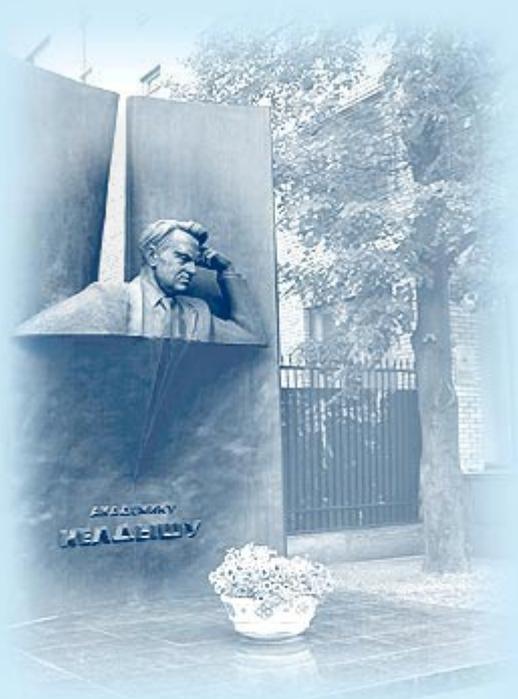




ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 86 за 2021 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

С.К. Григорьев, А.А. Бай,
Д.С. Бойков, Г.А. Багдасаров,
Ю.С. Шарова

Технология адаптивного
уточнения сетки для задач
механики сплошных сред

Рекомендуемая форма библиографической ссылки: Технология адаптивного уточнения сетки для задач механики сплошных сред / С.К. Григорьев [и др.] // Препринты ИПМ им. М.В.Келдыша. 2021. № 86. 19 с. <https://doi.org/10.20948/prepr-2021-86>
<https://library.keldysh.ru/preprint.asp?id=2021-86>

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

**С.К. Григорьев, А.А. Бай, Д.С. Бойков, Г.А. Багдасаров,
Ю.С. Шарова**

**Технология адаптивного уточнения
расчетных сеток для задач механики
сплошных сред**

Москва — 2021

С.К. Григорьев, А.А. Бай, Д.С. Бойков, Г.А. Багдасаров, Ю.С. Шарова

Технология адаптивного уточнения расчетных сеток для задач механики сплошных сред

Адаптивное уточнение сетки (Adaptive Mesh Refinement, AMR) широко применяется в настоящее время как средство решения задач механики сплошных сред, повышающее точность расчетов при экономии вычислительных ресурсов. В настоящей работе представлены новые структуры для хранения и обработки данных, предназначенные для адаптации сеток типа восьмеричное дерево в парадигме листовой модели. В текущей версии методики реализован набор алгоритмов, ориентированных на приложение к системам параллельных вычислений кластерного типа: динамическая адаптация сетки, синхронизация структуры данных при адаптации, балансировка нагрузки вычислительного комплекса. Разработанный инструментарий поддерживает функциональность, необходимую для реализации различных численных моделей механики сплошных сред. В качестве примера возможных приложений обсуждаются разностные схемы для уравнений теплопроводности и газовой динамики с использованием разработанной методики AMR. Представлены результаты численных экспериментов на модельных задачах.

Ключевые слова: Адаптивное уточнение сетки, балансировка нагрузки, проекционно-сеточная схема, интерфейс передачи сообщений.

S. K. Grigoriev, A. A. Bay, D. S. Boykov, G. A. Bagdasarov, Y. S. Sharova

Adaptive mesh refinement technique for continuum mechanics problems

Adaptive Mesh Refinement (AMR) is widely used today as a way to solve problems in the mechanics of continuous media, which makes it possible to increase the accuracy of the solution at an economical cost of computational resources. The paper presents new structures for storing and processing data for octree mesh adaptation in the leaf model paradigm. The current version of the technique implements a set of algorithms focused on applications to cluster-type parallel computing systems: dynamic mesh adaptation, data structure synchronization, and load balancing of the computing complex. The developed toolkit supports the functionality required for the implementation of various numerical models of continuum mechanics. As an example of possible applications, a difference schemes for heat conduction and gas dynamics using the developed AMR technique are discussed. The results of numerical experiments with model problems are presented.

Key words: Adaptive mesh refinement, load balancing, grid-projection scheme, message passing interface.

Работа выполнена при поддержке Российского научного фонда, проект 21-11-00362.

1. Введение

Технология Adaptive Mesh Refinement (AMR) предназначена для повышения точности компьютерного моделирования и снижения стоимости вычислительных ресурсов. Численное моделирование тепломассопереноса, гидродинамики, механики разрушения и т.д. связано с нелинейными коррелирующими процессами, которые существенно различаются по пространственным и временным масштабам. Необходимость моделирования подобных явлений стимулирует применение вычислительных технологий с использованием сеток нерегулярной структуры, среди которых сетки типа восьмеричное дерево (дерево октантов, англ. octree) являются наиболее известными и широко используемыми.

Этот подход был предложен более 40 лет назад [1] для задач гидродинамики и в настоящее время успешно используется как в компьютерной графике [2], так и в вычислительной физике [3]. В частности, AMR целесообразно использовать для расчета нестационарных процессов теплопередачи. Локальная адаптация вычислительных сеток может потребоваться для обеспечения отвода тепла от компактных источников и моделирования теплопередачи в сильно неоднородных средах, например, в сложных технологических структурах или анизотропных композитных материалах [4].

Такие вычисления выполняются на высокопроизводительных вычислительных кластерах с использованием геометрического параллелизма (декомпозиции областей) и технологии MPI для организации межпроцессорных обменов. В связи с этим возникают проблемы балансировки нагрузки, характерные для всех алгоритмов динамической адаптации и усложняемые необходимостью сохранения иерархической структуры сети при разделении на подобласти. Традиционно для этой цели используется заполняющая пространство кривая [5]. Мы предложили новые эффективные способы описания иерархических сеток, которые сохраняют структуру индексов и обеспечивают быстрый доступ к произвольному элементу. Разработаны новые алгоритмы статической (начальной) и динамической балансировки нагрузки вычислительных ядер на основе геометрического метода декомпозиции расчетной сетки.

Другой важной проблемой, возникающей при использовании AMR для численного решения уравнений и систем уравнений математической физики, является построение разностных схем с заданными свойствами на таких сетках. В частности, если речь идет о задачах теплопередачи, сеточные операторы должны не только обеспечивать аппроксимацию исходных уравнений с необходимой точностью, но и сохранять такие свойства дифференциальных операторов, как самосопряженность и положительная определенность, непрерывность тепловых потоков и др. законы сохранения. Для построения таких схем удобно использовать проекционные методы и метод конечных

элементов [6]. Эти методы широко используются для нерегулярных сеток, которыми могут быть назначены сетки типа восьмеричное дерево. Специальный выбор базисных функций позволил предложить новую формализованную процедуру построения схем заданного качества на сетках рассматриваемого типа для уравнения теплопроводности с коэффициентами тензорного типа.

