

Алгоритмы мелкозернистого распараллеливания в методике ТИМ-2D

А.А. Воропинов

Методика ТИМ-2D предназначена для решения задач механики сплошной среды на нерегулярных многоугольных сетках произвольной структуры. Для методики ТИМ-2D реализуется трехуровневое распараллеливание. На верхнем уровне осуществляется распараллеливание счета по математическим областям в модели распределенной памяти с использованием интерфейса передачи сообщений MPI. На втором уровне распараллеливается счет внутри счетной области по параобластям (счетная область разбивается на фрагменты с наложением в один слой ячеек), также с использованием MPI. На третьем (нижнем) уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Эти подходы могут использоваться как вместе в различных сочетаниях, так и отдельно при расчете одной задачи.

В докладе рассматриваются алгоритмы мелкозернистого распараллеливания (второй уровень) и взаимодействие алгоритмов мелкозернистого распараллеливания с распараллеливанием по областям и OpenMP распараллеливанием. Приводятся замеры эффективности распараллеливания на ряде тестовых расчетов.

1. Введение

Методика ТИМ-2D [1] предназначена для решения задач механики сплошной среды на нерегулярных многоугольных лагранжевых сетках произвольной структуры. Для методики ТИМ-2D реализуется трехуровневое распараллеливание [2]. На верхнем уровне осуществляется распараллеливание счета по математическим областям в модели распределенной памяти с использованием интерфейса передачи сообщений MPI. На втором уровне распараллеливается счет внутри математической области по параобластям (математическая область разбивается на фрагменты с наложением в один слой ячеек), также с использованием MPI. На третьем (нижнем) уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Эти подходы могут использоваться как вместе в различных сочетаниях, так и отдельно при расчете одной задачи. Ранее для методики ТИМ-2D было реализовано распараллеливание в модели общей памяти [3] и распараллеливание по областям [4].

В методике ТИМ-2D деление на математические области производится исходя из начальных данных и геометрии. При этом, не преследуется цель достижения высокой эффективности распараллеливания. В разных областях может использоваться существенно различное количество точек (иногда отличие может достигать нескольких порядков). 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 использование распараллеливания в модели общей позволяет увеличить количество задействованных процессоров почти на порядок (тем не менее, в реальных расчетах удается использовать не более нескольких десятков процессоров), но не решает проблему эффективности распараллеливания. Все это показывает, что такого распараллеливания оказывается недостаточно, поэтому необходима реализация мелкозернистого распараллеливания.

В докладе рассматриваются алгоритмы мелкозернистого распараллеливания (второй уровень) и взаимодействие алгоритмов мелкозернистого распараллеливания с распараллеливанием по областям и OpenMP распараллеливанием.

2. Декомпозиция данных для мелкозернистого распараллеливания

Принципы декомпозиции для мелкозернистого распараллеливания следующие:

- Декомпозиция осуществляется по ячейкам (ячейка – основной счетный элемент в ТИМ-2D).

- Каждая математическая область разбивается на компакты не зависимо от других.
- Все ячейки области распределяются по компактам, при этом каждая ячейка принадлежит одному и только одному компакт.
- Вдоль границ выделяется отдельный компакт, содержащий один или несколько слоев ячеек (это ограничение для корректного расчета контактных границ).

Непосредственно для декомпозиции используется библиотека ParMeTiS [5]. Декомпозиция выполняется следующим образом:

- На основе нерегулярной сетки строится граф, отображающий ее структуру. При этом ячейкам сетки соответствуют вершины графа, а соседству между ячейками – ребра графа.
- Приграничные вершины графа выделяются в отдельный компакт, состоящий из нескольких слоев ячеек (этот компакт образует замкнутое кольцо). Этот компакт является первым в данной области.
- Для графа (за исключением приграничных вершин, выделенных в отдельный компакт) вызывается одна из программ библиотеки ParMeTiS, которые формируют декомпозицию.

3. Формирование параобластей

На основе каждого компакта строится параобласть, при этом между параобластями формируется наложение в один слой ячеек. При этом в каждой параобласти выделяются следующие элементы:

1. Параграничные узлы и ребра. Ребро называется параграничным, если оно разделяет две ячейки, относящиеся к разным компактам; узел называется параграничным, если в нем сходится, хотя бы одно параграничное ребро.
2. Приграничные элементы сетки. Ячейка называется приграничной, если она относится к компакт соответствующему рассматриваемой параобласти и хотя бы одно из ее ребер является параграничным. Ребра и узлы, не являющиеся параграничными, называются приграничными, если хотя бы одна из ячеек их формирующих является приграничной.
3. Присоединенные элементы сетки. Ячейка называется присоединенной, если она не относится к компакт соответствующему рассматриваемой параобласти, но входит в нее из-за наложения между параобластями. Ребра и узлы, не являющиеся параграничными, называются присоединенными, если все ячейки их формирующие являются присоединенными.
4. Все остальные элементы сетки рассматриваемой параобласти называются внутренними.

