

УДК 519.6

МЕТОДЫ МЕЛКОЗЕРНИСТОГО РАСПАРАЛЛЕЛИВАНИЯ В МЕТОДИКЕ ТИМ-2D

А. А. Воропинов, И. Г. Новиков, С. С. Соколов
(РФЯЦ-ВНИИЭФ)

Рассматриваются методы мелкозернистого распараллеливания, применяемые в методике ТИМ-2D. Мелкозернистое распараллеливание осуществляется в модели распределенной памяти с использованием интерфейса MPI. При этом применяются два подхода: 1) с наложением параобластей в один слой ячеек; 2) без наложения параобластей с расчетом взаимодействия между ними. Приводится сравнение методов по эффективности распараллеливания на одном из тестовых расчетов.

Ключевые слова: методика ТИМ-2D, распараллеливание в модели распределенной памяти, интерфейс MPI, неструктурированные сетки.

Введение

Методика ТИМ-2D [1] предназначена для решения нестационарных задач механики сплошной среды на неструктурированных многоугольных лагранжевых сетках произвольного вида. Ячейки сетки представляют собой несамопересекающиеся многоугольники произвольной конфигурации, в узлах сетки может сходитьсь произвольное количество ребер.

Для методики ТИМ-2D используется трехуровневое распараллеливание [2]. На первом уровне осуществляется распараллеливание счета по математическим областям. На втором уровне распараллеливается счет внутри математической области по параобластям (математическая область разбивается на фрагменты). На этих двух уровнях используется модель распределенной памяти и интерфейс передачи сообщений MPI. На третьем уровне осуществляется распараллеливание итераций счетных циклов внутри параобластей в модели общей памяти с использованием интерфейса OpenMP. При расчете одной задачи эти подходы могут использоваться как вместе в различных сочетаниях, так и раздельно.

Ранее для методики ТИМ-2D реализовано распараллеливание в модели общей памяти [3] и распараллеливание по областям [4]. Деление на математические области в методике производится исходя из начальных данных, таких как пара-

метры геометрии и распределения веществ. При этом не преследуется цель достижения высокой эффективности распараллеливания. В разных областях может быть задано различное количество точек (иногда отличие может достигать нескольких порядков), используются разные вещества и приближения. Это приводит к тому, что, во-первых, на реальных задачах эффективность оказывается не слишком высокой, во-вторых, использовать такое распараллеливание удается для небольшого количества процессоров. Дополнительное распараллеливание в модели общей памяти позволяет увеличить количество задействованных процессоров почти на порядок. Тем не менее в реальных расчетах удается использовать не более нескольких десятков процессоров. Все это показывает, что такого распараллеливания недостаточно, поэтому необходима реализация мелкозернистого распараллеливания.

В статье рассматриваются алгоритмы мелкозернистого распараллеливания (второй уровень).

Декомпозиция данных

Эффективное использование счетных программ методик на *параллельных* машинах требует выполнения декомпозиции таким образом, чтобы ядра вычислительной системы были за-

гужены равномерно, а взаимодействия между ними были минимальны. В модели распределенной памяти задача декомпозиции состоит в распределении данных между процессами (декомпозиция данных) таким образом, чтобы количество и объем обменов между ними были минимальны. В методике ТИМ-2D для распараллеливания в модели распределенной памяти используется декомпозиция по пространству.

Принципы декомпозиции для мелкозернистого распараллеливания следующие:

- декомпозиция осуществляется по ячейкам (ячейка — основной счетный элемент сетки в методике ТИМ-2D);
- все ячейки области распределяются по компактам,* при этом каждая ячейка принадлежит одному и только одному компакту;
- каждая математическая область разбивается на компакты независимо от других.

Задача декомпозиции для мелкозернистого распараллеливания сводится к решению задачи о разрезании графа на подграфы. При этом используется следующий алгоритм:

1. На основе неструктурированной сетки строится граф, отображающий ее структуру. При этом ячейкам сетки соответствуют вершины графа, а соседству между ячейками — ребра графа.
2. Вершинам графа присваивается вес, отражающий вычислительную нагрузку, связанную с соответствующей ячейкой. Веса ребер графа используются для вытягивания компактов вдоль границ с целью придания декомпозиции дополнительного свойства — уменьшения количества обменов при расчете контактного взаимодействия.
3. Непосредственно задача о разрезании графа на подграфы решается при помощи алгоритмов библиотек ParMeTiS или SCOTCH [5–7].

