

УДК 519.6

МЕТОДЫ РАСПАРАЛЛЕЛИВАНИЯ РЕШЕНИЯ ДВУМЕРНЫХ ЗАДАЧ ГАЗОДИНАМИКИ НА НЕСТРУКТУРИРОВАННЫХ СЕТКАХ С ПЕРЕМЕННОЙ ТОПОЛОГИЕЙ В МЕТОДИКЕ МЕДУЗА

В. А. Пронин
(РФЯЦ-ВНИИЭФ)

Дается описание методов распараллеливания, используемых при расчете газодинамических задач на неструктурированных сетках. Основное внимание уделено алгоритмам распараллеливания при перестроении сетки. Результаты расчета тестовой задачи подтверждают применимость разработанных методов.

Ключевые слова: методы распараллеливания, неструктурированные сетки, корректировка сетки, методика МЕДУЗА.

Введение

В математическом отделении РФЯЦ-ВНИИЭФ осуществляется разработка *параллельной* версии комплекса программ МЕДУЗА [1] для расчета двумерных задач газовой динамики с теплопроводностью в режиме распараллеливания.

В методике МЕДУЗА используются уравнения движения среды, записанные в лагранжевых переменных, которые описывают ее поведение под действием массовых и/или поверхностных сил. Аппроксимация системы уравнений производится на неструктурированной лагранжевой сетке, способной изменять топологию сеточных связей в процессе счета. Ячейки сетки представляют собой несамопересекающиеся многоугольники (условие выпуклости не является обязательным). В методике используется однообластный подход.

Вычисление одного счетного шага можно условно разделить на два этапа, которые принято называть *лагранжевым* и *эйлеровым*. На лагранжевом этапе производится решение системы конечно-объемных уравнений, описывающих поведение среды под действием массовых и/или поверхностных сил. Второй этап — эйлеров — связан с улучшением качества счетной сетки. На этом этапе производится изменение сетки по конкретным алгоритмам и пересчет величин, припи-

санных к этой сетке. Пересчет величин происходит исходя из законов сохранения.

К лагранжеву этапу методики МЕДУЗА относится расчет таких процессов, как:

- газодинамика;
- упругопластика;
- кинетика горения и детонации ВВ;
- теплопроводность;
- турбулентное перемешивание.

К эйлерову этапу относятся операции корректировки сетки:

- раздвижка в ячейке;
- перестройка ребра;
- удаление ячейки из счета;
- добавление ячейки в счет.

Для расчета задач по методике МЕДУЗА в *параллельном* режиме необходимо было произвести распараллеливание всех процессов лагранжева этапа и всех операций эйлерова этапа. Разработка и внедрение алгоритмов распараллеливания для счетных модулей лагранжева этапа не представляет особой сложности. Первые работы в этом направлении описаны в [2, 3]. При создании параллельной версии методики основная работа заключалась в разработке и внедрении методов распараллеливания операций эйлерова этапа, изменяющих топологию связей.

Распараллеливание расчета задач в методике МЕДУЗА базируется на следующих принципах:

- применяется геометрический метод начальной декомпозиции данных по процессорам;
- один процессор рассчитывает одну параобласть;
- все процессоры равноправны (нет выделенного управляющего процессора);
- для реализации взаимодействия между процессорами используется библиотека MPI;
- все множество ячеек (точек, вокруг которых строятся эти ячейки) параобласти разделяется на пять непересекающихся подмножеств: *ядерные, предоболочечные, оболочечные, несчетные** первого и второго уровней (N1 и N2). При этом осуществляется поячеечное наложение двух слоев (предоболочечного и оболочечного множеств) на соответствующие несчетные множества ячеек, соседних (смежных) параобластей;
- для каждого процессора количество *смежных процессоров* (рассчитывающих смежные параобласти) может быть различным и изменяться в процессе счета;
- обмен данными производится только между смежными процессорами;
- производится совмещение вычислительной работы процессора с его обменами на всех этапах, где это возможно;
- любое изменение величин или связей в множествах оболочечных или предоболочечных ячеек тождественно (за счет обменов) отображается в соответствующие несчетные множества ячеек для смежных процессоров;
- алгоритмы корректировки сетки позволяют ячейкам "перемещаться" между соседними подмножествами и параобластями. Такую операцию перемещения будем называть *миграцией*.

Работа по распараллеливанию и отладке операций корректировки сетки, приводящих к изменению связей между объектами сетки, заняла длительный период времени. Первые работы относятся к 2004 году. В работе [4] были описаны основные типы миграций точек и этапы распараллеливания операции перестройки ребра. На тестовой задаче о падении ударной волны

*В данной статье *несчетным* называется множество ячеек (точек) параобласти, не рассчитываемых на процессоре, *счетным* — множество рассчитываемых на процессоре ячеек (точек).

на заполненную *тяжелым* газом область [5] были опробованы распараллеленные операции перестроек ребер ячеек на границе параобластей.

В дальнейших работах алгоритмы корректировки сетки были усовершенствованы и позволили посчитать эту задачу до конца, несмотря на присутствие сильного вихревого течения. Кроме того, результаты этой задачи дали новые направления развития параллельной версии методики МЕДУЗА. В процессе счета возникла ситуация, когда из-за миграций точек произошло локальное уменьшение толщины параобласти, вплоть до ее вырождения. Также в одной из параобластей этой задачи произошел отрыв части точек от основного множества и оно стало несвязным. Более того, оторвавшееся множество точек перемещалось внутри соседней параобласти и достигало границы следующей параобласти.

На рис. 1 (см. также цветную вкладку) приведен пример такой ситуации. Показан вид трех параобластей в зоне сильного вихревого течения. На цветном рисунке красным, синим и желтым цветом обозначены ядерные ячейки, а полукругами — множества граничных (оболочечных и предоболочечных) ячеек соответствующих параобластей. В красной окружности показаны оторвавшиеся ячейки параобласти 1, которые впоследствии достигают границы между второй и третьей параобластями. В момент достижения этой границы параобласти 1 и 3 должны "вступить в соседство" (до этого момента данные параобласти не являются соседними).

Опишем более подробно разработанные и внедренные схемы распараллеливания методики МЕДУЗА на лагранжевом и эйлеровом этапах.

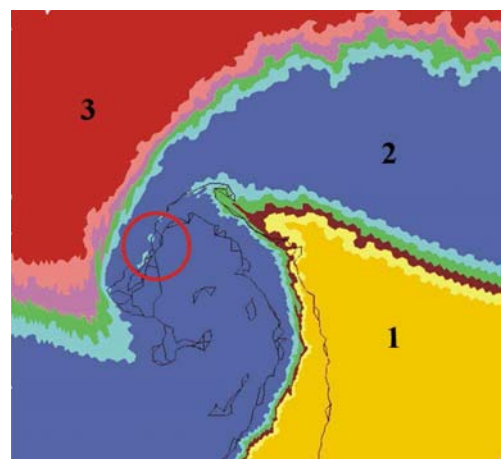


Рис. 1. Вид параобластей в зоне вихревого течения

Схема распараллеливания на лагранжевом этапе

В силу лагранжевых свойств ячейки сетки движутся вместе с веществом, при этом они деформируются под действием внешних или внутренних сил, приложенных к сторонам ячеек. Расчет уравнений движения среды производится по явной схеме, что позволяет рассчитывать величины на новом временном слое для каждой ячейки независимо. Этот факт лег в основу алгоритма распараллеливания при решении системы разностных уравнений. Кроме того, это позволяет совмещать вычислительную работу процессора с его обходами.