В счетных структурах параобласть представляется как математическая область с некоторой дополнительной информацией. При этом в параобласти устанавливается своя сквозная нумерация элементов сетки. Расчет величин производится для параграничных, приграничных и внутренних элементов сетки. Информация о присоединенных элементах обновляется на каждом счетном шаге с процессов их рассчитывающих.

При формировании параобласти в первую очередь нумеруются параграничные элементы сетки, затем приграничные, затем внутренние и в последнюю очередь присоединенные. Это позволяет в первую очередь рассчитывать параграничные и приграничные элементы сетки, чтобы начать выполнение обменов в асинхронном режиме, пока рассчитываются внутренние элементы, с другой стороны позволяет достаточно просто исключить из счета присоединенные элементы сетки, расчет которых производится в других параобластях.

Простейший пример разбиения области на две параобласти представлен на рисунках 1-2.

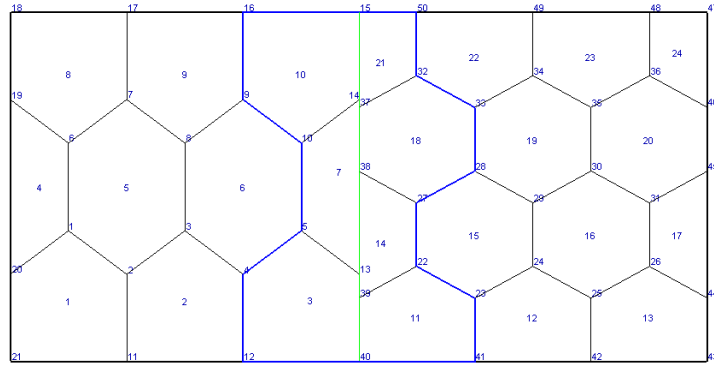


Рис. 1 – Математическая область из двух подобластей

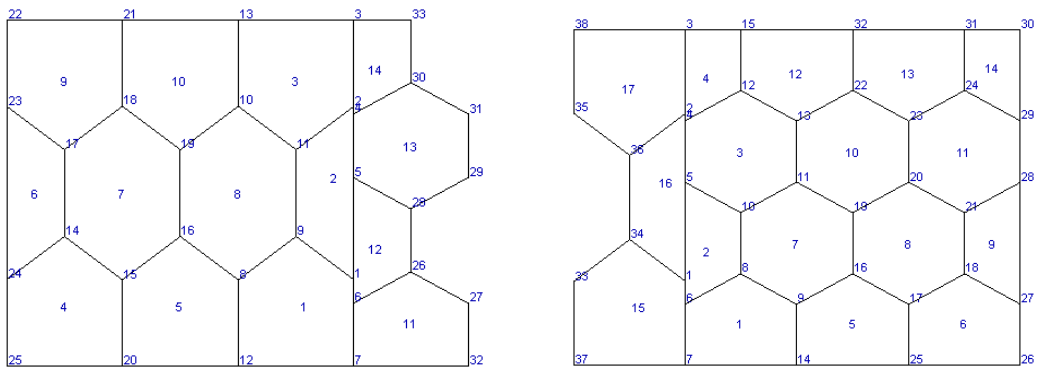


Рис. 2 – Вид сетки и нумерация элементов сетки в первой и второй параобласти

Набор элементов сетки одновременно существующих в двух параобластях и формирующих собой область наложения образуют собой параграницу, вдоль которой производятся обмены между параобластями. При этом параграница не является какой-то непрерывной линией, а представляется набором элементов сетки и может состоять из нескольких фрагментов, например в случае, когда компакт является не односвязным.

4. Алгоритмы мелкозернистого распараллеливания

В режиме мелкозернистого распараллеливания возникает задача обмена информацией об элементах сетки, находящихся в областях наложения между параобластями для корректного проведения расчетов.

В методике ТИМ-2D расчет газодинамических величин разделен на два основных этапа:

1. Расчет узловых величин – скорости, координаты (уравнение движения). При этом при расчете узла используется информация из соседних узлов и окружающих ячеек с n -ого момента времени.
2. Расчет ячейечных величин – плотность, давление, энергия (уравнение энергии). При этом используется информация о новом положении узлов рассматриваемых ячеек.

В режиме мелкозернистого распараллеливания схема алгоритмов, с учетом обменов для мелкозернистого распараллеливания, следующая:

1. В начале счетного шага на каждой параллельной границе запускается асинхронный прием данных для присоединенных элементов сетки (с процессов рассчитывающих соответствующие соседние параобласти).
2. Производится расчет узловых величин, состоящий из следующих этапов:
 - 2.1. Проводится расчет величин для всех приграничных и параграничных узлов области.
 - 2.2. Запускается асинхронная отправка величин приграничных узлов параобласти. Данные отправляются соседним параобластям (по одному обмену для каждой параграницы).
 - 2.3. Проводится расчет величин для внутренних узлов.
 - 2.4. Завершается асинхронный прием величин для присоединенных узлов.
3. Далее проводится расчет блока ячейечных величин:

- 3.1. Проводится расчет величин для всех приграничных ячеек области.
- 3.2. Запускается асинхронная отправка величин приграничных ячеек параобласти. Данные отправляются соседним параобластям (по одному обмену для каждой параграницы).
- 3.3. Проводится расчет величин для внутренних ячеек.
- 3.4. Завершается асинхронный прием величин для присоединенных ячеек.