Для задач, состоящих из нескольких математических областей, используется алгоритм двухпроходной декомпозиции. На первом проходе каждая область разбивается независимо от других на большое количество компактов. На втором проходе строится макрограф, вершинами

* Распределение по компактам является дополнительной информацией, на основе которой в дальнейшем производится формирование параобластей для исходной области.

которого являются компакты первого прохода и границы между областями. Ребра макрографа отражают соседство между компактами первого прохода и соседство с границами области. Для макрографа решается задача о разрезании на подграфы, и в результате получается распределение ячеек на требуемое количество компактов.

Подробное описание алгоритмов декомпозиции для методики ТИМ-2D приведено в работе [8].

Методы мелкозернистого распараллеливания

В методике ТИМ-2D расчет газодинамических величин разделен на два основных этапа, на которых соответственно рассчитываются:

1) *узловые величины* — скорости, координаты (расчет уравнения движения). При расчете узловых величин (для краткости — узлов) используется информация из соседних узлов и окружающих ячеек с предыдущего момента времени;

2) *ячеечные величины* — плотность, давление, энергия (расчет уравнения энергии). При этом используется информация о новом положении узлов рассматриваемых ячеек.

Основной вопрос, который возникает при мелкозернистом распараллеливании — это способ расчета узлов сетки, которые окружают ячейки, отнесенные к разным компактам (*параграничные узлы*).

Рассмотрим фрагмент сетки, представленный на рис. 1, где ячейки белого цвета отнесены к компакту 1, серого цвета — к компакту 2.

В соответствии с разностной схемой методики [1] для расчета узла строится замкнутый контур интегрирования, проходящий через центры

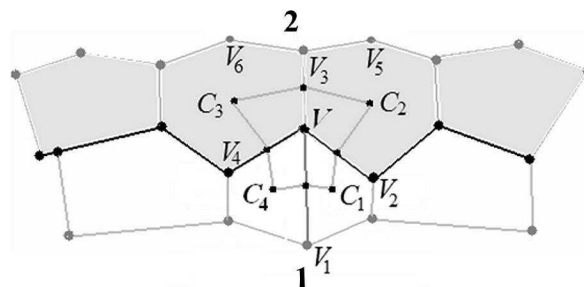


Рис. 1. Фрагмент сетки с разделением на компакты и с контуром интегрирования для узла

ячеек и *центры ребер* (центр ребра в цилиндрическом случае — точка, которая делит поверхность, образуемую вращением ребра вокруг оси симметрии, на две равные части; в плоском случае — это середина ребра). В контур интегрирования для узла V на рис. 1 включаются центры окружающих ячеек C_1, C_2, C_3, C_4 и центры ребер VV_1, VV_2, VV_3, VV_4 . Соответственно при расчете узла V используются величины, определенные в ячейках C_1, C_2, C_3, C_4 и узлах V, V_1, V_2, V_3, V_4 .

Если сохранять контур интегрирования для параграничного узла неизменным, то получается первый тип мелкозернистого распараллеливания — с наложением параобластей в один слой ячеек. При этом параобласть формируется из ячеек соответствующего компакта с добавлением прилегающего слоя ячеек из соседних компактов. Так, при формировании параобласти 1 (см. рис. 1) к компакт 1 добавляются ячейки серого цвета; аналогично, в параобласть 2 входят ячейки белого цвета. Для сохранения контура интегрирования со стороны первой параобласти не требуется полного формирования ячеек C_2, C_3 , т. е. не нужна информация об узлах V_5, V_6 . Однако если эти узлы не присоединять к первой параобласти, то сетка будет получаться не полной, нельзя будет выполнять некоторые операции с ячейками C_2, C_3 . В этом случае присоединенные ячейки придется особым образом описывать в структуре данных и учитывать в различных счетных алгоритмах. Поскольку одним из базовых принципов является максимальная прозрачность распараллеливания для счетных алгоритмов, то было решено включать в параобласти все узлы присоединенных ячеек (будем называть их также присоединенными). Таким образом, для параобласти 1 присоединенными являются узлы V_3, V_5, V_6 .

Совокупность ячеек, ребер и узлов слоя наложения двух соседних параобластей будем называть их *параллельной границей*.

