

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ЧИСЛЕННЫЕ МЕТОДЫ

УДК 519.6

БАЛАНСИРОВКА НАГРУЗКИ С ИСПОЛЬЗОВАНИЕМ ВЗВЕШЕННЫХ ДИАГРАММ ВОРОНОГО

© 2023 Р.В. Муратов^{1,*}, П.Н. Рябов^{1,**}, С.А. Дьячков²

¹Национальный исследовательский ядерный университет МИФИ, Москва, 115409, Россия

²Всероссийский научно-исследовательский институт автоматики им. Н.Л. Духова,
Москва, 101000, Россия

*e-mail: RVMuratov@mephi.ru

**e-mail: pnryabov@mephi.ru

Поступила в редакцию: 27.03.2023

После доработки: 06.04.2023

Принята к публикации: 14.04.2023

Решение задачи балансировки нагрузки актуально при проведении высокопроизводительных вычислений. Одним из оригинальных методов решения задачи балансировки является использование диаграмм Вороного для декомпозиции области. Сильной стороной данного подхода является возможность применения такой декомпозиции на различных типах данных. Балансировка может использоваться на эйлеровых сетках, лагранжевых сетках, совместно с методикой сглаженных частиц (SPH). Метод также может использоваться для балансировки адаптивных сеток, но в этом случае проявляется его недостаток: из-за сильной неоднородности нагрузки на адаптивных сетках метод динамической балансировки может быть неустойчивым или обладать плохой сходимостью. В работе предлагается усовершенствование алгоритма балансировки за счет перехода от использования обычных диаграмм Вороного к взвешенным диаграммам. Предложенный алгоритм реализован в виде программного комплекса. Тестирование предложенного алгоритма строится на базе ряда модельных задач и задач из области механики сплошной среды. Оценка эффективности балансировки строится на базе исследования поведения величины дисбаланса с учетом балансировки и без. Показано, что предложенный алгоритм успешно справляется со своей задачей, и величина дисбаланса в случае использования взвешенной диаграммы Вороного в 10–100 раз меньше, чем при использовании обычных диаграмм Вороного.

Ключевые слова: параллельные вычисления, балансировка нагрузки, взвешенные диаграммы Вороного, механика сплошной среды, численное моделирование.

DOI: 10.26583/vestnik.2023.252

1. ВВЕДЕНИЕ

Эффективное распределение вычислительных ресурсов во время выполнения программы называется динамической балансировкой нагрузки. Балансировка нагрузки требуется при проведении вычислений с использованием параллельности в системах с распределенной памятью. В данной статье балансировка нагрузки обсуждается в контексте вычислительных задач механики сплошных сред. Параллелизация с распределенной памятью при численном моделировании эволюци-

онных задач динамики сплошных сред обычно строится следующим образом.

Сначала вычислительная область разбивается на дискретные элементы: это могут быть ячейки эйлеровой или лагранжевой сеток, сглаженные частицы для метода SPH (*smooth- particles hydrodynamic*) или более экзотические элементы. Затем в основном цикле программы выполняются следующие этапы:

1. Выполняется шаг интегрирования по времени, при этом каждый вычислительный узел производит вычисление только на своей части элементов;

2. Осуществляется добавление новых элементов или удаление старых, изменение топологии сетки и другие операции;

3. Происходит перераспределение элементов между вычислительными узлами для выравнивания нагрузки на процессах.

На каждой итерации требуется синхронизация данных на всех процессах, обычно синхронизация осуществляется только для элементов, которые граничат с элементами на другом процессе. Множество элементов, для которых на каждом шаге требуется синхронизация, будем называть обменным слоем. Величина обменного слоя пропорциональна длине/площади границы между смежными областями. При оптимальном использовании возможностей параллельных вычислительных систем все вычислительные процессы должны выполнять свою часть работы за одинаковое время. В этом случае ни один процесс не будет простаивать в ожидании до следующей синхронизации. Сильный дисбаланс сводит на нет преимущества параллельности в приложении.

Важно отметить, что балансировка нагрузки не сводится к равномерному распределению дискретных элементов между вычислительными узлами. Т.е. вычислительная нагрузка на процесс не является линейной функцией от числа элементов в области. На время выполнения одной итерации может сильно влиять топология отдельной части сетки, длина границы области и многие другие факторы.

Динамическая балансировка требуется при наличии сильной неоднородности в структуре приложения. Неоднородность нагрузки может быть обусловлена следующими причинами:

1. Неоднородной нагрузкой от различных дискретных элементов. Такая ситуация возникает в тех случаях, когда расчет одной части элементов существенно сложнее, чем других. К примеру, когда для части элементов используется простое уравнение состояния идеального газа, в то время как для другой части используется табличное уравнение состояния;

2. Неоднородной нагрузкой на различные области пространства. Во многом похоже на предыдущий пункт, но возможны и другие варианты. К примеру, загруженными могут оказаться следующие области: область сильной деформации на лагранжевой сетке, где часто требуется перестройка сетки и переинтерполяция, область с сопряжением нескольких вычислительных методик

(лагранжевы сетки взаимодействуют с эйлеровыми), области с сеточной адаптацией (вызов дорогостоящих функций разбиения и огрубления ячеек);

3. Неоднородным распределением элементов в пространстве: сгущением сетки или частиц.

Последний случай является одним из самых распространенных, особенно при моделировании лагранжевых элементов (подвижных ячеек или частиц), когда они сгущаются. Но наиболее ярко проблема выражается на адаптивных сетках. При использовании сеточной адаптации достигается огромный дисбаланс в плотности элементов. К примеру, рассмотрим адаптацию на двумерной сетке из четырехугольных элементов. Каждый вызов адаптации разбивает ячейку на четыре части, таким образом, для пятого уровня адаптации плотность ячеек в адаптируемой области в $2^{10} \approx 1000$ раз больше, чем в области с базовой сеткой.

Разработка алгоритмов балансировки нагрузки для адаптивных сеток является основной целью данной работы, хотя предложенный алгоритм может быть применен и к другим структурам данных. При балансировке динамических адаптивных сеток возникают большие сложности. Дело в том, что адаптивные сетки (даже эйлеровы) в некотором смысле являются более подвижными, чем лагранжевы сетки или сглаженные частицы, поскольку уровень адаптации может перемещаться по адаптивной сетке с произвольной скоростью. При этом постоянно меняется топология сетки, связи между ячейками постоянно перестраиваются. Выбирать методику динамической балансировки нагрузки следует с учетом этих сложностей. Далее перечислим некоторые распространенные подходы для декомпозиции областей и балансировки нагрузки на них, выделим сильные и слабые стороны различных подходов.

Для начала рассмотрим методы декомпозиции на основе графов. Любую расчетную сетку можно представить в виде графа, где ячейки – вершины графа, а смежность ячеек – ребра графа. В виде графа можно представить даже набор сглаженных частиц: в этом случае частицы являются узлами графа, а ребра – связями между частицами. В случае с частицами граф будет более плотным, поскольку каждая сглаженная частица может иметь до сотни связей. Также вершинам графа и ребрам можно добавить весовые коэффициенты,

что позволяет создавать очень точную модель сеточных данных в виде графа.

Задача декомпозиции области сводится к задаче разбиения графа: требуется разбить граф на подграфы, которые имеют одинаковое число вершин и минимальную границу. Разбиение графа является *NP*-полной задачей, поэтому точное решение задачи не осуществимо за разумное время. В то же время существует большое количество эвристических алгоритмов, решающих данную задачу. Одна из популярных библиотек разбиения графов ParMETIS [6] используется преимущественно для декомпозиции сеток. Декомпозиция графа в некотором смысле является самой оптимальной для сеток, но, очевидно, требует для построения много времени. Разбиение графов выгодно использовать для декомпозиции стационарных задач, в которых топология сетки не меняется в ходе расчета. Но для динамической балансировки, когда необходима частая перестройка декомпозиции, алгоритмы плохо применимы, в этом случае декомпозиция области будет занимать большую часть расчетного времени.

Далее вспомним алгоритмы декомпозиции области типа рекурсивной координатной бисекции. В алгоритмах такого типа топология сетки не используется, способ декомпозиции чисто геометрический. Метод можно использовать на произвольных данных, если дискретным элементам можно присвоить некоторую координату. Классический алгоритм координатной бисекции для двумерного случая: вычислительная область разбивается прямой $y = \text{const}$ на две части с равным числом элементов, затем каждая из подобластей также разбивается на две части прямой $x = \text{const}$ с равным количеством элементов, разбиение применяется к подобластям рекурсивно, пока не будет получено необходимое количество областей.

Данный алгоритм несложно обобщить на произвольное число областей, существует множество вариаций и улучшений исходного алгоритма. Данный алгоритм уже будет работать существенно быстрее разбиения графа. Асимптотическая сложность алгоритма $O(\log N)$: столько операций необходимо, чтобы отсортировать элементы вдоль некоторой оси. Преимущество метода в том, что он полностью геометрический, а, значит, работает на любых данных. Недостаток по-прежнему в достаточно высокой вычислительной сложности. Также метод позволяет учесть только

количество элементов, но не реальную нагрузку в областях.

Для адаптивных сеток классическим алгоритмом декомпозиции является метод с использованием кривой Мортонa (z -кривой). Эта кривая отображает многомерные деревья в одномерную структуру. Кривая Мортонa проходит по всем ячейкам адаптивной сетки, упорядочивая их некоторым образом. Кривая строится за один обход по дереву ячеек, z -кривая сохраняет локальность, т.е. соседние ячейки сетки, вероятнее всего, будут друг рядом с другом на кривой. Для декомпозиции кривая просто разбивается на равные части по количеству ячеек. Такой метод декомпозиции максимально изящный и быстрый. Метод используется в популярной библиотеке *p4est*. Однако данный метод не применим к декомпозиции сглаженных частиц или лагранжевой сетки, также он завязан на число частиц, что не всегда является оптимальным.

Далее обсудим алгоритмы диффузионного типа. Вероятно, одним из первых алгоритм балансировки такого типа предложил Цибенко [1], который использовал алгоритм для балансировки нагрузки в сети. В алгоритмах диффузного типа нагрузка передается между вычислительными процессами небольшими частями против градиента нагрузки. В некотором смысле нагрузка в сети перераспределяется по аналогу закона Фурье для теплопроводности. Как и в уравнении диффузии, в результате итерационного применения диффузионного алгоритма нагрузка равномерно распределяется в области. В качестве примера можно привести диффузионный аналог рекурсивной ортогональной бисекции из работы [16]. В данной работе границы прямоугольных блоков перемещаются пропорционально градиенту нагрузки.

Одним из интересных вариантов диффузных алгоритмов является алгоритм балансировки нагрузки на основе диаграмм Вороного. Вероятно, одним из первых такой подход применил Корради [7] при численном моделировании в области молекулярной динамики. Диаграмма Вороного для множества из n точек (называемых генераторами) представляет собой разбиение области на n подобластей. То есть, по определению, декомпозиция области на n частей полностью определяется набором из n точек. Алгоритм балансировки нагрузки заключается в перемещении генераторов ячеек Вороного в направлении областей с вы-

сокой нагрузкой, что в результате приводит к равномерному распределению нагрузки.

Основным преимуществом метода является то, что он является чисто геометрическим, с его помощью можно провести декомпозицию ячеек, сглаженных частиц и других элементов. При этом алгоритм не завязан на количество дискретных элементов в области, обновление области может быть основано на времени вычислений на каждом процессе. Недостаток метода, который будет продемонстрирован в данной статье, заключается в том, что метод плохо справляется с существенной неоднородностью нагрузки. Метод зарекомендовал себя в расчетах по методике SPH [3], но в таких расчетах плотность расположения частиц обычно совпадает с физической плотностью сплошной среды, т.е. редко встречаются перепады плотности во много раз. Но для адаптивных сеток это является принципиальным, как мы выяснили, плотность дискретных элементов может отличаться в тысячи раз.

Ячейки Вороного классической диаграммы редко сильно отличаются по размерам; вероятно, в алгоритм можно добавить нелинейность, чтобы ячейки могли иметь существенно различные размеры. В данной работе мы попытаемся усовершенствовать схему балансировки обычной диаграммы и перейти к балансировке с использованием взвешенных диаграмм Вороного. Также будут продемонстрированы задачи, в которых взвешенные диаграммы справляются с балансировкой лучше по сравнению с обычными диаграммами.

В первом разделе работы приводится математическое описание диаграмм Вороного и обсуждаются некоторые их свойства. Во втором разделе изложен алгоритм балансировки диаграмм. В третьем разделе проводится тестирование алгоритма на серии искусственно сконструированных задач. Четвертый раздел содержит попытку применения алгоритма балансировки для декомпозиции сеток, полученных в реальных задачах механики сплошных сред.