2. Структуры данных для работы с сеткой

Широкое распространение обсуждаемой сеточной технологии в основном связано с простотой организации сетки (прямоугольная, с некоторым усложнением). Ниже описаны требования, которые использовались при разработке структур данных для работы с сетками восьмеричного дерева.

При генерации исходной регулярной сетки удобно использовать трехиндексную нумерацию ячеек (i, j, k) . Такая индексация позволяет на этапе первичной генерации расчетной сетки сразу вычислить всех соседей рассматриваемой ячейки. Такой расчет не означает реального существования ячейки, но позволяет построить исходный граф окрестностей ячеек. Необходимость в явном построении и хранении графа соседства ячеек обусловлена требованием быстрого доступа $O(1)$ к произвольному соседу рассматриваемой ячейки. Сохранение графа смежности ячеек в явном виде позволяет выполнять операции разделения и объединения ячеек также за $O(1)$ (указано без учета случая каскадного дробления). Операция разделения ячейки с учетом разделения соседних с рассматриваемой ячеек в соответствии со структурным критерием дробления, в общем случае, может быть $O(N)$, где N - общее количество ячеек сетки.

В качестве структурного критерия для дробления соседних ячеек необходимо соблюдать соотношение линейных размеров соседних элементов не более 1:2, т.е. уровни дробления соседних ячеек могут различаться не более, чем на 1.

Основными требованиями к сеточной структуре данных, которая работает с *o*stree-сеткой, являются: доступ к произвольному элементу сетки с оценкой не хуже, чем $O(1)$; обеспечение возможности быстрого доступа и обхода всех листовых и нелистовых элементов сетки; реализация доступа за $O(1)$ ко всем элементам, инцидентным заданной ячейке.

Кроме того, для создания решателей на основе методов динамической адаптации, а также генераторов соответствующих сеток необходимо предусмотреть операции разделения и обратного объединения ячеек (они применяются к набору ячеек, заданному перечислением или предикатом, и изменяют уровень дробления на единицу).

За исключением тривиального случая, когда структура прямоугольной сетки в точности соответствует трехмерному массиву и хранение значений

сеточных функций в таком массиве обеспечивает очевидное соответствие этих значений элементам сетки, для адаптивной динамически перестраивающейся сетки построить такой массив представляется затруднительным. При этом динамическая адаптация подразумевает появление или удаление сеточных элементов и требует соответствующего изменения в структуре данных, отвечающей за хранение сеточных функций. Для обеспечения возможности динамического перерасчета значений сеточных функций во время выполнения процедуры динамической адаптации, сеточные данные хранятся непосредственно в структурах, описывающих элемент сетки. Преимущество такой организации данных состоит в том, что доступ к сеточным значениям чрезвычайно прост и быстр, и, кроме того, соответствие между сеткой и набором значений не нарушается при изменении сетки. Поэтому для *ostree*-сеток этот метод сопоставления более привлекателен: *ostree*-сетка хранится в виде восьмеричного дерева ячеек, а физические данные хранятся в его листьях.

Таким образом, нам нужно сохранить индикацию, разделен ли элемент или нет, и, если он разделен, сохранить указатели на все дочерние ячейки, значение сеточных функций в элементе сетки, атрибут индикации границы и указатель на родительскую ячейку, если это необходимо.

Также реализована оригинальная разработка модифицированной кривой Мортон для геометрически произвольной области. Упорядочение по кривым заполнения пространства может использоваться для декомпозиции сетки, балансировки нагрузки и для некоторых вычислительных целей, например, при заполнении матрицы коэффициентов системы линейных алгебраических уравнений (СЛАУ) – обычно программы обращения матриц работают с массивами коэффициентов.

3. Параллельная реализация

Реализация параллельных вычислений в парадигме MPI предполагает декомпозицию адаптивных сеток, что связано с определенными трудностями из-за необходимости поддерживать правильную структуру сетки в распределенной системе. Кроме того, существует проблема балансировки нагрузки, поскольку при выполнении процедуры динамической адаптации количество сеточных элементов на разных MPI-процессах меняется, что естественным образом приводит к дисбалансу по вычислительной нагрузке.

При реализации балансировки нагрузки важно также учитывать необходимость поддержания вертикальных связей в структуре. Поля для межпроцессных взаимодействий формируются на ширину шаблона аппроксимации в терминах ячеек максимального размера. Единицей передачи информации при балансировке нагрузки является дерево ячеек, т.е. если хотя бы одна ячейка дерева попала на процесс в том или ином виде (как в качестве виртуальной, так и принадлежит текущему процессу), на процесс передаётся всё дерево, которому эта ячейка принадлежит.

Статическая начальная балансировка нагрузки выполняется с использованием индексной бисекции в трёхиндексном пространстве.

3.1. Синхронизация структуры при адаптации

При выполнении адаптации на распределённой вычислительной системе также возникает необходимость синхронизации структуры виртуальных ячеек между MPI-процессами. В силу того, что во время процедуры дробления и укрупнения сеточных элементов перерасчёт новых значений сеточных функций производится непосредственно во время изменения элементов, а также того, что для получения новых значений требуются значения сеточных функций из смежных элементов, независимая адаптация виртуальных ячеек не представляется возможной.

На основе алгоритма взаимодействий, описанного в предыдущем пункте, реализуется протокол обновления топологии, при котором каждый попарный обмен представляет собой обмен данными по принципу клиент-сервер, где один из процессов запрашивает состояние топологии своих виртуальных ячеек и получает в ответ сообщение, содержащее данные о состоянии структуры интересующих ячеек, а также всю необходимую информацию об изменении структуры, если такое изменение произошло.

Оценка количества независимых попарных обменов за одну итерацию выполнения сохраняется такой же, однако при большом количестве уровней дробления возможно каскадное дробление сетки в силу структурного критерия дробления, что приводит к необходимости множественных вызовов алгоритма синхронизации. В худшем случае оценка количества выполнений синхронизации структуры в том случае составляет $O(\min(p, \text{maxlevel}))$, где *maxlevel* – максимально допустимый уровень дробления ячеек.

Операция синхронизации данных содержит существенный дисбаланс по объёму передаваемых данных, обусловленный неравномерностью распределения изменяемых ячеек по процессам. В наихудшем случае количество передаваемых между различными процессами данных может отличаться в 64 раза (в случае, когда на одном из процессов все виртуальные ячейки были укрупнены, а на другом, наоборот, раздроблены).

3.2. Динамическая адаптация сетки

Процедура динамической адаптации выполняется в 3 этапа:

1. Вычисление значений функционального критерия адаптации в каждой листовой ячейке сетки;
2. Обход всех нелистовых элементов и укрупнение их при выполнении функционального критерия и отсутствии противоречий с другими критериями:
 - а. Обход всех нелистовых ячеек одного уровня;
 - б. Синхронизация полей виртуальных ячеек;
3. Обход всех листовых ячеек и дробление необходимых ячеек;

4. Синхронизация топологии.