Другой возможный вариант — представить контур интегрирования как совокупность замкнутых контуров со стороны каждой из ячеек (такой способ используется в разностной схеме для определения массы узла). Аналогично, можно представить контур интегрирования как совокупность замкнутых контуров со стороны каждой параобласти. Такой вариант показан на рис. 2 (см. также цветную вкладку); линия раздела контура интегрирования выделена (красным цветом). При разделении контура интегри-

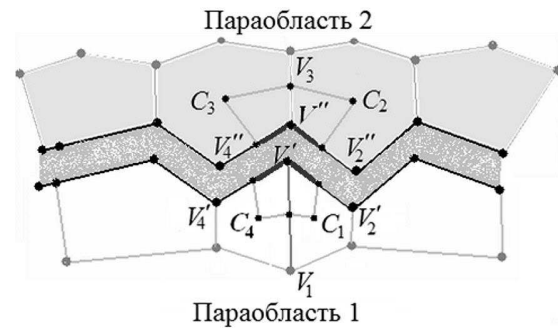


Рис. 2. Разрыв контура интегрирования для узла при мелкозернистом распараллеливании без наложения

рования каждый узел вдоль линии между параобластями разделяется на два (или на большее число узлов — по количеству параобластей, сходящихся в этом узле), например, узел V — на V' и V'' . Для каждого из узлов используется свой контур интегрирования для определения массы и ускорения — полученные значения в совокупности необходимы для расчета единой скорости.

Такое разделение контура интегрирования при расчете позволяет не использовать наложение между параобластями. Подобная схема используется при расчете контактной границы (КГ) *без скольжения*. Данная КГ в этом случае является и параллельной границей. Для мелкозернистого распараллеливания был использован программный блок расчета КГ [9] с небольшими доработками.

Формирование параобластей

При выполнении декомпозиции для каждой математической области формируется свое распределение ячеек по компактам. В результате декомпозиции создается целочисленный массив с числом элементов, равным количеству ячеек математической области. Значением элемента массива для соответствующей ячейки является номер компакта, к которому она относится. На основе этой информации формируются параобласти, т. е. происходит распределение ячеек, ребер и узлов по параобластям, введение новой нумерации элементов сетки в параобласти, подготовка информации для выполнения обменов. Параобласть, в отличие от компакта, предназначена для выполнения расчетных алгоритмов.

Формирование параобластей является головным процессом для математической области. Этот процесс выполняется в четыре основных

этапа. Первый этап — формирование структуры параобластей в рамках исходной математической области — затрагивает нумерацию элементов сетки, используемую в математической области. Второй этап — формирование параллельных границ между параобластями — также выполняется на уровне нумерации элементов математической области. На третьем этапе производится переход от глобальной нумерации к локальной нумерации в параобластях и соответствующих параллельных границах. На этом этапе производится непосредственное формирование параобластей. На последнем, четвертом этапе сформированные параобласти передаются процессам, которые будут их рассчитывать, где они сохраняются как обычные математические области с некоторой дополнительной информацией. При этом области присваивается сквозной номер в задаче, т. е. на разных процессах не может быть области (или параобласти) с одним и тем же номером.

Пример построения параобластей на примере сетки смешанного типа, изображенной на рис. 3, показан на рис. 4, 5. Здесь указаны номера ячеек, номера узлов и номера ребер. Жирными линиями выделен слой ячеек для режима мелкозернистого распараллеливания с наложением параобластей, пунктирными линиями — ребра, разделяющие ячейки из разных компактов. К первому компакту относятся ячейки с номерами от 1 до 9, ко второму — от 10 до 19. В результате построения получаются две параобласти: в режиме распараллеливания с наложением — такие, как на рис. 4, без наложения — как на рис. 5.

На рис. 4, 5 указана локальная нумерация ячеек и узлов рассматриваемых параобластей.

Алгоритмы мелкозернистого распараллеливания с наложением параобластей

Первый метод мелкозернистого распараллеливания основан на сохранении контуров интегрирования для параграничных узлов. В результате возникает необходимость наложения между параобластями в один слой ячеек. При этом параграничные узлы могут рассчитываться любым из процессов, рассчитывающих окружающие параобласти.