Рассмотрим более подробно разбиение всех ячеек параобласти на непересекающиеся подмножества. На рис. 2 (см. также цветную вставку) различным цветом представлены множества ячеек двух смежных параобластей. Фиолетовым и зеленым цветом показаны ячейки первой и второй параобласти соответственно, принадлежащие к множеству ядерных ячеек (эти ячейки являются чисто внутренними ячейками параобластей, т. е. не имеют образов в несчетных множествах смежных параобластей). Синим и оранжевым цветом показаны множества предоболочечных ячеек (эти ячейки лежат в непосредственной близости к границе параобластей и имеют образы в множествах Н2 смежных параобластей). Голубым и красным цветом показаны множества оболочечных ячеек (эти ячейки лежат на границе параобластей и имеют образы в множествах Н1 смежных параобластей).

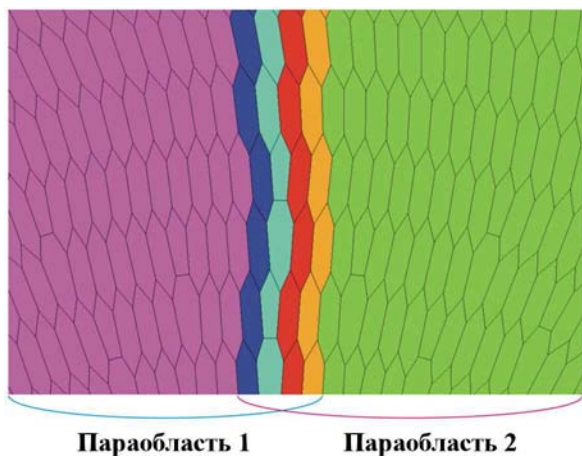


Рис. 2. Фрагмент сетки с разделением ячеек на подмножества и их перекрытия

После того, как произведена декомпозиция всей области задачи на процессоры, для каждого процессора имеется информация о количестве смежных процессоров, а также соответствующие им номера. Кроме того, программа декомпозиции производит разделение множества ячеек параобласти на пять непересекающихся подмножеств (см. Введение). При этом для каждой оболочечной и предоболочечной ячейки имеется информация о количестве ее образов на смежных параобластях (в несчетных множествах) и локальные номера соответствующих несчетных ячеек. Эта информация необходима для того, чтобы любые изменения величин или связей в счетных ячейках тождественно отображались на множества несчетных ячеек для смежных процессоров.

Совмещение вычислительной работы процессора с его обходами производится следующим образом. Сначала каждый процессор осуществляет асинхронный прием сообщений со смежных с ним процессоров, а затем рассчитывает соответствующее конечно-объемное уравнение в оболочечных и предоболочечных ячейках своей параобласти (при этом вычисляются значения величин на следующем временном слое). После этого формируется обменный буфер, в который заносятся новые значения величин из оболочечных и предоболочечных ячеек, и производится асинхронная отправка сообщений на смежные процессоры. Затем производится расчет величин на новом временном слое во внутренних (ядерных) ячейках параобласти. По окончании расчета соответствующего конечно-объемного уравнения во всех ячейках параобласти происходит ожидание завершения асинхронных обменов, после чего выполняется разбор принятых со смежных процессоров данных и обновление величин в ячейках несчетных множеств. Такую технологию обменов будем называть *встречными обходами*.

На рис. 3 приведена блок-схема расчета конечно-объемного уравнения движения. По технологии встречных обменов выполняется расчет и других конечно-объемных уравнений, входящих в систему уравнений газовой динамики.

Схемы распараллеливания на эйлеровом этапе

В методике МЕДУЗА в процессе счета задач применяются такие виды операций корректиров-

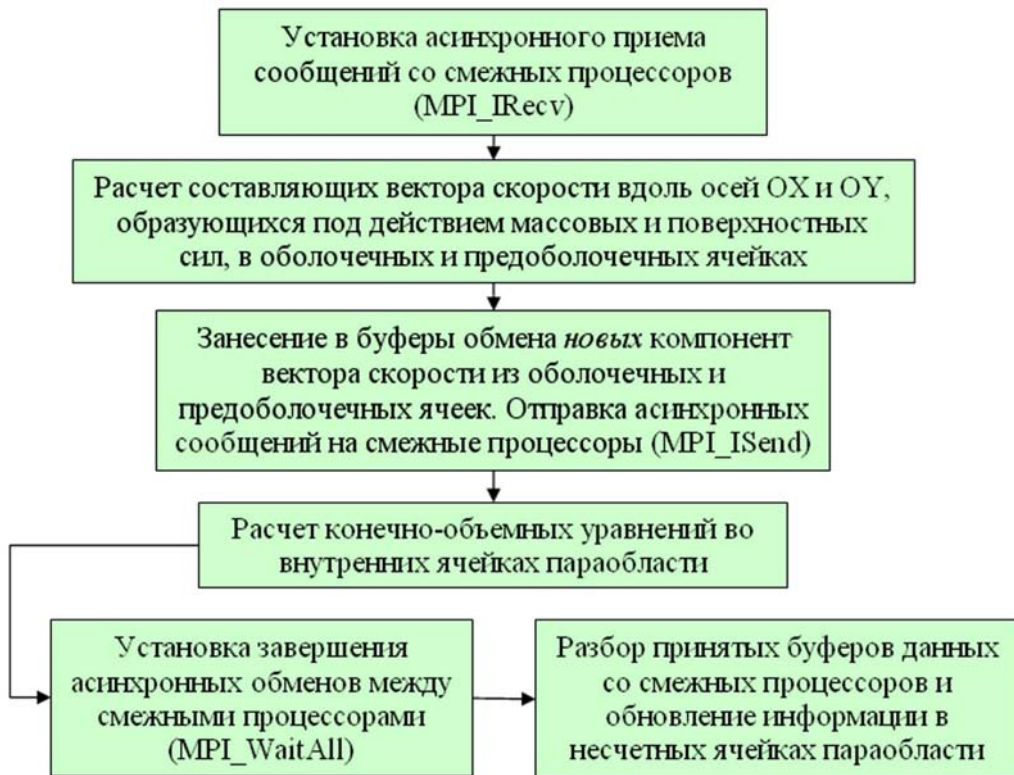


Рис. 3. Блок-схема технологии встречных обменов

ки сетки, как раздвижка в ячейке (операция отодвигания точки, близко подошедшей к границе своей ячейки), перестройка ребра (операция изменения соседства между ближайшими точками по принципу Дирихле), удаление ячейки, "держатель" значение временного шага, и добавление точки на ребро. Все эти операции сопровождаются корректировкой сетки и пересчетом сеточных величин исходя из законов сохранения. В методике МЕДУЗА используется реберная схема хранения информации о сетке. Под объектами сетки будем понимать точки и ребра, связанные между собой.

Отметим, что некоторые операции, такие как раздвижка в ячейке, не изменяют топологии связей между объектами сетки, и, следовательно, распараллеливание таких операций не представляет особой сложности. Другие виды операций корректировки сетки, такие как перестройка ребер и удаление точек из счета, требуют существенного усложнения методов распараллеливания в силу смены связей между объектами сетки.