Процедура адаптации запрещает изменение сеточного элемента более чем на 1 уровень за одно выполнение операции адаптации.

После завершения процедуры адаптации выполняется формирование сплошной глобальной нумерации листовых ячеек.

3.3. Балансировка нагрузки

Динамическая балансировка нагрузки выполняется сразу после адаптации расчетной сетки, таким образом, количество адаптаций совпадает с количеством балансировок. В дальнейшем подразумевается, что вызов алгоритма перестроения сетки автоматически вызывает алгоритм балансировки.

Балансировка нагрузки выполняется следующим образом:

- вычисление нового разбиения;
- формирование очереди обменов;
- обмен данными в соответствии со сформированной очередью:
 - сеанс попарных обменов балансируемыми элементами;
 - синхронизация виртуальных ячеек;
- очистка структур хранения от лишних элементов.

В силу необходимости поддержания межпроцессных связей в полях виртуальных ячеек на каждом этапе обменов данными возникает необходимость дополнительного сеанса обменов данными после каждого глобального попарного обмена балансируемыми элементами. Это приводит к общей оценке количества операций обменов данными на этом этапе в $O(p^2)$ для наихудшего случая. Наихудший случай предполагает такое неудачное распределение расчетной сетки по набору процессоров, при котором вся сетка разбита на последовательно прилегающие друг к другу слои, каждый из которых обрабатывается одним процессором, имеющим таким образом не более двух соседей. Используемые методы декомпозиции формируют иное трёхмерное разбиение сетки на фрагменты, в среднем имеющие $k \ll p$ соседей. В этом случае ожидаемая оценка числа операций составит $O(k^2)$, что не будет приводить к значительной деградации при большом числе процессоров. Дальнейшее снижение числа операций связано с пересмотром алгоритма формирования расписания обменов, что может быть выполнено на основе привлечения сокращённых сетей сортировки. В этом случае есть основания ожидать снижения числа выполняемых операций до величин $O((\log_2 p)^2)$ в худшем случае и $O(\log_2 p)$ в среднем, однако это требует дополнительных обоснований и проверки практикой.

3.4. Определение ячеек, перераспределяемых между процессорами на этапе динамической балансировки нагрузки

Для уменьшения дисбаланса нагрузки, возникающего в силу неравномерности измельчения расчетной сетки, применяется алгоритм, основанный на геометрической декомпозиции расчётной сетки [7, 8]. В критерии балансировки учитываются только листовые ячейки (ставится условие равномерности их распределения по процессорам), однако, средства используемой библиотеки позволяют построить отображение нового разбиения по процессам с учётом старого, таким образом, минимизируя, по возможности, количество передаваемых элементов на этапе перераспределения сетки.

3.5. Формирование глобальной нумерации листовых ячеек

Для реализации алгоритмов решения сеточных уравнений с использованием неявных схем может возникнуть необходимость сформировать сквозную нумерацию листовых ячеек. Построение нумерации листовых элементов происходит после завершения процедур адаптации и балансировки расчётной сетки. Глобальный номер листовой ячейки хранится как параметр ячейки. Вычисление номера производится следующим образом: сначала нумеруются листовые ячейки на текущем процессе, затем формируется глобальный массив смещений нумерации и, наконец, вычисляется глобальный номер листовой ячейки.

Никакого дополнительного упорядочения ячеек при построении нумерации не выполняется, и номера присваиваются листовым ячейкам в порядке их хранения в соответствующих структурах данных.

Построение сквозной нумерации выполняется сразу после выполнения балансировки нагрузки.

4. Реализация проекционной схемы для уравнения теплопроводности

Проекционно-сеточные схемы для уравнений, содержащих дифференциальные операторы второго порядка, строятся методом Бубнова-Галеркина, как описано в [9]. Запишем уравнение второго порядка в обобщенной форме

$$c \frac{\partial u}{\partial t} = \nabla \bullet (k \nabla \circ u) + fu \quad (1)$$

где \bullet и \circ – бинарные операторы умножения вектора на число, скалярного или векторного произведения векторов, $u = u(\vec{r}, t)$ – неизвестная скалярная или векторная функция, $f = f(\vec{r}, t)$, $c = c(\vec{r})$ – заданные функции, $\kappa = \kappa(\vec{r}, t)$ – скалярный или тензорный коэффициент.

Комбинируя значения операторов \bullet и \circ , можно получить ($\text{div } \kappa \text{ grad } u$), ($\text{grad } \kappa \text{ div } \mathbf{u}$) и ($\text{rot } \kappa \text{ rot } \mathbf{u}$) в правой части. Вспомогательная функция v и уравнение (1) запишутся в форме:

$$v = \nabla \circ u$$

$$c \frac{\partial u}{\partial t} = \nabla \bullet \kappa v + f u$$

Используя описанные в [9] базисные функции и соответствующие процедуры интегрирования, получим обобщенную форму записи итоговой схемы для листовой ячейки с номером i .

$$C_i V_i \frac{\partial u_i}{\partial t} = - \sum_{k \in \text{In}30(i)} \sum_{j \in \text{In}03(k)} \vec{S}_j(p_k) \bullet \frac{\kappa_k \vec{S}_i(p_k)}{V(p_k)} \circ u_j + f_i u_i V_i$$

Здесь V_i – объем i -й ячейки, $\vec{S}_i(p_k)$ – приузловая площадь узла p_k в ячейке i , $\text{In}30(i)$ – множество узлов ячейки i , $\text{In}03(k)$ – множество ячеек, содержащих узел k .

4.1. Полученные результаты

Для тестирования данной схемы решалась задача о распространении тепловой волны [10, стр. 448]

$$\frac{\partial T}{\partial t} + \varepsilon \frac{\partial^2 T}{\partial t^2} = \frac{\partial}{\partial x} \left(k_0 T^\alpha \frac{\partial T}{\partial x} \right) \quad (2)$$

Начальные условия:

$$T(x, t = 0) = \begin{cases} \left[\frac{\alpha V}{k_0} (s_0 - x) \right]^{\frac{1}{\alpha}}, & x \leq s_0 \\ 0, & x > s_0 \end{cases}$$

Граничные условия:

$$T(x = 0, t) = \left[\frac{\alpha V}{k_0} (Vt + s_0) \right]^{\frac{1}{\alpha}}, \quad t > 0$$

Автомодельное решение:

$$T(x, t) = \begin{cases} \left[\frac{\alpha V}{k_0} (Vt + s_0 - x) \right]^{\frac{1}{\alpha}}, & x \leq s_0 + Vt \\ 0, & x > s_0 + Vt \end{cases}$$

Параметры: $s_0 = 0.5$, $k_0 = 0.5$, $\alpha = 2.0$, $V = 5.0$.

