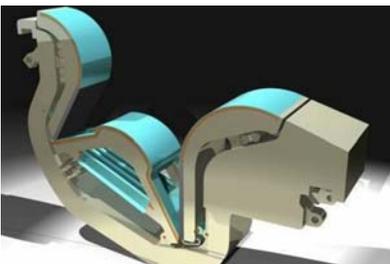
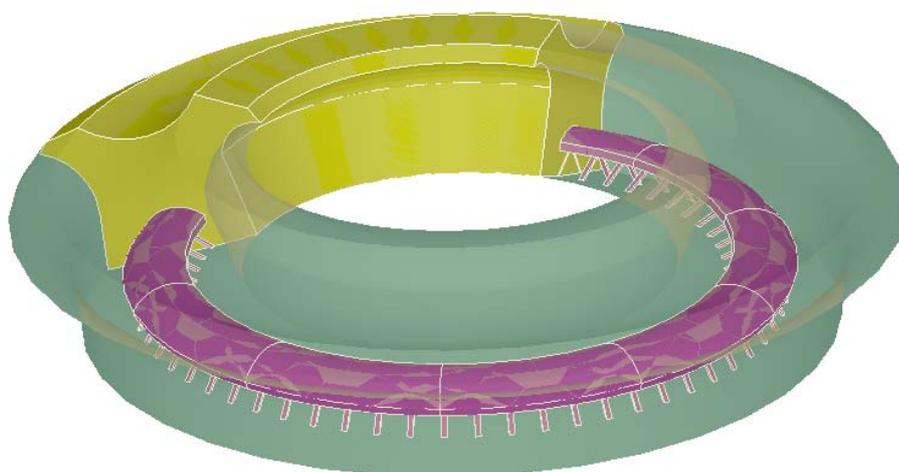


Рис. 17. Кассета дивертора ITER



*Рис. 18.* Дивертор токамака ITER - общий вид

Для моделирования газоплазменных потоков в диверторе токамака была построена принципиальная схема дивертора (см. рис. 19), максимально соответствующая доступному описанию дивертора ITER. Желтым цветом на рисунке выделена часть расчетной области. Лиловым обозначен купол дивертора.



*Рис. 19.* Геометрия дивертора

В расчетной области была построена тетраэдральная сетка порядка 3 миллионов ячеек. Для расчета задачи использовались разбиения сетки на 256 доменов. Решалась полная система уравнений радиационной газовой динамики с табличными уравнениями состояния с учетом диссипативных процессов и турбулентной вязкости. Справедливость такого приближения для расчета течений в диверторе ранее обсуждалась в ряде работ (см., например, [10]).

Расчет газоплазменных течений в диверторе токамака производился на суперкомпьютере «Helios» на 256 процессорных модулях.

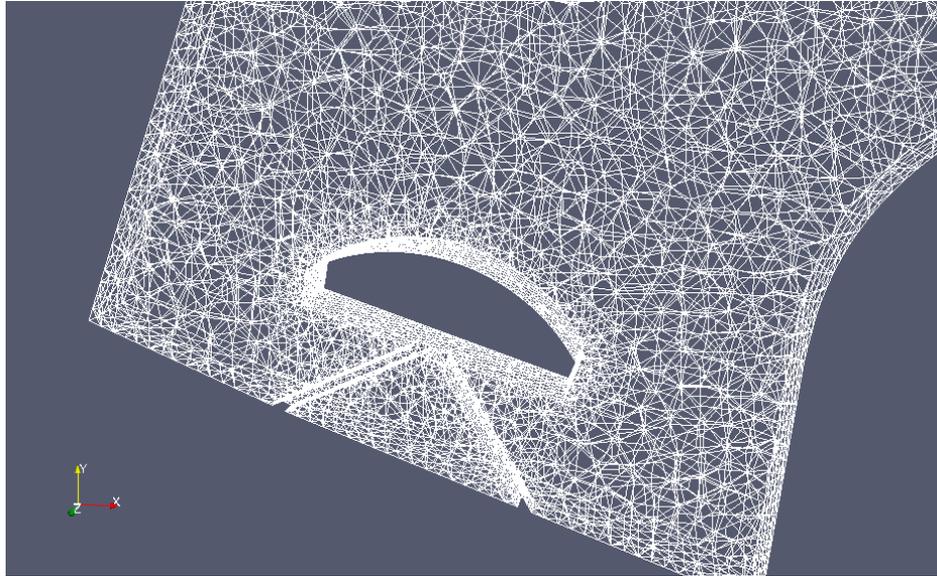


Рис. 20. Фрагмент сетки в районе купола дивертора

**Моделирование распространения ударной волны от приземного источника энергии взрывного типа.** Собственно взрыв, если под этим термином подразумевать первичное энерговыделение, развивается за время от нескольких микросекунд до миллисекунд. Однако чаще всего, говоря о взрыве, имеют в виду внешнюю картину – появление светящейся области, приход в точку наблюдения ударной волны, подъем облака взрыва и т.п.

Количественные значения параметров во фронте ударной волны (давление, плотность, температура воздуха, массовая скорость и др.) на стадии взрыва, когда можно пренебречь формой и размерами источника, удовлетворительно описываются автомодельным решением (модель точечного взрыва [11]), если учесть, что в ударную волну может быть передано до 50% энергии взрыва. Вначале давление во фронте ударной волны может превышать нормальное атмосферное до 20000 раз, а плотность – до 10 раз. При взрыве вблизи поверхности земли важную роль в распространении воздушной ударной волны играет взаимодействие с землей и образование отраженной волны. При взаимодействии со светящейся областью отраженной от земли ударной волны происходит нарушение сферической симметрии (нижняя кромка сферы деформируется).

Внешняя картина развития взрыва во многом зависит от высоты, на которой осуществляется взрыв. При более низких взрывах ударная волна может сместить светящуюся область как целое и увеличить размер по горизонтали. В момент отражения давление во фронте скачкообразно увеличивается в несколько раз относительно давления в падающей волне.

