

ДИНАМИЧЕСКАЯ БАЛАНСИРОВКА ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ ПРИ МОДЕЛИРОВАНИИ НЕУСТОЙЧИВЫХ ТЕЧЕНИЙ*

© 2025 А.М. Титова, Н.А. Михайлов, Ю.Ф. Юсупов

*Федеральное государственное унитарное предприятие «Российский Федеральный Ядерный
Центр – Всероссийский научно-исследовательский институт технической физики
имени академика Е.И. Забабахина»*

(ФГУП «РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина»)

(456770 Челябинская обл., Снежинск, ул. Васильева, д. 13)

E-mail: a.m.titova@vniitf.ru, n.a.mikhaylov@vniitf.ru, yusupovyuf@vniitf.ru

Поступила в редакцию: 19.04.2025

В работе описан алгоритм динамической балансировки вычислительной нагрузки при моделировании неустойчивых течений на динамически адаптивных расчетных сетках с помощью трехмерной эйлеровой газодинамической программы. Выравнивание вычислительной нагрузки между MPI-процессами производится независимо вдоль линий (наборов MPI-фрагментов, расположенных вдоль выделенного направления), и может выполняться в трех режимах: по числу ячеек, по времени счета, и в смешанном (или автоматическом) режиме. Для минимизации накладных расходов на адаптацию и балансировку используется адаптация с запасом, при этом строится последовательность из этапов адаптации, балансировки и счета в течение заданного количества шагов — цикл Адаптация — Балансировка — Счет (цикл А-Б-С). Оптимальное число шагов в цикле А-Б-С определяется путем минимизации времени счета задачи, которое зависит от общего числа ячеек. Рассматривается тестовая задача о сферическом сжатии легкого центрального вещества тяжелой оболочкой (тест Янгса). Источником развития неустойчивостей является начальное гармоническое возмущение на контактной границе веществ. Моделирование проводится с применением динамической сеточной адаптации второго уровня в области развития неустойчивостей. Оценивается ускорение времени счета задачи при включении балансировки вычислительной нагрузки в разных режимах относительно расчета без балансировки. В результате проведенных исследований показана эффективность применения адаптации с запасом в рамках цикла А-Б-С. При этом автоматический режим выравнивания нагрузки является наиболее практичным.

Ключевые слова: газовая динамика, неустойчивые течения, сферическое сжатие, динамическая адаптация сетки, динамическая балансировка, вычислительная нагрузка.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Титова А.М., Михайлов Н.А., Юсупов Ю.Ф. Динамическая балансировка вычислительной нагрузки при моделировании неустойчивых течений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 3. С. 59–76. DOI: 10.14529/cmse250304.

Введение

На контактной границе между веществами могут развиваться неустойчивые течения по разным причинам [1]. Например, неустойчивость Рэлея–Тейлора (РТ-неустойчивость, или НРТ) возникает при ускоренном движении двух разноплотных жидкостей (имеющих плоскую границу раздела) в направлении от легкого вещества к тяжелому. Частным случаем РТ-неустойчивости является неустойчивость Рихтмайера–Мешкова (НРМ), которая воз-

*Статья рекомендована к публикации программным комитетом Всероссийской научной конференции с международным участием «Параллельные вычислительные технологии (ПаВТ) 2025».

никает при прохождении ударной волны через контактную границу двух различных газов, на которой имеются малые возмущения границы. В случае, когда на контактной границе имеется разрыв тангенсальной компоненты скорости, возникает неустойчивость Кельвина—Гельмгольца (неустойчивость КГ, или НКГ).

В работе [2] проводится моделирование турбулентного перемешивания в сферическом секторе. Авторы описывают расчет сжатия легкого вещества тяжелой оболочкой, при этом на контактной границе веществ задаются гармонические возмущения (тест Янгса). В результате исследований показано, что степень подробности разностной сетки влияет на ширину зоны перемешивания, и для моделирования турбулентного перемешивания необходимо использовать достаточно подробную сетку, которая сможет разрешать основные масштабы течения.

Радиус моделируемого шара в тесте Янгса — 15 см. Прицельные расчеты с помощью трехмерной эйлеровой газодинамической программы [3] показали, что сетки с размером ячеек 0.05 мм достаточно для описания зоны перемешивания с хорошей точностью. Если использовать статическую сетку и проводить расчет в $1/8$ шара, то число ячеек достигает 14 миллиардов. Такая сетка требует огромного количества вычислительных ресурсов.

В целях экономии моделирование необходимо проводить не на статической, а на динамически адаптивной расчетной сетке. Она позволяет сохранять требуемый размер ячеек лишь в зоне перемешивания, где это необходимо. В остальной области сетка может быть в несколько раз грубее. При этом, по мере продвижения зоны, при необходимости ячейки измельчаются или загрубляются. Тестовые оценки показали, что пиковое значение количества ячеек на адаптивной сетке на порядок меньше, чем на статической.

Однако, использование динамической адаптации сетки приводит к дисбалансу вычислительной нагрузки на MPI-фрагментах, ведь, например, в случае адаптации второго уровня, число ячеек на параллельных областях (параобластях) может отличаться до 16 раз в двумерном случае и до 64 раз в трехмерном. В следствие этого менее загруженные MPI-процессы будут простаивать в ожидании других, более загруженных процессов. Поэтому, необходимо иметь функционал, который будет выравнивать вычислительную нагрузку между MPI-процессами, тем самым обеспечивая более равномерное время работы процессов и уменьшение простоев.

Существуют открытые библиотеки, в которых реализована динамическая балансировка, например SCOTCH, PT-SCOTCH [4], ParMETIS [5]. Однако, тестирование данных библиотек выявило несколько недостатков. Во-первых, миграция большого количества ячеек между MPI-процессами, вплоть до глобальной передислокации, значительно увеличивает временные затраты на сетках с числом ячеек больше 10 млн. Во-вторых, работа алгоритма оказалась нестабильной при включении адаптации второго уровня. Именно это и стало поводом для создания собственного модуля балансировки вычислительной нагрузки.

Статья организована следующим образом. В разделе 1 приведены основные особенности газодинамической методики, расчетной сетки и декомпозиции на параллельные области. В разделе 2 описан алгоритм реализации модуля балансировки вычислительной нагрузки в трех режимах: балансировка по числу ячеек, по фактическому времени счета, смешанный режим. В разделе 3 приведены результаты серий тестовых расчетов в двумерной и трехмерной постановках. В заключении проведен анализ результатов тестирования и сделан вывод о том, какой режим балансировки вычислительной нагрузки наиболее оптимален при проведении расчетов задач моделирования неустойчивых течений на динамически адаптивной расчетной сетке.

1. Газодинамическая программа

Модуль балансировки вычислительной нагрузки внедряется в трехмерную программу, которая решает уравнения радиационной газовой динамики на неструктурированных эйлеровых сетках, тем самым, моделируя произвольные вихревые течения. В качестве численной схемы применяется схема годуновского типа [6], которая позволяет избежать использования искусственной вязкости. Для того, чтобы обеспечить монотонность численной схемы, на гранях ячеек производится кусочно-линейная TVD-реконструкция величин [7]. Отсутствие численной диффузии обеспечивается наличием механизма восстановления контактной границы (геометрический метод Volume of Fluid). Верификация данной методики пройдена на задачах моделирования различных режимов развития неустойчивостей [8]. Результаты расчетов сравнивались с открытыми газодинамическими кодами [3].