Расчет узлов является относительно дешевой операцией в методике ТИМ-2D, поэтому для дополнительного контроля расчет параграничных узлов дублируется. Второй важный момент — обновление информации для присоединенных элементов сетки (ячеек и узлов). Это обновление производится путем асинхронных обменов информацией для параграничных, приграничных и присоединенных элементов. Вызовы процедур выполнения обменов расположены в программе так, чтобы необходимая информация обновлялась до ее использования.

Блок-схема расчета шага с учетом мелкозернистого распараллеливания с наложением приведена на рис. 6. Серым цветом выделены этапы обращения к процедурам выполнения асинхронных обменов; пунктирными линиями отображены фактические потоки передаваемых данных

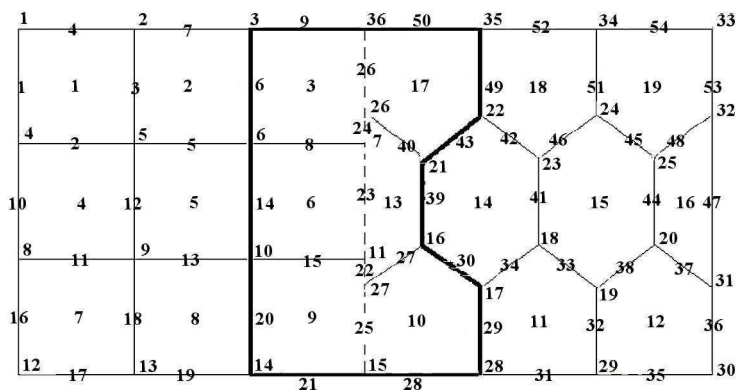


Рис. 3. Пример сетки для построения параобластей

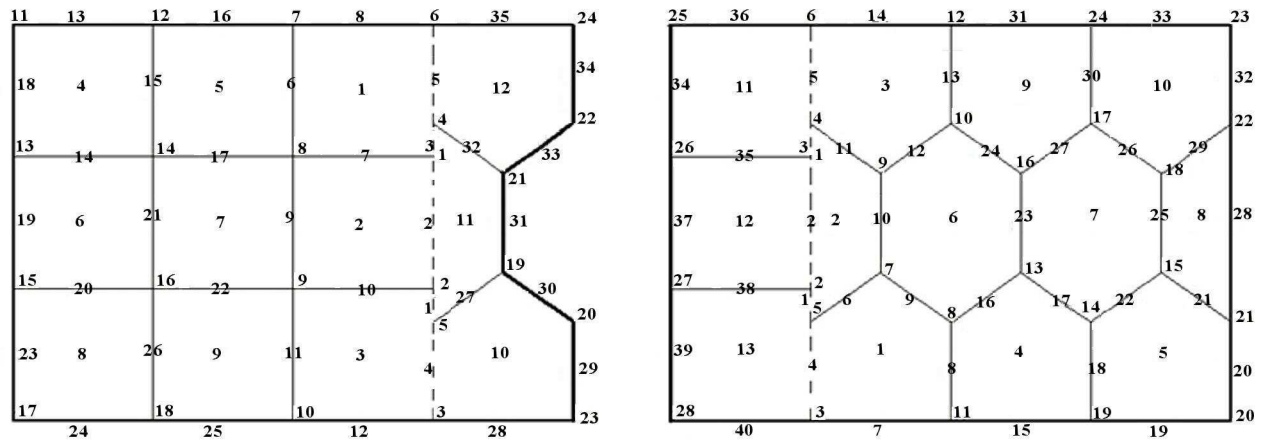


Рис. 4. Параобласти для мелкозернистого распараллеливания с наложением

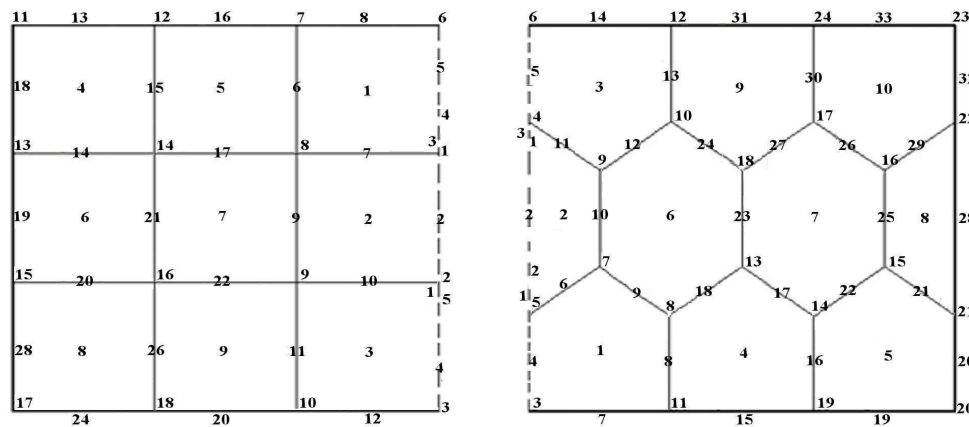


Рис. 5. Параобласти для мелкозернистого распараллеливания без наложения

между процессами. Как видно из блок-схемы, обмены выполняются одновременно с расчетом внутренних ячеек и узлов. При выполнении обменов используются упаковка и распаковка значений величин в буферный массив.

Алгоритмы мелкозернистого распараллеливания без наложения параобластей

Второй метод мелкозернистого распараллеливания основан на разбиении контуров интегрирования параграничных узлов. В результате между параобластями не требуется наложения по ячейкам. При разрыве контура интегрирования в каждой параобласти появляется свой параграничный узел со своим контуром интегрирования. Для восстановления исходного контура интегрирования необходимо их объединение.

В этом алгоритме для каждого из появившихся параграничных узлов должно рассчитываться ускорение, после чего производится обмен рассчитанными ускорениями и массами и расчет единой скорости. Получение единой скорости является относительно дешевой операцией, поэтому для уменьшения количества обменов она дублируется процессами.