2. ВЗВЕШЕННЫЕ ДИАГРАММЫ ВОРОНОГО

Пусть в m -мерном евклидовом пространстве R^m задано множество из n различных точек $G = \{g_i\}_{i=1}^n$. *Диаграммой Вороного* множества точек G называется разбиение некоторой области Ω пространства на n подобластей $\{\Omega_i\}_{i=1}^n$, при кото-

ром каждая подобласть Ω_i состоит из точек, которые ближе к g_i , чем к любой другой точке из G . Подобласти разбиения Ω_i называются ячейками диаграммы Вороного, а точки g_i – генераторами или центрами ячеек Вороного. Простыми словами – ячейка Ω_i содержит точки, для которых генератор g_i ближе всех остальных генераторов:

$$\begin{aligned} \Omega_i &= \{x \notin \Omega: \forall j \in \overline{1, n} \Rightarrow \\ &\Rightarrow |x - g_i| \leq |x - g_j|\}. \end{aligned} \quad (2.1)$$

Классическая диаграмма Вороного показана на рис. 2.1,а. Отметим некоторые простые свойства диаграмм Вороного:

- границы ячеек диаграммы Вороного образованы гиперплоскостями (прямыми линиями в двумерном случае);
- ячейки Вороного являются выпуклыми многогранниками;
- грань между двумя смежными ячейками Вороного перпендикулярна отрезку, который соединяет генераторы этих ячеек, и равноудалена от генераторов этих ячеек.

Все три свойства следуют непосредственно из определения.

В приведенном выше определении (2.1) диаграмма Вороного строится на основе естественной метрики в евклидовом пространстве R^m , аналогичным образом можно определить диаграмму Вороного в произвольном метрическом пространстве. Более того, для построения диаграммы достаточно определить только «расстояние» от точки пространства x до каждого генератора g_i . Обозначим расстояние как $d(x, g_i)$ и определим ячейки диаграммы следующим образом:

$$\begin{aligned} \Omega_i &= \{x \in \Omega: \forall j \in \overline{1, n} \Rightarrow \\ &\Rightarrow d(x, g_i) \leq d(x, g_j)\}. \end{aligned} \quad (2.2)$$

В приложениях часто используются диаграммы, в которых расстояния до генераторов определяются с использованием весовых коэффициентов генераторов. Такие диаграммы называют взвешенными. Среди взвешенных диаграмм наиболее часто используются диаграммы двух типов: аддитивные и мультипликативные взвешенные диаграммы.

Пусть помимо множества генераторов ячеек Вороного $\{g_i\}_{i=1}^n$ задано также множество весов $\{\omega_i\}_{i=1}^n$ соответствующих генераторам ячеек.

Определим расстояние в мультипликативной взвешенной диаграмме по формуле

$$d(x, g_i) = \omega_i |x - g_i|,$$

где весовые коэффициенты должны быть положительными.

Диаграмма остается неизменной при умножении всех весовых коэффициентов на произвольную положительную константу. Границы ячеек Вороного в мультипликативной диаграмме имеют форму окружностей. При значении весов генераторов больше нуля ячейки никогда не вырождаются в точку, т.е. всегда содержат некоторую окрестность своего генератора. Пример мультипликативной диаграммы приведен на рис. 2.1,б.

Ячейки Вороного мультипликативной диаграммы не всегда являются односвязными областями. В этом можно убедиться при рассмотрении простейшего случая: диаграммы для двух генераторов с различными весами на плоскости. Поскольку граница между ячейками является окружностью, диаграмма Вороного для двух ге-

нераторов будет разделять плоскость на две части: некоторый круг и внешность круга (внешность круга является двусвязной). При большом количестве генераторов возможно получить диаграммы, в которых ячейки вовсе не являются связными областями.

Возможная неодносвязность ячеек мультипликативной диаграммы повлияла на выбор аддитивных диаграмм для декомпозиции двумерных областей. Односвязность и выпуклость областей является более предпочтительной, поскольку многие алгоритмы работают наиболее эффективно в выпуклых односвязных областях с минимальной границей (т.е. близких к кругу). Во введении уже упоминалось, что уменьшение границы между ячейками позволяет уменьшить обменный слой и издержки на пересылку данных между различными процессами.

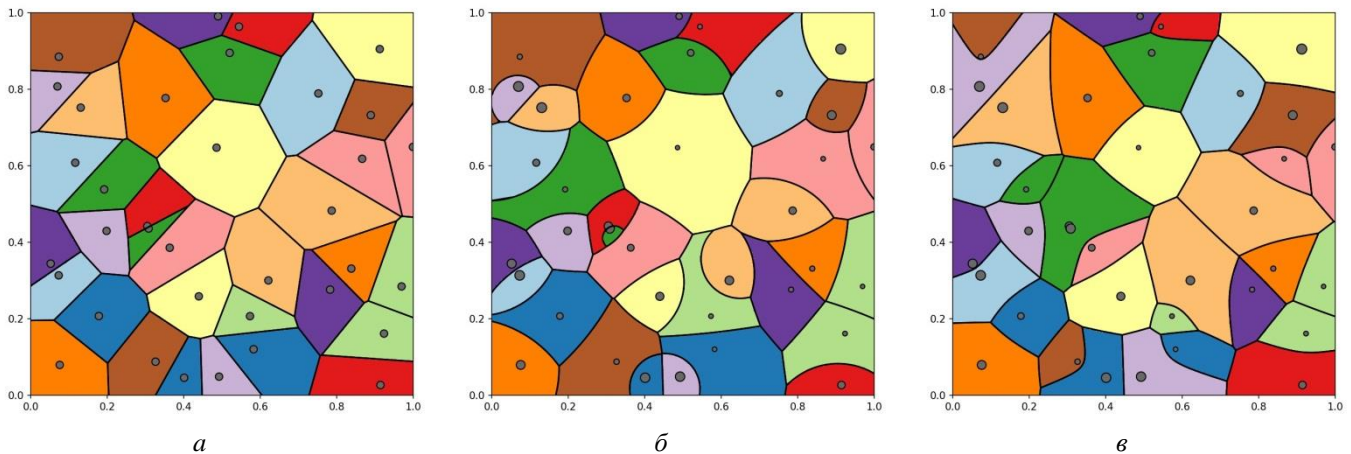


Рис. 2.1. Сравнение трех типов диаграмм. Генераторы всех диаграмм заданы случайно, но совпадают для всех диаграмм, весовые коэффициенты также задаются случайно:
а) классическая диаграмма Вороного; б) мультипликативная диаграмма Вороного;
в) аддитивная диаграмма Вороного

Для аддитивной диаграммы расстояние вводится по формуле

$$d(x, g_i) = |x - g_i| - \omega_i. \quad (2.3)$$

Границы ячеек аддитивной диаграммы Вороного имеют форму гипербол с фокусами в центрах ячеек. Весовые коэффициенты могут принимать как положительные, так и отрицательные значения. При изменении всех весов на константу диаграмма не изменяется, поэтому без ограничения общности можно рассматривать диаграммы только с положительными весовыми коэффициентами или диаграммы, в которых сумма весовых коэффициентов равна нулю.

При равенстве весов у всех генераторов обе взвешенные диаграммы переходят в классическую. А при равенстве весов у двух соседних генераторов граница между соответствующими ячейками переходит в прямую. На рис. 2.2 показаны три вида диаграмм. На диаграммах кругами обозначены генераторы ячеек Вороного, для всех трех диаграмм положения генераторов совпадают. Радиусы меток генераторов пропорциональны весам генераторов.

Как говорилось ранее, граница между смежными ячейками равноудалена от их центров. Такое строгое положение границы наводит на мысль о том, что диаграммы Вороного не явля-

ются универсальным способом представления сбалансированной декомпозиции области. Продемонстрируем данный факт в одномерном случае.

Пусть на отрезке $[a, b]$ задана интегрируемая функция $\rho(x) > 0$ (плотность нагрузки), суммарная нагрузка $L = \int_a^b \rho(x)dx$. Сбалансированной декомпозицией области будем называть разбиение отрезка на n подобластей набором точек $a = x_0 < x_1 < \dots < x_n = b$, при котором нагрузка на каждую часть равна

$$\forall i = \overline{1, n}, \quad \int_{x_{i-1}}^{x_i} \rho(x)dx = \frac{L}{n}. \quad (2.4)$$

Несложно доказать следующие утверждения.

Утверждение 1. Для произвольной функции $\rho(x) > 0$, интегрируемой на отрезке $[a, b]$, существует единственная сбалансированная декомпозиция.

Действительно, введем первообразную $F(x) = \int_a^x \rho(x)dx$. Первообразная $F(x)$ является строго возрастающей функцией и принимает на отрезке $[a, b]$ значения от 0 до L . Тогда, по теореме о

промежуточном значении, для любого $k \in \{1, \dots, n-1\}$ существует единственное значение x_k такое, что $F(x_k) = kL/n$.

Утверждение 2. Не любая сбалансированная декомпозиция $\{x_i\}_{i=0}^n$ отрезка может быть представлена в виде классической диаграммы Вороного.

Проще говоря, не любой упорядоченный набор точек на прямой является границами некоторой одномерной диаграммы Вороного. Обозначим координаты генераторов ячеек Вороного на прямой как $\{g_i\}_{i=1}^n$, где $g_i \in (x_{i-1}, x_i)$, $i = \overline{1, n}$. Границы ячеек Вороного равноудалены от генераторов, а, значит, величины должны удовлетворять условию $g_i + g_{i+1} = 2x_i$ для i от 1 до $n-1$. Итого, имеется система из $n-1$ линейных уравнений на n неизвестных g_i с ограничениями в виде неравенств $g_1 < g_2 < \dots < g_n$. Данная система не всегда имеет решение, пример декомпозиции отрезка, которая не может быть представлена в виде одномерной диаграммы Вороного, представлен на рис. 2.2,б.

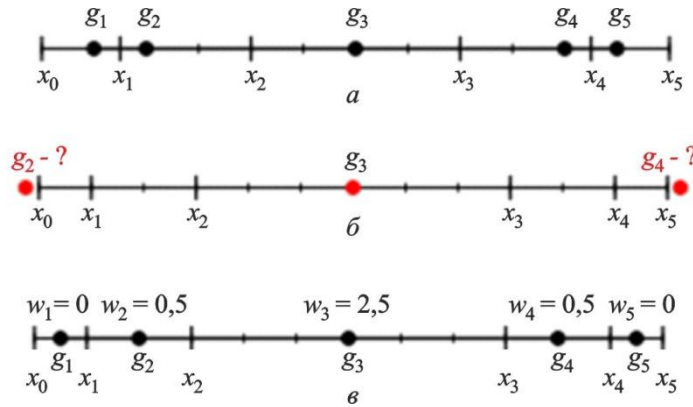


Рис. 2.2. Диаграммы Вороного в одномерном случае:

а) одномерная диаграмма Вороного; б) декомпозиция отрезка не представима в виде классической диаграммы Вороного; в) декомпозицию можно представить в виде аддитивной диаграммы Вороного

Из утверждений выше следует, что классическая диаграмма Вороного не всегда способна сбалансировать нагрузку в одномерном случае. А пример ко второму утверждению показывает, что проблемы возникают при сильном дисбалансе нагрузки, когда требуется, чтобы смежные ячейки Вороного отличались более, чем в два раза. Есть предположение, что добавление новых параметров, таких как веса ячеек Вороного, позволит сбалансировать любое распределение нагрузки в одномерном случае.

Гипотеза. Любая сбалансированная декомпозиция $\{x_i\}_{i=0}^n$ отрезка может быть представлена в виде взвешенной аддитивной диаграммы Вороного.

Авторам неизвестно доказательство данной гипотезы, но она звучит довольно убедительно. Положения генераторов $\{g_i\}_{i=1}^n$ и веса $\{\omega_i\}_{i=1}^n$ должны удовлетворять системе из $n-1$ линейных уравнений вида $g_i + g_{i+1} + \omega_i - \omega_{i+1} = 2x_i$, $i = \overline{1, n-1}$, при этом теперь у нас $2n$ переменных, а количество ограничений не измени-

лось (веса задаются произвольно). Пример декомпозиции отрезка с аддитивной диаграммой Вороного показан на рис. 2.2, в.

Утверждения выше не следует экстраполировать с одномерного случая на двумерный или трехмерный. Возможно и такое, что двумерные диаграммы способны сбалансировать практически любое распределение нагрузки в области. Однако одномерные примеры позволяют выявить слабое место при декомпозиции классическими диаграммами Вороного: балансировка сильно неоднородной нагрузки.