На рис. 1 показано численное решение на момент времени $t_{\max} = 0.4$. Справа – фронт тепловой волны в увеличенном масштабе.

Красная кривая – решение на равномерной сетке 50 узлов $h = 0.06$.

Синие точки – решение на равномерной сетке 100 узлов $h = 0.03$.

Зеленая кривая – решение на адаптивной сетке. Использовался критерий дробления-слияния на основе известного аналитического решения. В окрестности фронта тепловой волны $x_0 = s_0 + Vt$ с шагом по времени

$\Delta t_{\text{адапт}} = 0.03$ для всех ячеек $x_0 - 0.2 < x < x_0 + 0.2$ выполнялось одноуровневое дробление исходной равномерной сетки 50 узлов $h = 0.06$.

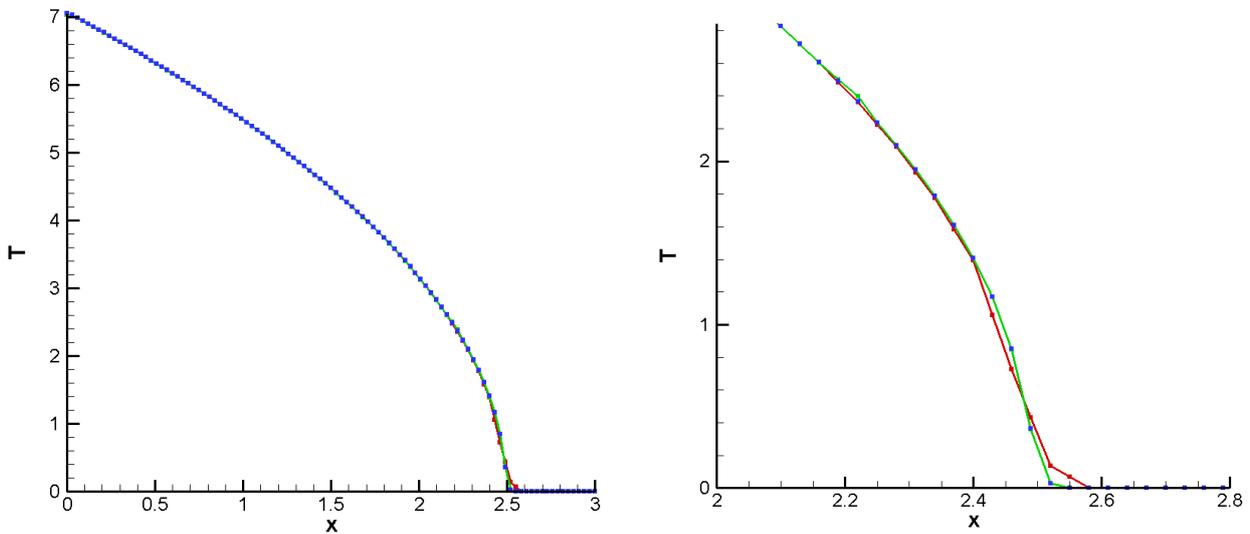


Рис. 1 Решение задачи о распространении тепловой волны.

Локальное измельчение сетки в области большого градиента вблизи фронта тепловой волны обеспечивает такую же точность решения, как и равномерная сетка с вдвое меньшим шагом.

Одномерный расчет выполнялся по трехмерной программе на трехмерной сетке, дробление ячеек выполнялось по всем трем координатам (на 8 ячеек), см. рис. 2.

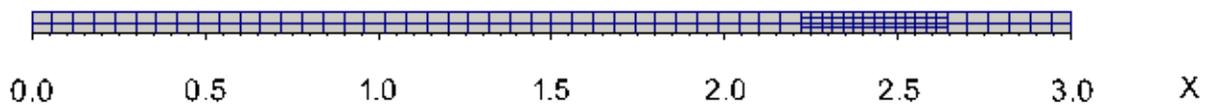


Рис. 2 Пример раздробленной сетки.

Во всех трех случаях выполнялось решение гиперболического [11] уравнения (2) по трехслойной явной схеме для интегрирования по времени. Запись уравнения теплопроводности в такой форме позволяет существенно (по сравнению с классическим параболическим уравнением) повысить шаг по времени для явной схемы с $\Delta t \sim h^2$ до $\Delta t \sim h^{3/2}$. Использование явной схемы дает большие преимущества при расчетах на сетках древовидной структуры, особенно в многопроцессорном режиме, поскольку позволяет избежать решения распределенных систем линейных уравнений и сложных итерационных процедур, возникающих при реализации неявных по времени схем. В данном расчете $\varepsilon = 0.005$, $\Delta t = 3 \cdot 10^{-4}$. Максимальное допустимое значение параметра ε : $\varepsilon_{\text{max}} = h/V = 0.006$. Оценка из максимального допустимого

$$\Delta t \leq \sqrt{\frac{\varepsilon}{\kappa}} h = 4.2 \cdot 10^{-4}$$

значения шага по времени из линейной теории:

5. Реализация схемы для уравнений газовой динамики

Для численного решения использовалась консервативная форма уравнений газодинамики:

$$\partial_t U + \partial_x F + \partial_y G + \partial_z H = 0,$$

где $U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}$ $F = \begin{pmatrix} \rho \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{pmatrix}$ $G = \begin{pmatrix} \rho \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (E + p)v \end{pmatrix}$ $H = \begin{pmatrix} \rho \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (E + p)w \end{pmatrix}.$

Система дополняется уравнением состояния и уравнением полной энергии. Уравнение состояния для идеального газа имеет вид:

$$p = (\gamma - 1)\rho\epsilon, \quad E = \rho\epsilon + \rho \frac{(u^2 + v^2 + w^2)}{2},$$

где ϵ – удельная внутренняя энергия.

Все физические величины вычисляются в центрах ячеек, с помощью метода наименьших квадратов [12] или метода Piecewise Parabolic Method (PPM) [13] проводится интерполяция величин на грани, далее решается задача Римана о распаде разрыва с помощью метода Harten–Lax–van Leer Discontinuities Contact (HLLC) [14]. Для дискретизации уравнений Эйлера используется схема Годунова [15] и схема предиктор-корректор.

Интерполяция методом наименьших квадратов заключается в построении полинома заданного порядка, минимизирующего следующее соотношение:

$$E(P_j) = \sum_{i \in \mathcal{N}_j} \left(\frac{1}{V_i} \int_{\omega_i} P_j(x, y, z) dx dy dz - Q_i \right)^2,$$

где $P_j(x, y, z)$ – полином для j -ой ячейки, \mathcal{N}_j – некоторый набор ячеек, по которым строится полином в j -ой ячейке, ω_i – i -ая ячейка, Q_i – значение в i -ой ячейке, V_i – объем i -ой ячейки. В схеме используется интерполяция полиномом первого порядка, в шаблон интерполяции \mathcal{N}_j входят все ячейки, которые имеют общую грань с j -ой ячейкой.