На рис. 4 представлено графическое пояснение к операции раздвижки. Операция раздвижки сводится к изменению координаты точки, вокруг которой строится ячейка. Точка a смещается

в новое положение a' в направлении центра тяжести многоугольника, обведенного на рисунке жирным контуром. Серым цветом обозначена старая ячейка (до изменения координаты узла a). Как видно из рис. 4, после применения операции раздвижки изменяется только форма самой ячейки и форма ближайших соседних ячеек, а связи между объектами не меняются.

Как правило, операции раздвижки в ячейке выполняются на протяжении всего расчета зада-

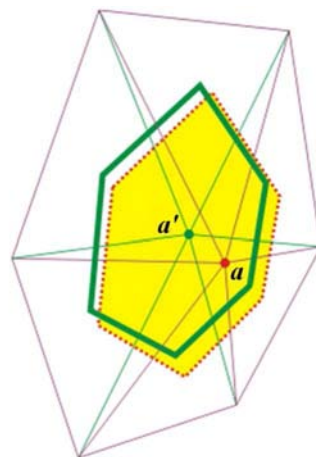


Рис. 4. Действие операции раздвижки в ячейке

чи через некоторый временной интервал, заданный пользователем, и применяются при слабых деформациях сетки.

Для распараллеливания этой операции необходимо понять ее суть. Операция раздвижки в ячейке состоит из двух этапов. Сначала проверяются критерии изменения сетки. Если хотя бы один из критериев выполняется, происходит вычисление нового положения точки. Номер смещаемой точки, а также номера ближайших к ней ячеек помечаются соответствующим признаком. Второй этап состоит из пересчета величин для помеченных на первом этапе ячеек. Пересчет производится исходя из наложения контуров измененных ячеек на контуры старых ячеек и подсчета соответствующих долей величин. На обоих этапах вычислительная работа процессора совмещается с обменов данными со смежными процессорами. Схема совмещения аналогична блок-схеме, изображенной на рис. 3.

При сильных сдвиговых деформациях требуются алгоритмы корректировки, связанные с изменением топологии связей между объектами сетки. В методике МЕДУЗА к таким операциям относятся перестройка ребра, удаление ячейки из счета, добавление ячейки в счет.

На рис. 5 показано графическое пояснение к операции перестройки ребра. На рис. 5, *а* показано состояние сетки до перестройки ребра V_1V_2 , связывающего ячейки V_1 и V_2 . На рис. 5, *б* показано состояние сетки после перестройки этого ребра. Из рис. 5, *б* видно, что ребро между ячейками V_1 , V_2 удаляется, а между ячейками V_3 , V_4 добавляется новое ребро, образуя соседство между ними.

На рис. 6 показано графическое пояснение к операции удаления ячейки. На рис. 6, *а* показано состояние сетки до удаления ячейки O . Операцию удаления можно рассматривать как перемещение точки O вдоль некоторого ребра (OA) к вершине A . На рис. 6, *б* приведено состояние сетки после удаления ячейки. Видно, что "внутренность" удаленной ячейки распределилась между ближайшими соседними с ней ячейками.

Еще одной операцией корректировки неструктурированной сетки является добавление ячейки — операция, обратная операции удаления. Особенностью алгоритма является то, что добавление ячейки производится в центр ребра. На рис. 7 показано графическое пояснение к этой операции. Операцию добавления можно ассоциировать с добавлением к элементам сетки одной точки и трех ребер.

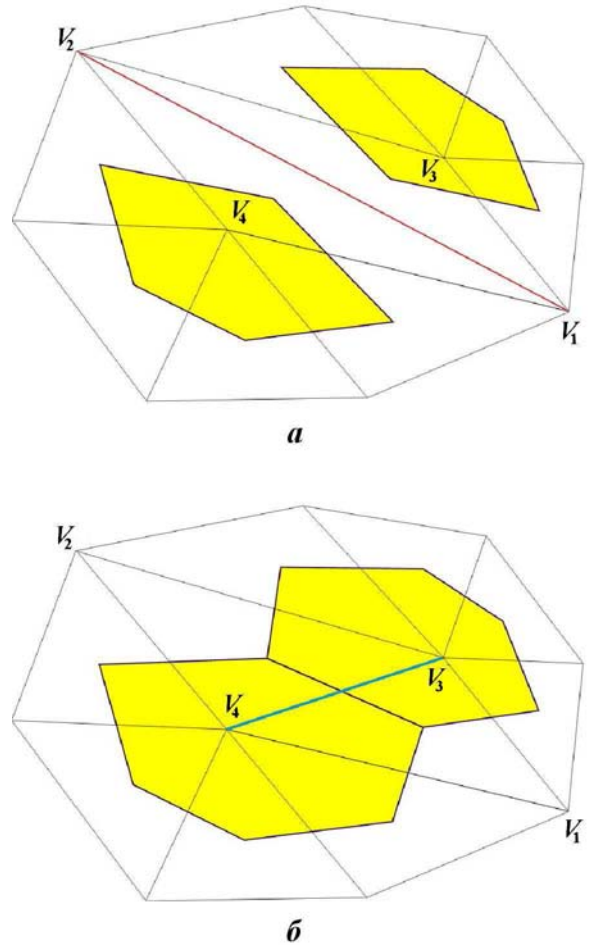


Рис. 5. Состояние сетки до (*а*) и после (*б*) перестройки ребра между точками V_1 , V_2

Как видно из представленных рисунков, применение операций перестройки ребра, удаления и добавления ячейки приводит к изменению не только формы ячеек, но и связей между объектами сетки.

Распараллеливание этих операций явилось одной из самых сложных задач в методике МЕДУЗА. Это было вызвано несколькими причинами:

1. В силу распределенности данных процессоры "не знают", как меняется сетка на соседних процессорах и в каком именно месте.
2. Должна быть возможность применения алгоритмов корректировки к любым объектам сетки. Если операция по изменению связей неструктурированной сетки проводится на границе параобластей, то на смежных процессорах должны проводиться идентичные изменения.
3. В методике МЕДУЗА существуют две базы данных для хранения сеточных связей и термодинамических величин. Они необхо-