Для непосредственной передачи данных используется предварительная упаковка данных в буферные массивы, а после прием распаковка из буферных массивов.

Необходимо отметить, что в данной схеме не учитываются алгоритмы поддержания счетной сетки, используемые в методике ТИМ-2D [1]. Для их использования в режиме мелкозернистого распараллеливания, необходимо провести в дальнейшем доработку, как самих алгоритмов, так и данной схемы распараллеливания.

5. Взаимодействие мелкозернистого распараллеливания с распараллеливанием по областям

Один из вопросов, возникающих при реализации мелкозернистого распараллеливания это сочетание этих алгоритмов с алгоритмами распараллеливания по областям, реализованные в методике ТИМ-2D ранее [2].

5.1 Согласование нумерации параобластей с матобластями

При этом один из важных вопросов – способ нумерации параобластей и согласование этой нумерации с нумерацией математических областей. Для решения этого вопроса используется следующая схема.

Параобласти в счетных структурах нумеруются в списке математических областей. При этом параобласти получают номера после математических областей. Математическая область, из которой были порождены параобласти, из счета исключается. При этом параобласти нумеруются в соответствии с возрастанием номеров родительских математических областей.

Рассмотрим нумерацию параобластей на простом примере: в задаче три математических области, из первой области формируется 2 параобласти, из второй области параобласти не формируются совсем (для этой области сформирована декомпозиция из одного компакта), а из третьей формируется 3 параобласти. В результате параобласти полученные из первой матобласти получают номера 4, 5, а параобласти из 3 матобласти номера 6, 7, 8. Общее количество областей в структуре данных устанавливается равным 8. Отметим, что исходные математические области 1 и 3 сохраняют за собой номера, но из счета исключаются. Математическая область 2 рассчитывается под своим собственным номером – параобласти для нее не формируются.

5.2 Расчет контактных границ

Другим важным вопросом при реализации мелкозернистого распараллеливания является расчет контактных границ между матобластями. Способ расчета контактных границ определяется типом их учета в алгоритмах декомпозиции. В настоящее время реализован алгоритм расчета с выделением всех приграничных ячеек в отдельный компакт.

При этом подходе вдоль всех границ области в отдельный компакт группируется несколько слоев ячеек. В результате в этой параобласти содержатся все граничные точки данной области и выполнение операций сшивания фрагментов границы не требуется, также не требуется выполнения дополнительных обменов. Таким образом, для данного компакта можно использовать полностью те же самые алгоритмы расчета контактного взаимодействия и схему обменов, что и для распараллеливания по математическим областям.

Недостатком этого подхода является то, что достаточно сложно сбалансировать нагрузку между сформированной таким образом приграничной параобластью и внутренними параобластями. В случае использования математических областей с различным количеством граничных

точек, эта проблема становится практически не разрешимой. По этим причинам данный подход может рассматриваться как первый вариант декомпозиции, предназначенный для расчета задач с небольшим количеством областей и параобластей.

6. Взаимодействие мелкозернистого распараллеливания с OpenMP распараллеливанием

Взаимодействие между мелкозернистым распараллеливанием и OpenMP распараллеливанием логически организуется достаточно просто, так как в этих режимах используются разные парадигмы. В OpenMP распараллеливаются итерации счетных циклов, такие как циклы по ячейкам или по узлам. Поскольку в параобласти вводится локальная нумерация, то для OpenMP распараллеливания параобласть идентична матобласти и для OpenMP никаких модификаций не требуется. Тем не менее, одно обстоятельство потребовало специальной доработки алгоритмов.

Для мелкозернистого распараллеливания необходимо выполнение обменов приграничными элементами сетки между соседними параобластями. При этом используются асинхронные обмены для того чтобы наложить вычисления и обмены. Для того чтобы как можно раньше начать отправку данных о приграничных элементах сетки при формировании параобластей они нумеруются раньше внутренних. В результате в цикле по ячейкам или узлам они обходятся раньше и можно начать их отправку соседней параобласти и совместить обмен данных с расчетом внутренних элементов. Вот тут и возникает конфликт с OpenMP. Дело в том, что при распараллеливании итераций счетных циклов нарушается последовательность обработки этих самых итераций, то есть, например, 101 итерация может быть выполнена раньше 100. Нарушение последовательности итераций происходит из-за их распределения между нитями OpenMP. Упорядочить выполнение фрагментов счетного цикла можно при помощи флага ORDERED для директивы `!$OMP DO`, однако, опыт ее использования показал, что в этом случае очень сильно снижается эффективность OpenMP распараллеливания. Ввести некоторую закономерность в последовательность выполнения итераций можно при помощи флага SCHEDULE (также для директивы `!$OMP DO`). Однако это, во-первых, полностью проблему не решает, во-вторых, уменьшает степень наложения между расчетом и обменами, в-третьих, может снижать эффективность OpenMP распараллеливания.