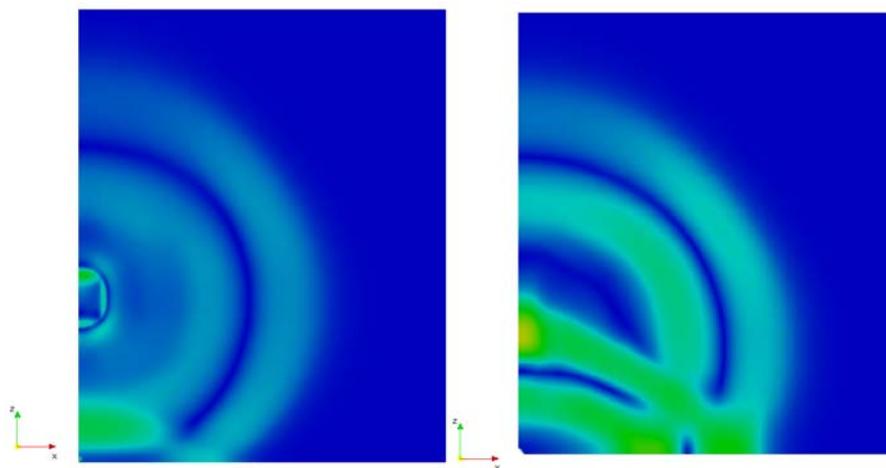


Рис. 21. Приземный взрыв, результаты моделирования

Распространение воздушной ударной волны, трансформация светящейся области в облако взрыва и дальнейший подъем облака с распределенными в нем радиоактивными продуктами взрыва сопровождаются крупномасштабными газодинамическими течениями и загрязнением среды в районе взрыва.

Для моделирования приземного взрыва была выбрана кубическая область. В области была построена гексаэдральная сетка (скошенные кубы) порядка 61 миллиона ячеек со сгущением в области взрыва. Для проведения расчетов сетка была разбита на 4096 доменов. Задача считалась с учетом радиационного и кондуктивного теплопереноса. Турбулентные потоки не учитывались. Начальную стадию взрыва в рамках газодинамической модели описать невозможно, поэтому в качестве начальных данных использованы справочные данные по взрывам в зависимости от массы взорванного вещества. Расчет процесса формирования и распространения ударной волны от приземного взрыва производился на суперкомпьютере «Ломоносов» на 4096 процессорных модулях.

**Моделирование распространения ударной волны от взрыва химического взрывчатого вещества в протяженном сооружении с нетривиальной геометрией.** Воздушная ударная волна химического взрыва вблизи границы раздела воздух-земля является фактором, к изучению которого имеется повышенный интерес в фундаментальных и прикладных исследованиях. Механическое действие воздушной ударной волны на объекты представляет собой многоплановую проблему из-за разнообразия самих объектов, различия условий воздействия и степени разрушения объекта в процессе воздействия. Действие ударной волны на преграду (объект) формирует динамическую нагрузку, которая определяется параметрами ударной волны, формой и размерами преграды, а также ориентацией преграды относительно вектора скорости движения фронта волны. Процесс взаимодействия ударной волны с преградой обычно разделяют на две характерные фазы:

- фаза дифракции (начальный период) – от момента соприкосновения фронта волны с преградой до установления сравнительно стабильного процесса обтекания преграды потоком сжатого воздуха;
- фаза квазистационарного обтекания – после окончания фазы дифракции до момента окончания действия положительной фазы волны на преграду.

При взрывах химического взрывчатого вещества возникают воздушные волны умеренной интенсивности, в которых нет интенсивного излучения, но может оказаться существенным турбулентный тепло- массоперенос.

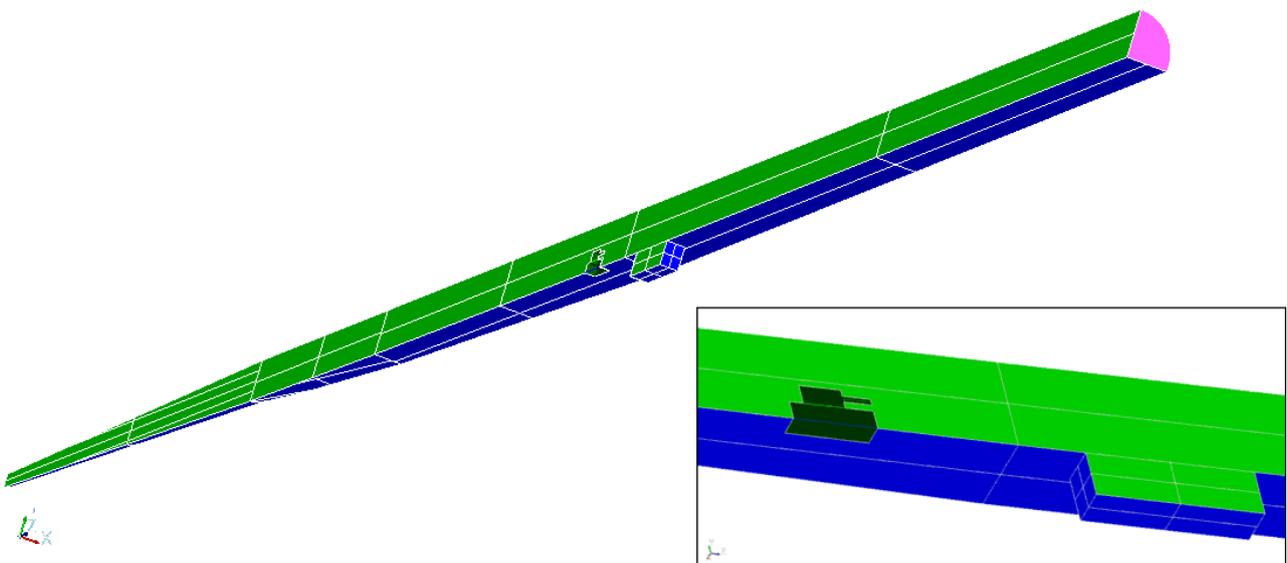


Рис. 22. Геометрия расчетной области

Для моделирования процесса формирования и распространения воздушной ударной волны в протяженном сооружении была построена принципиальная схема ударной трубы. Для рассмотрения процесса обтекания воздушной ударной волной препятствий в трубу помещен некоторый объект испытаний. Объект рассматривается как жесткое тело. В расчетной области была построена тетраэдральная сетка порядка 26 миллионов ячеек со сгущением вблизи области взрыва, в районе инженерной секции и вокруг объекта. Для проведения расчетов сетка была разбита на 4096 доменов. Решалась полная система уравнений магнитной газовой динамики с табличными уравнениями состояния с учетом турбулентной вязкости. Радиационный теплоперенос не учитывался. Расчет процесса формирования и распространения ударной волны в протяженном сооружении производился на суперкомпьютере «Ломоносов» на 4096 процессорных модулях.