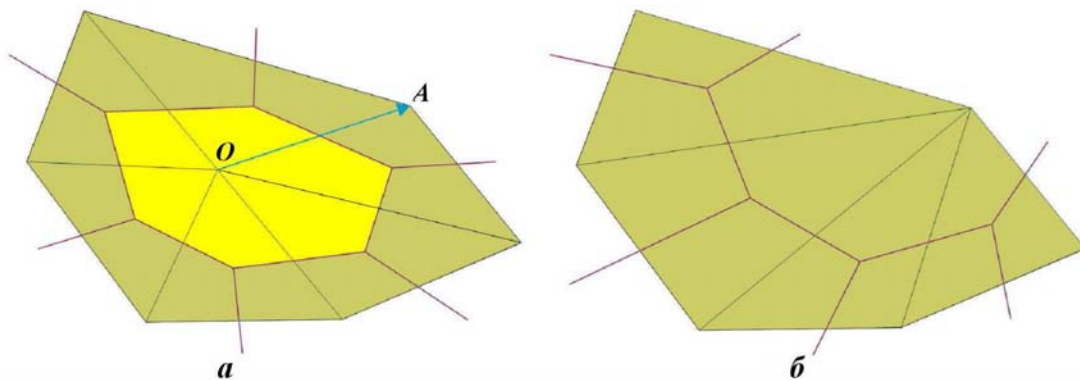


Рис. 6. Состояние сетки до (а) и после (б) удаления ячейки O

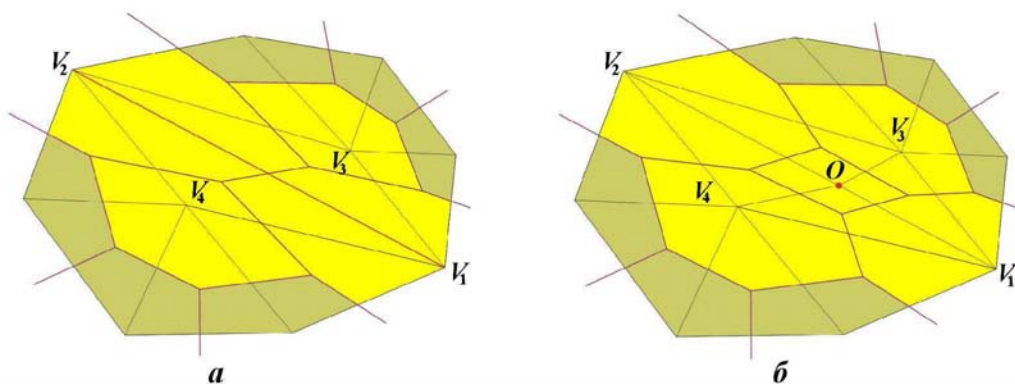


Рис. 7. Состояние сетки перед (а) и после (б) добавления ячейки O

димы для устранения зейделевского эффекта при пересчете величин.

4. Каждому локальному номеру элемента сетки на текущем процессоре ставится в соответствие уникальный глобальный номер. Кроме того, существует и обратная ссылка (с глобального номера объекта сетки на его локальный номер на текущем процессоре).
5. Для каждого процессора существуют связанные множества ячеек сетки на границе параобластей. Изменение связей между этими ячейками может приводить к потере целостности соответствующих множеств. Необходимо анализировать данную ситуацию на каждой операции корректировки и проводить восстановление множеств.
6. Изменение связей между объектами сетки на границе параобластей может приводить к локальному вырождению внутренних ячеек, вплоть до потери связности области задачи, рассчитываемой на данном процессоре. В этом случае необходимо проводить работу по обновлению соседства смежных процессоров.

Для взаимодействия процессоров между собой на этапе корректировки сетки с изменением связей была разработана специальная технология, получившая название *ведущий — ведомый*. Применение данной технологии позволило решить многие сложные вопросы взаимодействия процессоров и однозначно производить изменения сеточных связей на границе параобластей.

Смысл технологии *ведущий — ведомый* состоит в том, что операции корректировки сетки проводятся поэтапно. На каждом этапе некоторые процессоры объявляются ведущими, а остальные — ведомыми. В окружении каждого ведущего процессора имеются только ведомые. На ведущем процессоре имеется возможность инициализировать его собственные корректировки сетки и отправлять изменения на смежные с ним ведомые процессоры. На ведомых процессорах не производится собственных сеточных изменений, а лишь принимаются сообщения от смежных ведущих процессоров и проводятся соответствующие изменения только в множестве несчетных объектов. На следующих этапах ведущие и ведомые процессоры меняются ролями. На протя-

жении всех этапов процессор имеет статус ведущего только один раз. Количество этапов равняется числу хроматичности *процессорного графа*. Изменение соседства в графе влечет за собой изменение его раскраски. Новая раскраска производится на одном процессоре, а затем каждому процессору отправляется *его* номер цвета. Этот номер определяет номер этапа, на котором данный процессор является ведущим. В общем случае граф может быть не планарным. Число хроматичности вычисляется приближенно. В качестве примера на рис. 8 показаны четыре этапа смены статусов 20 процессоров при матричной декомпозиции данных. Серым цветом показаны ведущие процессоры, белым — ведомые.

Операции корректировки сетки с изменением топологии связей между ее объектами имеют единую схему распараллеливания. На рис. 9 представлена блок-схема, поясняющая принцип распараллеливания операции перестройки ребра, лежащего на границе смежных параобластей. Практически на всех этапах, представленных на блок-схеме, присутствуют обмены между процессорами.

Другие операции корректировки сетки с изменением связей между объектами производятся по аналогичной схеме. Все эти операции разделены между собой и реализованы в виде независимых блоков. Внутри каждого блока произво-

дятся действия над сеткой, пересчет величин и обмены.

Для унифицирования операций корректировки была выбрана стратегия, при которой изменяемые объекты помечаются соответствующими признаками. Установку этих признаков производит программа проверки критериев. После проверки критериев могут возникнуть ситуации, когда некоторые помеченные ребра окажутся близлежащими друг к другу и их перестройка может привести к переклесту сетки (ребра 01 и 02 на рис. 10, см. также цветную вкладку). В связи с этим производится исправление признаков для ребер (некоторые признаки обнуляются). Кроме того, "снятие" признаков позволяет исключить зейделевскую зависимость от обхода ребер, делая перестройку текущего ребра локальной, не зависящей от перестройки других ребер. На рис. 11 (см. также цветную вкладку) показан пример, когда ребро 1 перестраивается, а для ребер 2—33 вокруг него производится снятие признаков. Перестройка ребра 1 сопровождается изменением связей между четырьмя ячейками (на цветном рисунке они обозначены красными точками) и ребрами 1, 2, 8, 9, 14.

Распараллеливание операции перестройки ребер состоит из нескольких этапов. Сначала происходит верификация перестроения ребер на границе смежных параобластей. Верификация

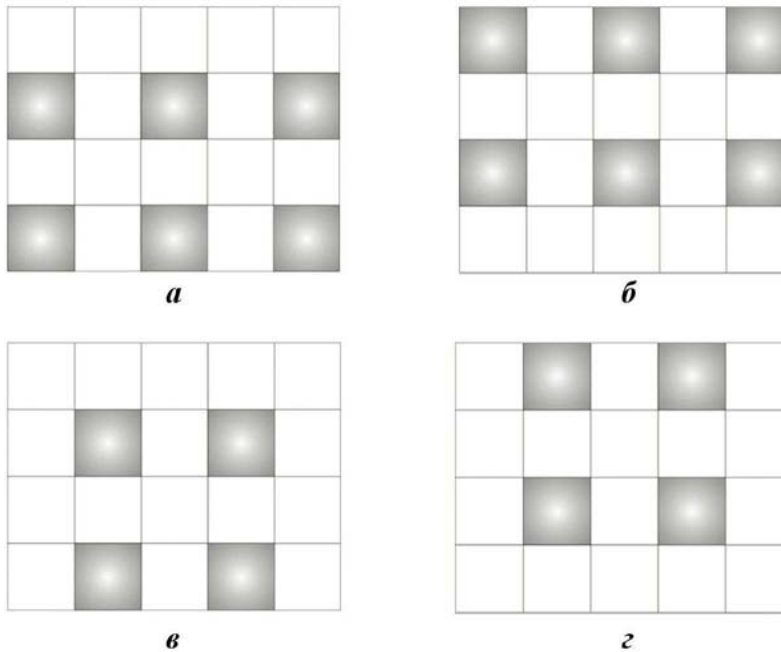


Рис. 8. Ведущие и ведомые процессоры: а — 1-й этап; б — 2-й этап; в — 3-й этап; г — 4-й этап