Выходом из положения стало разделение счетных циклов на два. Первый цикл выполняется по граничным и приграничным элементам сетки. Второй цикл по внутренним элементам. При этом оба цикла заключены в одну параллельную область OpenMP. Между циклами выполняются обмены. При этом обмены выполняются внутри параллельной области OpenMP.

7. Исследование эффективности распараллеливания

В качестве характеристик эффективности распараллеливания использовались следующие функции: $S_p = \frac{t_1}{t_p}$ – ускорение счета; $E_p = \frac{t_1}{p \cdot t_p} \cdot 100\%$ – эффективность распараллеливания,

где t_1 – время расчёта на одном процессоре используемой параллельной машины (последовательный режим счета), t_p – время счёта на p процессорах.

Для исследования эффективности распараллеливания была использована задача об обжатии цилиндра равномерным давлением.

Начальная геометрия: Область представляет собой цилиндр. Размеры области 1×1 . Все границы, кроме верхней являются жесткими стенками.

Вещество: Идеальный газ с показателем адиабаты $\gamma = 3$.

Начальные условия: $\rho_0 = 1 \frac{\Gamma}{\text{см}^3}$ $E_0 = P_0 = 0$ начальная скорость:

$$v(x, y, 0) = 0; w(x, y, 0) = 0$$

$$(\vec{u} = \vec{u}(v, w)).$$

Граничные условия: На боковой поверхности цилиндра задается постоянное по времени давление $P_{sp} = 5$ ГПа.

Математическая постановка задачи: В качестве счётной сетки, выбрана четырёхугольная счётная сетка из 1 000 000 ячеек.

Декомпозиция выполнялась при помощи библиотеки ParMeTiS. Пример декомпозиции на 100 параобластей приведен на рисунке 3.

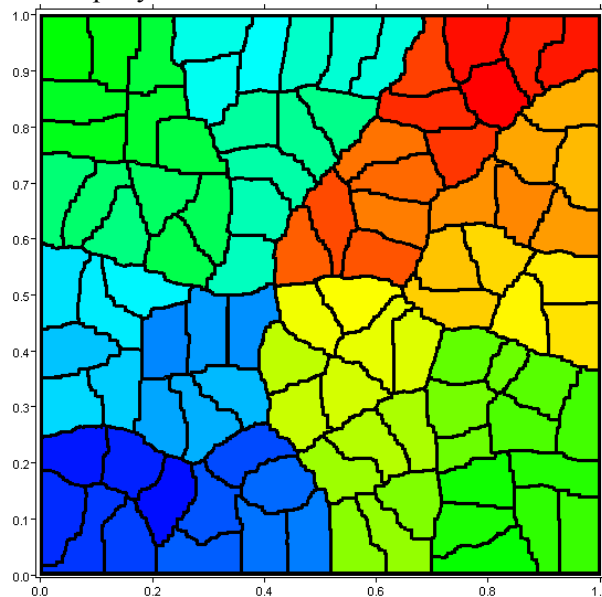


Рис. 3 – Декомпозиция на 100 процессов

Замеры времени выполнения в модели распределенной памяти представлены в таблице 1. Из данных, приведенных в таблице видно, что максимальное ускорение составляет чуть менее 30 раз. Эффективность оказывается не очень высокой, особенно при увеличении количества процессоров. Причем расчет на 100 процессорах оказывается даже медленнее чем на 64. Это обстоятельство объясняется тем, что вдоль границ строится отдельный компакт (параобласть) и при большом количестве процессоров эта параобласть оказывается в счете самой дорогой. Для улучшения эффективности при использовании большого количества процессоров необходимо реализовывать алгоритмы декомпозиции, допускающие фрагментирование контактных границ.

Таблица 1 – Режим счета в модели распределенной памяти (MPI)

Количество процессоров	1	8	16	32	50	64	100
Общее время счета, сек.	9573,7	1371,5	746,5	456,36	344,4	325,5	395,0
Ускорение, разы	1,00	6,98	12,83	20,98	27,80	29,42	24,24
Эффективность, %	100	87,26	80,19	65,56	55,6	45,96	24,24

На рисунке 4 приведена зависимость эффективности от количества процессоров. Из графика видно, что эффективность падает практически линейно при увеличении количества процессоров.