**В качестве инструмента исследования** был выбран параллельный программный комплекс MARPLE3D [12]. Пакет прикладных программ MARPLE3D создан в ИПМ им. М.В.Келдыша РАН на основе численных методик с применением неструктурированных расчетных сеток. Предметной областью пакета являются задачи двухтемпературной радиационной магнитной

гидродинамики. Для таких задач на каждом временном шаге необходимо вычислять несколько десятков физических величин в каждой ячейке расчетной сетки. Для моделирования существенно трехмерных процессов с учетом реальной геометрии устройств необходимы расчетные сетки, содержащие от нескольких миллионов до десятков и сотен миллионов ячеек. Решение задач такой размерности возможно только с использованием распределенных вычислений на базе современных высокопроизводительных ЭВМ. Поэтому пакет MARPLE3D был сконструирован как программное обеспечение для высокопроизводительных параллельных вычислительных систем. Для выполнения расчетов на массивно-параллельных системах с распределенной памятью применяется известный подход геометрического параллелизма, поддержка которого встроена в дискретную модель MARPLE3D на уровне структур сеточных данных. В пакет MARPLE3D встроена внутренняя система журналирования, с помощью которой и производились измерения (фиксировалось астрономическое время выполнения операций на каждом вычислительном узле).

## Результаты

Вычисления производились на кластерах К-100, "Ломоносов" и «Helios».

На описанных выше физических задачах проведено тестирование разбиений, полученных методами разработанного программного комплекса (далее - GRIDSPIDERPAR), пакета PARMETIS, пакета ZOLTAN и пакета PT-SCOTCH. Сравнивалась эффективность параллельного счета физических задач при распределении сеток по процессорам (ядрам) в соответствии с различными разбиениями.

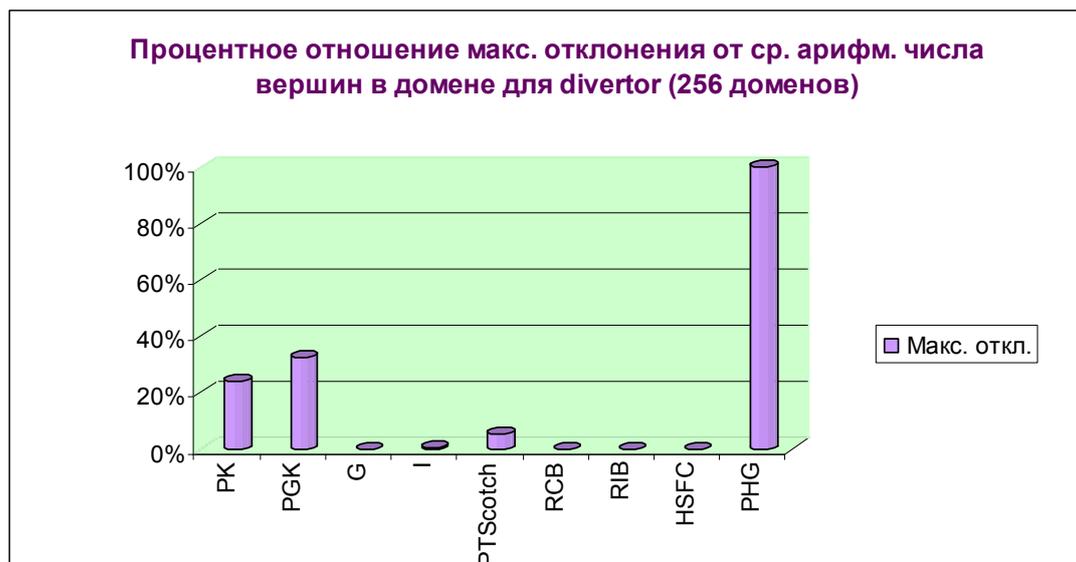
Для всех расчетных сеток были построены дуальные графы, учитывающие связи между ячейками через ребро. Число вершин в графах  $2.8 \cdot 10^6$  -  $6.1 \cdot 10^7$ , число ребер  $2.3 \cdot 10^7$  -  $5.4 \cdot 10^8$ .

Разбиения графов на домены были получены следующими методами: PartKway (на графиках обозначен PK) и PartGeomKway (на графиках PGK) пакета PARMETIS, методами GeomDecomp (параллельный алгоритм геометрической декомпозиции, на графиках – G) и IncrDecomp (параллельный инкрементный алгоритм, на графиках – I) программного комплекса GRIDSPIDERPAR, иерархическим диффузионным алгоритмом пакета PT-SCOTCH и методами RCB, RIB, HSFC и PHG пакета ZOLTAN. Методы PartKway и PartGeomKway пакета PARMETIS основаны на иерархическом алгоритме разбиения графов. В PartGeomKway выполняется предварительное геометрическое разбиение. Метод RCB пакета ZOLTAN основан на алгоритме рекурсивной координатной бисекции, метод RIB – на алгоритме рекурсивной инерциальной бисекции. Метод HSFC выполняет геометрическое разбиение с использованием кривой Гильберта. Метод PHG разбивает гиперграф.