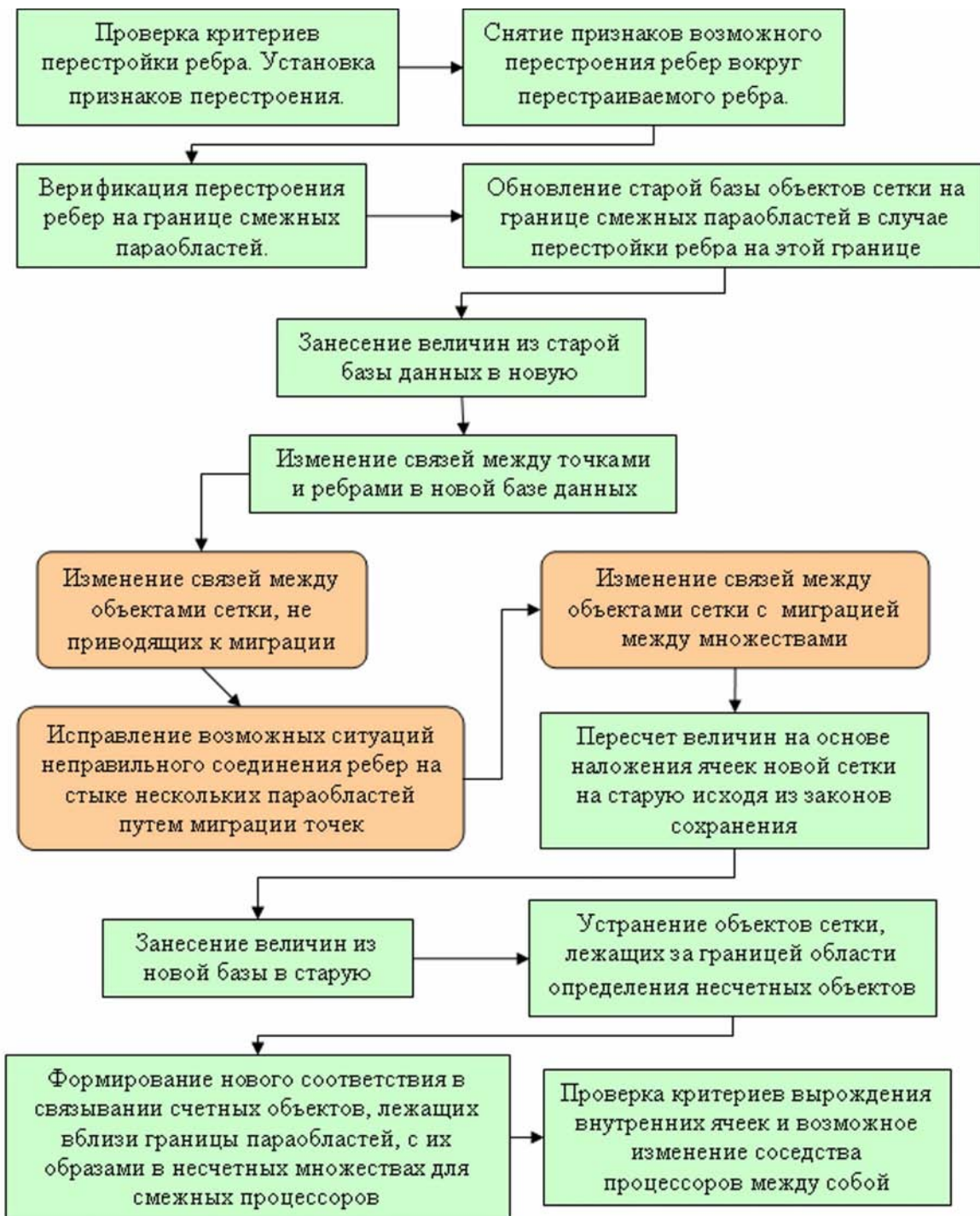


Рис. 9. Блок-схема операции перестройки ребра на границе параобластей

проводится путем обмена информацией со смежными процессорами и передачи на них глобальных номеров перестраиваемых ребер, у которых одна точка лежит в множестве оболочечных точек, а другая — в множестве N_1 точек. Этот этап необходим для идентичного перестроения сетки на границе смежных параобластей.

На следующем этапе организуется цикл по перестраиваемым ребрам, лежащим на границе параобластей, перестройка которых не разрывает соответствующих счетных множеств (т. е. не производится перемещения ячеек между множествами). Затем перестраиваются внутренние ребра параобласти. На всех вышеперечисленных

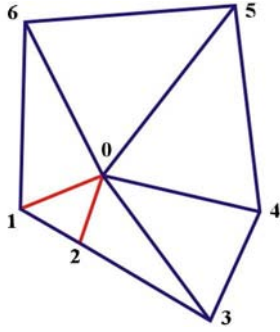


Рис. 10. Перестроение двух смежных ребер (01 и 02), приводящее к перехлесту сетки

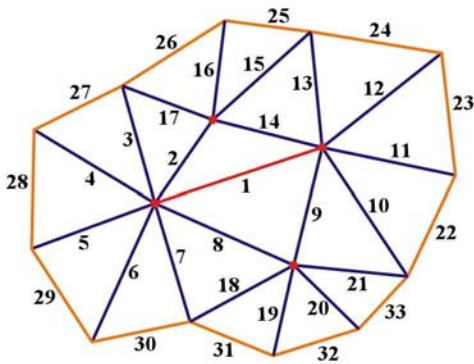


Рис. 11. Перестроение ребра 1 со снятием с ребер 2—33 возможных признаков перестроения

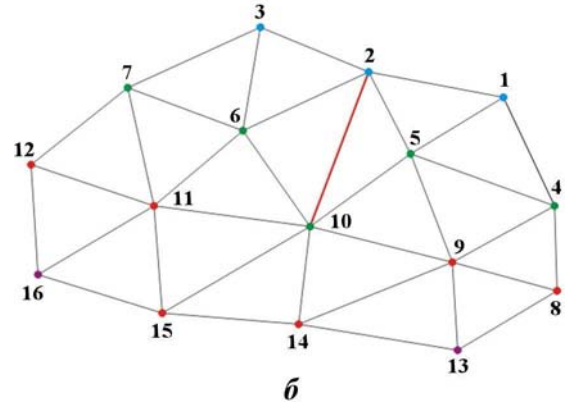
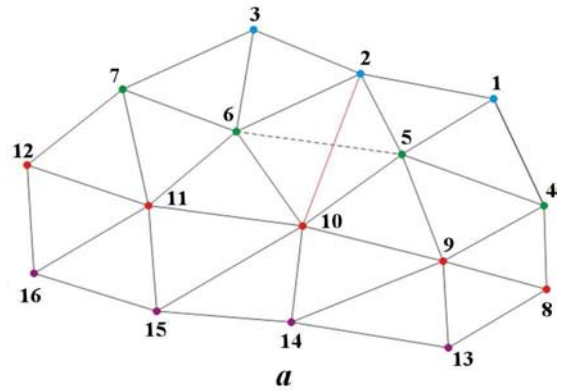


Рис. 12. Состояние множеств точек до (а) и после (б) перестройки ребра

этапах реализуется технология встречных обменов.

На следующем этапе перестраиваются такие ребра, которые влекут за собой разрыв счетных множеств. Для их восполнения используется алгоритм миграции [4]. Перестройка таких ребер реализована по технологии обменов *ведущий — ведомый*. Перестраиваемые ребра такого типа соединяют две оболочечные или две предоболочечные точки.