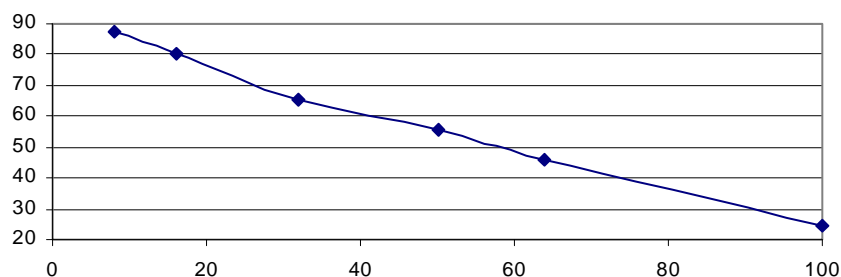


Рис. 4 – Зависимость эффективности от количества процессоров в MPI распараллеливании

В модели общей памяти был выполнен расчет на 8 процессорах, приведенный в таблице 2 в столбце с заголовком «8». В этом расчете ускорение составило 5,3 раза, а эффективность распараллеливания в модели общей памяти около 66%.

Таблица 2 – Режим расчета в смешанной модели памяти (MPI + OpenMP)

Кол-во процессоров	1	8	64	128	256	400	512	800
MPI × OpenMP	1 × 1	1 × 8	8 × 8	16 × 8	32 × 8	50 × 8	64 × 8	100 × 8
Время расчета, сек	9573,73	1799,5	382,77	255,4	296,64	283,43	294,63	345,12
Ускорение, разы	1	5,32	25,01	37,49	32,27	33,78	32,49	27,74
Эффективность, %	100	66,5	39,08	29,29	12,61	8,44	6,35	3,47

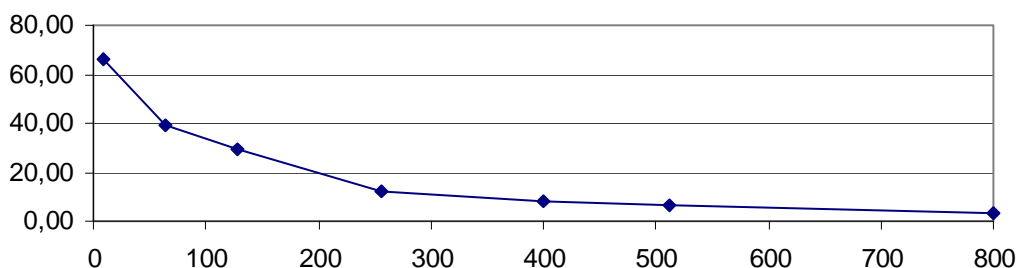


Рис. 5 – Зависимость эффективности от количества процессоров в смешанном распараллеливании

Расчеты в смешанной модели памяти проводились на различном количестве процессоров от 64 до 800. При этом в узле в модели общей памяти всегда использовалось 8 нитей OpenMP. Результаты замеров времени проведения этих расчетов приведены в таблице 2.

Как видно из приведенных результатов максимальное ускорение составляет 37,5 раз на 128 процессорах, когда область разбивается на 16 компактов.

С другой стороны, из таблицы видно, что замедление в счете возникает уже на 256 процессорах при количестве компактов 32. Это объясняется тем, что начиная с 32 компактов самым дорогой в расчете становится приграничная параобласть.

Таблица 3 – Режим счета в модели распределенной памяти (MPI)

Количество процессоров	64	72	80	88	96	100
Общее время счета, сек.	325,5	345,81	366,11	378,06	390	395
Ускорение, разы	29,41	27,69	26,15	25,32	24,55	24,24
Эффективность, %	45,96	38,45	32,69	28,78	25,57	24,24

Таблица 4 – Режим расчета в смешанной модели памяти (MPI + OpenMP)

Кол-во процессоров	64	72	80	88	96	104	112	120	128
MPI × OpenMP	8 × 8	9 × 8	10 × 8	11 × 8	12 × 8	13 × 8	14 × 8	15 × 8	16 × 8
Время счета, с.	382,8	354,2	326,2	302,2	290,7	274,5	271,0	260,8	255,4
Ускорение, р.	25,01	27,03	29,35	31,68	32,94	34,87	35,33	36,71	37,49
Эффективн., %	39,08	37,54	36,69	36,00	34,31	33,53	31,54	30,59	29,29

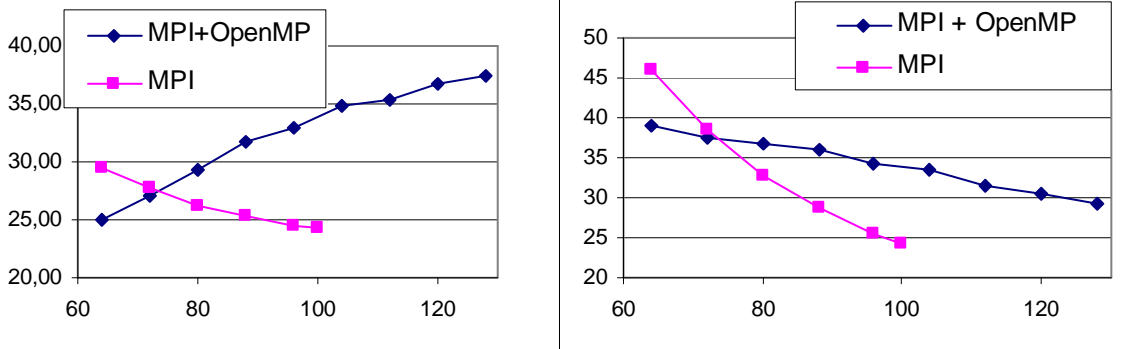


Рис. 6 – Сравнение ускорения (слева) и эффективности (справа) в MPI и MPI+OpenMP распараллеливании при количестве процессоров более 60