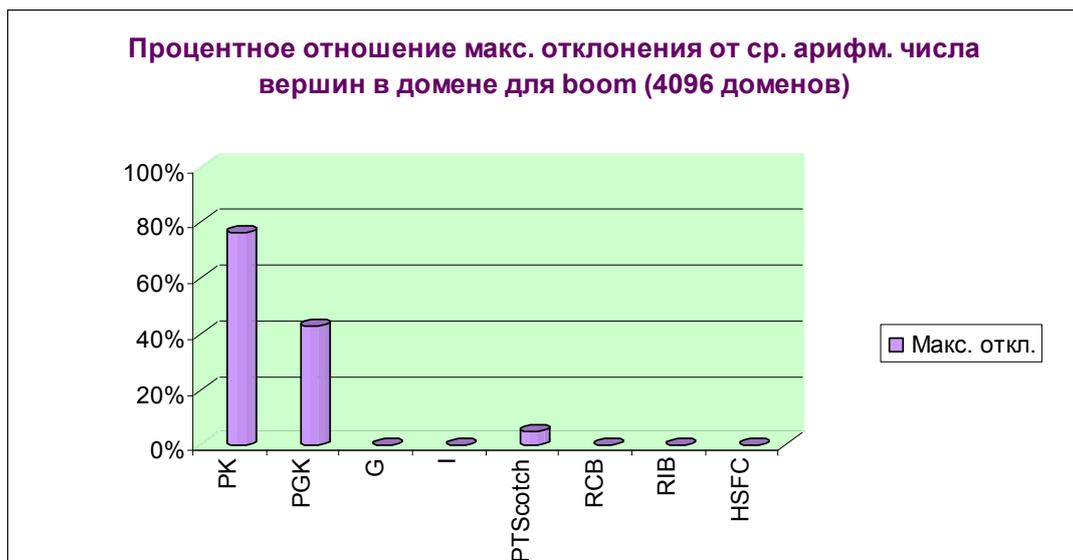
Далее в тексте приводятся краткие обозначения для физических задач и соответствующих им графов: divertor – моделирование газоплазменных потоков

в диверторе токамака, boom – моделирование распространения ударной волны от приземного источника взрывного типа, tube – моделирование распространения ударной волны от взрыва химического вещества в протяженном сооружении.

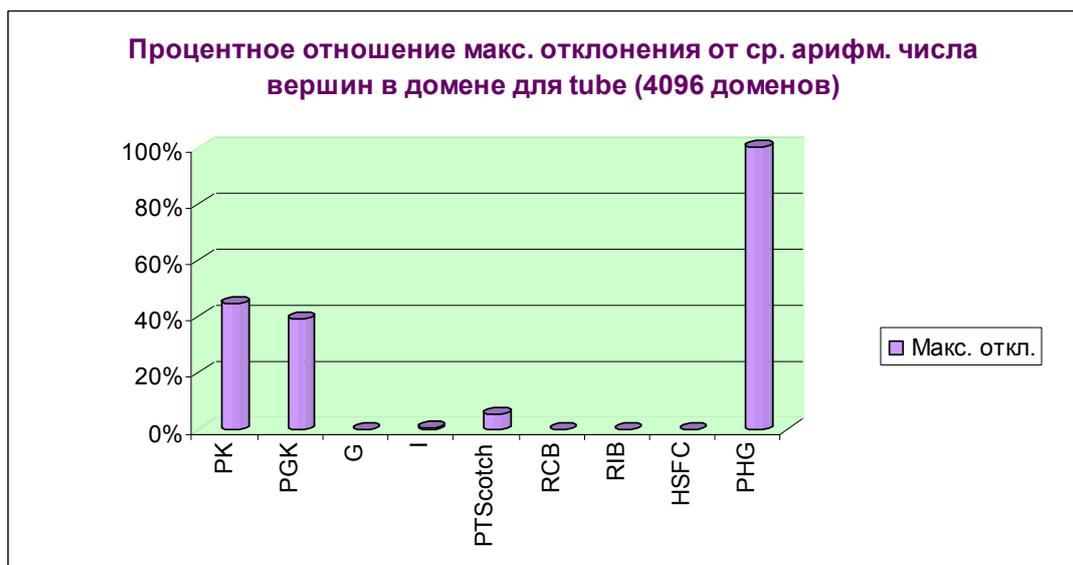
На рис. 23 – 25 представлено процентное отношение максимального модуля отклонения от среднего арифметического числа вершин в домене для разбиений графов. Наибольший дисбаланс числа вершин в доменах обнаружен в разбиениях, полученных методами PartKway и PartGeomKway пакета PARMETIS (до 80 %), наименьший (с разницей в одну вершину) – в разбиениях геометрическими методами GeomDecomp, RCB, RIB и HSFC. Пакетом PT-SCOTCH получены разбиения с отклонением от среднего арифметического порядка 5 %, методом IncrDecomp программного комплекса GRIDSPIDERPAR – до 0.1 %. В разбиениях, полученных методом PHG пакета ZOLTAN, присутствуют домены с нулевым числом вершин, поэтому вычисление физических задач на данных разбиениях не проводилось.



*Рис. 23.* Процентное отношение максимального модуля отклонения от среднего арифметического числа вершин в домене в разбиениях графа divertor на 256 доменов



*Рис. 24.* Процентное отношение максимального модуля отклонения от среднего арифметического числа вершин в домене в разбиениях графа boom на 4096 доменов



*Рис. 25.* Процентное отношение максимального модуля отклонения от среднего арифметического числа вершин в домене в разбиениях графа tube на 4096 доменов

На рис. 26–29 представлено число несвязных доменов. Во всех разбиениях, за исключением полученных методом HSFC пакета ZOLTAN, практически все домены оказались связными. Разбиения HSFC содержат до 40% несвязных доменов.