Расчеты проводятся на сетке с ячейками, которые имеют разные уровни дробления. Для метода PPM необходимо брать соседей одного уровня дробления, поэтому, если сосед интересующей нас ячейки раздроблен, необходимо заранее найти значения физических величин в центре родительской ячейки соседа. Такое же правило действует и в обратную сторону: если интересующая нас ячейка имеет уровень дробления больше, чем ее сосед, необходимо заранее подготовить значения в дочерних ячейках соседа. Для этого применяются следующие формулы [16]:

- Вычисление физических величин в ячейках после дробления

$$U_{l,m,n}^{l+1} = U_{i,j,k}^l + \sum_{d=1}^3 \frac{v_d^{l+1} - v_d^l}{v_{d,+}^l - v_{d,-}^l} \Delta_d U_{i,j,k}^l,$$

где v_d^l – координата центра ячейки на уровне дробления l в направлении d , $v_{d,+/-}^l$ – координаты центров левой и правой граней ячейки, $\Delta_d U_{i,j,k}^l$ – центральная разность. Во избежание появления новых локальных экстремумов, использовался лимитер MinMod.

- Вычисление физических величин в ячейках после слияния

$$U_{i,j,k}^l = \sum_{m=0}^7 \frac{v_i^{l+1} U_m^{l+1}}{v_{i,j,k}^l},$$

Для получения новых значений на уровне дробления l консервативные решения на уровне $l+1$ суммируются и усредняются. $v_{i,j,k}^l$ – объем ячейки на уровне дробления l .

Для получения разности потоков у ячейки с гранями разного уровня дробления применяется формула:

$$\delta F_d^l = -A_{d,-}^l F_{d,-}^l + \sum_{i=0}^7 A_{d,+}^{l+1} F_{d,i}^{l+1},$$

где i – индекс ячейки с уровнем дробления $l+1$, $A_{d,+/-}^{l+1}$ – площади левой и правой граней ячеек.

Потоки из ячеек с уровнем дробления $l+1$ в ячейку с меньшим уровнем дробления l не должны нарушать консервативность (см. рис. 3).

После чего разность потоков учитывается в уравнении для получения новых значений на следующем временном слое:

$$U_{i,j,k}^{t+1} = U_{i,j,k}^t - \Delta t \frac{\delta F_d^l}{v_{i,j,k}},$$

где $v_{i,j,k}$ – объем ячейки.

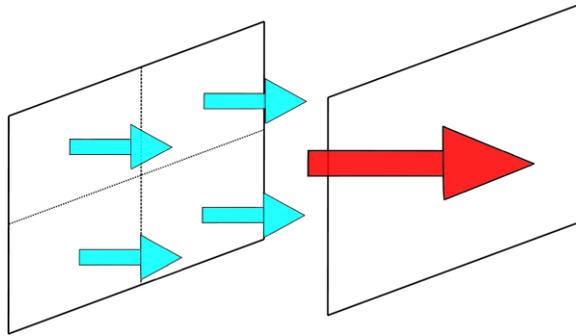


Рис.3 Потоки из ячеек с уровнем дробления $l+1$ в ячейку с уровнем дробления l

Для адаптивного уточнения сетки реализован критерий дробления [16], который использует отношение второй производной к первой производной:

$$s(U) = \sqrt{\frac{\sum_d |\Delta_{d,+1/2}\sigma - \Delta_{d,-1/2}\sigma|^2}{\sum_d (|\Delta_{d,+1/2}\sigma| + |\Delta_{d,-1/2}\sigma| + \varepsilon \sigma_{d,ref})^2}},$$

где $\sigma \equiv \sigma(U)$ – это функция консервативных величин, $\Delta_{d,\pm 1/2}\sigma$ – дифференциальная разность «вверх» и «вниз по потоку» в направлении d , $\Delta_{d,\pm 1/2}\sigma = \pm(\sigma_{i\pm 1} - \sigma_i)$. Последний член в знаменателе $\sigma_{d,ref}$ действует как фильтр и предотвращает объединение ячеек в областях малых возмущений, $\sigma_{d,ref} = |\sigma_{i+1}| + 2|\sigma_i| + |\sigma_{i-1}|$. Константа $\varepsilon=0.01$. Такие же выражения используются, когда $d = y$, $d = z$.

Если максимальная величина отношения $s(U) > s_2$, то ячейка помечается для дробления, если $s(U) < s_1$, то к ячейке следует применить слияние. Константы $s_1 = 0.2, s_2 = 0.8$. Частота проверки критерия дробления зависит от решаемой задачи и может меняться.

5.1. Полученные результаты

В качестве тестовой решалась задача о распространении ударной волны, возникшей в результате сильного взрыва (моментального выделения большого количества энергии в малом объеме) – задача Седова, имеющая автомодельное решение [17].

Задача рассматривается в плоской постановке в области $[0, 150] \times [0, 1] \times [0, 1]$. В начальный момент времени во всей области расчета плотность и скорость одинаковы $\rho = 1.0, v = 0.0$, давление везде, кроме области выделения энергии, $p = 10^{-4}$. Выделение энергии $E = \rho \varepsilon V = \frac{p}{(\gamma-1)} V = 100$ происходит мгновенно в области $[0, 1] \times [0, 1] \times [0, 1]$.

Было выполнено 3 варианта расчета:

- 1) на равномерной сетке $150 \times 1 \times 1$ без дробления;
- 2) на равномерной сетке $600 \times 4 \times 4$ без дробления;
- 3) на сетке $150 \times 1 \times 1$ с 2 уровнями дробления на фронте волны.

Для всех расчетов число Куранта $q = 0.1, \gamma = 1.4$.

На рис. 4–6 приведены результаты расчета на сетке $600 \times 4 \times 4$ на момент времени $t=125$.