Для моделирования задач сферического сжатия используются квазисферические динамически адаптивные расчетные сетки [9]. Существует несколько подходов для динамической адаптации сеток: перемещение узлов сетки без изменения ее топологии [10] (лагранжево-эйлеровый подход), покрытие необходимой области вложенной сеткой нужного разрешения [11] (блочно-структурированный подход) или, как в нашем случае, измельчение или укрупнение любой ячейки независимо от других в соответствии с критериями. При произвольной форме зоны перемешивания, используется локальная адаптация ячеек — режим, при котором крупные ячейки, в случае необходимости, измельчаются, а измельченные, когда это не противоречит критерию адаптации, укрупняются. Однако, при моделировании задач, в которых зона перемешивания локализована и движется слоем в течение расчета, можно производить адаптацию сетки слоями (см. рис. 1). Такой подход лучше описывает зону перемешивания, так как разрешение ячеек в поперечном к направлению роста струй направлении остается одинаковым. При этом, он позволяет адаптировать ячейки с запасом, прогнозируя положение зоны перемешивания через заданное количество счетных шагов [12].

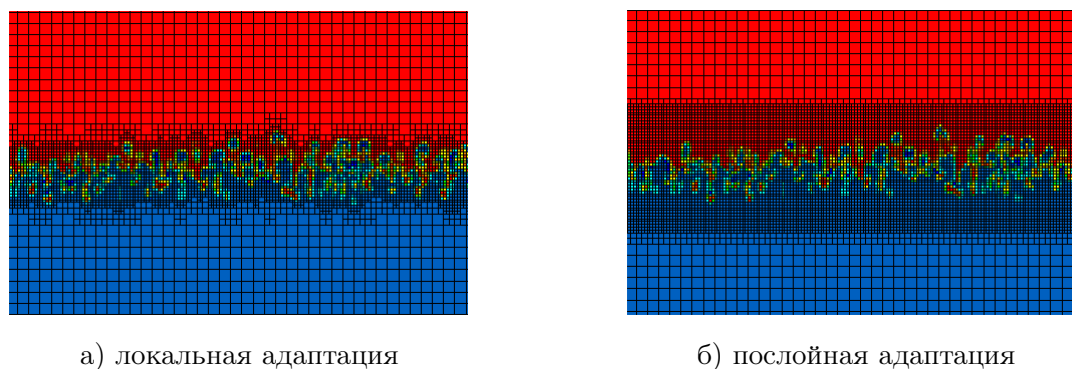


Рис. 1. Динамическая адаптация сетки 2-ого уровня

Распараллеливание программного кода на распределенной памяти выполнено в рамках стандартного подхода пространственной декомпозиции сетка и обмена данными с помощью библиотеки MPI [13]. Расчетная сетка декомпозируется по углу и по радиусу на сферические сектора, как показано на рис. 2. Набор MPI-фрагментов, расположенных вдоль радиуса, называется линией. Число таких линий в каждом угловом направлении совпадает с максимальным количеством параобластей по углу. Так как MPI-фрагмент может лежать в

нескольких линиях, то вводится его вес — коэффициент, который определяет вклад этого процесса в вычислительную нагрузку на всей линии.

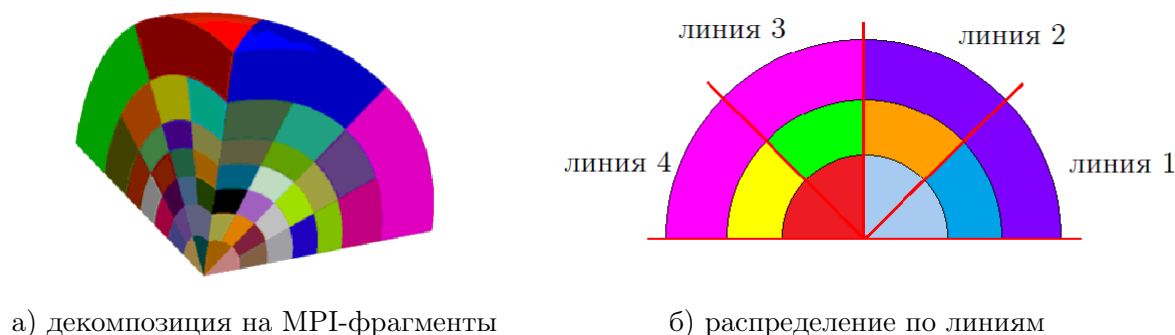


Рис. 2. Сферическая декомпозиция расчетной сетки

В начальный момент времени число ячеек в параобластях одинаково. Однако, измельчение сетки в зоне перемешивания при адаптации приводит к тому, что число ячеек на некоторых MPI-фрагментах становится больше, чем на других, и время на работу таких фрагментов увеличивается. Таким образом, возникает необходимость иметь функционал, который будет балансировать вычислительную нагрузку между MPI-процессами и обеспечивать выровненное время работы всех MPI-процессов.

2. Реализация алгоритма

2.1. Общая схема работы

Так как зона адаптации является квазисферической, и геометрия параобластей также обладает сферичностью, балансировка вычислительной нагрузки производится вдоль линий. При этом, каждый процесс может передавать нагрузку на процессы, которые находятся выше или ниже по радиусу, и не передает нагрузку процессам слева или справа по углу. Балансировка вычислительной нагрузки производится после адаптации расчетной сетки. Алгоритм балансировки включает в себя следующие этапы:

1. Формирование линий.

Так как в каждой линии по углу располагается не более одного процесса, то число линий в системе определяется произведением максимальных количеств MPI-фрагментов по полярному и азимутальному углам. В случае процессного каскада (случая, когда процесс располагается в нескольких линиях), определяется вес процесса, который учитывается при вычислении целевой нагрузки на MPI-фрагменте в линии. Этап формирования линий запускается один раз в начале счета задачи.

2. Определение вычислительной нагрузки на каждом MPI-процессе.

В случае, если балансировка вычислительной нагрузки производится по ячейкам, то вычислительная нагрузка определяется как число ячеек на MPI-фрагменте. При балансировке по времени вычислительная нагрузка есть фактическое время счета шага для MPI-процесса.

3. Вычисление коэффициента разбалансированности системы.

Коэффициент разбалансированности определяется следующим образом:

$$K = \frac{\text{макс. нагрузка}}{\text{средняя нагрузка}}. \quad (1)$$

Максимальная нагрузка определяется с учетом всех MPI-фрагментов в линии. Средняя нагрузка определяется как сумма вычислительных нагрузок на MPI-процессах с учетом их весов в линии, деленная на количество MPI-процессов в линии. Если K оказывается больше максимально разрешенного коэффициента разбалансированности, заданного пользователем, то запускается алгоритм балансировки.

4. Построение списка количества нагрузки, которую должен отдать или принять каждый MPI-процесс.

Данный этап является единственным последовательным алгоритмом во всем модуле балансировки и производится мастер-процессом. Он собирает информацию о дисбалансе вычислительной нагрузки (разнице между текущей вычислительной нагрузкой и средней) на каждом MPI-процессе и определяет, сколько ячеек принять или отдать каждому MPI-процессу. В цикле по всем линиям, начиная с самого близкого к центру MPI-фрагмента в линии, происходит вычисление количества нагрузки, которое необходимо отдать с текущего фрагмента на следующий или принять от него. Если соседний MPI-фрагмент не имеет достаточного количества нагрузки, то аналогичная процедура проводится для MPI-фрагмента, следующего за соседом. Так происходит до тех пор, пока для все MPI-процессов не будут определены списки приема-передачи вычислительной нагрузки. После этого, построенный на мастер-процессе двумерный массив рассылается всем слейв-процессам.

5. Маркировка ячеек на каждом MPI-фрагменте с учетом построенного списка приема-передачи ячеек.

Получив информацию о том, каким количеством нагрузки необходимо обменяться, каждый MPI-процесс определяет, какие именно ячейки будут передаваться. Ячейки маркируются по очереди, начиная с ближайшего к процессу-приемнику слоя еще неотмаркированных ячеек. Таким образом, если процесс-приемник находится выше по радиусу, чем текущий MPI-фрагмент, то маркировка ячеек начинается со слоя ячеек сверху, причем для самого удаленного фрагмента ячейки маркируются первыми. Схематично этап маркировки ячеек изображен на рис. 3.