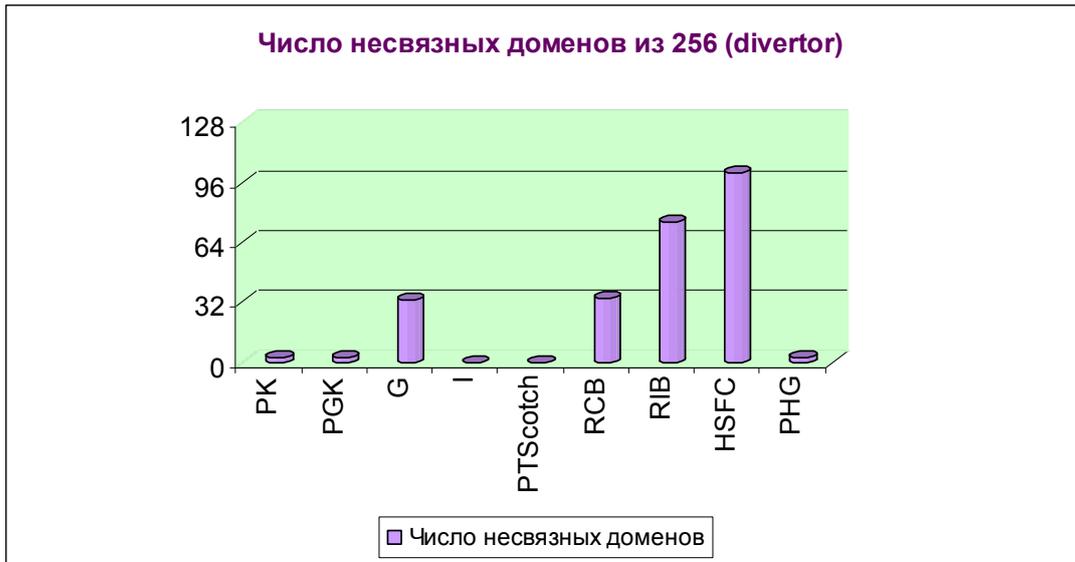


Рис. 26. Число несвязных доменов в разбиениях графа divertor на 256 доменов

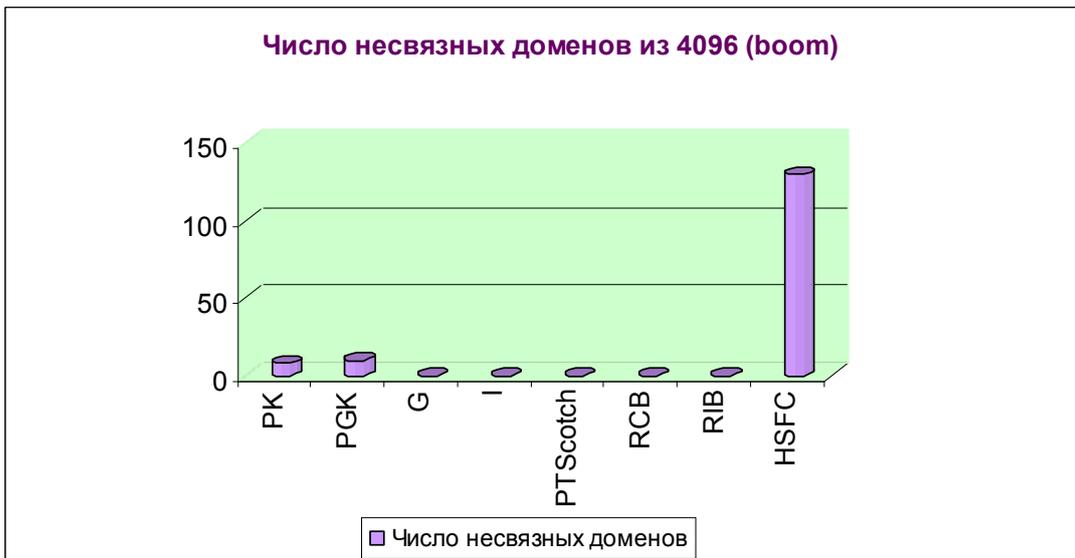


Рис. 27. Число несвязных доменов в разбиениях графа boom на 4096 доменов

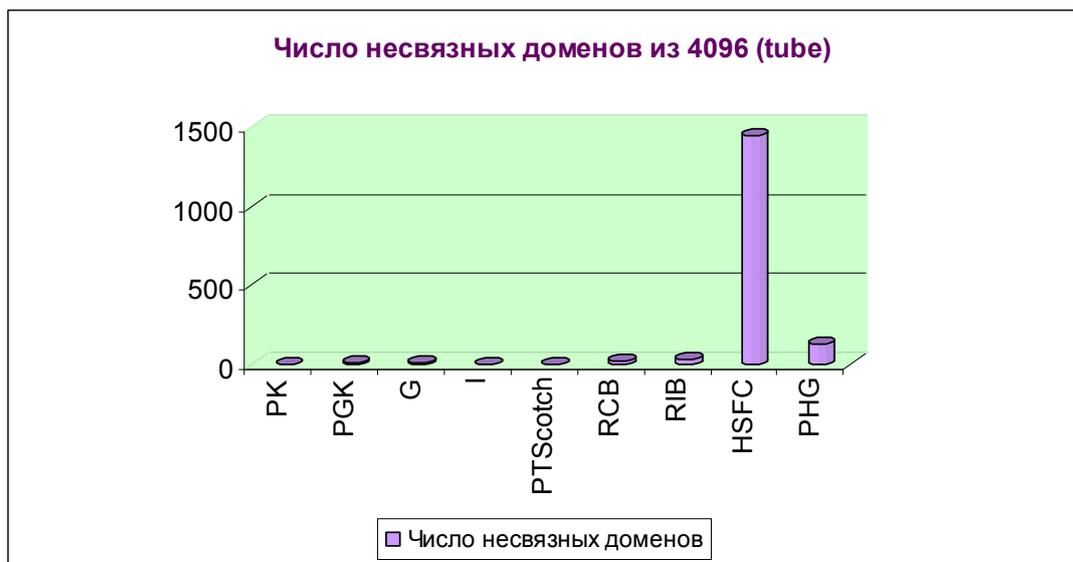


Рис. 28. Число несвязных доменов в разбиениях графа tube на 4096 доменов

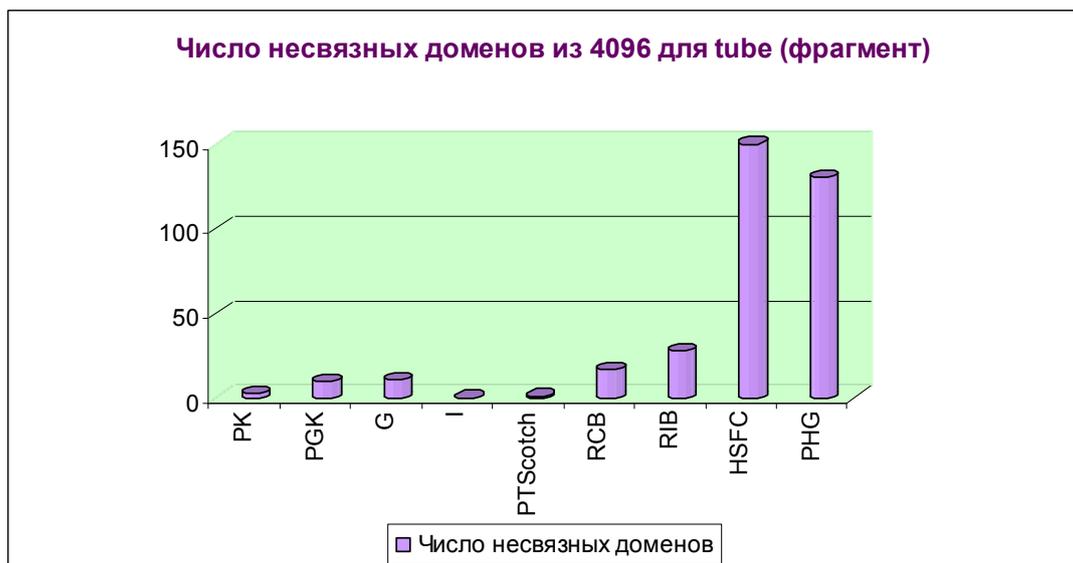


Рис. 29. Число несвязных доменов в разбиениях графа tube на 4096 доменов (фрагмент)

На рис. 30–32 представлено число разрезанных ребер между доменами. Наименьшее число разрезанных ребер получено методами PartKway и PartGeomKway пакета PARMETIS (для двух из трех графов). В разбиениях, полученных пакетом PT-SCOTCH, число разрезанных ребер немного больше, чем у методов PartKway и PartGeomKway пакета PARMETIS. Далее идут разбиения, полученные методом IncrDecomp комплекса программ GRIDSPIDERPAR. Методы геометрической декомпозиции GeomDecomp, RCB, RIB и HSFC, не учитывающие связи между вершинами, дают разбиения с большим числом разрезанных ребер, чем методы разбиения графов. При этом методом GeomDecomp комплекса программ GRIDSPIDERPAR получены разбиения с меньшим числом разрезанных ребер, чем аналогичным методом

RCB пакета ZOLTAN. Наибольшее и сильно выделяющееся число разрезанных ребер получено методом HSFC пакета ZOLTAN.