Из графика зависимости эффективности от количества процессоров, приведенном на рисунке 5, видно, что в режиме смешанного распараллеливания падение эффективности происходит не линейно в отличие от чистого MPI распараллеливания (смотри **Ошибка! Источник ссылки не найден.**). В связи с этим в районе 64 процессоров по эффективности MPI режим и смешанный практически сравниваются. Это обстоятельство было исследовано более подробно. Результаты замеров времени счета, ускорения и эффективности распараллеливания приведены в таблицах 3 – 4 и в графическом виде на рисунке 6. Из приведенных результатов видно, что при увеличении количества процессоров более 64 в режиме MPI распараллеливания происходит только замедление (смотри таблицу 3), в то время как ускорение в смешанном режиме счета продолжает расти (таблица 4). В результате на 72 процессорах режимы счета практически сравниваются между собой, а на 80 и более процессорах режим счета MPI + OpenMP начинает опережать чистый MPI. При этом необходимо отметить, что по общей скорости работы смешанный режим на 88 и более процессорах работает быстрее, чем MPI режим на любом количестве процессоров.

8. Тестовый расчет

По мимо тестовых расчетов приведенных в предыдущем разделе, проводился тестовый расчет задачи о полете несферической оболочки. Данный тест предназначен для проверки правильности работы алгоритмов контактного взаимодействия при двумерном движении системы.

Начальная геометрия задачи представлена на рисунке 7. Начальная форма оболочки $R(t=0, \theta) = 7 + 0.875 \cos^3 \theta - 0.525 \cos \nu$ см, толщина $\Delta R = 0.16$ см. Материал – свинец:

$\rho(0) = 11.4 \frac{\Gamma}{\text{см}^3}$. В начальный момент оболочка имеет скорость к центру:

$$v = 5.70686 \cdot 10^6 \frac{\text{см}}{\text{сек}}$$

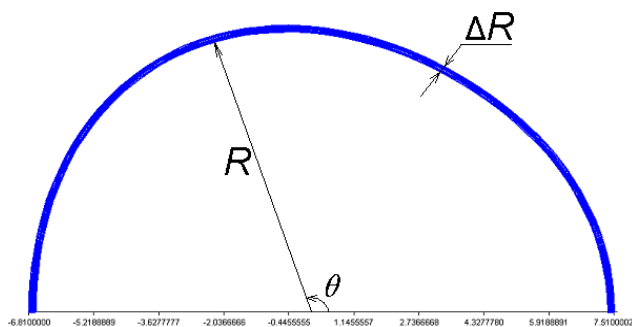


Рис. 7 – Начальная геометрия задачи о полете не сферической оболочки

Давление на наружной поверхности оболочки задается постоянным от времени $P_E = 0$, давление на внутренней поверхности зависит от времени – $P_i = 3.495 \cdot 10^6 \cdot (V(t))^{-5/3}$ см³, где $V(t)$ – объём внутренней части полости, меняющийся со временем на стадии полёта.

Сетка сформирована из четырехугольников с 4 связями в узле. Количество ячеек 16800 (шесть областей с количеством точек по радиусу и углу задана 20×140 , и одна область с количеством точек по радиусу и углу 120×140).

При проведении расчетов вдоль границ строился компакт шириной в 2 ячейки. В многообластном расчете каждая область разбивалась на 4 компакта. В однообластном на 6 компактов. Расчеты этой серии проводились с использованием смешанного распараллеливания, то есть дополнительно использовалось OpenMP распараллеливание. Соответственно многообластной расчет проводился на 184 процессорах (24 MPI процессов \times 8 OpenMP нитей), а однообластной на 48 процессорах (6 MPI процессов \times 8 OpenMP нитей).

На этом тесте проверялось работоспособность общей схемы трехуровневого распараллеливания. При этом в многообластном расчете варьировалось распределение управляющих функций для математических областей по разным процессам (при этом соответствующим образом изменялось и количество задействованных процессоров). Во всех параллельных расчетах получено полное совпадение результатов, как между собой, так и с последовательным расчетом.