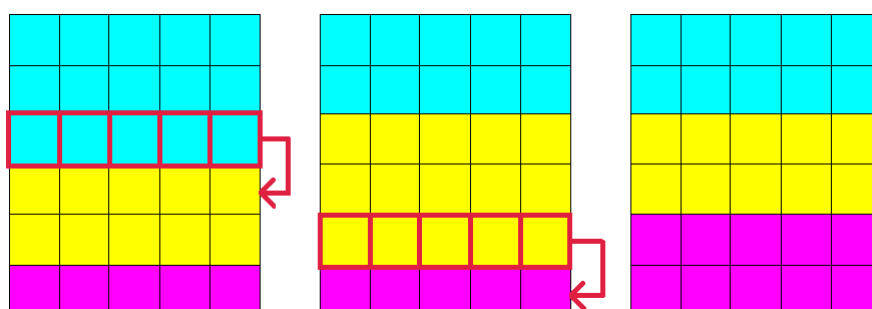


Рис. 3. Схематическое изображение работы этапа маркировки ячеек для трех MPI-фрагментов (ячейки с одного фрагмента окрашены одним цветом)

Стоит отметить, что все поколения адаптированной ячейки передаются вместе с родительской ячейкой, что может привести к небольшому отланию нового значения вычислительной нагрузки от целевого.

2.2. Балансировка по числу ячеек

Динамическая адаптация расчетной сетки приводит к тому, что число ячеек на параобластях становится разным. Это влечет за собой замедление времени счета шага, так как некоторые из MPI-процессов простаивают в ожидании окончания работы загруженных процессов. Поэтому балансировка числа ячеек между фрагментами является необходимым инструментом при проведении расчетов на динамически адаптивных сетках, который позволяет ускорить общее время счета задачи.

При балансировке по числу ячеек вычислительная нагрузка равняется числу ячеек на MPI-фрагменте. Целевое число ячеек на MPI-фрагменте P в линии L после балансировки по числу ячеек определяется следующим образом:

$$\langle N_P^L \rangle = \frac{\sum_{p \in L} N_p \cdot W_P^L}{\sum_{p \in L} W_P^L} \cdot W_P^L, \quad (2)$$

где W_P^L — вес MPI-фрагмента P в линии L , который равен:

$$W_P^L = \frac{N_P^L}{N_P}. \quad (3)$$

Здесь N_P — число ячеек на MPI-фрагменте P , N_P^L — число ячеек на MPI-фрагменте P , которые попали в линию L .

На рис. 4 схематично представлен пример работы алгоритма балансировки по числу ячеек при адаптации первого уровня. После измельчения сетки число ячеек на параобластях стало отличаться в 4 раза. После балансировки число ячеек вновь стало одинаковым, при этом дочерние ячейки находятся на одном MPI-фрагменте с родительскими.

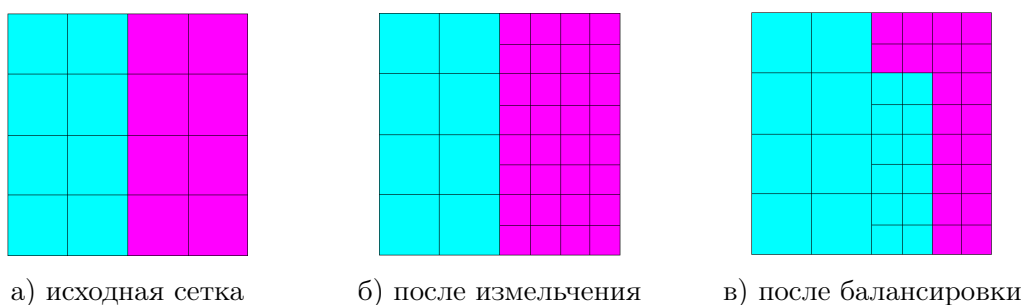


Рис. 4. Пример распределения ячеек между MPI-фрагментами

2.3. Балансировка по времени счета

Несмотря на то, что число ячеек на MPI-фрагментах одинаковое, длительность работы каждого MPI-процесса может отличаться. Время счета шага определяется как сумма времен счета основных этапов для каждой ячейки с MPI-фрагмента. К основным этапам относятся: вычисление гудоновских потоков, восстановление контактной границы, обновление величин в ячейках, расчет давления и температуры из уравнения состояния, при этом время работы для каждой ячейки на некоторых этапах зависит от количества веществ в ней. Как видно из рис. 5, это приводит к тому, что счет смешанных ячеек дольше, чем чистых.

Таким образом, при балансировке вычислительной нагрузки необходимо выравнивать не количество ячеек на MPI-фрагментах, а фактическое время их работы, которое определя-

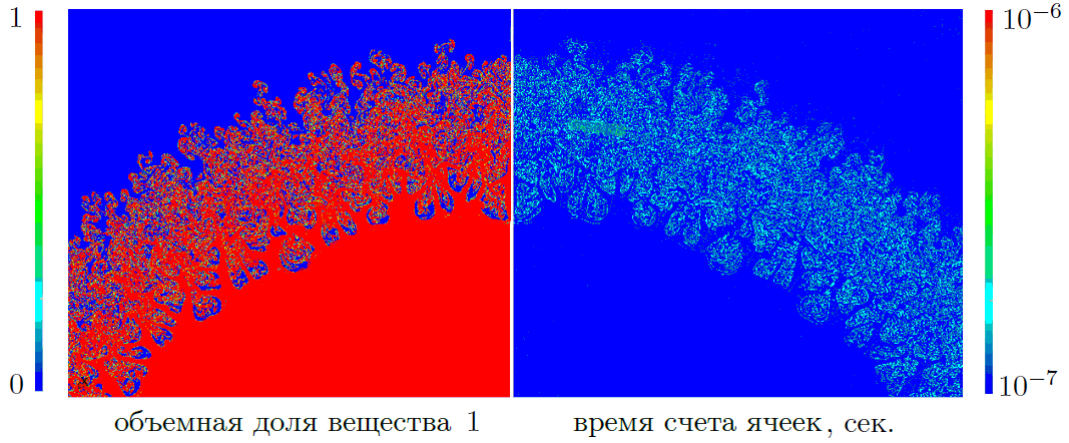


Рис. 5. Объемная доля вещества и время обработки ячеек для тестовой задачи Янгса

ется как:

$$t_{cell} = t_{cell}^{flux} + t_{cell}^{КГ} + t_{cell}^{EOS} + t_{cell}^{update}, \quad (4)$$

где t_{cell}^{flux} — время счета годуновских потоков для ячейки, $t_{cell}^{КГ}$ — время работы механизма восстановления контактной границы для ячейки, t_{cell}^{EOS} — время работы с уравнением состояния, t_{cell}^{update} — время обновления величин в ячейке. Вес MPI-фрагмента в линии равен:

$$W_P^L = \frac{t_P^L}{t_P}, \quad (5)$$

где t_P^L — суммарное время счета ячеек фрагмента P, лежащих в линии L, t_P — суммарное время счета всех ячеек на MPI-фрагменте. В таком случае, целевое время работы MPI-процесса P в линии L после балансировки по времени есть:

$$\langle t_P^L \rangle = \frac{\sum_{p \in L} t_P \cdot W_P^L}{\sum_{p \in L} W_P^L} \cdot W_P^L. \quad (6)$$

Таким образом, реализовано два режима балансировки вычислительной нагрузки: по числу ячеек и по фактическому времени счета MPI-процесса. Балансировка по числу ячеек обеспечивает одинаковое заполнение памяти MPI-фрагментов и позволяет избежать ошибок переполнения на MPI-процессах при измельчении сетки и появлении новых ячеек на этапах адаптации. При этом из-за того, что смешанные и чистые ячейки считаются разное количество времени, время работы MPI-процессов отличается. Балансировка по фактическому времени счета позволяет сделать время работы всех MPI-процессов одинаковым и уменьшить простои. Однако, при таком режиме число ячеек на фрагментах может отличаться значительно. При этом, если на уже заполненном ячейками MPI-фрагменте произведется этап адаптации, то имеется риск выхода за пределы оперативной памяти. Комбинирование двух режимов балансировки в процессе счета задачи позволит наиболее эффективно использовать сильные стороны реализованных алгоритмов и обойти слабые.