3. АЛГОРИТМ БАЛАНСИРОВКИ НАГРУЗКИ

Сформулируем математически задачу балансировки нагрузки. Пусть задана область в m -мерном пространстве $\Omega \in R^m$, в которой определена интегрируемая функция плотности нагрузки $\rho(x) > 0$. Нагрузка μ на определенную подобласть V может быть вычислена как объемный интеграл:

$$\mu(V) = \int_V \rho(x) dx. \quad (3.1)$$

Для некоторой декомпозиции области $\{\Omega_i\}_{i=1}^n$ можно определить дисбаланс нагрузки по формуле

$$I = \frac{\max_i L_i}{\bar{L}} - 1, \quad \bar{L} = \frac{1}{n} \sum_{i=1}^n L_i = \frac{\mu(\Omega)}{n}. \quad (3.2)$$

Требуется построить в области Ω классическую или взвешенную диаграмму Вороного, для которой дисбаланс I минимален.

Алгоритм нахождения декомпозиции должен удовлетворять некоторым условиям. Алгоритм должен быть сконструирован в виде итерационной процедуры, которая бы приводила произвольную диаграмму Вороного к сбалансированной, при этом подразумевается, что вид функции $\rho(x)$ нам не известен, а на каждой итерации в распоряжении есть только нагрузка L_i на каждую из ячеек Вороного.

В прикладных задачах нагрузка на процесс L_i может считаться произвольным образом. К примеру, в качестве L_i можно выбрать количество дискретных элементов внутри области Ω_i . Если в расчете встречаются элементы нескольких типов, которые считаются по-разному, то в качестве нагрузки следует рассматривать линейную комбинацию от числа элементов каждого типа.

Наиболее естественным способом определения нагрузки L_i является время выполнения процессом одной итерации (из этого времени необходимо вычесть время на пересылку, ожидание и синхронизацию данных между процессами). Поскольку балансировка проводится один раз за несколько итераций, хорошей мерой нагрузки является среднее время выполнения одной итерации.

Если в качестве нагрузки рассматривается время, то дисбаланс, введенный по формуле (3.2), имеет определенный физический смысл. Если предположить, что мы можем перераспределить всю нагрузку $\sum_i L_i$ равномерно по \bar{L} на каждый процесс, тогда \bar{L} показывает, сколько времени будет потрачено на одну итерацию, а $\max_i L_i$ показывает, сколько времени тратится на итерацию при текущей декомпозиции. Таким образом, дисбаланс I показывает, на сколько процентов текущая декомпозиция хуже идеальной. К примеру, если в расчете сохраняется дисбаланс I около 50 %, это значит, что расчет будет длиться на 50 % дольше, чем мог бы при идеальном балансе.

Наибольший дисбаланс достигается в том случае, когда вся нагрузка оказывается сосредоточена на одном процессе, в то время как другие простаивают. В случае, когда расчет проводится на n процессорах, худший дисбаланс составляет $I = n - 1$. На практике удовлетворительным является показатель дисбаланса порядка 2–10 %, на такой показатель нацелен предложенный алгоритм балансировки.

3.1. Шаг 1. Базовый принцип перемещения генераторов ячеек

Построение алгоритма начнем с балансировки с использованием только перемещений генераторов ячеек Вороного. Данный метод применим для балансировки классической диаграммы Вороного, общие черты алгоритма заимствованы у авторов [3, 17]. В основе алгоритма обычная схема итерации

$$g_i^{n+1} = g_i^n + \Delta g_i \quad (3.3)$$

зависит только от нагрузки в i -й ячейке Вороного и смежных с ней ячеек. Обозначим как $\omega(i)$ множество смежных ячеек к i -й. Алгоритм является локальным в том смысле, что для определения смещения генератора g_i требуется только информация с соседних ячеек. Величина Δg_i должна стремиться к нулю при стремлении дисбаланса

к нулю, это является необходимым условием сходимости алгоритма к нулевому дисбалансу.

Следуя идеям оригинального метода, смещения генераторов должны быть пропорциональны дисбалансу между соседними ячейками. Введем относительный дисбаланс нагрузки между соседними ячейками Вороного:

$$I_{ij} = \frac{L_j - L_i}{L_j + L_i}. \quad (3.4)$$

Можно выделить несколько бесполезных фактов об этой величине: функция является антисимметричной, обращается в нуль тогда и только тогда, когда нагрузки на смежные ячейки равны, модуль относительного дисбаланса меньше единицы:

$$|I_{ij}| \leq 1, I_{ij} = -I_{ji}, I_{ij} = 0 \Leftrightarrow L_i = L_j. \quad (3.5)$$

Определим также максимальный относительный дисбаланс для i -го процесса:

$$I_{i,\max} = \max_{j \in \omega(i)} |I_{ij}|. \quad (3.6)$$

Это значение характеризует то, насколько сильно ячейка выделяется среди своих соседей.

Зная относительный дисбаланс между соседними ячейками, считаем следующую величину:

$$\delta g_i = \sum_{j \in \omega(i)} I_{ij} \frac{g_j - g_i}{|g_j - g_i|}. \quad (3.7)$$

Величина δg_i является безразмерной и указывает направление с максимальной нагрузкой. Фактически это безразмерный градиент. Смещение Δg_i должно быть пропорционально δg_i но эту величину требуется ограничить. Введем ограничитель:

$$\Delta g_i = v_g D_{g,i} H(I_{i,\max}) \delta g_i. \quad (3.8)$$

Здесь v_g – некоторая константа меньше единицы, $D_{g,i}$ – ограничение на смещение генератора, единственная размерная величина (имеет размерность длины); $H(I)$ – некоторая функция от 0 до 1, которая регулирует скорость сходимости.

Функция $H(I)$ необходима, чтобы замедлить движение генераторов при небольших значениях дисбаланса. При маленьких значениях $I_{i,\max} \ll 1$ нет необходимости делать какую-либо балансировку, лучше совсем не двигать генератор для улучшения стабильности алгоритма.

Можно подбирать различные варианты функции $H(I)$, но она должна обладать некоторыми свойствами. $H(I)$ должна монотонно возрастать,

находиться в пределах от 0 до 1: $H(1) \approx 1$; также желательно, чтобы $H(0) = 0$. Можно ничего не придумывать и использовать просто $H(I) = 1$. Будто функции и нет. В работе же используется

$$H(I) = \frac{I}{I + I_0}, I \geq 0. \quad (3.9)$$

Это правая половина простой сигмоидальной функции. Здесь I_0 – некоторый допустимый уровень дисбаланса, при таком значении дисбаланса $H(I_0) = 1/2$, а при меньших функция резко убывает. Таким образом, при достижении дисбаланса I_0 балансировка замедляется, что улучшает стабильность алгоритма. Как говорилось ранее, такого условия достаточно для практических целей: при уменьшении дисбаланса до 1 % нет смысла в дальнейшем улучшении, лучше поддерживать стабильность. В работе используется значение $I_0 = 10^{-2}$, но это не значит, что алгоритм остановится на дисбалансе 10^{-2} . В различных тестах будут продемонстрированы значения дисбаланса меньше 10^{-2} .

Ограничение $D_{g,i}$ имеет размерность длины – это минимум из нескольких величин: расстояния от генератора g_i до границы ячейки в направлении δg_i среднего размера каждой из смежных ячеек. Первое условие гарантирует, что генератор при смещении не покинет область ячейки, второе ограничение запрещает резкое движение генераторов крупных ячеек, когда вблизи них есть маленькие ячейки. В работе [3] роль размерного параметра $D_{g,i}$ в формулах играет ширина обменного слоя между смежными ячейками. Но это условие привязывает алгоритм к расчетной методике, а также может сильно замедлять сходимость, поскольку обменный слой часто имеет ширину существенно меньше характерных размеров ячейки.

Алгоритм для смещения генераторов, приведенный выше, подойдет для балансировки классических диаграмм Вороного.

3.2. Шаг 2. Принцип изменения весов у генераторов

Следуя тем же принципам, построим итерационную процедуру для обновления весов генераторов:

$$\omega_i^{n+1} = \omega_i^n + \Delta \omega_i. \quad (3.10)$$

Веса генераторов меняем пропорционально дисбалансу:

$$\delta \omega_i = \sum_{j \in \omega(i)} I_{ij}. \quad (3.11)$$

На веса также накладываем некоторые ограничения:

$$\Delta w_i = v_\omega D_{\omega,i} H(I_{i,\max}) \delta \omega_i, \quad (3.12)$$

здесь v_ω – некоторая константа меньше единицы, используется такая же функция $H(I)$, как при вычислении смещений генераторов. Ограничитель $D_{\omega,i}$ является размерным (имеет размерность длины) и описывает естественные границы изменения веса ячейки Вороного, которые будут определены далее.

Граница между смежными ячейками взвешенной диаграммы Вороного является гиперболой. В предельном случае, когда веса смежных генераторов сильно отличаются, гипербола вырождается в прямую, и одна ячейка полностью поглощает другую. В этом случае одна из ячеек Вороного становится пустым множеством:

$$|\omega_i - \omega_j| < |g_i - g_j|. \quad (3.13)$$

Таким образом, естественными ограничениями для весов можно считать условия, при которых ни один генератор не поглощает другой.

Обозначим α наименьший угол между асимптотой гиперболы и осью, соединяющей фокусы гиперболы, $\alpha \in [0, \pi/2]$. Угол α обозначен на рис. 3.1. При поглощении одной ячейкой другой ячейки, ветви гиперболы сливаются, в этом случае угол $\alpha = 0$. При значении $\alpha = \pi/2$ гипербола переходит в прямую. Нас не интересуют сильно вытянутые ячейки Вороного, поэтому мы можем ввести минимальный угол α_0 , который допускается между ветвями гиперболы. Выясним, какие ограничения при этом накладываются на веса генераторов.

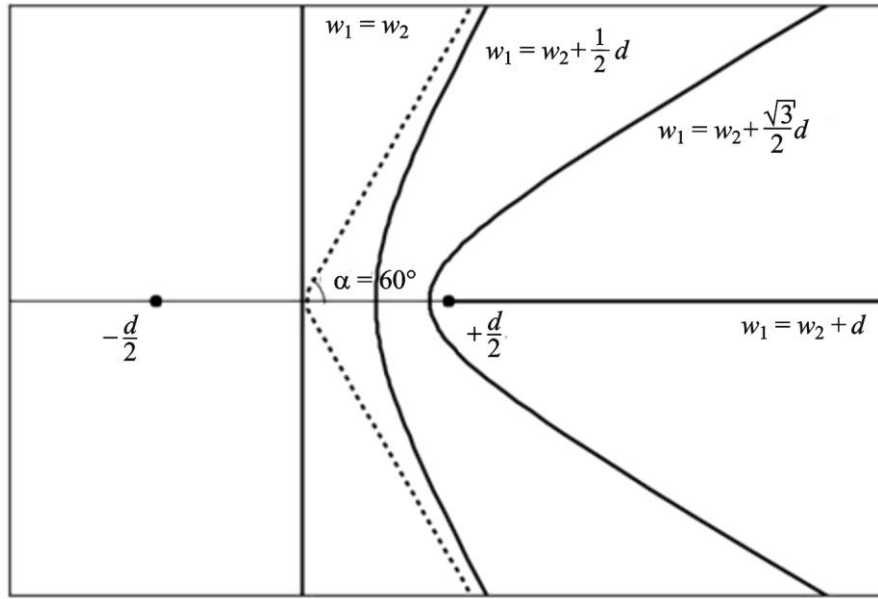


Рис. 3.1. Аддитивная диаграмма Вороного для двух точек

Построим диаграмму Вороного для двух генераторов, которые расположены на оси x слева и справа от начала координат на расстоянии $d/2$ и имеют веса ω_1 и ω_2 соответственно (см. рис. 3.1). Граница между ячейками Вороного описывается уравнением

$$\begin{aligned} & \sqrt{\left(x + \frac{d}{2}\right)^2 + y^2} - \omega_1 = \\ & = \sqrt{\left(x - \frac{d}{2}\right)^2 + y^2} - \omega_2. \end{aligned} \quad (3.14)$$

Уравнение двух ветвей гиперболы в каноническом виде:

$$\frac{4x^2}{(\omega_2 - \omega_1)^2} - \frac{4y^2}{d^2 - (\omega_2 - \omega_1)^2} = 1. \quad (3.15)$$

Уравнение справедливо при условии $d > |\omega_2 - \omega_1|$. Косинус угла α между асимптотой гиперболы и осью x :

$$\cos \alpha = \frac{|\omega_2 - \omega_1|}{d}. \quad (3.16)$$

Таким образом, угол между ветвью гиперболы и осью x больше α_0 при выполнении следующего условия на веса генераторов:

$$|\omega_i - \omega_j| \leq \cos \alpha_0 |g_i - g_j|. \quad (3.17)$$

При использовании $\alpha_0 = 0$, когда не допускается только полное поглощение, получаем условие (3.13). При значении $\alpha_0 = \pi/2$ получаем ограничение $|\omega_i - \omega_j| = 0$, при выполнении которого все генераторы диаграммы оказываются равны, и взвешенная диаграмма переходит в классическую.