На рис. 12 (см. также цветную вкладку) показана перестройка обозначенного пунктиром ребра (5б), приводящая к разрыву множества оболочечных ячеек. На цветном рисунке зеленым цветом помечены оболочечные точки, красным — предоболочечные, фиолетовым — ядерные. Красной линией показано соединение перестроенного ребра.

Следующий этап алгоритма перестроек связан с пересчетом величин путем наложения ячеек измененной сетки на старые. Алгоритм пересчета проводится по стандартному сценарию методики МЕДУЗА, поэтому на нем останавливаться не будем. Отметим лишь, что обмены на этом этапе

реализованы с помощью технологии встречных обменов. После пересчета величин новая база данных переписывается в старую.

Последний этап — проверка на изменение соседства процессоров между собой. Изменение соседства процессоров производится в том случае, когда образ несчетной точки принадлежит процессору, не являющемуся соседним для текущего процессора.

Основная идея выполненной работы состоит в том, что операции корректировки сетки могут проводиться независимо от взаимного положения границ параобластей. Кроме того, реализована возможность изменения смежности параобластей, что значительно повышает уровень безаварийного счета задач. Для каждой параобласти содержится информация о количестве и номерах смежных параобластей. При смене топологии параобластей производится новая раскраска процессорного графа. В качестве критерия для смены топологии параобластей выбран случай, когда оболочечная или предоболочечная точка имеет несколько образов на соседних процессорах, которые не смежны между собой. Та-

кая ситуация представлена на рис. 13, на котором изображены четыре параобласти. Параобласть 1 содержит оболочечную точку O , имеющую образы на параобластях 2, 3, 4. При этом параобласти 2 и 4, а также 3 и 4 не смежны между собой. После смены топологии параобластей указанные пары окажутся смежными.

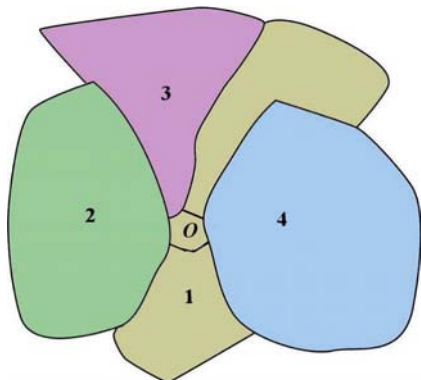


Рис. 13. Иллюстрация необходимости смены топологии параобластей

Результаты расчета тестовой задачи

Для подтверждения применимости алгоритмов был проведен расчет тестовой задачи о падении плоской ударной волны на заполненную тяжелым газом прямоугольную область. Постановка задачи описана в [5]. Расчет проводился на сетке, содержащей около 1 млн ячеек (мноугольной формы), с использованием 200 процессоров. Таким образом, после декомпозиции на каждом процессоре было около 5 000 счетных ячеек. Начальная декомпозиция задачи представлена на рис. 14 (см. также цветную вкладку).

В методике МЕДУЗА реализованы различные алгоритмы декомпозиции. Некоторые из них описаны в работах [6, 7].

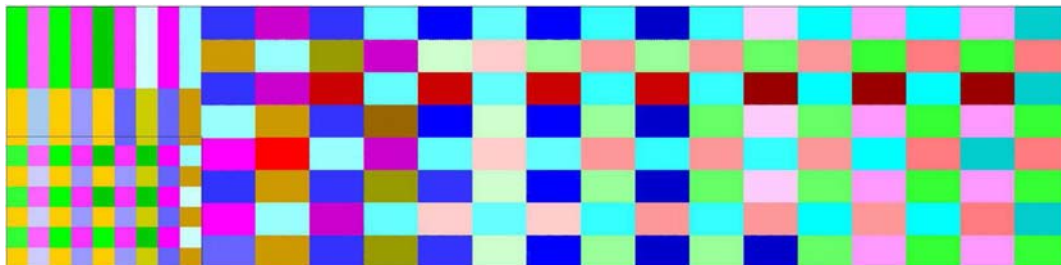


Рис. 14. Начальная декомпозиция задачи

Данная задача характеризуется тем, что в ней формируется плоская ударная волна, которая впоследствии ударяется о прямоугольную область, заполненную более тяжелым газом. В результате прохождения волны развивается неустойчивость Рихтмайера–Мешкова, приводящая к сильному вихревому течению. На этапе вихревого течения активно применяются все механизмы корректировки сетки без каких-либо ограничений, которые могут быть связаны с границами параобластей. Кроме того, в результате действия этих механизмов может нарушаться целостность (связность) параобласти. Целью проведения расчета тестовой задачи являлась демонстрация возможности безаварийного счета сложных течений с использованием всевозможных алгоритмов корректировки сетки в параллельном режиме счета.

Отметим, что в начале счета задачи эффективность распараллеливания составила около 45 %. Это связано с тем, что из 5 000 ячеек параобласти внутренних ячеек (ядерных) примерно 4 400, остальные ячейки имеют образы на смежных процессорах, и количество смежных процессоров для текущего процессора около 8. Кроме того, все алгоритмы корректировки сетки использовались с самого начала расчета, несмотря на то, что до 0,8 мс течение представляет собой плоскую ударную волну. Таким образом, при выбранном количестве ячеек процессорные обмены преобладают над вычислительной работой.

На рис. 15 (см. также цветную вкладку) приведена динамика развития течения и вид параобластей в зоне вихревого течения.

В моменты времени 1,20 мс и 1,36 мс с начала счета задачи проводилась передекомпозиция. Это связано с тем, что в области сильного вихревого течения из-за алгоритмов корректировки сетки параобласти подвергались сильной деформации, вплоть до потери связности. При этом изменялось количество приграничных (обло-