2.4. Автоматический режим

Тестовые расчеты показали, что накладные расходы на адаптацию и балансировку могут составлять до 50% от общего времени счета задачи. Для того, чтобы их минимизиро-

вать, рассматривается подход адаптации с запасом на заданное количество шагов. Прогнозирование положения зоны перемешивания делается в предположении постоянной скорости движения контактной границы в течение этих шагов. Скорость движения определяется по изменению положения границ зоны на предыдущем интервале [12].

Измельчение сетки с запасом приводит к росту количества ячеек, что увеличивает время счета. Для того, чтобы минимизировать общее итоговое время счета, необходимо решить задачу оптимизации количества временных шагов, на которые делается прогноз. Общее время счета задачи в зависимости от количества запасных шагов N_{ABC} в приближении роста ширины зоны перемешивания оценивается как:

$$T(N_{ABC}) = \sum_{j=1}^{N_T/N_{ABC}} \left(t_{cell} \cdot (N_{cells}^0 + j \cdot N_R \cdot N_{ABC}) \cdot N_{ABC} + t_{AB}^{cell} \cdot (N_{cells}^0 + j \cdot N_R \cdot N_{ABC}) \right), \quad (7)$$

где T — время счета задачи из N_T временных шагов, N_{ABC} — количество шагов прогнозирования, t_{cell} — время счета одной ячейки на одном временном шаге, N_{cells}^0 — начальное количество ячеек в задаче, N_R — количество ячеек, добавляемое при прогнозировании на один временной шаг, t_{AB}^{cell} — накладные расходы на одну ячейку. Минимум данной функции дает N_{ABC}^* , получающаяся из следующего квадратного уравнения:

$$\frac{dT}{dN_S}(N_{ABC}^*) = 0, \quad (8)$$

или

$$t_{cell} \cdot N_R \cdot N_{ABC}^2 + \frac{N_R t_{AB}^{cell}}{2} N_{ABC} - \left(\frac{N_R t_{AB}^{cell}}{2} \cdot N_T + t_{AB}^{cell} \cdot N_{cells}^0 \right) = 0. \quad (9)$$

Из расчетов тестовой задачи были получены следующие значения параметров построенной модели: $t_{cell} = 1.7 \cdot 10^{-7}$, $N_R = 2 \cdot 10^3$, $t_{AB}^{cell} = 4 \cdot 10^{-7}$.

Для двумерного расчета с начальным числом ячеек $N_{cells}^0 = 10^6$ ячеек и числом временных шагов $N_T = 5 \cdot 10^4$ оптимальное количество шагов в цикле А-Б-С равно $N_{ABC}^* = 244$. Данное значение дает примерно 2% накладных расходов.

Таким образом, строится цикл Адаптация — Балансировка — Счет (цикл А-Б-С) — последовательность из этапов адаптации, балансировки и счета на заданном количестве N_{ABC} счетных шагов.

Цикл строится таким образом, что сначала производятся все этапы измельчений, количество которых определяется максимальным уровнем адаптации. При этом после каждого из этих этапов производится балансировка вычислительной нагрузки по числу ячеек. Далее укрупняются все ячейки, которые можно укрупнить. После укрупнения производится балансировка по числу ячеек и счетный шаг, на котором замеряются времена счета ячеек. Далее делается балансировка по времени счета и счет N_S числа шагов (N_S — количество счетных шагов в цикле А-Б-С). Общее число шагов в цикле А-Б-С есть сумма счетных шагов, шагов измельчения и укрупнения, и шагов балансировки. Число шагов в цикле задается пользователем. Пример порядка шагов при адаптации 2-го уровня представлен на рис. 6.

Здесь И — этап измельчения ячеек, У — этап укрупнения ячеек, Б — этап балансировки вычислительной нагрузки (зеленым — балансировка по числу ячеек, розовым — балансировка по времени счета), С — этап счета.

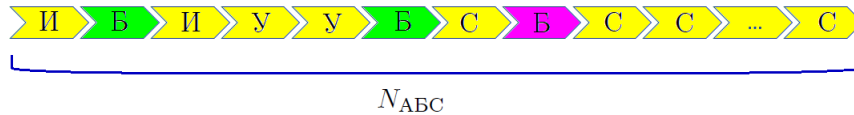


Рис. 6. Порядок расположения шагов в цикле А-Б-С при адаптации 2-го уровня

Таким образом, построен численный алгоритм, который позволяет балансировать вычислительную нагрузку в автоматическом режиме, используя комбинацию из балансировок по ячейкам и по фактическому времени счета. При этом накладные расходы на адаптацию и балансировку минимизируются за счет использования цикла Адаптация — Балансировка — Счет с прогнозированием положения зоны перемешивания.

3. Тестирование алгоритма

3.1. Постановка тестовой задачи

Рассматривается тестовая задача Янгса о сжатии легкого центрального вещества плотной оболочкой [14]. При наличии шероховатости на контактной границе между первым и вторым веществом развивается неустойчивость Рихтмайера—Мешкова, поэтому данная задача часто используется для отработки технологии проведения расчетов перемешивания. Для моделирования граничных условий по эйлеровой программе используется буферная область (вещество 3). Распределение веществ и начальные характеристики представлены на рис. 7 и в табл. 1.

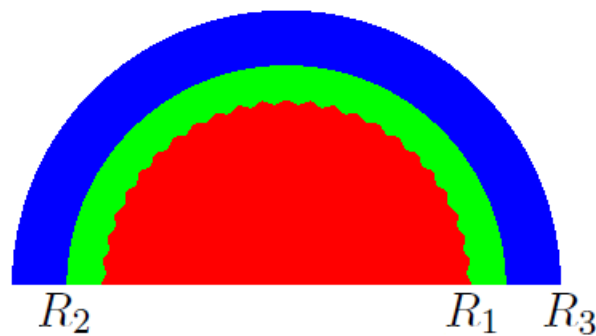


Рис. 7. Геометрия тестовой задачи

Таблица 1. Параметры веществ в тестовой задаче

вещество	R , см	ρ , г см ³	C_v	e , эрг
1	10	0.05	2	3
2	12	1	0.1	0.15
3	15	0.1	100	150

Уравнение состояния веществ — идеальный газ с показателем адиабаты $5/3$. Внутренняя удельная энергия на внешней границе оболочки (вещества 2) до $t = 0.5$ сек поддерживалась постоянной и равнялась 150 эрг. Далее она снижалась по линейному закону до

1.5 эрг к моменту времени $t = 3$ сек. Затравкой для развития неустойчивостей на контактной границе между веществами служит гармоническое возмущение:

$$P(\theta) = A \cdot \cos(m \cdot \theta), \quad (10)$$

здесь амплитуда $A = 1.25$ см, номер моды $m = 48$. Конечное время счета $t_k = 2.5$ сек — момент максимального сжатия центрального вещества. В трехмерной постановке начальное возмущение контактной границы «центральное вещество — оболочка» задавалось квазидвумерным: гармонические возмущения задавались только по одному угловому направлению. На рис. 8 представлено распределение веществ и слой оболочки на начальный момент времени.