Раскрывая модуль в условии (3.17) и применяя ограничения для всех смежных ячеек, получаем следующие ограничения на вес i -го генератора

$$\begin{aligned} \omega_{i,\min} &= \max_{j \in \omega(i)} (\omega_j - \cos \alpha_0 |g_i - g_j|), \\ \omega_{i,\max} &= \min_{j \in \omega(i)} (\omega_j + \cos \alpha_0 |g_i - g_j|). \end{aligned} \quad (3.18)$$

Здесь минимум и максимум не перепутаны. Если $\omega_i \in (\omega_{i,\min}, \omega_{i,\max})$, то граница между смежными ячейками будет иметь небольшой изгиб (угол между асимптотами гиперболы больше $2\alpha_0$).

После оценки предельных значений для ω_i можно ввести ограничитель $D_{\omega,i}$:

$$D_{\omega,i} = \begin{cases} \max(0, \omega_{i,\max} - \omega_i^n), & \delta\omega_i > 0, \\ \max(0, \omega_i^n - \omega_{i,\min}), & \delta\omega_i \leq 0. \end{cases} \quad (3.19)$$

Данная формула имеет довольно простой смысл: если вес на данном шаге увеличивается ($\delta\omega_i > 0$), то после изменения он не должен превысить $\omega_{i,\max}$, если же вес на данном шаге уменьшается, то он не должен стать меньше $\omega_{i,\min}$. Также в расчетах может произойти ситуация, когда $\omega_{i,\max} < \omega_{i,\min}$, в этом случае вес устанавливается на среднее значение между $\omega_{i,\min}$ и $\omega_{i,\max}$.

На этом описание алгоритма для обновления весов закончено.

3.3. Шаг 3. Смещение генераторов к центру масс

Дополнительной полезной опцией алгоритма является добавка смещения генераторов ячеек Вороного в сторону центра масс элементов в данной области.

Пусть c_i – координата центра масс дискретных элементов в области ячейки Вороного Ω_i (среднее

от координат элементов). Введем смещение Δ_2 в направлении от генератора к центру масс:

$$\Delta_2 g_i = c_i - g_i. \quad (3.20)$$

В качестве итогового смещения будем использовать взвешенную сумму

$$g_i^{n+1} = g_i^n + (1 - \sigma)\Delta_1 g_i + \sigma\Delta_2 g_i, \quad (3.21)$$

где $\Delta_1 g_i$ – смещение в соответствии с формулой (3.8); σ – параметр варьируется от 0 до 1. При $\sigma \rightarrow 0$ влияние смещения к центру масс стремится к нулю.

Смещение к центру масс также используется в работе [3], такая добавка выполняет сразу несколько полезных функций:

1. Связывает движение ячеек Вороного с движением лагранжевых элементов: если элементы перемещаются внутри области, то они увлекают за собой и генераторы ячеек Вороного;

2. Если неоднородность нагрузки связана с плотностью ячеек, тогда центр масс находится близко к плотным областям, генераторы ячеек Вороного устремляются туда.

3. Генераторы ячеек Вороного стремятся к центру масс ячеек Вороного, т.е. отходят от границы вглубь области. Это делает ячейки более выпуклыми и с более правильной формой, что повышает устойчивость алгоритма.

Преимущества использования смещения к центру масс будут рассматриваться далее.

Важно отметить, что при $\sigma > 0$ не решается исходная задача минимизации дисбаланса. То есть любой алгоритм при $\sigma > 0$ не будет минимизировать дисбаланс.

Фактически алгоритм будет сходиться к декомпозиции, при которой выполняется следующее условие:

$$(1 - \sigma)\Delta_1 g_i + \sigma\Delta_2 g_i = 0, \quad (3.22)$$

при котором уравниваются дисбаланс и тяга генератора к центру масс. Иначе говоря, такое решение является компромиссным. Чем больше σ , тем сильнее влияние стягивания генераторов к центру масс. При больших σ невозможно достичь хорошего баланса нагрузки, но наличие параметра σ сказывается положительно на декомпозиции.

Рассмотрим влияние параметра на примере.

В квадратной области $[0, 1]^2$ расположены точки с нормальным распределением, которые моделируют вычислительную нагрузку – всего 10^5 точек. Требуется построить декомпозицию

области на 36 процессов с использованием взвешенной диаграммы Вороного таким образом, чтобы дисбаланс (по числу точек внутри) был минимален.

Результаты балансировки для значений параметра $\sigma = 0, 10^{-1}, 10^{-3}$ приведены на рис. 3.2. На рисунках подписаны значения конечного дисбаланса (после 5000 итераций балансировки) во всех случаях. Результаты моделирования подтверждают утверждение: чем больше значение σ , тем больше дисбаланс. Но важно отметить, что дисбаланс незначительно растет при увеличении σ , даже при большом значении $\sigma = 0,1$ достигает-

ся приемлемый дисбаланс около 1 %. При этом, жертвуя идеальным балансом, мы получаем хорошие красивые ячейки. Главное преимущество в том, что генераторы ячеек находятся ближе к центру масс ячеек, при этом ячейки становятся более выпуклыми, круглыми, увеличивается стабильность алгоритма. Из примера видно, что для получения хорошего результата достаточно использовать очень маленькое значение σ порядка 10^{-3} . В следующем разделе влияние параметра σ будет рассмотрено более подробно.

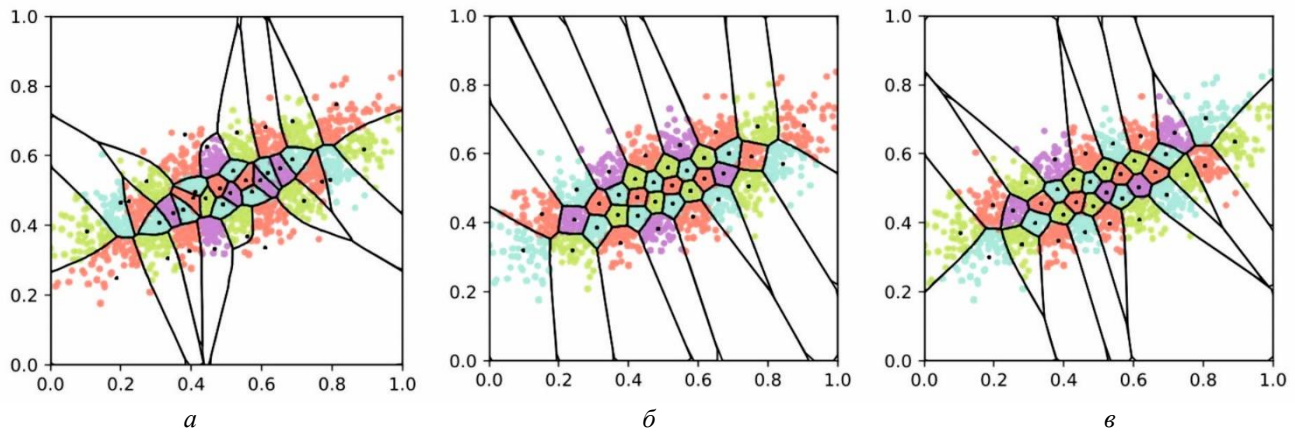


Рис. 3.2. Декомпозиция области при различных значениях параметра σ :

а) $\sigma = 0,0$, дисбаланс $I = 0,04$ %; б) $\sigma = 10^{-3}$, дисбаланс $I = 0,4$ %; в) $\sigma = 0,1$, дисбаланс $I = 1,2$ %

4. ВЕРИФИКАЦИЯ АЛГОРИТМА НА МОДЕЛЬНЫХ ЗАДАЧАХ

Для тестирования алгоритма попробуем решить несколько модельных задач. Рассмотрим квадратную область $\Omega \in [0, 1]^2$, которая содержит множество точек. Каждая точка моделирует нагрузку, т.е. для произвольной ячейки Вороного нагрузка определяется как количество точек, расположенных внутри ячейки. Требуется сбалансировать нагрузку (т.е. точки) в области с использованием классической или взвешенной диаграммы Вороного. В результате балансировки в каждой ячейке Вороного должно содержаться примерно одинаковое число точек.

Начальные положения генераторов диаграммы Вороного задаются случайным образом в области Ω . Начальный вес генераторов равен нулю. Итоговый дисбаланс проверяется на 5000 итерации балансировки нагрузки.

Далее используются параметры $\alpha_0 = 45^\circ$, $v_g = 0,2$, $v_\omega = 0,02$, $I_0 = 10^{-2}$, $\sigma = 10^{-3}$, вз-

де, где не оговорено обратное. Рассуждать, почему веса v_g , v_ω именно такие, нет смысла: такие значения были проверены для большинства задач.

4.1. Равномерное распределение нагрузки

Пусть точки, моделирующие нагрузку, равномерно распределены в области Ω . Общее число точек равно 10^5 , требуется получить декомпозицию области на 64 подобласти. В дискретном случае идеальная декомпозиция не возможна, для данной задачи при идеальном балансе на каждую ячейку Вороного должно приходиться по 1562.5 точек. Минимально возможный для данной задачи дисбаланс составляет $I = 1563/1562.5 - 1 = 0.032$ %.

На рис. 4.1, а, б показана итоговая декомпозиция при использовании классических и взвешенных диаграмм Вороного. Цветом выделены точки, моделирующие нагрузку (показаны не все, а только две тысячи случайных точек). Взвешенная

диаграмма Вороного слабо отличается от классической: границы ячеек близки к прямым, т.е. весовые коэффициенты слабо отличаются друг от друга.

На рис. 4.1,в показаны зависимости дисбаланса от итерации при балансировке двух типов диаграмм. Из графика следует, что, не смотря на слабые различия в итоговой декомпозиции, итоговый дисбаланс отличается для классической и взвешенной диаграмм. Итоговый дисбаланс для классической диаграммы колеблется около 1.9 %, а для взвешенной диаграммы составляет около 0.2 %. Дисбаланс 0.2 % соответствует разнице в три точки между соседними ячейками Вороного.

В любом случае это избыточная точность. Данный тест показывает, что с балансировкой равномерной нагрузки одинаково хорошо справляется как взвешенная диаграмма Вороного, так и обычная. Две зависимости на рис. 4.1,в идут рядом и слабо отличаются до достижения дисбаланса около 1 %.

4.2. Существенный дисбаланс нагрузки

Снова рассматривается квадратная область Ω , в которой распределены точки, моделирующие нагрузку, но теперь они имеют более сложное распределение. В квадрате выделены три круга, внутри которых плотность точек (а, соответственно, и плотность нагрузки) существенно больше, чем снаружи. В первом круге с центром (0.25, 0.25) и радиусом 0.15 плотность точек в 2^6 раз больше, во втором круге с центром (0.80, 0.40) и радиусом 0.15 плотность нагрузки в 2^6 раз больше, в третьем круге с центром (0.40, 0.80) и радиусом 0.1 плотность нагрузки в 2^8 раз больше. Всего в области располагается 10^5 точек.

Коэффициенты 2^8 и 2^6 выбраны не случайно. Во столько раз будет отличаться плотность ячеек на базовой сетке и на двумерной адаптивной сетке с четвертым или третьим уровнем адаптации. Плотность распределения нагрузки хорошо отражена на рис. 4.2–4.4. Как видно из графиков, большая часть точек сосредоточена в трех кругах.