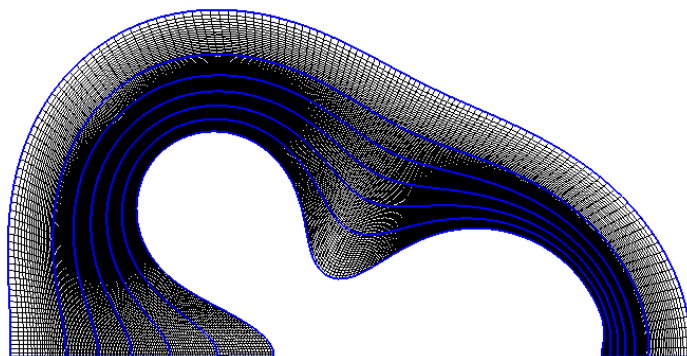


Рис. 8 – Состояние счетной сетки и положение границ для задачи о полете не сферической оболочки в многообластной постановке на момент времени $t = 0.11$ мкс с выделением контактных границ между областями

На рисунке 8 показано состояние сетки, и положение границ в многообластном расчете на момент времени 0.11 мкс. На рисунке 9 приведена заливка по номеру компакта, когда каждая область разбивалась на 4 компакта. Поскольку математические области имеют подобные сетки, то и декомпозиция для них получилась одинаковая (для MeTiS использовались параметры по умолчанию). Вид параобластей построенных из первой матобласти приведены на рисунке 10. Результаты однообластного расчета приведены на рисунке 11. Вид декомпозиции для однообластного расчета приведен на рисунке 12. На рисунке 13 приведено сравнение формы оболочки для однообластного расчета в последовательном и параллельном режимах.

Как видно из рисунков результаты расчетов хорошо согласуются между собой по форме внутренней оболочки как в многообластном, так и однообластном расчетах.

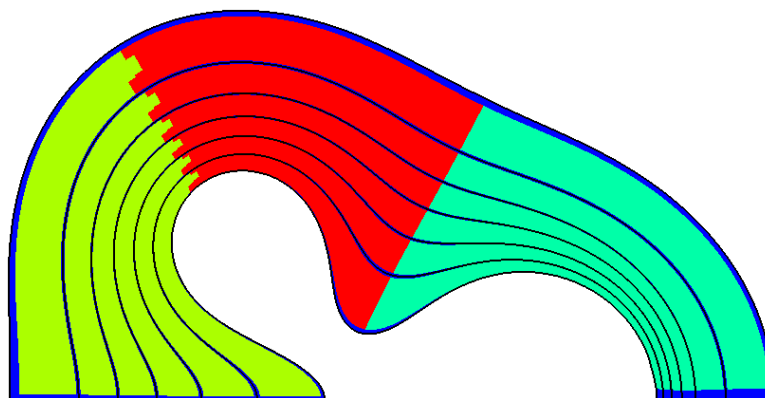


Рис. 9 – Декомпозиция задачи для задачи о полете не сферической оболочки на момент времени $t = 0.11$ мкс

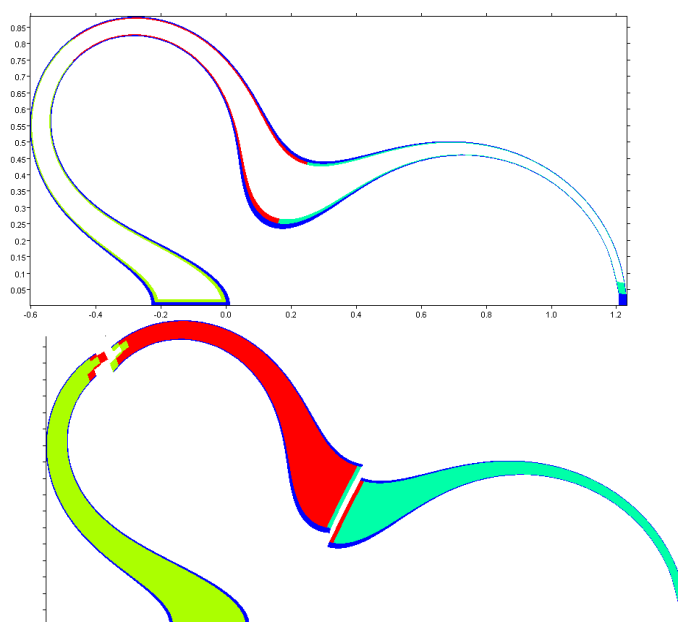


Рис. 10 – Вид параобластей, построенных из первой матобласти в шестиобластной задаче о полете не сферической оболочки с заливкой по номеру компакта