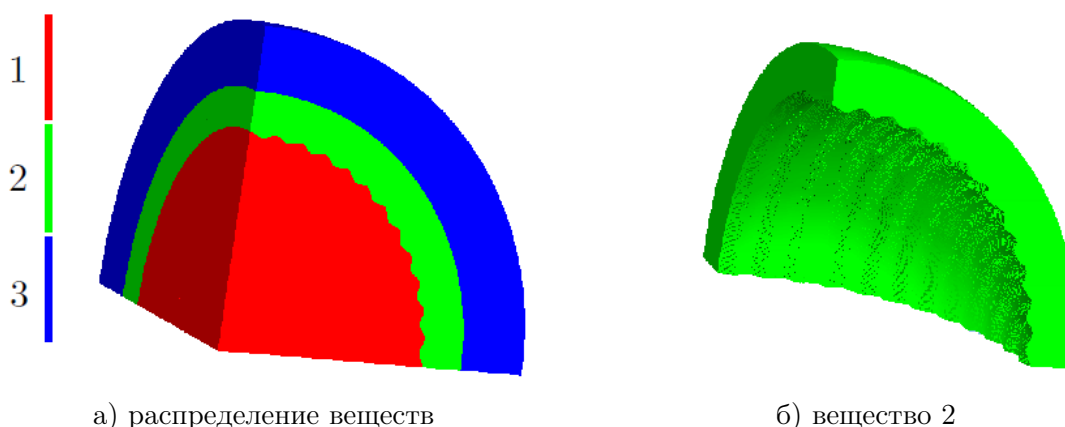


Рис. 8. Распределение веществ в задаче Янгса в 3D постановке

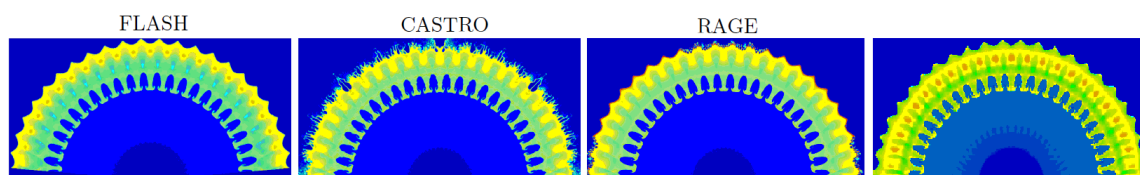
3.2. Характеристики расчетов

В двумерном случае расчеты проводились в осесимметричной постановке в секторе $0 \leq \theta \leq 180^\circ$. Использовалась квазисферическая сетка с базовым размером ячеек 0.02 см. При этом в области третьего вещества размер ячеек увеличивался до 0.04 см. Режим адаптации — послойная, 2-ого уровня, что обеспечивает размер ячеек в зоне перемешивания 0.05 мм.

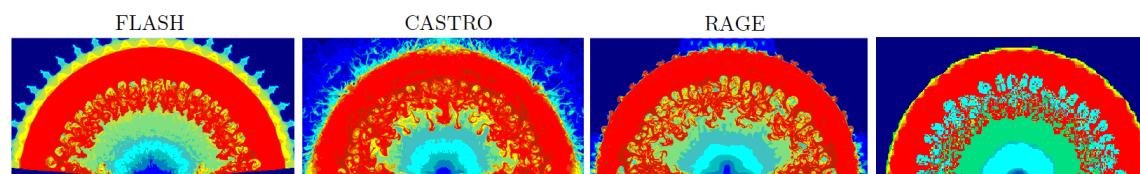
В трехмерной постановке рассматривался сектор $0 \leq \theta \leq 90^\circ$ и $0 \leq \phi \leq 90^\circ$. Аналогично двумерному случаю, использовалась квазисферическая сетка с базовым размером ячеек 0.02 см и 0.04 см на периферии системы. В зоне перемешивания была включена адаптация 2-го уровня с прогнозированием положения зоны перемешивания.

Коэффициент разбалансированности в расчетах $K_{max} = 1.2$. Число шагов в цикле $N_{ABC} = 300$.

На рис. 9 представлено распределение плотности в системе до и в момент максимального сжатия центрального вещества ($t = 1.5$ сек и $t = 2.5$ сек). Сравнение с открытыми CFD-программами (FLASH, RAGE, CASTRO) [14] показывает хорошее сходство.



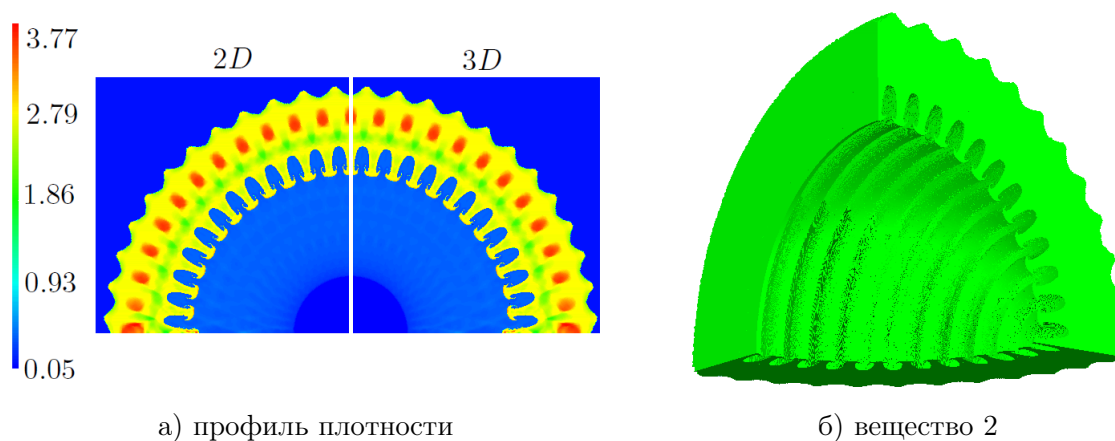
а) профиль плотности при $t = 1.5$ сек



б) профиль плотности при $t = 2.5$ сек

Рис. 9. Профиль плотности на для разных программ. Правый столбец — расчет программы, рассматриваемой в рамках данной статьи

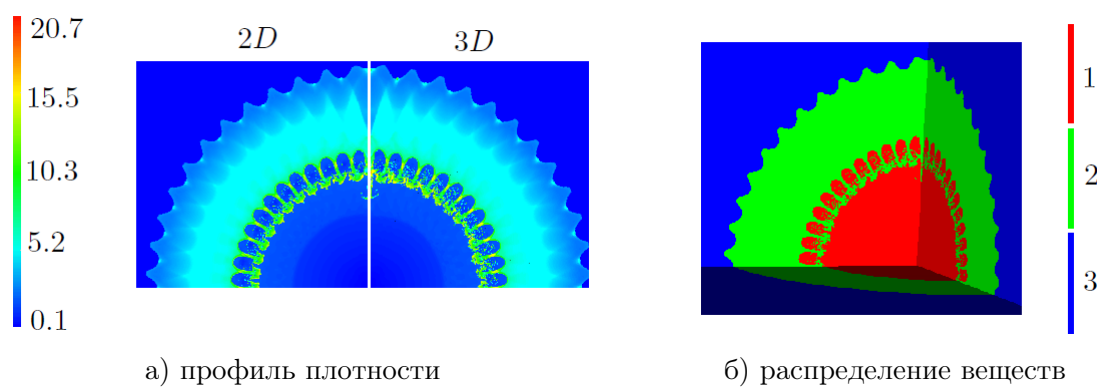
На рис. 10 и 11 представлен результат моделирования в трехмерной постановке. Как видно, результат трехмерного расчета согласуется с двумерным.