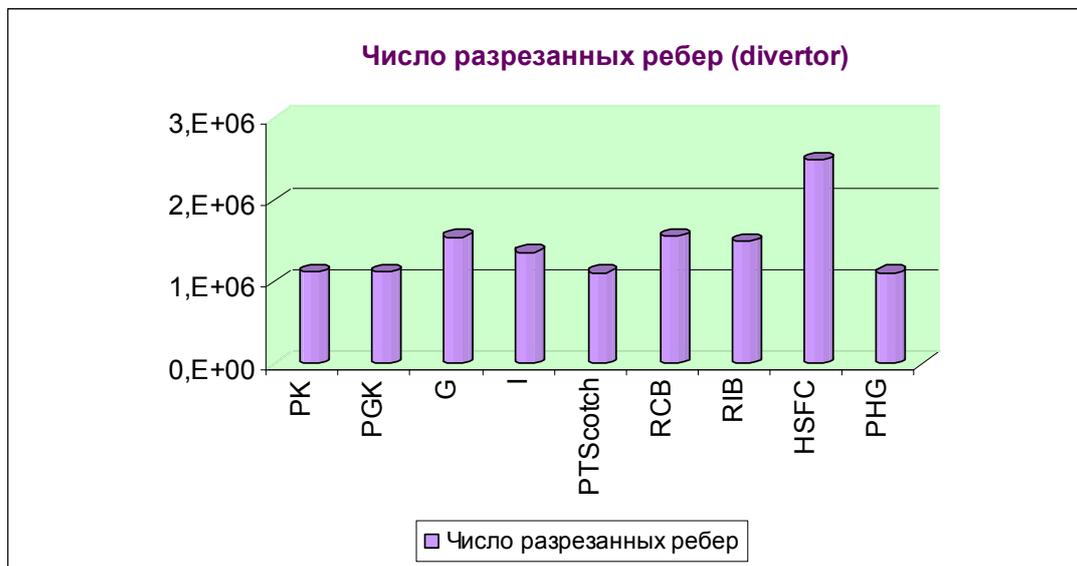


Рис. 30. Число разрезанных ребер в разбиениях графа divertor на 256 доменов

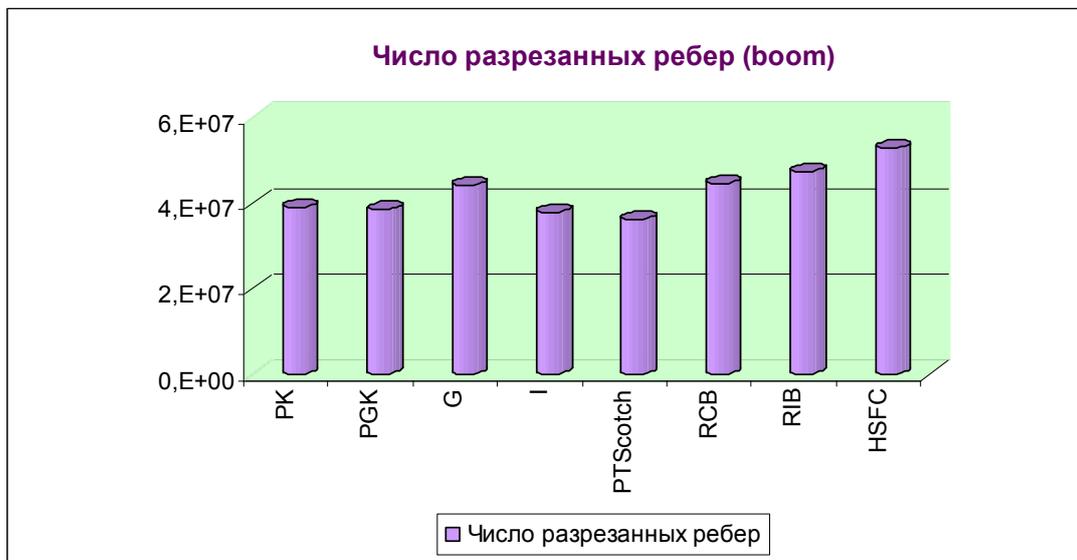


Рис. 31. Число разрезанных ребер в разбиениях графа boom на 4096 доменов

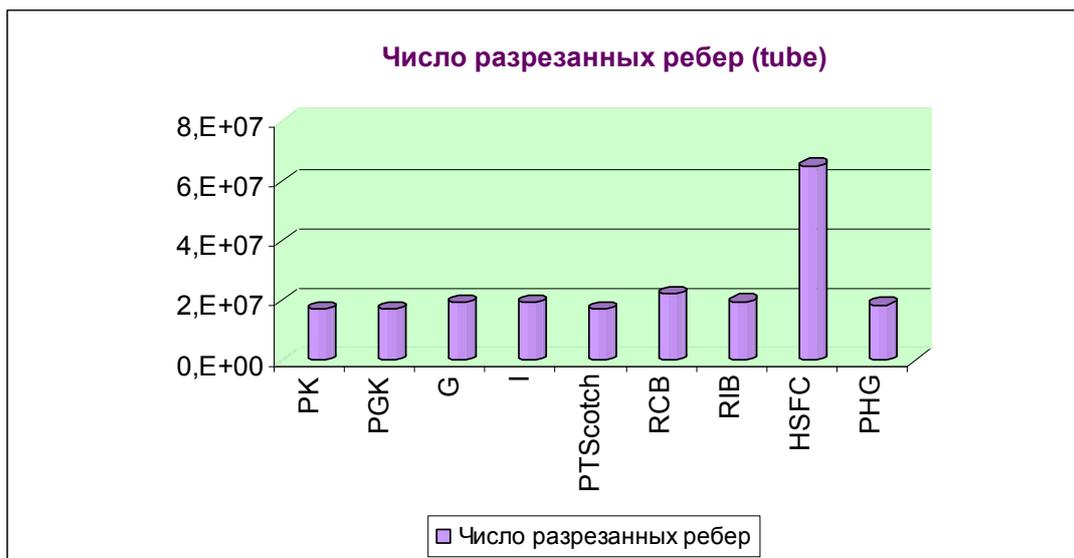


Рис. 32. Число разрезанных ребер в разбиениях графа tube на 4096 доменов

В соответствии с разбиениями дуальных графов были получены разбиения вершин сеток. Каждая физическая задача пускалась на одинаковое время на всех разбиениях. Были получены числа итераций и физические времена, до которых досчитали задачи. Распределения разбиений по числу итераций и физическим временам оказались аналогичными, поэтому ниже приводятся только распределения по физическим временам. Распределения представлены на рис. 33 – 36.



Рис. 33. Физические времена для divertor



Рис. 34. Физические времена для divertor (фрагмент)