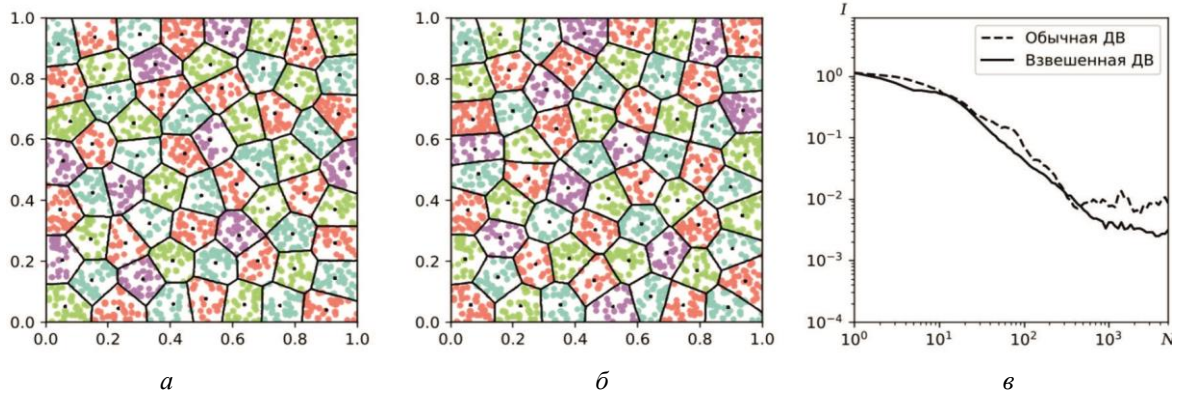


Рис. 4.1. Результат балансировки в модельной задаче с неравномерным распределением нагрузки: а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

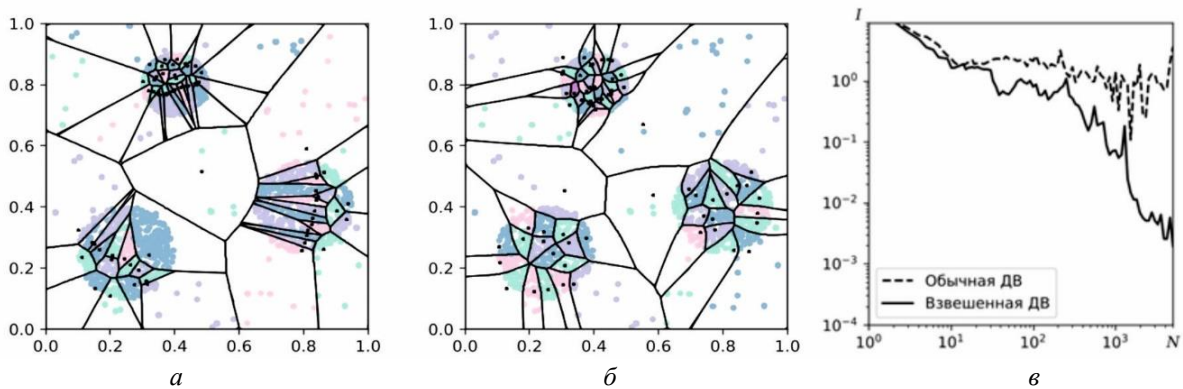


Рис. 4.2. Балансировка в тесте с существенным дисбалансом нагрузки, параметр $\sigma = 0$: а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

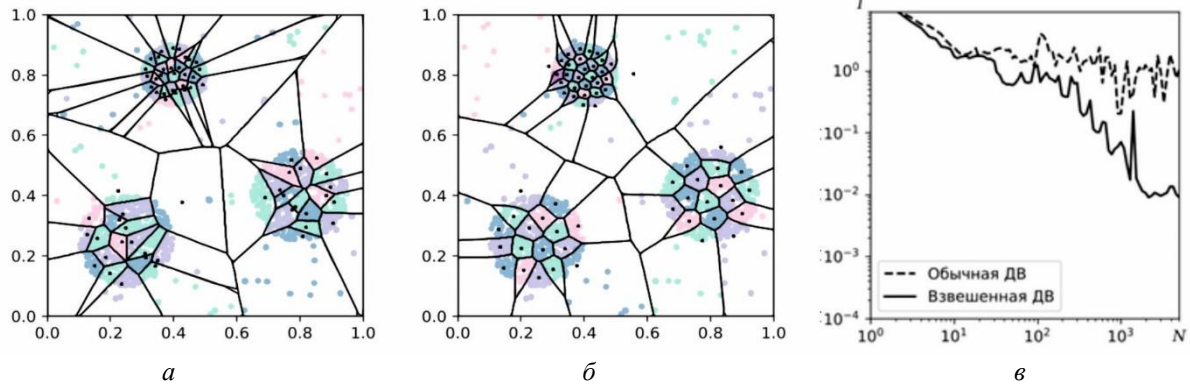


Рис. 4.3. Балансировка в тесте с существенным дисбалансом нагрузки, параметр $\sigma = 10^{-3}$:
а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

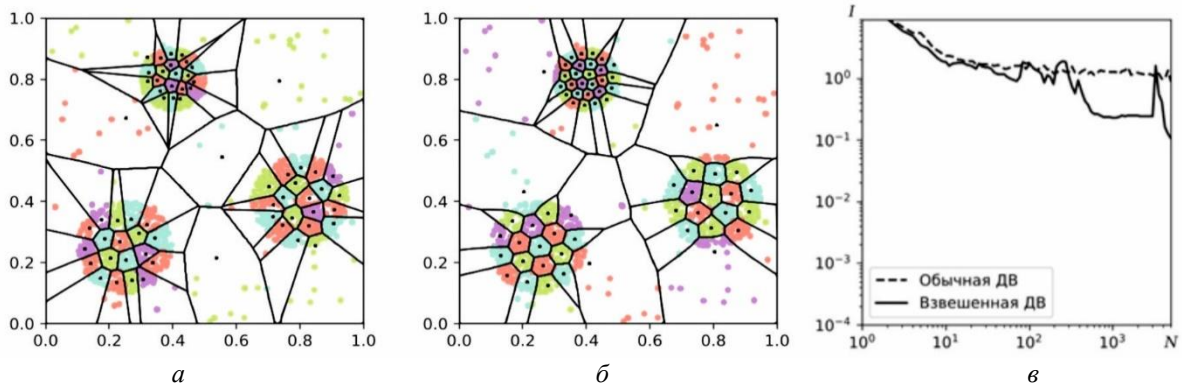


Рис. 4.4. Балансировка в тесте с существенным дисбалансом нагрузки, параметр $\sigma = 0.1$ дисбаланс составляет около 10 %: а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

В тесте с равномерной нагрузкой использовалось «стандартное» значение $\sigma = 10^{-3}$, хотя могло быть использовано значение равное нулю, что не сильно повлияло бы на результат. В тесте с существенным дисбалансом влияние параметра σ более значительное, поэтому рассмотрим три теста: без движения к центру масс ($\sigma = 0$), с небольшим стягиванием к центру масс ($\sigma = 10^{-3}$) и с сильным стягиванием ($\sigma = 0.1$). Результаты всех тестов приведены рис. 4.2–4.4. Балансировка нагрузки проводится как для взвешенной диаграммы, так и для классической.

На рис. 4.2 показаны результаты балансировки нагрузки с параметром $\sigma = 0$. При визуальной оценке декомпозиции сразу выделяется наличие большого количества тонких вытянутых ячеек Вороного. Также можно заметить, что для многих ячеек генераторы располагаются близко к границам ячеек. Особенно эти утверждения справедливы для классической диаграммы. Подобные декомпозиции являются неустойчивыми, поскольку малейшее изменение положений генераторов,

расположенных около границы, приводит к сильному изменению декомпозиции.

Неустойчивость алгоритма хорошо демонстрирует график дисбаланса на рис. 4.3,в. Существенные колебания дисбаланса связаны именно с частыми и значительными изменениями диаграммы. Классическая диаграмма при этом совсем не приходит к балансу. В то же время дисбаланс для аддитивной диаграммы снижается до 0.2 %.

Другой предельный случай (большое значение σ) показан на рис. 4.4. Ячейки диаграммы очень выпуклые. Диаграммы состоят практически целиком из шестиугольников с правильным замощением типа сот. Однако при такой красивой декомпозиции не достигается хороший баланс. Дисбаланс для классической диаграммы составляет порядка 100 %, т.е. одна из ячеек Вороного содержит в два раза больше точек, чем должно быть в идеальном случае. Для взвешенной диаграммы Вороного результаты тестирования лучше: итоговый дисбаланс составляет около 10 %.

Оптимальный случай (небольшое значение $\sigma = 10^{-3}$) показан на рис. 4.3. В этом тесте объединены лучшие черты предыдущих двух расчетов. Ячейки Вороного выпуклые, среди них мало вытянутых ячеек и ячеек с генераторами у границы, при этом достигается хороший баланс нагрузки. Классическая диаграмма, конечно, как и ранее, не справляется с задачей балансировки, а для взвешенной диаграммы достигается устойчивое значение дисбаланса около 1 %.

Результаты теста показывают, что при наличии существенного дисбаланса нагрузки классические диаграммы Вороного не могут справиться с задачей балансировки нагрузки. При использовании классических диаграмм дисбаланс вообще не стремится к нулю, генераторы ячеек ведут себя хаотично, а дисбаланс колеблется на неприемлемом уровне > 100 %. При этом взвешенные диаграммы Вороного позволяют сбалансировать такую нагрузку с высокой точностью. Также установлено, что добавление параметра σ с небольшим значением позволяет сделать алгоритм более устойчивым, при несущественной потере в точности.

5. БАЛАНСИРОВКА НАГРУЗКИ ПРИ МОДЕЛИРОВАНИИ ДИНАМИКИ СПЛОШНЫХ СРЕД НА ЛОКАЛЬНО АДАПТИВНЫХ СЕТКАХ

В следующих разделах рассмотрим несколько тестовых задач из области механики сплошных сред. Численное моделирование задач проводится с использованием адаптивных сеток, после чего на полученных сетках проверяется алгоритм балансировки нагрузки. В данных тестах не идет речь о динамической балансировке нагрузки: все задачи считаются на одном вычислительном узле, а алгоритм балансировки применяется к сетке на конкретных временных шагах.

Для декомпозиции, как и в предыдущих тестах, используется классическая диаграмма Вороного и взвешенная. До завершения балансировки проводится 5000 итераций, после которых фиксируется итоговая декомпозиция и дисбаланс. Также дисбаланс сохраняется на каждой итерации, чтобы сравнить поведение алгоритмов для классической и взвешенной диаграмм. Введем некоторую терминологию для описания адаптивных сеток. *Базовой сеткой* называется исходная сетка до применения адаптации. Базовая сетка состоит из четырехугольных ячеек и является

конформной¹. По определению уровни адаптации ячеек базовой сетки предполагаются равными нулю.

В течение расчета каждая ячейка может быть разбита на четыре четырехугольные. Части, полученные при разбиении ячейки, называются *дочерними*. Дочерние ячейки имеют уровень адаптации на единицу выше, чем у родительской ячейки. Допускается и обратная операция: любые четыре ячейки, имеющие общую родительскую ячейку, могут быть объединены (либо удалены с заменой на родительскую). Структура сетки в любой момент расчета описывается как двумерное дерево. *Листовые ячейки* находятся на вершине иерархии, это ячейки, которые не имеют дочерних (совпадает с определением листовых узлов для деревьев из теории графов). Листовые ячейки полностью покрывают расчетную область, и только они участвуют в расчете на каждом временном шаге.

Введем также такое понятие, как *референсная сетка* – так будем называть сетку, полученную путем адаптации базовой сетки до максимального уровня.

Понятно, что максимальный уровень ничем не ограничен, максимальный уровень адаптации фиксируется для конкретного расчета или серии расчетов. Референсная сетка также является конформной. Размеры референсной сетки ограничивают величину сетки при проведении расчетов.

5.1. Проблема схлопывания пузырька газа

Задача о схлопывании пузырька под действием ударной волны является популярной тестовой задачей для гидродинамических кодов, моделирующих динамику гетерогенных сред. В данной работе рассматривается постановка задачи с пузырьком газа под водой. Для моделирования динамики гетерогенных сред используется схема второго порядка по времени и пространству, основанная на методе Годунова и решении составной задачи Римана (CRP [11]).

Вычислительная область имеет форму квадрата 1.2×1.2 см. Область заполнена водой, а в центре области располагается круглый пузырек газа

¹Ячейки полностью покрывают расчетную область: если две ячейки пересекаются, то либо они пересекаются в одной точке, либо по общей грани. Это означает, что ячейки не наслаиваются друг на друга, а в сетке отсутствуют дырки и «висящие» вершины.

радиусом 6 мм. Задача решается в плоской постановке. В начальный момент времени фронт ударной волны располагается слева от пузырька и имеет координату $x = 0.6$ мм. Начальные данные приведены в табл. 5.1.

Таблица 5.1. Начальные данные в задаче о схлопывании пузырька газа

Параметр	В газе	Перед фронтом ударной волны	За фронтом ударной волны
Плотность, кг/м ³	1.0	1000.0	1323.6
Скорость, м/с	0.0	0.0	681.58
Давление, МПа	0.1	0.1	1900.0

Для газа используется уравнение идеального газа с показателем $\gamma = 1.4$, для воды используется двучленное уравнение состояния с параметрами $\gamma = 4$ и $p_\infty = 0,6$ ГПа. Такая постановка задачи используется, к примеру, в работе [14] (в литературе также встречаются постановки задачи с другими материалами). В ходе расчета ударная волна проходит через пузырек газа, и он схлопывается. Динамика процесса отражена на рис. 5.1, на котором показано распределение плотности в расчет-

ной области в различные моменты времени. Области газа и воды на рисунках можно визуально отличить, поскольку плотность в них отличается в 1000 раз.