Скорость узлов определяется следующим образом. Для узла i_0 (рис. 7)

$$u_{i_0}^{n+1} = u_{i_0}^n + \left((a_x)_{i_0} m_{i_0} + \sum_{\xi=1}^k (a_x)_{l_\xi} dm_{l_\xi} \right) \tau \frac{1}{M_{i_0}};$$

$$v_{i_0}^{n+1} = v_{i_0}^n + \left((a_y)_{i_0} m_{i_0} + \sum_{\xi=1}^k (a_y)_{l_\xi} dm_{l_\xi} \right) \tau \frac{1}{M_{i_0}};$$

$$M_{i_0} = m_{i_0} + \sum_{\xi=1}^k m_{l_\xi},$$



Рис. 7. Фрагмент контактной границы

где k — количество точек с другой стороны границы, попавших на поверхность, относящуюся к узлу i_0 (в осесимметричном случае она ограничена серединами поверхностей вращения i_0i_1 и i_0i_2 , а в плоском случае — серединами отрезков i_0i_1 и i_0i_2); $(dm)_{i_1}, \dots, (dm)_{i_k}$ — доли масс узлов l_1, \dots, l_k , попавших в окрестность узла i_0 ; a_x, a_y — компоненты векторов ускорений узлов, вычисленные по формуле, согласованной с формулой для внутренних узлов [1, 9], в предположении, что на границе задан нулевой тензор напряжений с каждой из сторон КГ.

После вычисления скорости движения КГ определяется ее положение на новый, $(n + 1)$ -й, момент времени по известным координатам на предыдущий, n -й, момент времени:

$$x_{k_0}^{n+1} = x_{k_0}^n + u_{k_0}^{n+1} \tau; \quad y_{k_0}^{n+1} = y_{k_0}^n + v_{k_0}^{n+1} \tau.$$

Перед описанием алгоритмов расчета контактного взаимодействия в параллельном режиме необходимо определить, каким образом распределяется расчет контуров между процессами. Основной принцип: КГ между двумя областями рассчитывается одним из двух процессов, к которым отнесены области, формирующие эту границу.

Расчет контактного взаимодействия в параллельном режиме производится по следующему алгоритму (распараллеленные шаги отмечены звездочкой):

1. Для каждого контура формируются *свои* стороны (в последовательном режиме все стороны — свои).
- 2.* Производится обмен информацией о сторонах контуров (свои отправляются, чужие принимаются).

Все последующие операции выполняются только для тех контуров, которые рассчитываются текущим процессом.

3. Совмещение сторон контура.
4. Формирование основного и вспомогательного слоя узлов на контуре.