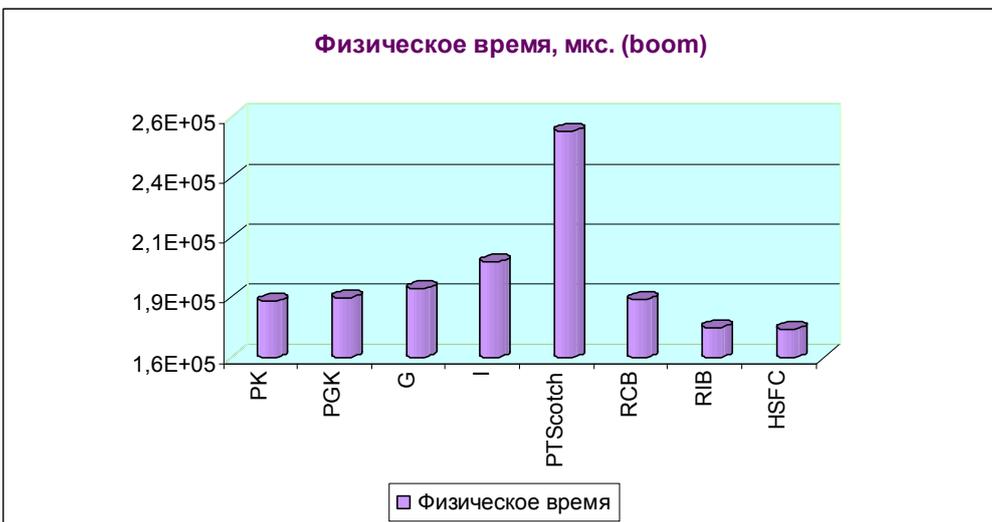


Рис. 35. Физические времена для boom

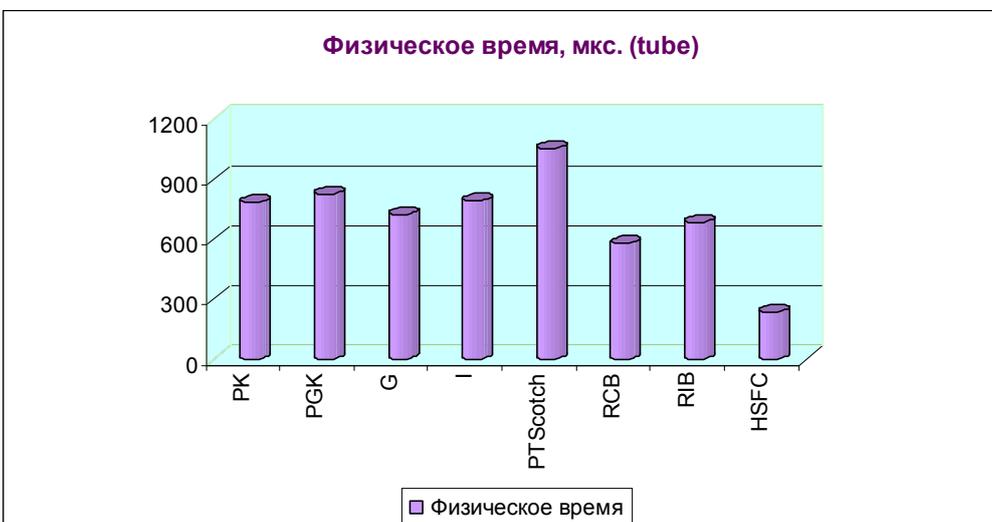


Рис. 36. Физические времена для tube

При сравнении получены следующие результаты. Наибольшие физические времена получены на разбиениях пакета PT-SCOTCH. Разбиения метода IncrDecomp комплекса программ GRIDSPIDERPAR дали результаты, превышающие аналогичные одного или всех двух методов пакета PARMETIS (PartKway и PartGeomKway). На разбиениях метода GeomDecomp комплекса программ GRIDSPIDERPAR полученные физические времена больше, чем на разбиениях похожего метода RCB пакета ZOLTAN, что обусловлено меньшим числом разрезанных ребер. Причем на двух сетках они больше даже, чем полученные на разбиениях методами пакета PARMETIS, что объясняется сильным дисбалансом числа вершин в доменах у разбиений пакета PARMETIS. На разбиениях HSFC пакета ZOLTAN, отличающихся очень большими числами несвязных доменов и разрезанных ребер, получены наименьшие физические времена. Разбиения пакета PT-SCOTCH имеют немного худший дисбаланс числа вершин в доменах, чем разбиения, полученные IncrDecomp, но число разрезанных ребер не намного больше, чем у алгоритмов пакета PARMETIS, что в итоге оказалось самым оптимальным для рассмотренных задач.

## **Заключение**

На модельных газодинамических задачах проведено тестирование разбиений расчетных сеток, полученных методами комплекса программ GRIDSPIDERPAR, пакета PARMETIS, пакета ZOLTAN и пакета PT-SCOTCH. Сравнивалась эффективность параллельного счета физических задач при распределении сеток по процессорам (ядрам) в соответствии с различными разбиениями. Каждая физическая задача пускалась на одинаковое время на всех разбиениях. Были получены физические времена, до которых досчитали задачи. Самыми оптимальными для рассмотренных задач оказались разбиения, полученные пакетом PT-SCOTCH. Разбиения метода IncrDecomp комплекса программ GRIDSPIDERPAR дали результаты, превышающие аналогичные одного, или всех двух методов пакета PARMETIS (PartKway и PartGeomKway). На разбиениях метода GeomDecomp комплекса программ GRIDSPIDERPAR полученные физические времена больше, чем на разбиениях похожего метода RCB пакета ZOLTAN.

Выигрыш пакета PT-SCOTCH относительно остальных алгоритмов вызывает вопросы, но следует отметить, что в нашем случае, когда численная модель включает большое количество различных по ресурсоемкости и организации вычислений солверов, а расчетная сетка имеет нерегулярную структуру, имеет место существенная пространственная неоднородность моделируемых физических процессов, баланс загрузки вычислительных ядер может нарушаться, поэтому точная теоретическая оценка эффективности разбиения затруднительна. В дальнейшем предполагается улучшение результатов метода IncrDecomp путем уменьшения числа разрезанных ребер между доменами. Также предполагается изучить влияние архитектуры