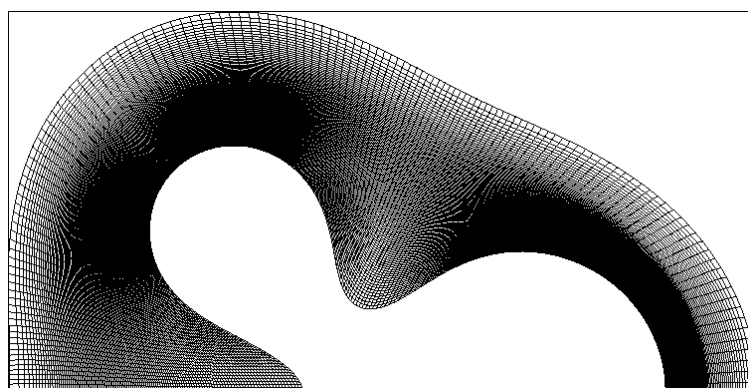


Рис. 11 – Состояние счетной сетки задачи о полете не сферической оболочки в однообластной постановке на момент времени $t = 0.11$ мкс

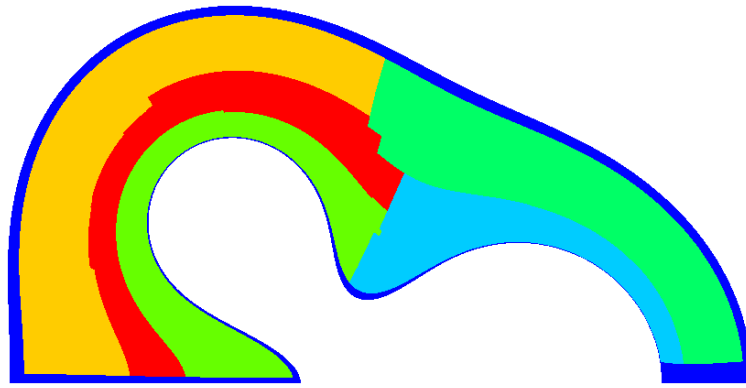


Рис. 12 – Декомпозиция данных для задачи о полете не сферической оболочки в однообластной постановке на момент времени $t = 0.11$ мкс

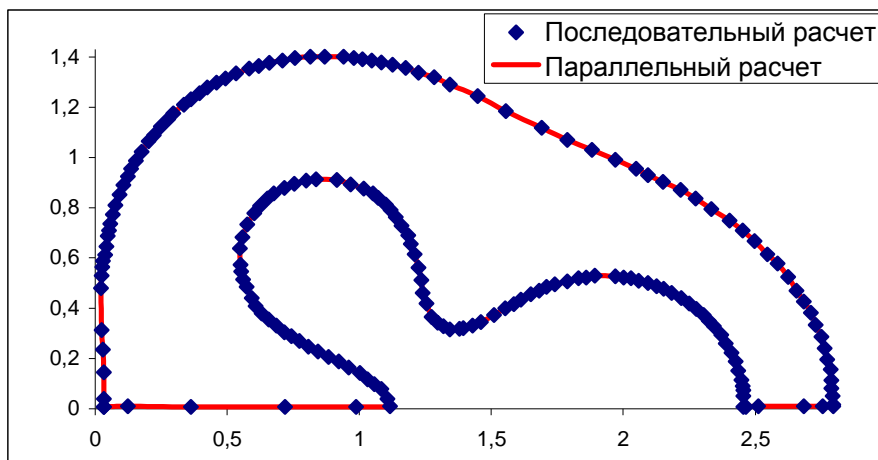


Рис. 13 – Сравнение формы оболочки в последовательном и параллельном расчетах

9. Заключение

В докладе описываются алгоритмы мелкозернистого распараллеливания в модели распределенной памяти с использованием интерфейса обмена сообщениями MPI в методике ТИМ-2D для расчета задач газовой динамики на нерегулярных многоугольных сетках. Мелкозернистое распараллеливание осуществляется внутри математической области на основе разбиения области на компакты.

Приводится ряд тестов, демонстрирующих правильность работы программ и алгоритмов. Для тестов использовались четырехугольные сетки с 4 связями в узле и многоугольные сетки с 3 связями в узле. На одной из тестовых задач приводятся замеры эффективности распараллеливания в различных режимах счета (MPI, OpenMP, MPI+OpenMP). При этом максимальное ускорение в 37,5 раз достигнуто в режиме смешанного распараллеливания MPI + OpenMP на 128 процессорах (эффективность около 30%). Приводится сравнение режимов между собой – на количестве процессоров более 64, смешанный режим распараллеливания оказывается более эффективным, чем MPI режим.

Для дальнейшего развития необходимо реализовать алгоритмы с разбиением границ по нескольким компактам и реализовывать алгоритмы поддержания счетной сетки в режиме мелкозернистого распараллеливания.

Литература

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): Труды международной научной конференции (Санкт-Петербург, 28 января - 1 февраля 2008 г.). Челябинск: Изд. ЮУрГУ, 2008. С. 69 - 79.
3. Воропинов А.А., Новиков И.Г., Соболев И.В., Соколов С.С. Распараллеливание методики "ТИМ" в модели общей памяти с использованием интерфейса OpenMP // Вычислительные методы и программирование. 2007. Том 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. ParMETIS - Parallel Graph Partitioning and Fill-reducing Matrix Ordering [Electronic resource]. Mode of access: [<http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview> 13.09.2007].