Наиболее интересным в данной задаче является формирование мелкомасштабных структур при коллапсе пузырька. Струя воды (*secondary jet*), проникающая в пузырек, отчетливо видна на рис. 5.2,б. Для получения хорошего сеточного разрешения в данной области необходимо использовать адаптацию сетки по контактной границе между материалами. На рис. 5.2,а,б цветом показан материал ячеек, а затемнением – уровни адаптации ячеек. При расчете данной задачи использовалась базовая сетка размером 300×300 ячеек, т.е. ячейка базовой сетки имеет размер 40 мкм. Контактная граница между водой и газом адаптировалась до пятого уровня, а, значит, ячейки на интерфейсе имеют размер 1,25 мкм. Ячейки такого размера сложно разглядеть на рис. 5.2,в даже при хорошем приближении. Если бы вся сетка состояла из ячеек такого размера, то она содержала бы более 90 млн ячеек. В данном же расчете в последний момент времени (см. рис. 5.1,в и 5.2,в) сетка содержит всего около 175 тыс. ячеек.

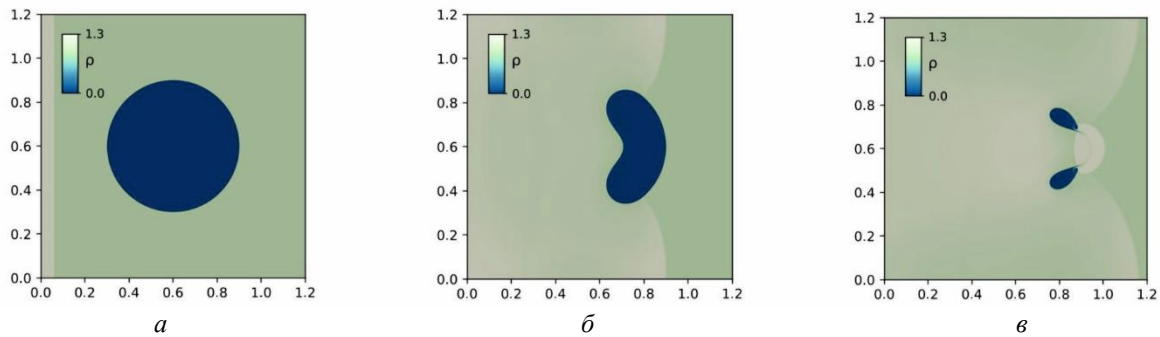


Рис. 5.1. Плотность среды в различные моменты времени:

а) $t = 0.0$ с; б) $t = 3.0$ с; в) $t = 4.0$ с

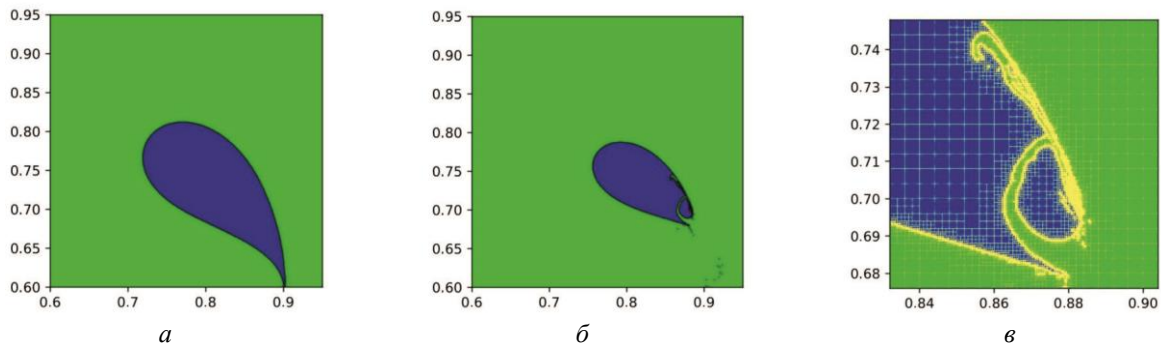


Рис. 5.2. Схлопывание пузырька вблизи. Цветом показан материал, затемнением показаны уровни адаптации ячеек: а) $t = 3.7$ с; б) $t = 4.0$ с; в) $t = 4.0$ с

В данной задаче из-за использования сеточной адаптации наблюдается существенный дисбаланс нагрузки. Как было сказано выше, количество ячеек в данной задаче в последний момент времени составляет около 175 тыс., примерно такое же число ячеек сохраняется на протяжении всего расчета. Из этих ячеек 25–35 % приходится на ячейки самого высокого уровня адаптации. То есть до трети ячеек сосредоточены на контактной границе между материалами.

На рис. 5.3 показан результат балансировки ячеек сетки на 64 части с использованием классической и взвешенной диаграмм Вороного. На

рис. 5.4 показан дисбаланс для обеих диаграмм. Из рисунка следует, что классическая диаграмма не справляется с задачей балансировки нагрузки: дисбаланс колеблется около 10 %. Хотя это не так много, но постоянные колебания плохо бы сказались на реальном расчете. Колебания дисбаланса связаны с неустойчивостью алгоритма. Для взвешенной диаграммы результат существенно лучше, дисбаланс уменьшился до 0.6 %. Для других временных шагов получены аналогичные результаты: классическая диаграмма не может сбалансировать ячейки сетки, но это возможно при использовании взвешенной диаграммы.

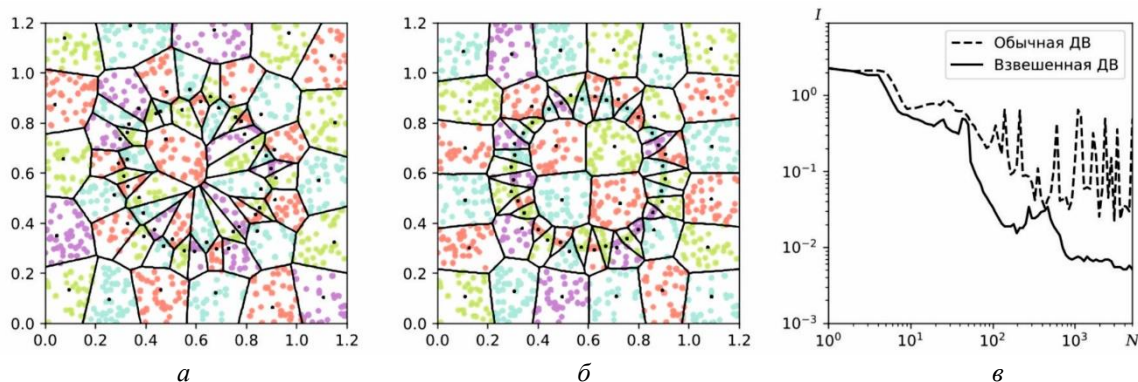


Рис. 5.3. Балансировка ячеек в начальный момент времени:
а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

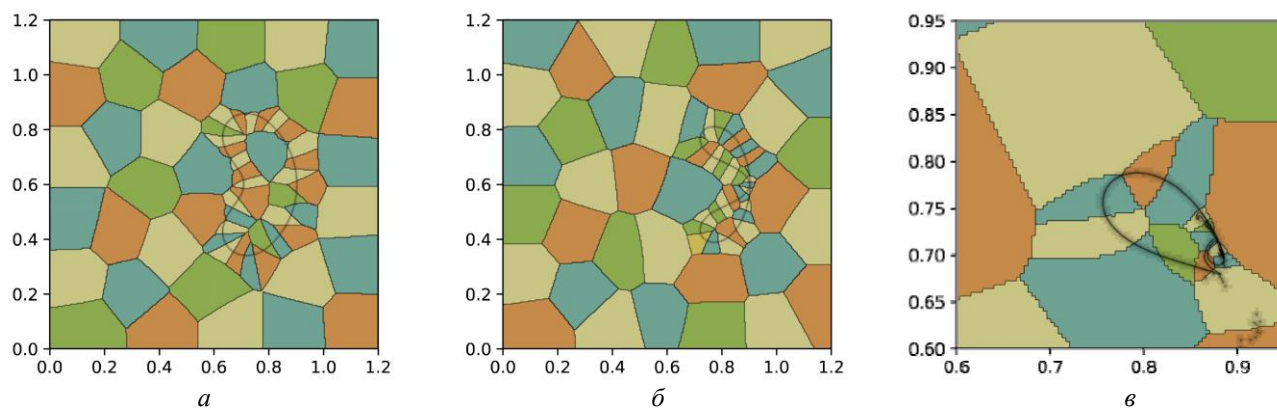


Рис. 5.4. Декомпозиция ячеек в различные моменты времени:
а) $t = 3.0$ мкс; б) $t = 3.7$ мкс; в) $t = 4.0$ мкс

На рис. 5.4 показана декомпозиция расчетной области на 64 подобласти в различные моменты времени с использованием взвешенной диаграммы Вороного. На рис. 5.4,в высокое разрешение позволяет разглядеть границу между базовыми ячейками. Для всех диаграмм на рис. 5.4 дисбаланс не превышает 2 %.

5.2. Задача с тройной точкой

Другим популярным тестом для гидрокодов является задача с тройной точкой. В качестве теста задача встречается в работах [2, 5, 15]. В статье [15] содержатся результаты моделирования задачи с очень высоким сеточным разрешением, в

статьях [2, 5] на сетках с низким разрешением тестируются ALE-методы.

В начальный момент времени прямоугольная область $[0, 7] \times [0, 3]$ разбита на три подобласти, заполненные различными материалами. В первой области $x < 1$ находится газ с $\gamma = 1.4$ и параметрами $p = 1$, $\rho = 1$. Во второй области $x > 1$, $y > 1.5$ находится газ с $\gamma = 1.4$ и параметрами $p = 0.1$, $\rho = 0.125$. Оставшаяся область заполнена газом с $\gamma = 1.5$ и параметрами $p = 0.1$, $\rho = 1$. Поскольку в первой и второй области газ имеет одинаковый показатель адиабаты, можно считать, что области

содержат одинаковый газ и рассматривать задачу как двуматериальную.

Распределение плотности в некоторые моменты времени показано на рис. 5.5. В ходе расчета ударная волна набегает на второй материал, и на границе двух сред формируется неустойчивость Кельвина-Гельмгольца. Для отслеживания развития возмущений в хорошем разрешении недостаточно использовать адаптацию только по контактной границе между средами, как в предыдущей задаче.

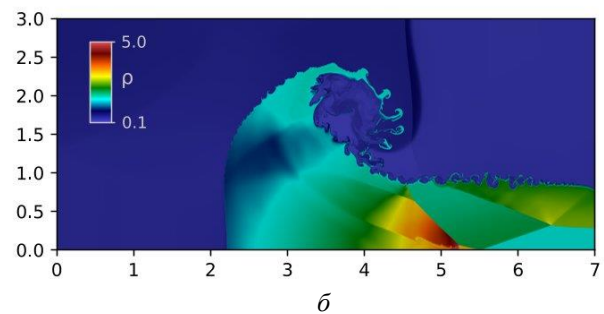
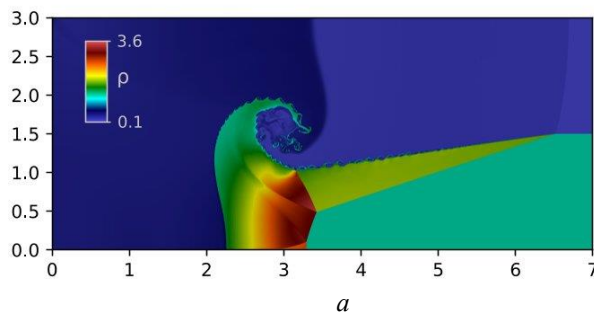


Рис. 5.5. Плотность среды в различные моменты времени: а) $t = 3,0$; б) $t = 5,0$

В данной задаче используется более продвинутая техника адаптации: ячейки адаптируются в соответствии с χ -критерием, который введен в работе [10] для метода конечных элементов, для эйлеровых сеток критерий описан в статье [4]. В данной работе используется формула из статьи [16]. χ -критерий позволяет отслеживать возмущения плотности и других параметров среды. Благодаря использованию критерия, в задаче на высоком уровне адаптации поддерживаются ударные волны, волны разрежения, контактные разрывы, а также область возмущений вблизи контактных границ.

На рис. 5.6 цветом показаны два материала, которые участвуют в задаче, а затемнением обозначены уровни ячеек. Поскольку адаптируются только узкие области вблизи границы, уровни ячеек на рисунке выглядят просто как черные линии. Эти черные линии как раз и подсвечивают ударные волны, волны разрежения и границы между средами. На рис. 5.5 отчетливо видно, какое высокое разрешение достигается на ударных волнах.

Как и для прошлой задачи, попробуем сбалансировать ячейки сетки с использованием классической и взвешенной диаграмм Вороного. Для этой задачи нужно получить декомпозицию на 72 подобласти. Результаты декомпозиции сетки для момента времени $t = 2.0$ приведены на рис. 5.7. Хотя визуально диаграммы отличаются не сильно, декомпозиция классической диаграммы существенно хуже. Для классической диаграммы дисбаланс колеблется от 20 до 70 %, в то время как для взвешенной диаграммы дисбаланс составляет менее 1 %.