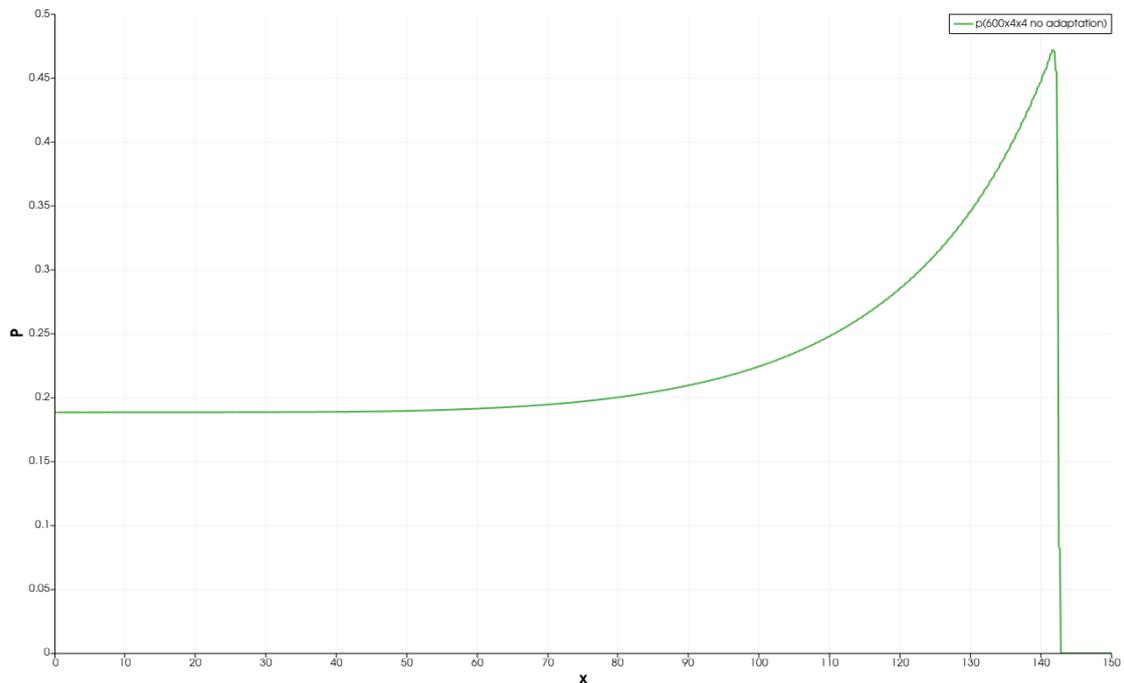


Рис. 4 Профиль давления.

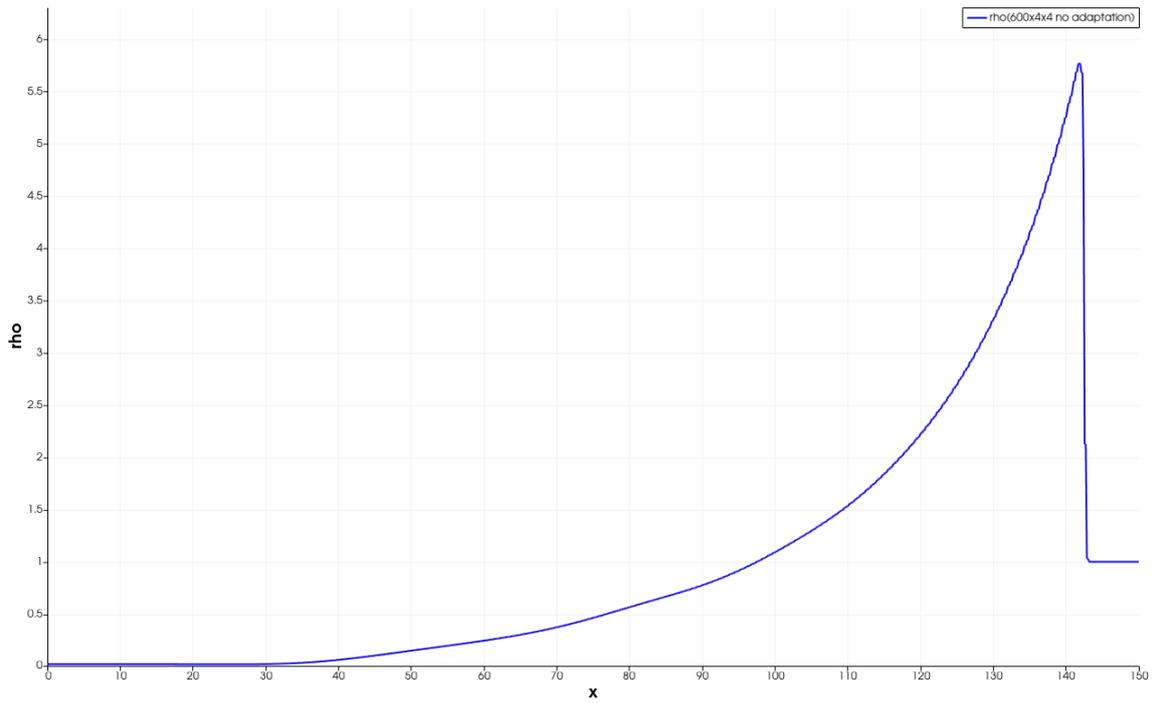


Рис. 5 Профиль плотности.

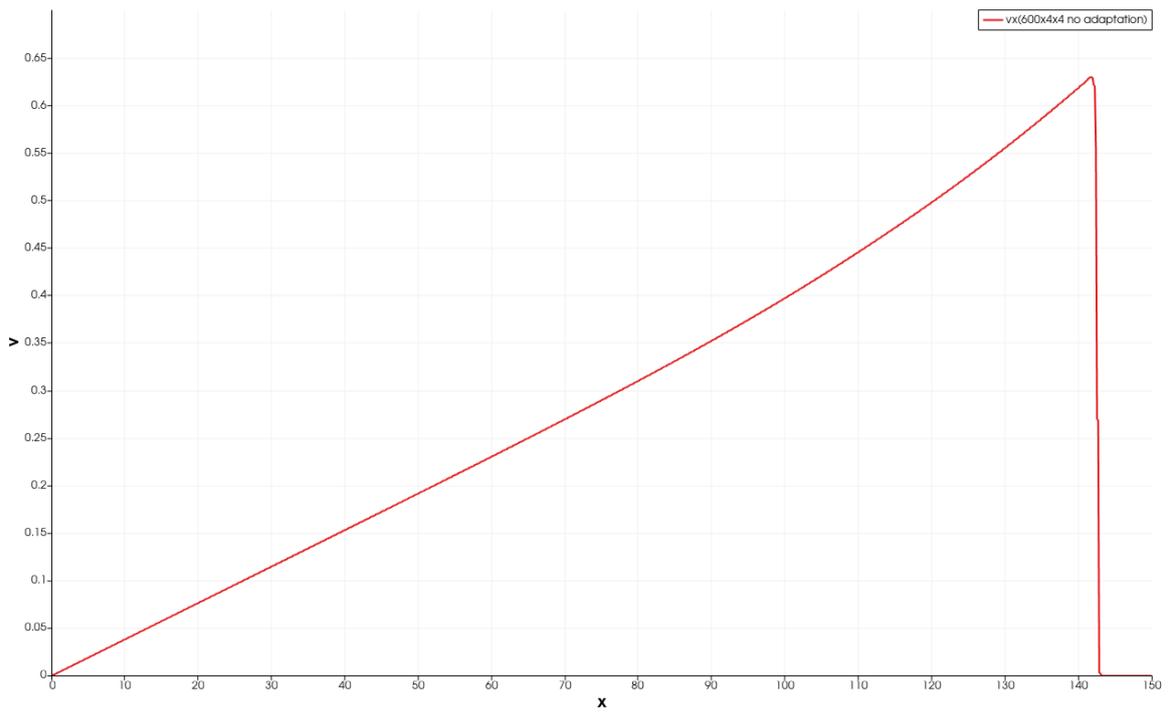


Рис. 6 Профиль скорости.