5. Расчет расстояний между соседними точками на контуре.
6. *Посадка* узлов вспомогательного слоя на основную слой.
- 7.* Обмен информацией о сформированных контурах между процессами.
8. Расчет ускорений и скоростей для граничных узлов своих областей.
- 9.* Получение процессом, рассчитывающим контур, информации о чужой стороне контура. (Соответственно эта сторона подготавливается и отправляется процессом, рассчитывающим соседнюю область).
10. Расчет взаимодействия сторон контура. Вычисление новых координат и скоростей точек контура.
- 11.* Пересылка рассчитанного контура.

Заметим, что обмены информацией производятся только для тех контуров, которые разделяют области, рассчитываемые разными процессами. Если разделяемые области рассчитываются одним и тем же процессом, то и контур рассчитывается этим же процессом. Аналогично, внешний контур области задачи (типа *свободная граница* или *жесткая стенка*) рассчитывается процессом, рассчитывающим эту область.

Исследование эффективности распараллеливания

В качестве характеристик эффективности распараллеливания использовались следующие функции: $S_p = t_1/t_p$ — ускорение счета; $E_p = t_1/pt_p \cdot 100\%$ — эффективность распараллеливания, где t_1 — время расчета на одном процессоре используемой *параллельной* машины (последовательный режим), t_p — время счета на p процессорах.

Для исследования эффективности мелкозернистого распараллеливания использовалась задача о плоской волне в однообластной постановке. Размеры области — 1×1 . Все границы области, кроме левой, являются жесткими стенками. Вещество — идеальный газ с параметрами $\gamma = 3$, $\rho_0 = 1$. На левой границе задано постоянное по времени давление $P_{гр} = 5$.

Использовалась сетка диаграммы Вороного из 1 млн ячеек со случайной расстановкой центров. Фрагмент сетки приведен на рис. 8. Виды декомпозиций для различного количества процессов приведены на рис. 9 (см. также цветную вкладку).

В табл. 1, 2 приведены значения ускорения и эффективности распараллеливания. Расчеты проводились в модели смешанной памяти. За единицу принят счет на одном узле с использованием OpenMP-распараллеливания. Сравнение эффективности режимов мелкозернистого распараллеливания приведено на рис. 10.

Как видно из представленных результатов, методы дают близкую эффективность распараллеливания. Однако при мелкозернистом распараллеливании без наложения параобластей пока нет совмещения обменов и вычислений. Можно ожидать, что при реализации такого совмещения для этого метода будет достигаться более высо-

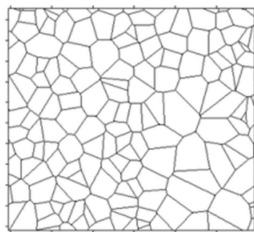


Рис. 8. Фрагмент сетки диаграммы Вороного

Таблица 1

Характеристики эффективности мелкозернистого распараллеливания с наложением параобластей

Кол-во ядер (MPI×OpenMP)	S_p	$E_p, \%$
8(1×8)	1	100
64(8×8)	6,8	85
80(10×8)	8,26	82
128(16×8)	11,5	72
256(32×8)	18,7	58
400(50×8)	24,5	49

Таблица 2

Характеристики эффективности мелкозернистого распараллеливания без наложения параобластей

Кол-во ядер (MPI×OpenMP)	S_p	$E_p, \%$
8(1×8)	1	100
64(8×8)	6,9	86
80(10×8)	8,4	84
128(16×8)	12,3	77
256(32×8)	17,5	54
400(50×8)	22,1	44

кая эффективность распараллеливания за счет меньшего объема передаваемой информации.

Тестовый расчет

Для тестирования смешанных режимов распараллеливания была выбрана задача о полете несферической оболочки [10]. Начальная геометрия задачи представлена на рис. 11. Начальная форма оболочки: $R(t = 0, \theta) = 7 + 0,875 \cos^3 \theta - 0,525 \cos \theta$; толщина $\Delta R = 0,16$. Материал — свинец; $\rho_0 = 11,4$. Оболочка имеет скорость, направленную к центру: $v = 5\,706\,860$. Давление на наружной поверхности оболочки $P_e = 0$; давление на внутренней поверхности зависит от времени: $P_i = 3,495 \cdot 10^6 (V(t))^{-5/3}$, где $V(t)$ — объем внутренней части полости.

Результаты расчетов приведены на рис. 12, 13 (см. также цветную вкладку). Во всех расчетах получено полное совпадение результатов. Сравнение формы оболочки, полученной при расчетах в последовательном и параллельном режимах, показано на рис. 12.