а) профиль плотности

б) вещество 2

Рис. 10. Сравнение решений в 2D и 3D постановках на момент времени $t = 1.5$ сек



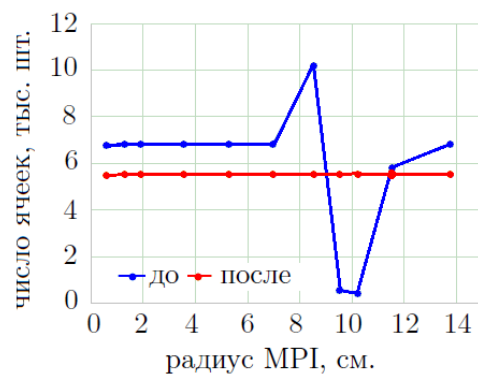
а) профиль плотности

б) распределение веществ

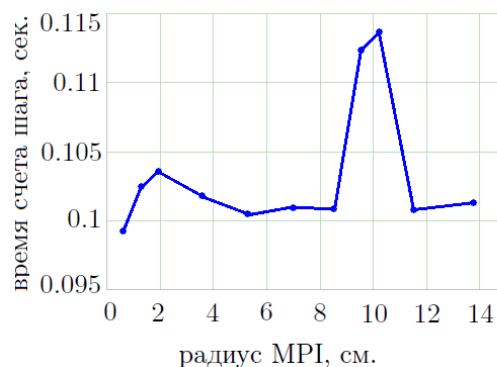
Рис. 11. Сравнение решений в 2D и 3D постановках на момент времени $t = 2$ сек

3.3. Результаты тестирования

Как видно из рис. 12, количество ячеек на MPI-фрагментах вдоль выделенной линии после балансировки по числу ячеек становится одинаковым. При этом время счета шага этих MPI-процессов отличается значительно.



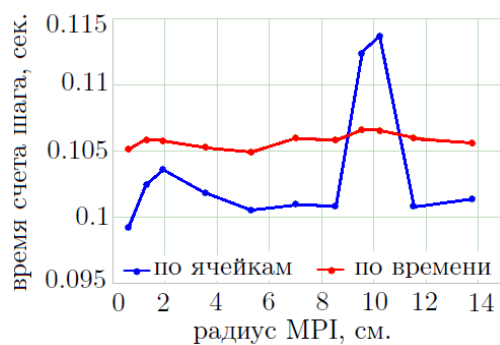
а) число ячеек на MPI-фрагментах



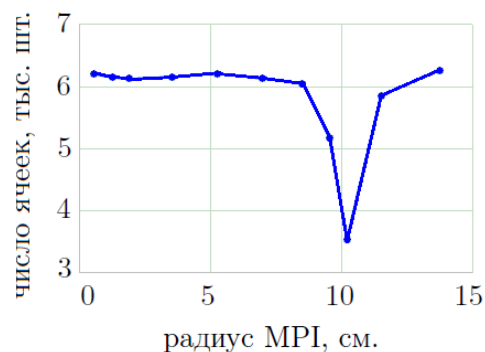
б) время счета шага

Рис. 12. Число ячеек на MPI-фрагментах вдоль выделенного направления и время счета шага при балансировке по числу ячеек

На рис. 13 слева представлено распределение времени счет шага для MPI-процессов, расположенных вдоль выделенного направления, при балансировке вычислительной нагрузки по фактическому времени счета. При этом число ячеек на этих MPI-фрагментах отличается (см. рис. 13 справа).



а) время счета шага



б) число ячеек на MPI-фрагментах

Рис. 13. Время счета шага и число ячеек на MPI-фрагментах при балансировке по времени

По представленным графикам можно сделать вывод о том, что реализованные алгоритмы балансировки вычислительной нагрузки по ячейкам и по времени работают корректно, выравнивая число ячеек на MPI-фрагменте или время их счета.

Для анализа эффективности работы реализованного алгоритма балансировки вычислительной нагрузки было проведено несколько серий сравнительных расчетов тестовой задачи в двумерной постановке.

В первой серии расчетов сравнивалось полное время счета тестовой задачи без прогнозирования положения зоны перемешивания для двух режимов балансировки: по числу ячеек и по времени счета. Оценивалось ускорение времени счета задачи относительно рас-

чета без балансировки (время счета без балансировки — 17.7 ч.). Результаты исследования приведены в табл. 2.

Таблица 2. Время счета тестовой задачи в 2D постановке при разных режимах балансировки вычислительной нагрузки без прогнозирования

режим	время счета, час	ускорение, раз
по ячейкам	10.4	1.7
по времени	9.2	1.9

При проведении расчета без прогнозирования положения зоны перемешивания балансировка вычислительной нагрузки сокращает время счета почти в 2 раза. При этом расчет с балансировкой по фактическому времени счета прошел на 11.5% быстрее, чем с балансировкой по числу ячеек.

Во второй серии расчетов сравнивалось время счета задачи для разных режимов балансировки при включении прогнозирования положения зоны перемешивания. Расчет без балансировки прошел за 16.3 ч.

Таблица 3. Временные характеристики расчета тестовой задачи в 2D постановке при разных режимах балансировки вычислительной нагрузки с прогнозированием

режим	время счета, час	ускорение, раз
по ячейкам	8.9	1.8
по времени	5	3.3
автоматический	6.5	2.5

Как видно из табл. 3, расчет с адаптацией с запасом прошел на 10–15% быстрее, чем расчет без прогнозирования. При этом наибольшее ускорение в расчете относительно расчета без балансировки вычислительной нагрузки было получено при балансировке по времени (в 3.3 раза). Балансировка по числу ячеек дает наименьшее ускорение (1.8 раз).

В третьей серии расчетов оценивалось практическое значение наиболее оптимального количества шагов в цикле А-Б-С. Для этого в расчете тестовой задачи число шагов N_{ABC} варьировалось от $N_{ABC} = 50$ до $N_{ABC} = 900$. Из графика 14 видно, что минимум времени счета достигается при $N_{ABC} = (200; 400)$, что согласуется с теоретическим значением $N_{ABC}^{теор.} = 244$.

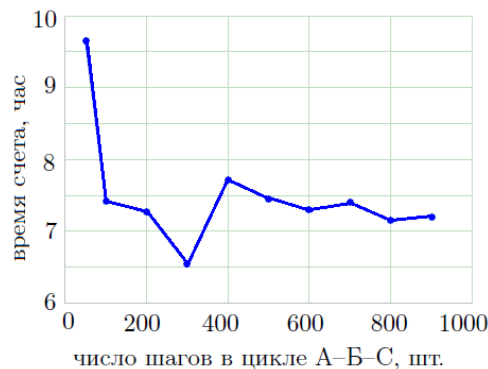


Рис. 14. Зависимость времени счета задачи от числа шагов в цикле А-Б-С

На первой и второй диаграммах на рис. 15 показано, какую долю общего времени счета задачи занимают расходы на адаптацию и балансировку.