Полученный в результате расчета закон движения фронта волны асимптотически стремится к аналитическому решению [17] $x(t) t^{2/3}$, как показано на рис. 7.

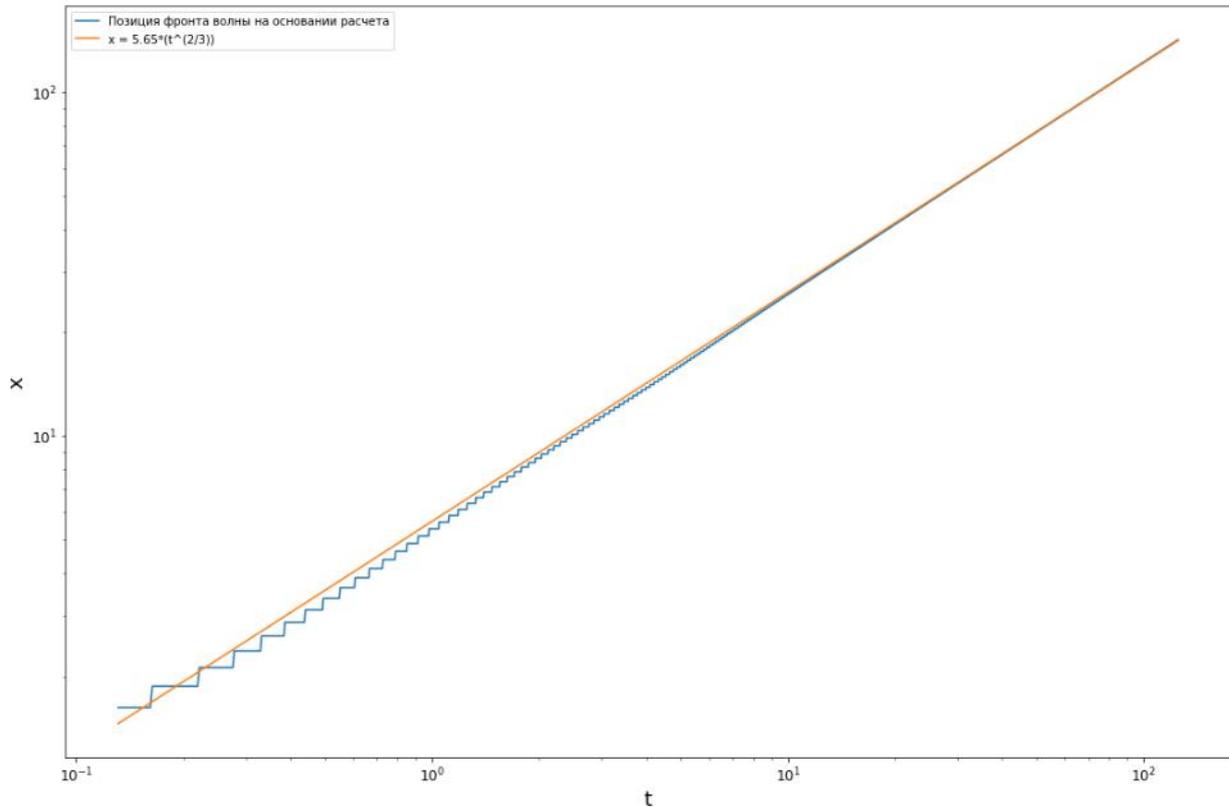


Рис. 7 Координата фронта ударной волны в зависимости от времени.

Адаптация на фронте ударной волны производилась следующим образом. До первого уровня дробления адаптировалась область $5.65t^{2/3} - 7 < x < 5.65t^{2/3} + 6$. До второго уровня дробления адаптировалась область $5.65t^{2/3} - 4 < x < 5.65t^{2/3} + 4$. Адаптация производилась с интервалом $\Delta t_a = 0.5$.

На рис. 8 приведены результаты расчета на момент времени $t=125$.

Красная кривая – решение на равномерной сетке 150 узлов $h = 1$.

Синяя кривая – решение на равномерной сетке 600 узлов $h = 0.25$.

Зеленая кривая – решение на адаптивной сетке.

Расчет на мелкой сетке занял 2859 секунд, на крупной сетке с адаптацией – 737 секунд. При этом, как видно на рис. 8, решение на сетке с динамической адаптацией воспроизводит фронт ударной волны с такой же точностью, как решение на мелкой сетке.