Заключение

В статье дано описание двух методов мелкозернистого распараллеливания, применяемых в методике ТИМ-2D. Первый подход заключается в сохранении контура интегрирования для узлов и использовании наложения в один слой ячеек между параобластями. В слое наложения производится обмен информацией об узлах и ячейках. Второй подход заключается в разделении контура интегрирования для узлов и расчете взаимодействия между параобластями.

Методы дают близкую эффективность распараллеливания. Однако при мелкозернистом распараллеливании без наложения параобластей не реализовано совмещение обменов и вычислений. Можно ожидать, что при реализации такого совмещения мелкозернистое распараллеливание без наложения параобластей даст более высокую эффективность распараллеливания за счет меньшего объема передаваемой информации.

Разработанные алгоритмы дополняют распараллеливание по областям и OpenMP-распараллеливание, реализованные для методики ТИМ-2D ранее.

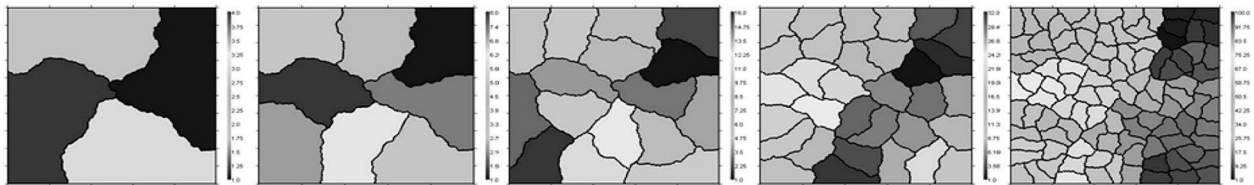


Рис. 9. Виды декомпозиций для фрагмента сетки диаграммы Вороного

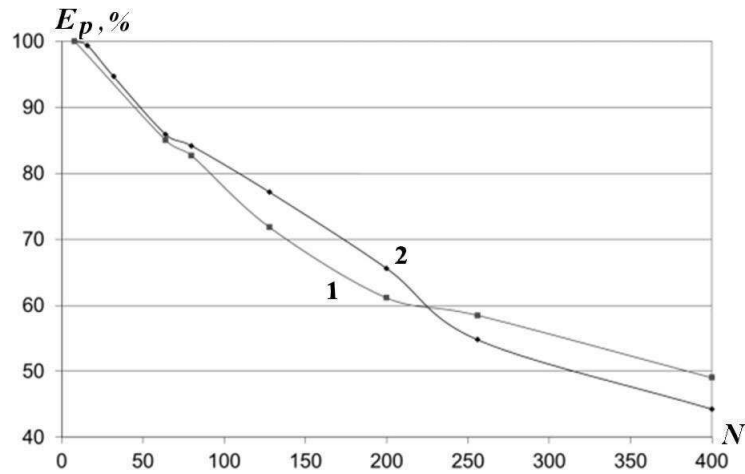


Рис. 10. Эффективность мелкозернистого распараллеливания: 1 — с наложением параобластей; 2 — без наложения параобластей (N — количество процессоров)