Аналогичные результаты получены и для других моментов времени. Декомпозиция сетки в различные моменты времени показана на рис. 5.8. Разрешение базовой сетки составляет 150×350 ячеек, т.е. базовая сетка содержит около 53 тыс. ячеек. Максимальный уровень адаптации равен пяти. Если бы вся сетка имела максимальное разрешение, то она состояла бы из 53 млн ячеек. На практике количество ячеек существенно меньше. Число ячеек в задаче максимально в последний момент времени и составляет около 2 млн. Из этих точек максимальный уровень адаптации имеют 1.5 млн точек.

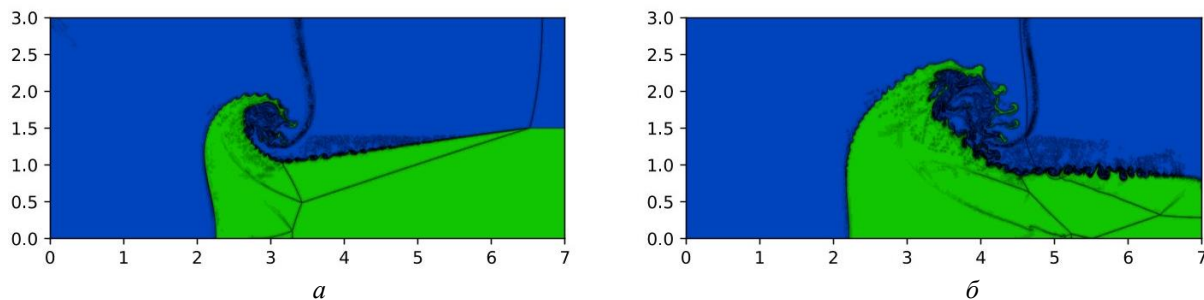


Рис. 5.6. Задача с тройной точкой: а) $t = 3.0$; б) $t = 5.0$

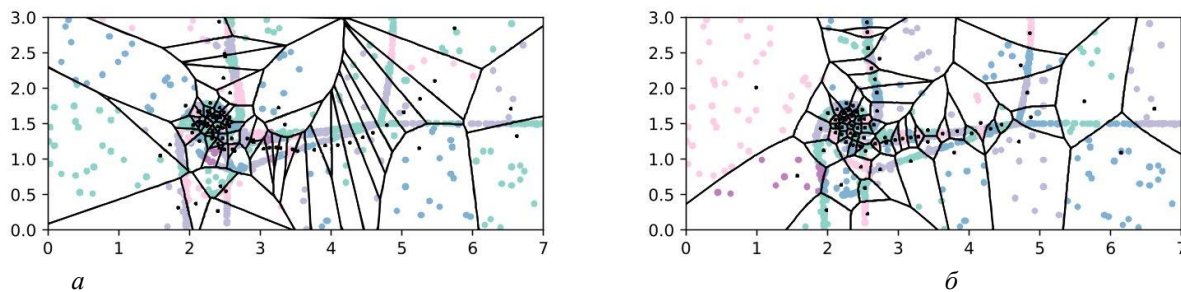


Рис. 5.7. Декомпозиция классической и взвешенной диаграмм Вороного, $t = 2.0$:
а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного

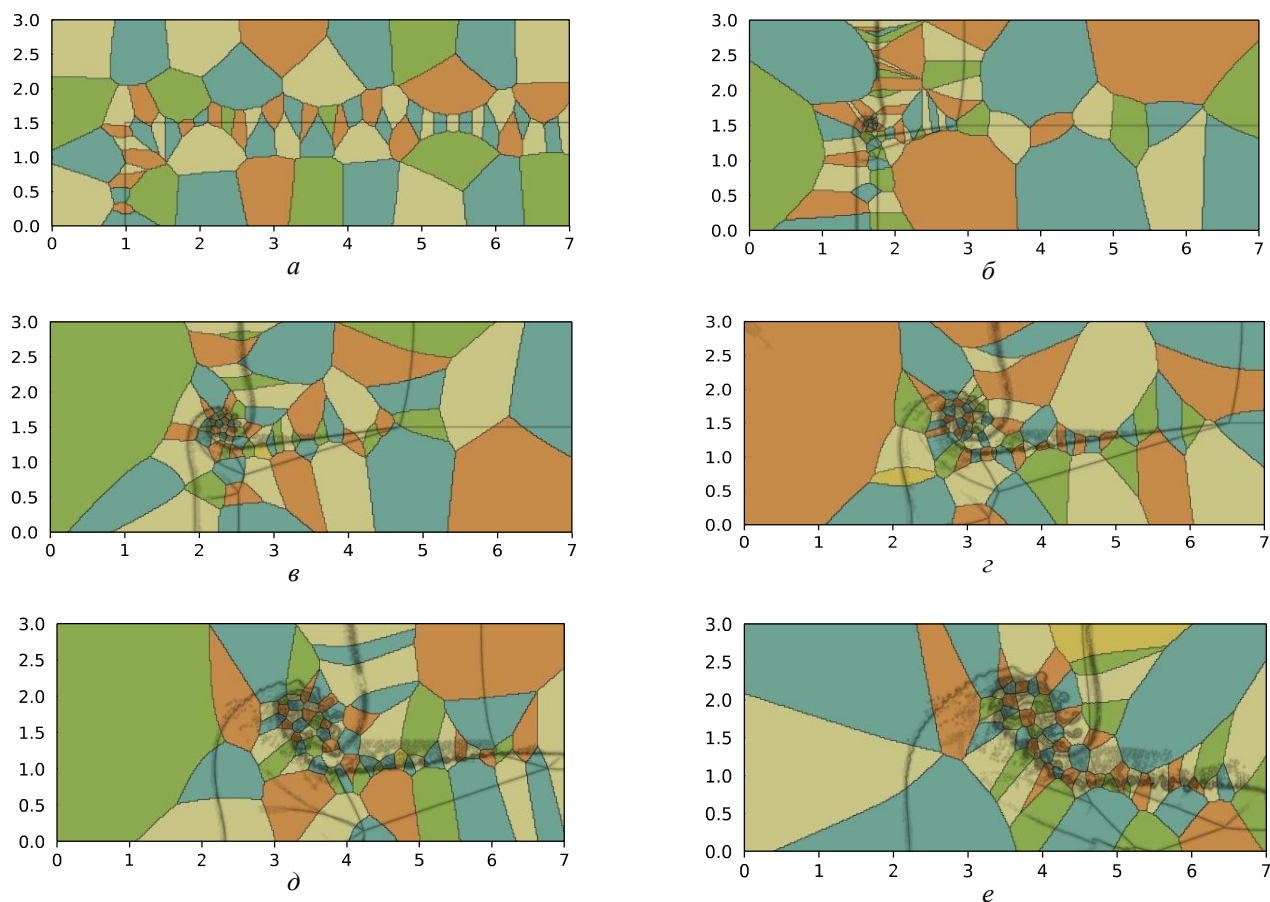


Рис. 5.8. Декомпозиция области в различные моменты времени:
а) $t = 0.0$; б) $t = 1.0$; в) $t = 2.0$; г) $t = 3.0$; д) $t = 4.0$; е) $t = 5.0$

5.3. Формирование множественных полос адиабатического сдвига

Полоса адиабатического сдвига – это узкие области шириной порядка 10–500 мкм, в которых за короткий промежуток времени достигаются значительные температуры и деформации. Данные полосы образуются при высокоскоростном сдвиге в упруго-пластичных материалах. В ряде работ установлено явление формирования множественных полос сдвига, которые взаимодействуют между собой. Данное явление является существенно нелинейным и сопровождается экстремальными значениями теплофизических параметров задачи. Примеры исследования коллективного поведения полос сдвига можно найти в работах [8, 9, 12, 13].

Для исследования коллективного поведения используется следующая постановка задачи. В квадратном блоке случайно распределено напряжение. Скорость распределена линейно. Нижняя часть блока зафиксирована, верхняя сдвигается с постоянной скоростью. Случайное распределение напряжения моделирует внутренние дефекты в материале и приводит к формированию множественных полос сдвига. Такая постановка одномерной задачи встречается в работах [9, 13]. Двумерная постановка задачи и численный алгоритм для моделирования представлены в работе [12].

Результаты численного моделирования сдвига квадратного образца 10×10 мм из урана показаны на рис. 5.9. Скорость сдвига верхней поверхности образца составляет 200 м/с.

Локализация полос происходит в момент времени $t = 10$ мкс, когда верхняя часть блока сместилась на 2 мм. На графике температуры (см. 5.9,а) отчетливо видны тонкие полосы, в которых достигается температура около 1200 °С, близкая к температуре плавления урана, – это и есть полосы сдвига. На графике скорости (см. рис. 5.9,б) полосам соответствуют области со скачком скорости.

Полосы адиабатического сдвига имеют ширину порядка 10 мкм, в то время как характерные размеры областей, в которых исследуется данный процесс, составляют от 1 см. Таким образом, данная задача является многомасштабной. Для ускорения счета оптимальным является поддержание высокого сеточного разрешения только в областях с высокими деформациями и температурами, в которых и протекают интересующие нас процессы. Методы с динамической адаптацией сетки уже применялись для исследования локализации пластических деформаций [8]. Однако в работе [8] использовался метод динамического сгущения одномерной сетки, не применимый для двумерных задач.

В данной работе для дискретизации используется двумерная адаптивная сетка с иерархической структурой. На рис. 5.10 показана сетка в конечный момент времени. На самом высоком уровне адаптации поддерживаются области, в которых температура превышает некоторое заданное пороговое значение. На рис. 5.11 показана оптимальная декомпозиция такой сетки с использованием взвешенной диаграммы Вороного.

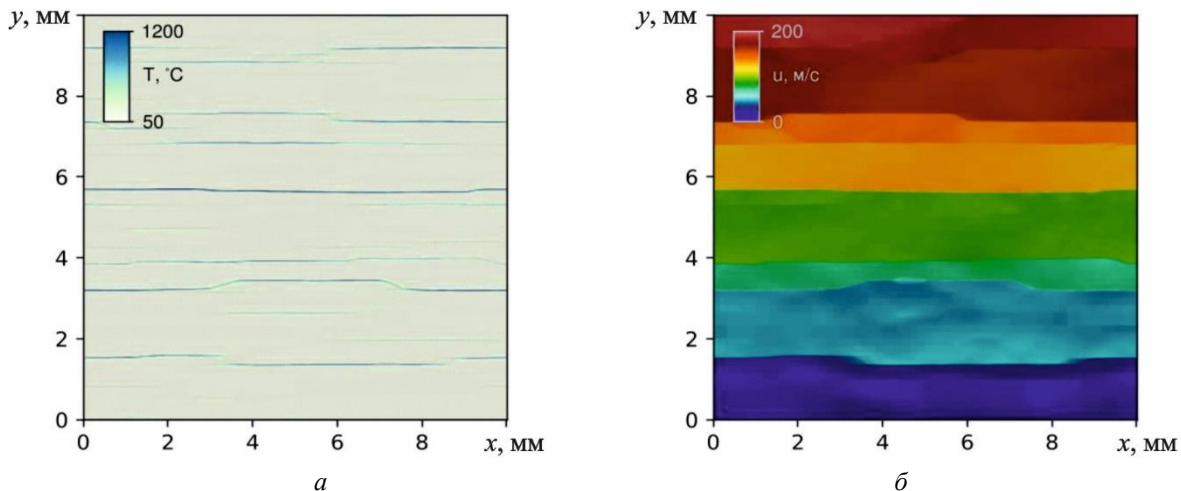


Рис. 5.9. Распределение температуры и скорости в образце в конце расчета:
а) температура; б) скорость

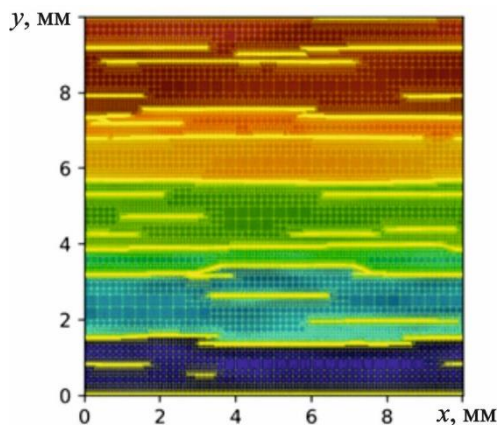


Рис. 5.10. Расчетная сетка адаптируется в зонах с высокой деформацией

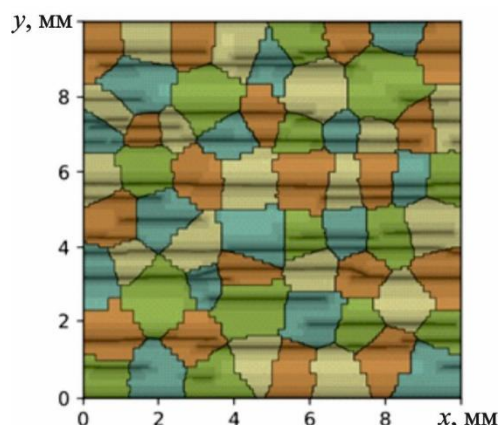


Рис. 5.11. Декомпозиция области: затемнением показаны уровни ячеек

Как и для других задач, оптимальная декомпозиция для данной сетки строилась с использованием обычной и взвешенной диаграмм Вороного. Результаты работы алгоритма балансировки показаны на рис. 5.12. Поскольку нагрузка в данной задаче распределена достаточно равномерно (на

масштабах нескольких миллиметров), визуально диаграммы мало отличаются: ячейки распределены в области равномерно и имеют границы, близкие к прямым. В то же время график дисбаланса дает понять, что взвешенные диаграммы лучше справляются с балансировкой нагрузки.