кластера, на котором проводится расчет, на результаты вычислительного эксперимента.

## Литература

1. Hendrickson B., Kolda T.G. Graph partitioning models for parallel computing // *Parallel Computing*. 2000. V. 26, P. 1519-1534.
2. Pelegrini F. A parallelizable multi-level banded diffusion scheme for computing balanced partitions with smooth boundaries // *ENSEIRB, LaBRI and INRIA Futurs, Universite Bordeaux I*.
3. Якобовский М.В. Инкрементный алгоритм декомпозиции графов // *Вестник Нижегородского университета им. Н.И.Лобачевского. Серия «Математическое моделирование и оптимальное управление»*. Нижний Новгород: Издательство ННГУ. 2005. Вып. 1(28). С. 243-250.
4. Hendrickson B., Leland R. A Multilevel Algorithm for Partitioning Graphs // *SAND93-1301, Unlimited Release, 1993*.
5. Boman E., Devine K., Catalyurek U., Bozdog D., Hendrickson B., Mitchell W.F., Teresco J. Zoltan: Parallel Partitioning, Load Balancing and Data-Management Services. Developer's Guide, Version 3.3 // Sandia National Laboratories, Copyright © 2000-2010, [http://www.cs.sandia.gov/Zoltan/dev\\_html/dev.html](http://www.cs.sandia.gov/Zoltan/dev_html/dev.html) .
6. Preis R., Diekmann R. PARTY – A Software Library for Graph Partitioning // *Advances in Computational Mechanics with Parallel and Distributed Processing, CIVIL-COMP PRESS, 1997, pp. 63-71*.
7. Кнут Д.Э. Искусство программирования, т.3.: Сортировка и поиск. - 2-е изд.: Пер. с английского – М.: Издательский дом «Вильямс», 2001.
8. Якобовский М.В. Параллельные алгоритмы сортировки больших объемов данных // *Фундаментальные физико-математические проблемы и моделирование технико-технологических систем: Сб. науч. тр., Выпуск 7 / Под ред. Л.А. Уваровой. – М.: Издательство "Янус-К", 2004. – С. 235-249*.
9. Vahala G., Vahala L., Morrison J., Krashennnikov S., Sigmar D. - *K-ε compressible 3D neutral fluid turbulence modeling of the effect of toroidal cavities on flame-front propagation in the gas-blanket regime for tokamak divertors // J. Plasma Physics, 1997, vol. 57, part 1, pp. 155-173*.
10. ITER - the way to new energy. - URL: <http://www.iter.org/>.
11. Зельдович Я.Б., Райзер Ю.П. Физика ударных волн и высокотемпературных гидродинамических явлений, М.: Наука, 1966.
12. Гасилов В.А. и др. Пакет прикладных программ MARPLE3D для моделирования на высокопроизводительных ЭВМ импульсной магнитоускоренной плазмы // *Матем. моделирование, 2012, Т. 24, № 1, С. 55–87*.

## Оглавление

Введение.....	3
Комплекс программ параллельной декомпозиции сеток.....	5
Инкрементный алгоритм декомпозиции графов.....	6
Параллельный инкрементный алгоритм декомпозиции графов.....	9
Параллельный алгоритм геометрической декомпозиции сеточных данных .....	13
Подготовка вычислительного эксперимента.....	16
Результаты.....	22
Заключение.....	30
Литература.....	31