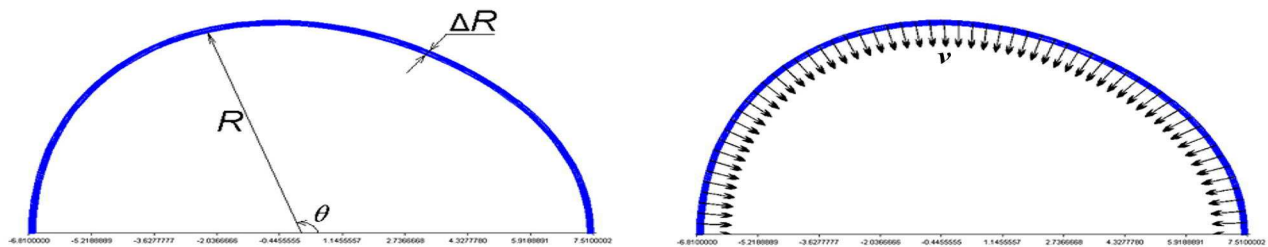


Рис. 11. Начальные форма и скорость оболочки в тестовой задаче

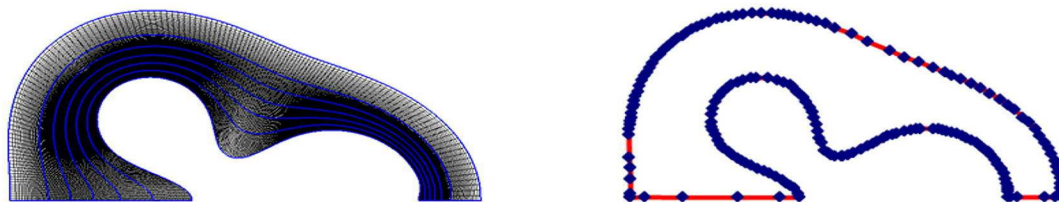


Рис. 12. Вид расчетной сетки на момент времени $t = 1,1$ и сравнение формы оболочки, полученной при расчетах в последовательном (\blacklozenge) и параллельном (—) режимах

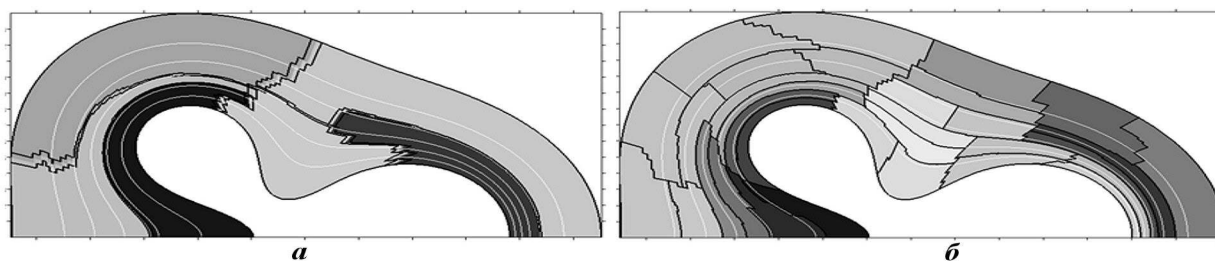


Рис. 13. Форма области и разделение на параобласти: *a* — с наложением; *b* — без наложения

Список литературы

1. Соколов С. С., Воропинов А. А., Новиков И. Г. и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29—43.
2. Воропинов А. А., Соколов С. С., Новиков И. Г. Распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике ТИМ-2D // Параллельные вычислительные технологии (ПаВТ'2008): Тр. межд. науч. конф. (Санкт-Петербург, 28 января — 1 февраля 2008 г.). Челябинск: Изд-во ЮУрГУ, 2008. С. 69—79.
3. Воропинов А. А., Новиков И. Г., Соболев И. В. и др. Распараллеливание методики ТИМ в модели общей памяти с использованием интерфейса OpenMP // Вычислительные методы и программирование. 2007. Т. 8, № 1. С. 134—141.
4. Воропинов А. А., Соколов С. С., Новиков И. Г. Двухуровневое распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2008. Вып. 1. С. 51—59.
5. Половникова Т. Н., Воропинов А. А. Опыт использования библиотек SCOTCH и MeTiS для декомпозиции неструктурированных сеток в методике ТИМ // Супервычисления и математическое моделирование. Тр. XII межд. семинара / Под ред. Р. М. Шагалиева. Саров: РФЯЦ-ВНИИЭФ, 2011. С. 282—288.
6. ParMETIS — Parallel Graph Partitioning and Fill-reducing Matrix Ordering. <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>.
7. Pellegrini F. SCOTCH: Static Mapping, Graph, Mesh and Hypergraph Partitioning, and Parallel and Sequential Sparse Matrix Ordering Package. <http://www.labri.fr/perso/pelegrin/scotch/>.
8. Воропинов А. А. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества // Вестник ЮУрГУ. Сер. Математическое моделирование и программирование. 2009. № 37(170). Вып. 4. С. 40—50.
9. Воропинов А. А., Новиков И. Г., Соколов С. С. Расчет контактного взаимодействия между счетными областями в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2008. Вып. 2. С. 5—20.
10. Sofronov I. D., Rasskazova V. V., Nesterenko L. V. The use of nonregular nets for solving two-dimensional nonstationary problems in gas dynamics // Numerical Methods in Fluid Dynamics / Ed. by N. N. Yanenko, Ju. I. Shokin. Moscow: Mir Publishers, 1984. С. 82—121.

Статья поступила в редакцию 05.12.11.