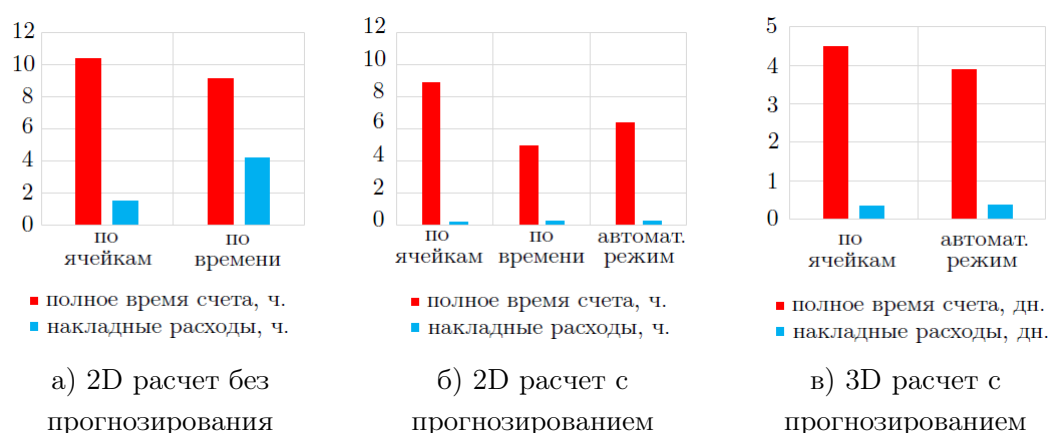


Рис. 15. Накладные расходы на адаптацию и балансировку

В случае, когда не используется адаптация с запасом, накладные расходы на составляют до 45% времени счета задачи. При использовании прогнозирования в рамках цикла А-Б-С накладные расходы снижаются и не превышают 5%.

Также была проведена серия трехмерных расчетов с использованием прогнозирования положения зоны, в которой оценивалось время счета задачи на характерном временном интервале $t = 1$ сек до $t = 2$ сек при балансировке по числу ячеек, по времени счета и в автоматическом режиме. В табл. 4 представлены времена счета задачи.

Таблица 4. Время расчета интервала $t \in (1, 2)$ сек в 3D постановке при разных режимах балансировки вычислительной нагрузки

режим	время счета, дни	накладные расходы на АБ, %
по ячейкам	4.5	7.6
по времени	—	—
автоматический	3.9	9.1

При проведении расчета с использованием балансировки по времени наличие сильно неравномерного количества данных на MPI-процессах привело к выходу за пределы оперативной памяти на вычислительном узле. Действительно, дисбаланс числа ячеек на MPI-фрагментах при балансировке по времени в 3D расчете достигает ~ 10 , а при адаптации 2-го уровня число ячеек на параобласти может увеличиться еще до 64 раз. Именно в таком случае и происходит выход за пределы памяти. Проведение расчета с балансировкой по времени возможно лишь с увеличением вычислительных ресурсов. Именно для того, чтобы минимизировать риски переполнения памяти, в автоматическом режиме между этапами измельчения и укрупнения сетки производятся балансировки по числу ячеек, и лишь перед счетными шагами цикла А-Б-С делается балансировка по времени счета.

Результаты трехмерных расчетов показали, что расчет характерного интервала прошел на 15% быстрее при балансировке в автоматическом режиме. Накладные расходы на адаптацию и балансировку в трехмерных расчетах отражены в табл. 4 и на рис. 15в.

Заключение

В трехмерной газодинамической эйлеровой программе реализован функционал динамической балансировки вычислительной нагрузки вдоль линий, который может работать в трех режимах: выравнивать нагрузку по числу ячеек, по фактическому времени счета, или в автоматическом режиме с построением цикла Адаптация—Балансировка—Счет. Для анализа работоспособности алгоритма проводилось моделирование неустойчивого течения, возникающего в задаче о сферическом сжатии легкого центрального вещества тяжелой оболочкой (задача Янгса). Было проведено несколько серий тестовых расчетов на квазисферической динамически адаптивной сетке, которая имела сферическую декомпозицию на MPI-фрагменты. Адаптация проводилась в послойном режиме.

Для двумерной постановки было проведено несколько серий тестовых расчетов. В первой серии сравнивалось полное время счета задачи при разных режимах балансировки вычислительной нагрузки без использования прогнозирования положения зоны перемешивания. Анализ результатов показал, что балансировка вычислительной нагрузки по времени счета на 10% сокращает время счета задачи. При этом накладные расходы на адаптацию и балансировку составляют значительную часть от общего времени (до 45% для балансировки по времени).

Во второй серии расчетов оценивалось полное время счета тестовой задачи при активации прогнозирования положения зоны перемешивания на $N_{\text{АБС}} = 300$ счетных шагов. Анализ результатов показал, что внедрение адаптации с запасом в рамках цикла А-Б-С позволяет снизить накладные расходы на адаптацию и балансировку до 5%. При этом общее время счета тестовой задачи сократилось на 30% относительно расчета без прогнозирования.

В третьей серии расчетов варьировалось число шагов в цикле А-Б-С, на которое делался запас. Результат показал, что теоретическая оценка оптимального количества шагов $N_{\text{АБС}}^{\text{теор.}} = 244$ в цикле А-Б-С совпадает с практической.

В трехмерной постановке была проведена одна серия расчетов, которая показала, что расчеты практичнее проводить либо с балансировкой по ячейкам, либо с использованием автоматического режима. При балансировке по времени счета дисбаланс количества ячеек может приводить к выходу за пределы оперативной памяти. При использовании автоматического режима расчет измеряемого характерного интервала прошел на 13 % быстрее, чем балансировка по времени. При этом накладные расходы на балансировку и адаптацию менее 9%.

В результате проведенных исследований можно сделать вывод о том, что режим балансировки по фактическому времени счета наиболее эффективен для двумерных расчетов. В трехмерных же расчетах практичнее выбирать автоматический режим балансировки вычислительной нагрузки при моделировании неустойчивых течений.

Литература

1. Янилкин Ю.В., Стаценко В.П., Козлов В.И. Математическое моделирование турбулентного перемешивания в сжимаемых средах. Том 2: учебное пособие. Саров: Российский федеральный ядерный центр – Всероссийский НИИ экспериментальной физики, 2020. 407 с.
2. Youngs D.L., Williams R.J. Turbulent Mixing in Spherical Implosions // Int. J. Numer. Methods Fluids. 2008. Vol. 56, no. 8. P. 1597–1603. DOI: 10.1002/fld.1594.

3. Глазырин И.В., Михайлов Н.А. Метод укрупнения контактных границ для моделирования трехмерных многофазных сжимаемых течений в эйлеровых переменных // Забабахинские научные чтения: Сборник тезисов XIII международной конференции, Снежинск, 20–24 марта 2017. Снежинск: Из-во РФЯЦ-ВНИИТФ, 2017. С. 326.
4. Chevalier C., Pellegrini F. PT-Scotch: A Tool for Efficient Parallel Graph Ordering // Parallel Computing. 2008. Vol. 34, no. 6. P. 318–331. DOI: 10.1016/j.parco.2007.12.001.
5. Karypis G. METIS and ParMETIS // Encyclopedia of Parallel Computing. Springer US, 2011. P. 1117–1124. DOI: 10.1007/978-0-387-09766-4_500.
6. Toro E. Rieman Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction. Berlin, Heidelberg: Springer Verlag, 2009. 721 p. DOI: 10.1007/b79761.
7. Darwish M., Moukalled F. TVD-schemes for Unstructured Grids // International Journal of Heat and Mass Transfer. 2003. Vol. 46, no. 8. P. 599–611. DOI: 10.1016/S0017-9310(02)00330-7.
8. Глазырин И.В., Михайлов Н.А. Конечно-объемная схема для многокомпонентных сжимаемых течений на неструктурированной сетке в трехмерной программе Фокус // ЖФММФ. 2021. Т. 61, № 6. С. 1019–1033. DOI: 10.31857/S0044466921060041.
9. Балашов Н.М., Глазырин И.В., Михайлов Н.А. Двумерная динамическая адаптация сетки в газодинамической программе // Параллельные вычислительные технологии – XIV международная конференция, ПаВТ’2020, Пермь, 31 марта – 2 апреля 2020. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2020. С. 325.
10. Jasak H., Tukovic Z. Automatic Mesh Motion for the Unstructured Finite Volume Method // Transactions of FAMENA. 2006. Vol. 30, no. 2. P. 1–20.
11. Colella P., Berger M. Local Adaptive Mesh Refinement for Shock Hydrodynamics // Journal of Computational Physics. 1989. Vol. 82, no. 1. P. 64–84. DOI: 10.1016/0021-9991(89)90035-1.
12. Глазырин И.В., Глазырина Н.В., Михайлов Н.А., Писклова М.А. Использование динамической адаптации сетки для расчетов неустойчивостей Рэлея–Тейлора // Забабахинские научные чтения: сборник материалов XVI Международной конференции 29 мая – 2 июня 2023. Снежинск: Изд-во РФЯЦ-ВНИИТФ, 2023. С. 364.
13. Оленев Н.Н. Основы параллельного программирования в системе MPI. Москва: Вычислительный центр А.А. Дородницына, 2005. 79 с.
14. Joggerst C.C., Nelson A., Woodward P., *et al.* Cross-code Comparison of Mixing During the Implosion of Dense Cylindrical and Spherical Shells // Journal of Computational Physics. 2014. Vol. 82, no. 1. P. 154–173. DOI: 10.1016/j.jcp.2014.06.037.