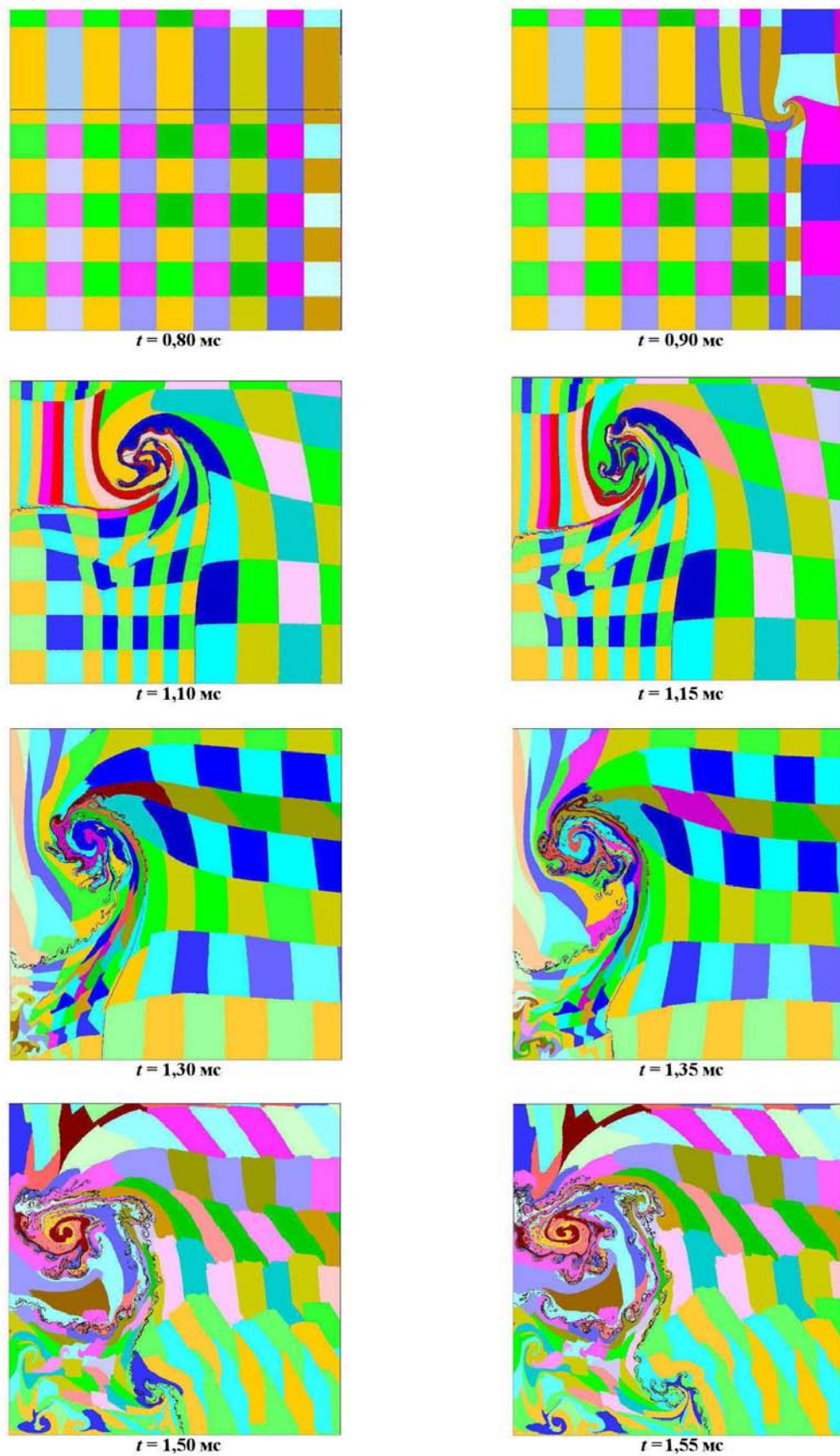


Рис. 15. Вид параобластей на различные моменты времени в зоне вихревого течения

чечных и предоболочечных) ячеек, увеличивалось количество смежных процессоров. Эти обстоятельства приводили к тому, что эффективность распараллеливания становилась ниже 20%. Таким образом, счет задачи как таковой не завершался аварийным остановом, но она считалась неэффективной. Алгоритмы повторной декомпозиции изменяют формы параобластей, и они снова приобретают компактный вид. На протяжении всего расчета соседство процессоров изменялось примерно 20 раз.

На рис. 16 представлена динамика деформации одной из параобластей.

На рис. 17 (см. также цветную вкладку) приведены графики количества ячеек, рассчитываемых на процессорах, в различные моменты времени, откуда следует, что с течением времени оно меняется. В начальный момент отклонение от среднего значения по количеству ячеек

составляет менее 5%, а при $t = 1,57$ мс — около 16%.

На рис. 18 приведена форма некоторых параобластей вблизи главного вихря на момент времени $t = 1,57$ мс.

Заключение

В настоящее время методика МЕДУЗА позволяет проводить расчеты задач со сложными течениями в параллельном режиме счета с использованием всевозможных алгоритмов корректировки сетки. Разработанные технологии обмена между смежными процессорами позволяют изменять топологию связей между объектами неструктурированной сетки без каких-либо ограничений, которые могут быть связаны с границами параобластей.