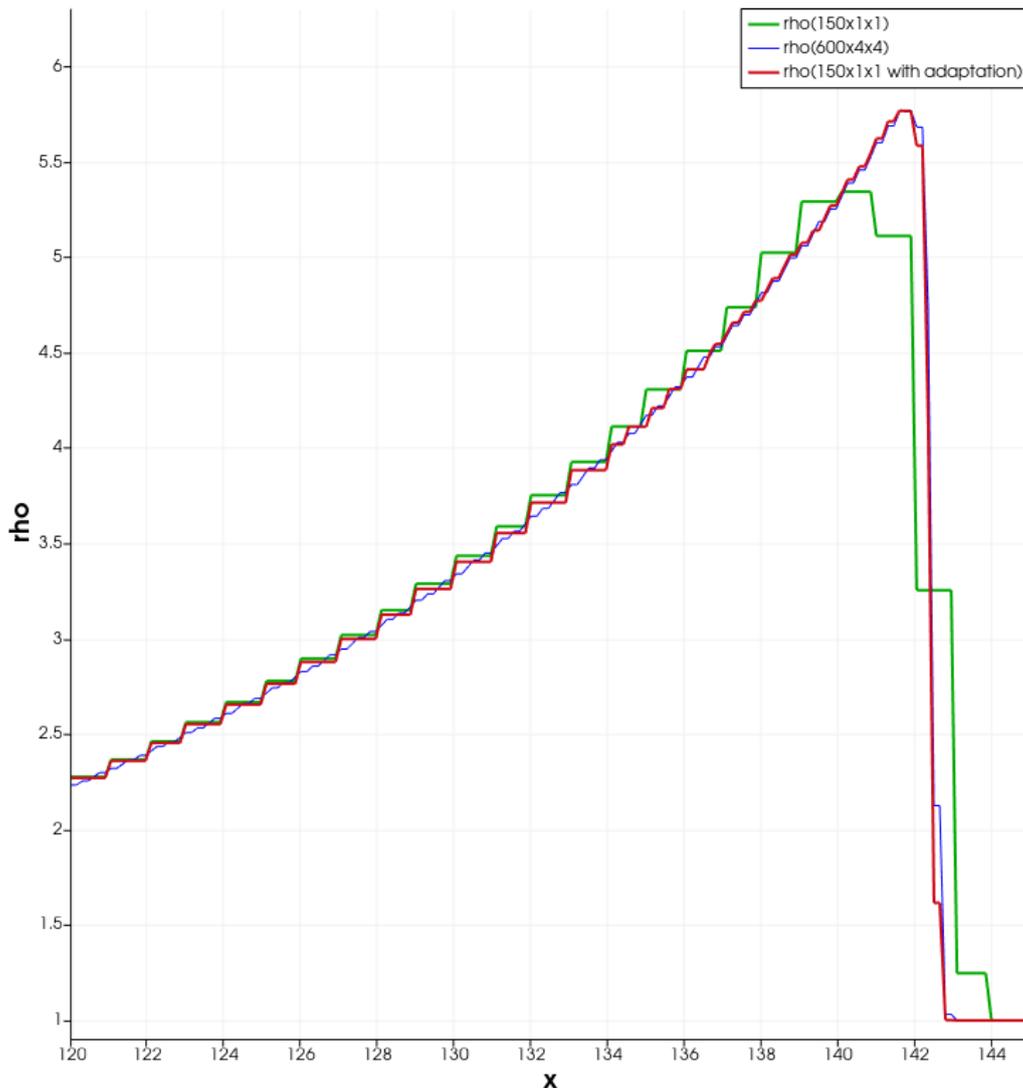


Рис. 8 Сравнение профилей плотности около фронта волны для расчетов на разных сетках.

6. Заключение и план дальнейшей работы

Техника, представленная в этой статье, обеспечивает стратегию распараллеливания приложений с локально-адаптивной сеткой. Предложен эффективный способ хранения сеточных данных, динамические алгоритмы для адаптивного уточнения сетки и балансировки нагрузки. Наш подход обеспечивает относительно простой для разработчика переход от последовательного приложения к параллельному MPI-приложению, поскольку фреймворк требует лишь небольшого числа дополнительных входных данных. Работа наших сеточных структур была протестирована на численном решении типовых задач математической физики: уравнение теплопроводности с нелинейным коэффициентом и уравнения газовой динамики. Показано, что разработанная технология позволяет приложениям, выполняемым на стороне клиента, повышать точность решения с помощью адаптивного уточнения сетки.

Наш алгоритм балансировки нагрузки обеспечивает практическую равномерность загрузки вычислительных ядер. Масштабируемость алгоритмов адаптации и созданного программного обеспечения в целом требует тщательного изучения и оптимизации. Эти вопросы являются одним из важнейших планируемых направлений нашей работы. Мы считаем, что с очень небольшими изменениями мы сможем улучшить производительность нашей платформы. Дальнейшая работа будет направлена на создание инфраструктуры, позволяющей подключать сторонние алгоритмы балансировки нагрузки, и адаптацию структуры сеточных данных под конкретную архитектуру процессора.

7. Библиографический список

1. Berger M. J., Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Comput. Phys.* **82**, 64–84 (1989) [https://doi.org/10.1016/0021-9991\(89\)90035-1](https://doi.org/10.1016/0021-9991(89)90035-1)
2. Luebke D.P. *Level of Detail for 3D Graphics*. Morgan Kaufmann Publishers Inc., San Francisco (2003).
3. Peraire J., Patera A.T. Bounds for linear-functional outputs of coercive partial differential equations: local indicators and adaptive refinement. In: *Advances in Adaptive Computational Methods in Mechanics 47 Studies in Applied Mechanics*, pp. 199–216 (1998).
4. Boyard N. *Heat Transfer in Polymer Composite Materials: Forming Processes*. John Wiley & Sons, Inc., Hoboken (2016) DOI:10.1002/9781119116288
5. Sagan H. *Space-Filling Curves*. Springer, New York (1994). <https://doi.org/10.1007/978-1-4612-0871-6>
6. Fletcher C. A. J. *Computational Galerkin Methods*. Springer, Heidelberg (1984). <https://doi.org/10.1007/978-3-642-85949-6>
7. Головченко Е.Н. Параллельный пакет декомпозиции больших сеток // *Математическое моделирование*, 23, 10, 3-18, 2011.
8. Головченко Е.Н., Якобовский М.В. Пакет параллельной декомпозиции больших сеток GridSpiderPar // *Вычислительные методы и программирование*, 16, 507-517, 2015. DOI: 10.26089/NumMet.v16r448
9. Ольховская О.Г. Проекционно-сеточные схемы для аппроксимации уравнений в частных производных второго порядка на нерегулярных сетках // *Препринт ИПМ № 226, Москва, 2018 г., 24 стр.* <http://library.keldysh.ru/preprint.asp?id=2018-226>
10. Самарский А.А. *Теория разностных схем*. М., Наука, 1977, 656 стр.
11. Четверушкин Б.Н., Ольховская О.Г. Моделирование процесса лучистой теплопроводности на высокопроизводительных вычислительных системах // *Доклады Российской Академии Наук. Математика, Информатика, Процессы Управления*, 2020, том 491, с. 111–114 DOI: 10.31857/S2686954320020083

12. Калиткин Н.Н. Численные методы. Учебное пособие для студентов высших учебных заведений. Под редакцией А.А. Самарского. Москва: «Наука». Главная редакция физико-математической литературы, 1978 г., 512 стр.
13. Phillip Colella and Paul R Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.
14. Eleuterio F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer Science & Business Media, 2009.
15. Годунов С.К. Разностный метод численного расчета разрывных решений уравнений гидродинамики. *Матем. сб.*, 47(89)(3):271–306, 1959
16. The Pluto code for adaptive mesh computations in astrophysical fluid dynamics, A. Mignone, C. Zanni, P. Tzeferacos, B. van Straalen, P. Colella, and G. Bodo, *The Astrophysical Journal Supplement Series*, 198:7, 2012 January.
17. Чернов И.А. Трактовка решения Седова как серии промежуточных асимптотик в течении от сильного взрыва // *Изв. вузов «ПНД»*, т. 18, No 4, 2010.

Оглавление

1. Введение3
2. Структуры данных для работы с сеткой4
3. Параллельная реализация5
 - 3.1. Синхронизация структуры при адаптации6
 - 3.2. Динамическая адаптация сетки6
 - 3.3. Балансировка нагрузки7
 - 3.4. Определение ячеек, перераспределяемых между процессорами на этапе динамической балансировки нагрузки8
 - 3.5. Формирование глобальной нумерации листовых ячеек8
4. Реализация проекционной схемы для уравнения теплопроводности8
 - 4.1. Полученные результаты9
5. Реализация схемы для уравнений газовой динамики11
 - 5.1. Полученные результаты13
6. Заключение и план дальнейшей работы16
7. Библиографический список17