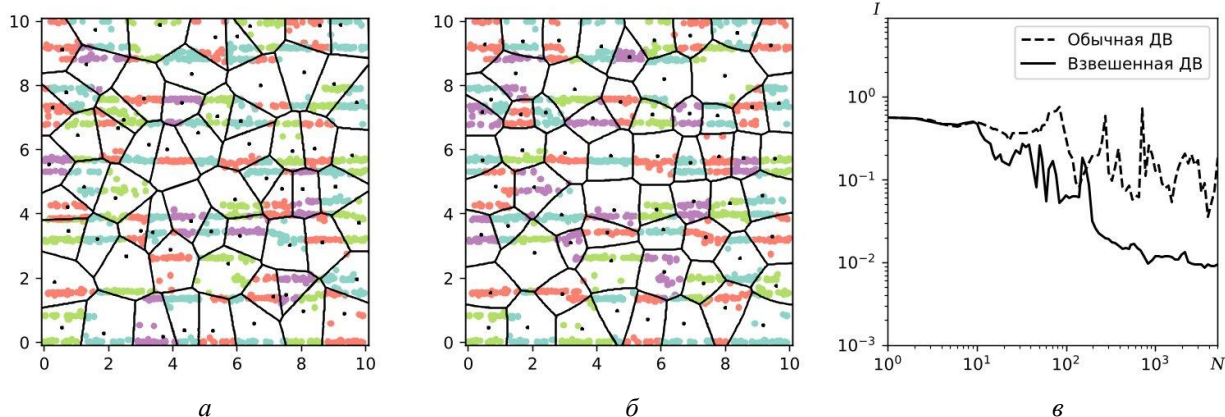


Рис. 5.12. Балансировка ячеек сетки в конце расчета: а) классическая диаграмма Вороного; б) взвешенная диаграмма Вороного; в) дисбаланс

ЗАКЛЮЧЕНИЕ

В работе предложен новый алгоритм балансировки нагрузки для различных задач вычислительной математики, предполагающих применение адаптивной сетки, сетки Эйлера и Лагранжа и т.д. Данный алгоритм основан на использовании взвешенных диаграмм Вороного.

Эффективность предлагаемого алгоритма была продемонстрирована на серии модельных задач. Результаты численных экспериментов показали, что применение данного алгоритма для балансировки нагрузки позволяет уменьшить величину

дисбаланса в 10–100 раз в зависимости от решаемой задачи. Помимо этого, предложенный алгоритм позволяет сократить количество ячеек с наивысшим уровнем адаптации, что позволяет повысить эффективность всего численного алгоритма в целом.

БЛАГОДАРНОСТИ

Исследование выполнено за счет гранта Российского научного фонда (проект № 21-71-00102).

СПИСОК ЛИТЕРАТУРЫ

1. *Cybenko G.* Dynamic load balancing for distributed memory multiprocessors // *Journal of Parallel and Distributed Computing*, 1989. V. 7. P. 279–301.
2. *Dobrev V.A., Kolev T.V., Rieben R.N.* High-order curvilinear finite element methods for lagrangian hydrodynamics // *SIAM Journal on Scientific Computing*, 2012. V. 34. P. B606–B641.
3. *Egorova M.S., Dyachkov S.A., Parshikov A.N., Zhakhovsky V.* Parallel sph modeling using dynamic domain decomposition and load balancing displacement of voronoi subdomains // *Computer Physics Communications*, 2019. V. 234. P. 112–125.
4. *Fryxell B., Olson K., Ricker P., Timmes F., Zingale M., Lamb D., MacNeice P., Rosner R., Truran J., Tufo H.* Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes // *The Astrophysical Journal Supplement Series*, 2000. V. 131. P. 273.
5. *Galera S., Maire P.-H., Breil J.* A two-dimensional unstructured cell-centered multi-material ale scheme using vof interface reconstruction // *Journal of Computational Physics*, 2010. V. 229. P. 5755–5787.
6. *Karypis G., Schloegel K., Kumar V.* Parmetis: Parallel graph partitioning and sparse matrix ordering library, 1997.
7. *Koradi R., Billeter M., Guentert P.* Point-centered domain decomposition for parallel molecular dynamics simulation // *Computer Physics Communications*, 2000. V. 124. P. 139–147.
8. *Kudryashov N., Muratov R., Ryabov P.* The collective behavior of shear strain localizations in dipolar materials // *Applied Mathematics and Computation*, 2018. V. 338. P. 164–174.
9. *Kudryashov N.A., Ryabov P.N., Zakharchenko A.S.* Self-organization of adiabatic shear bands in ofhc copper and hy-100 steel // *Journal of the Mechanics and Physics of Solids*, 201576. P. 180–192.
10. *Lohner R.* An adaptive finite element scheme for transient problems in cfd // *Computer Methods in Applied Mechanics and Engineering*, 1987. V. 61. P. 323–338.
11. *Menshov I., Zakharov P.* On the composite riemann problem for multi-material fluid flows // *International Journal for Numerical Methods in Fluids*, 2014. V.76. P. 109–127.
12. *Muratov R., Kudryashov N., Ryabov P.* A finite volume method for numerical simulations of adiabatic shear bands formation // *Communications in Nonlinear Science and Numerical Simulation*, 2021. V.101. P. 105858.
13. *Muratov R.V., Kudryashov N.A., Ryabov P.N.* Application of the adaptive mesh refinement technique to study the plastic flow localization process // *Bulletin of the National Research Nuclear University MEPhI*, 2019. V.8. P. 361–369.
14. *Nourgaliev R.R., Dinh T.-N., Theofanous T.G.* Adaptive characteristics-based matching for compressible multifluid dynamics // *Journal of Computational Physics*, 2006. V. 213. P. 500–529.
15. *Pan S., Han L., Hu X., Adams N.A.* A conservative interface-interaction method for compressible multi-material flows // *Journal of Computational Physics*, 2018. V. 371. P. 870–895.
16. *Smirnov N., Kiselev A., Zakharov P., Muratov R., Bukharinskaya D.* The usage of adaptive mesh refinement in simulation of high-velocity collision between impactor and thin-walled containment // *Acta Astronautica*, 2022. V. 194. P. 401–410.
17. *Zhakhovskii V., Nishihara K., Fukuda Y., Shimojo S., Akiyama T., Miyanaga S., Sone H., Kobayashi H., Ito E., Seo Y., Tamura M., Ueshima Y.* A new dynamical domain decomposition method for parallel molecular dynamics simulation, in *CCGrid 2005 // IEEE International Symposium on Cluster Computing and the Grid*, 2005. V. 2, 2005. P. 848–854.

ONE METHOD OF LOAD BALANCING USING WEIGHTED VORONOI DIAGRAM

R.V. Muratov^{1,*}, P.N. Ryabov^{1,}, S.A. Dyachkov²**¹*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Moscow, 115409 Russia*²*NL Dukhov All-Russian Research Institute of Automation
Moscow, 101000 Russia***e-mail: RVMuratov@mephi.ru****e-mail: pnryabov@mephi.ru*

Received March 27, 2023, revised April 6, 2023; accepted April 14, 2023

The solution of load balancing problem in parallel programming is very actual problem. One of the original methods for solving the balancing problem is the use of Voronoi diagrams. The main advantage of the following method that it can be used on the data of different types. For example, this method can be used for balancing the Eulerian and Lagrangian computational grids with the smooth-particles hydrodynamic method. Also, this approach can be used for adaptive grids. However, in that case, due to the strong inhomogeneity of the load on adaptive grids, the such load balancing method can be unstable or have poor convergence. In the present work we give an improvement of the load balancing method in terms of weighted Voronoi diagrams. The proposed algorithm is implemented as a software package. Testing of the proposed algorithm is based on a number of model problems and problems from the field of continuum mechanics. Evaluation of the balancing efficiency is based on the study of the behavior of the imbalance value with and without balancing. It is shown that the proposed algorithm successfully copes with its task and the magnitude of the imbalance in the case of using a weighted Voronoi diagram is 10–100 times less than when using classical Voronoi diagrams.

Keywords: parallel programming, load balancing, weighted Voronoi diagrams, continuum mechanics, numerical simulation.

REFERENCES

1. Cybenko G. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*. 1989. Vol. 7. P. 279–301.
2. Dobrev V.A., Kolev T.V., Rieben R.N. High-order curvilinear finite element methods for lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*. 2012. Vol. 34. P. B606–B641.
3. Egorova M.S., Dyachkov S.A., Parshikov A.N., Zhakhovsky V. Parallel sph modeling using dynamic domain decomposition and load balancing displacement of voronoi subdomains. *Computer Physics Communications*. 2019. Vol. 234. P. 112–125.
4. Fryxell B., Olson K., Ricker P., Timmes F., Zingale M., Lamb D., MacNeice P., Rosner R., Truran J., Tufo H. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *The Astrophysical Journal Supplement Series*. 2000. Vol. 131. P. 273.
5. Galera S., Maire P.-H., and Breil J. A two-dimensional unstructured cell-centered multi-material ale scheme using vof interface reconstruction. *Journal of Computational Physics*. 2010. Vol. 229. P. 5755–5787.
6. Karypis G., Schloegel K., Kumar V. Parmetis: Parallel graph partitioning and sparse matrix ordering library. 1997.
7. Koradi R., Billeter M., Guñtert P. Point-centered domain decomposition for parallel molecular dynamics simulation. *Computer Physics Communications*. 2000. Vol. 124. p. 139–147.
8. Kudryashov N., Muratov R., Ryabov P. The collective behavior of shear strain localizations in dipolar materials. *Applied Mathematics and Computation*. 2018. Vol. 338. P. 164–174.
9. Kudryashov N.A., Ryabov P.N., Zakharchenko A.S. Self-organization of adiabatic shear bands in ofhc copper and hy-100 steel. *Journal of the Mechanics and Physics of Solids*. 201576. P. 180–192.

10. *Lohner R.* An adaptive finite element scheme for transient problems in cfd. *Computer Methods in Applied Mechanics and Engineering*. 1987. Vol. 61. P. 323–338.

11. *Menshov I., Zakharov P.* On the composite riemann problem for multi-material fluid flows. *International Journal for Numerical Methods in Fluids*. 2014. Vol. 76. P. 109–127.

12. *Muratov R., Kudryashov N., Ryabov P.* A finite volume method for numerical simulations of adiabatic shear bands formation. *Communications in Nonlinear Science and Numerical Simulation*. 2021. Vol. 101. P. 105858.

13. *Muratov R.V., Kudryashov N.A., Ryabov P.N.* Application of the adaptive mesh refinement technique to study the plastic flow localization process. *Bulletin of the National Research Nuclear University MEPhI*. 2019. Vol. 8. P. 361–369.

14. *Nourgaliev R.R., Dinh T.-N., Theofanous T.G.* Adaptive characteristics-based matching for compressible

multifluid dynamics. *Journal of Computational Physics*, 2006. Vol. 213. P. 500–529.

15. *Pan S., Han L., Hu X., Adams N.A.* A conservative interface-interaction method for compressible multi-material flows. *Journal of Computational Physics*. 2018. Vol. 371. P. 870–895.

16. *Smirnov N., Kiselev A., Zakharov P., Muratov R., Bukharinskaya D.* The usage of adaptive mesh refinement in simulation of high-velocity collision between impactor and thin-walled containment. *Acta Astronautica*. 2022. Vol. 194. P. 401–410.

17. *Zhakhovskii V., Nishihara K., Fukuda Y., Shimojo S., Akiyama T., Miyanaga S., Sone H., Kobayashi H., Ito E., Seo Y., Tamura M., Ueshima Y.* A new dynamical domain decomposition method for parallel molecular dynamics simulation, in *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid*. 2005. Vol. 2. P. 848–854.