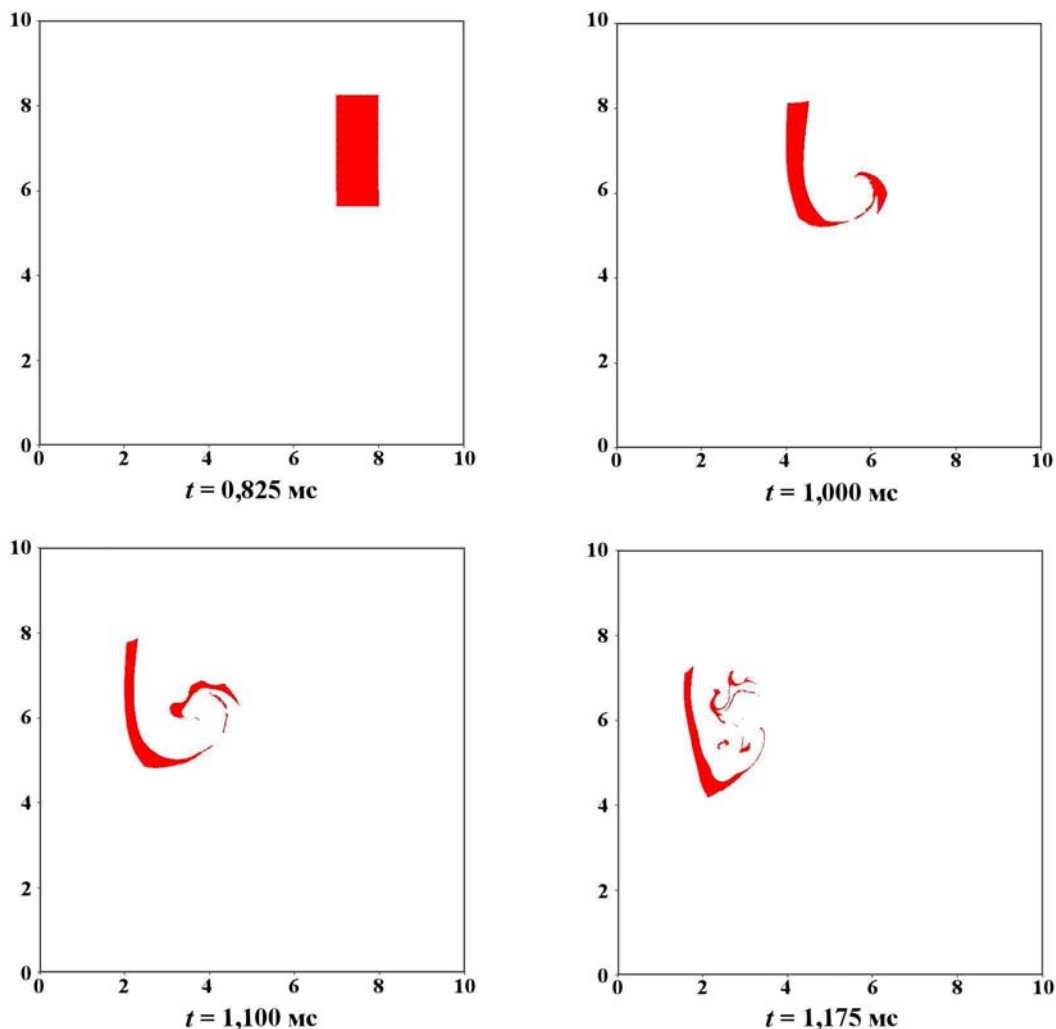


Рис. 16. Динамика изменения формы одной из параобластей

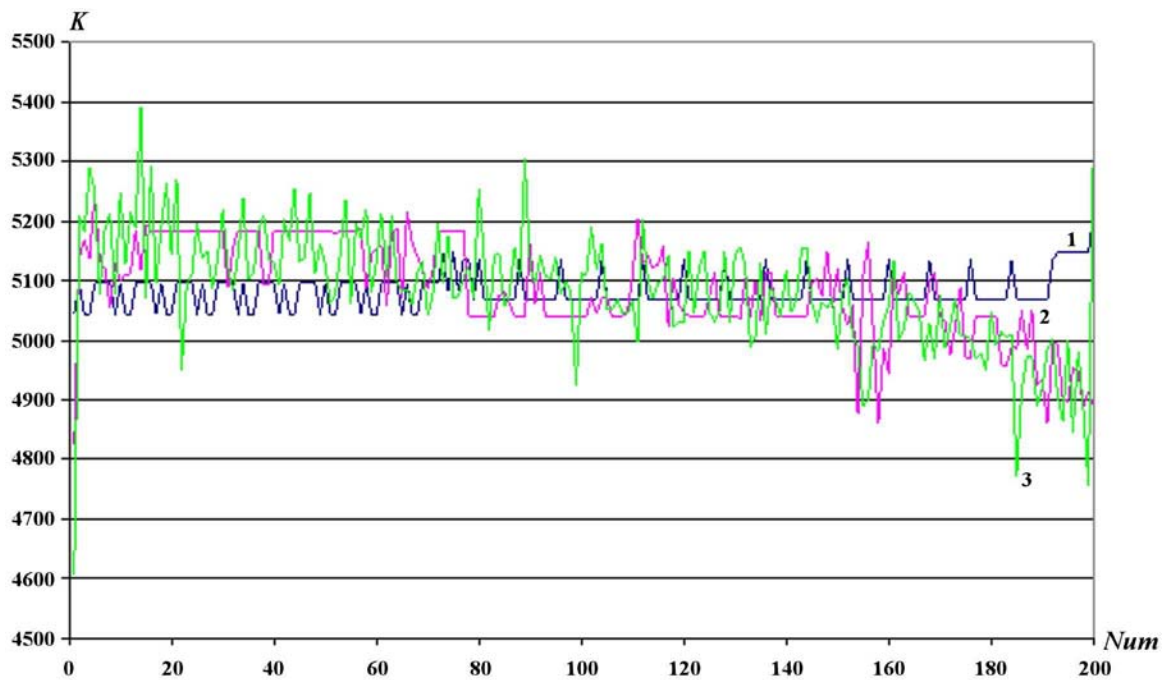


Рис. 17. Количество ячеек K на процессорах с номерами Num в различные моменты времени: 1 — $t = 0$; 2 — $t = 1,25$ мс; 3 — $t = 1,57$ мс

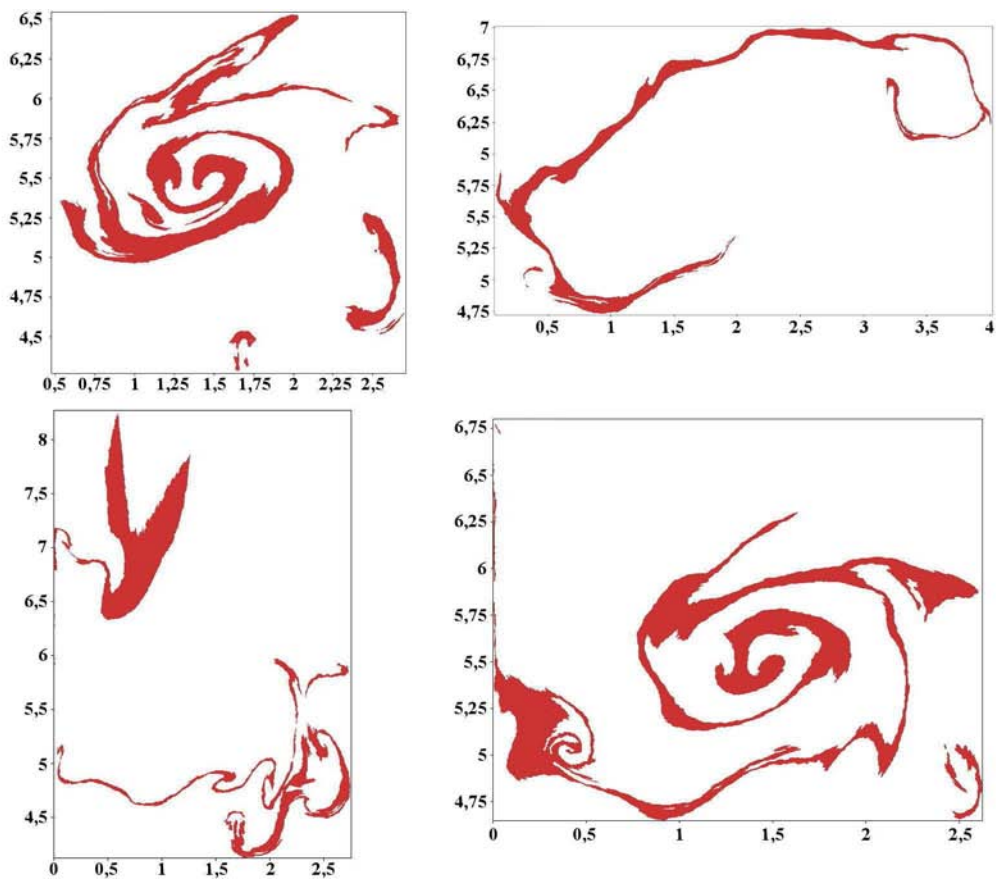


Рис. 18. Форма параобластей вблизи главного вихря, $t = 1,57$ мс

Список литературы

1. Глаголева Ю. П., Жогов Б. М., Кирьянов Ю. Ф., Мальшаков В. Д., Нестеренко Л. В., Подливаев И. Ф., Софронов И. Д. Основы методики МЕДУЗА численного расчета двумерных нестационарных задач газодинамики // Числ. методы мех. спл. среды. 1972. Т. 3, № 2. С. 18—55.
2. Butnev O. I., Gorbunov A. A., Malshakov V. D., Voronin B. L., Volkov S. G., Zhogov B. M., Pronin V. A., Sofronov I. D. Parallelization of 2D heat conduction problem computation on unstructured grids on distributed-memory computer system // PDPTA-2000. Las-Vegas, USA. Vol. V. P. 2891—2898.
3. Barabanov R. A., Butnev O. I., Voronin B. L., Volkov S. G., Zhogov B. M., Pronin V. A., Sofronov I. D. 2D gas dynamics problem computation parallelization on unstructured grids on distributed-memory computer // Ibid. P. 2899—2906.
4. Бутнев О. И., Пронин В. А. Организация миграции точек между процессорами в параллельных расчетах по методике Медуза // Сб. докл. Второй науч.-тех. конф. "Молодежь в науке". Саров: РФЯЦ-ВНИИЭФ, 2005. С. 143—148.
5. Жогов Б. М., Клопов Б. А., Мешков Е. Е., Пастернак В. М., Толшмяков А. И. Численный расчет и сравнение с экспериментом задачи о прохождении плоской ударной волны через "тяжелый" угол // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1992. Вып. 3. С. 59—65.
6. Бутнев О. И., Пронин В. А. Алгоритмы декомпозиции нерегулярной сетки с учетом вычислительной нагрузки // Там же. 2004. Вып. 4. С. 97—103.
7. Сидоров М. Л. Применение метода натянутых сеток для геометрической декомпозиции нерегулярной сетки в методике МЕДУЗА // Сб. докл. Седьмой науч.-тех. конф. "Молодежь в науке". Саров: РФЯЦ-ВНИИЭФ, 2008. С. 206—210.

Статья поступила в редакцию 10.09.10.