Титова Александра Михайловна, научный сотрудник, научно-теоретическое отделение 2, ФГУП «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» (Снежинск, Российская Федерация)

Михайлов Никита Анатольевич, к.ф.-м.н., начальник лаборатории, научно-теоретическое отделение 2, ФГУП «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» (Снежинск, Российская Федерация)

Юсупов Юлиус Фаритович, научный сотрудник, научно-теоретическое отделение 2, ФГУП «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» (Снежинск, Российская Федерация)

DOI: 10.14529/cmse250304

DYNAMIC COMPUTATIONAL LOAD BALANCING DURING SIMULATIONS OF INSTABILITY FLOWS

© 2025 A.M. Titova, N.A. Mikhaylov, Y.Y. Yusupov

*Federal State Unitary Enterprise “Russian Federal Nuclear Center – All-Russian Research
Nuclear Institute of Technical Physics named after Academician E.I. Zababakhin”,*

(FSUE “RFNC-VNIITF named after Academ. E.I. Zababakhin”)

(Vasiliev st. 13, Snezhinsk, Chelyabinsk region, 456770 Russian Federation)

E-mail: a.m.titova@vniitf.ru, n.a.mikhaylov@vniitf.ru, yusupovyuf@vniitf.ru

Received: 19.04.2025

In this work we describe the algorithm of dynamic computational load balancing during modeling of instability flows over dynamically refined mesh using three-dimensional eulerian gas dynamic program. Balancing of computational load between MPI-fragments is executed along lines (set of MPI-fragments that are situated along preferential direction) independently and may be carried out in three regimes: accounting number of cells, accounting real calculation time, and automatic (or composite) regime. We use adaptation with reservation to minimize overheads for adaptation and balancing: we construct a sequence of adaptation, balancing and calculation stages during fixed count of steps – “Adaptation — Balancing — Calculation” cycle (A–B–C cycle). The optimal count of steps in A–B–C cycle is determined by minimization of task calculation time that depends on count of cells. For testing of realized algorithm we considered a spherical implosion of a light material by a dense shell (Youngs’ problem). A source of instabilities was the initial harmonic perturbation on the contact boundary. The calculations are made using second-level dynamically adaptive mesh refinement in the region of instability growth. We analyze the calculation time of the problem using different regimes of load balancing algorithm. The result of the research proves the efficiency of using adaptive mesh refinement with reservation within A–B–C cycle. The automatic regime of load balancing is the most stable for three-dimensional modelling.

Keywords: gas dynamics, instability flows, spherical implosion, adaptation mesh refinement, load balancing, computational load.

FOR CITATION

Titova A.M., Mikhaylov N.A., Yusupov Y.Y. Dynamic Computational Load Balancing During Simulations of Instability Flows. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 3. P. 59–76. (in Russian) DOI: 10.14529/cmse250304.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Yanilkin Y.V., Stacenko V.P., Kozlov V.I. Mathematical Modeling of Turbulent Mixing in Compressible Medium. Vol. 2: Training manual. Sarov: Russia Research Nuclear Center – All-Russian SRI of Experimental Physics, 2020. 407 p. (in Russian).

2. Youngs D.L., Williams R.J. Turbulent Mixing in Spherical Implosions. *Int. J. Numer. Methods Fluids*. 2008. Vol. 56, no. 8. P. 1597–1603. DOI: 10.1002/fld.1594.
3. Glazyrin I.V., Mikhaylov N.A. The Torsion Method for Contact Bounds for Modeling of Three-dimensional Multi-component Compressible Flows in Eulerian Variables. *Zababakhin Scientific Readings: Collection of Abstracts XIII International Conference, Snezhinsk, 20–24 March 2017*. Snezhinsk: Publishing of RFNC VNIITF, 2017. P. 326. (in Russian).
4. Chevalier C., Pellegrini F. PT-Scotch: A Tool for Efficient Parallel Graph Ordering. *Parallel Computing*. 2008. Vol. 34, no. 6. P. 318–331. DOI: 10.1016/j.parco.2007.12.001.
5. Karypis G. METIS and ParMETIS. *Encyclopedia of Parallel Computing*. Springer US, 2011. P. 1117–1124. DOI: 10.1007/978-0-387-09766-4_500.
6. Toro E. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*. Berlin, Heidelberg: Springer Verlag, 2009. 721 p. DOI: 10.1007/b79761.
7. Darwish M., Moukalled F. TVD-schemes for Unstructured Grids. *International Journal of Heat and Mass Transfer*. 2003. Vol. 46, no. 8. P. 599–611. DOI: 10.1016/S0017-9310(02)00330-7.
8. Glazyrin I.V., Mikhailov N.A. Finite-Volume Scheme for Multicomponent Compressible Flow on Unstructured Meshes in the Focus 3D Code. *JCMMP*. 2021. Vol. 61, no. 6. P. 1019–1033. (in Russian) DOI: 10.31857/S0044466921060041.
9. Balashov N.M., Glazyrin I.V., Mikhaylov N.A. Two-dimensional Adaptive Mesh Refinement in Gas Dynamics Program. *Parallel Computational Technologies – XIV International Conference, PCT'2020, Perm, 31 March – 2 April 2020. Short articles and posters' description*. Chelyabinsk: Publishing of the South Ural University, 2020. P. 325. (in Russian).
10. Jasak H., Tukovic Z. Automatic Mesh Motion for the Unstructured Finite Volume Method. *Transactions of FAMENA*. 2006. Vol. 30, no. 2. P. 1–20.
11. Colella P., Berger M. Local Adaptive Mesh Refinement for Shock Hydrodynamics. *Journal of Computational Physics*. 1989. Vol. 82, no. 1. P. 64–84. DOI: 10.1016/0021-9991(89)90035-1.
12. Glazyrin I.V., Glazyrina N.V., Mikhaylov N.A., Pisklova M.A. Using Dynamic Mesh Adaptation for Rayleigh-Taylor Instability Calculations. *Zababakhin Scientific Readings: Collection of Materials XVI International Conference, 29 May – 2 June 2023*. Snezhinsk: Publishing of RFNC VNIITF, 2023. P. 364. (in Russian).
13. Olenov N.N. *Basics of MPI Parallel Programming*. Moscow: Calculating center A.A. Dorodnitsina, 2005. 79 p. (in Russian).
14. Joggerst C.C., Nelson A., Woodward P., *et al.* Cross-code Comparison of Mixing During the Implosion of Dense Cylindrical and Spherical Shells. *Journal of Computational Physics*. 2014. Vol. 82, no. 1. P. 154–173. DOI: 10.1016/j.jcp.2014.06.037.