

АЛГОРИТМЫ ДИНАМИЧЕСКОЙ БАЛАНСИРОВКИ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ ДЛЯ МНОГОМЕРНЫХ РАСЧЕТОВ В МЕТОДИКЕ ТИМ

Т. Н. Половникова, А. А. Воропинов

Российский Федеральный Ядерный Центр –
Всероссийский НИИ экспериментальной физики, г. Саров

Методика ТИМ [1–2] предназначена для решения многомерных нестационарных задач механики сплошных сред на неструктурированных лагранжевых сетках произвольного вида.

По методике ТИМ можно рассчитывать широкий набор процессов: газовую динамику и упругопластичность, детонацию, теплопроводность, магнитную гидродинамику с учетом диффузии магнитного поля, многопоточковую газодинамику и другие процессы.

Ячейками сетки являются в двумерном случае произвольные несамопересекающиеся многоугольники, в трехмерном случае – произвольные несамопересекающиеся многогранники, которые не обязательно выпуклые. Грани состоят из произвольного количества узлов, в узлах сетки сходится произвольное количество ребер.

Для повышения точности проводимых расчетов необходимо производить численное моделирование на сетках с большим количеством ячеек. Проведение таких расчетов требует значительного календарного времени. Один из путей сокращения сроков – проведение расчетов в параллельном режиме счета.

Для методики ТИМ используется метод трехуровневого распараллеливания [3–4]. На первом (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 также в модели распределенной памяти с использованием интерфейса передачи сообщений MPI (параобласть – фрагмент счетной области, рассчитывающийся на одном узле). На третьем (нижнем) уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Эти подходы могут использоваться, как совместно в различных сочетаниях, так и раздельно при расчете одной задачи.

Эффективное выполнение счетных программ на многопроцессорных машинах с распределенной памятью требует декомпозиции данных по процессам таким образом, чтобы распределение вычислительной нагрузки было равномерным, а количество межпроцессных обменов минимальным. Получение качественной декомпозиции является актуальной проблемой, особенно для методик, использующих неструктурированные сетки.

В методике ТИМ задача выполнения декомпозиции сводится к решению задачи о разрезании графа на подграфы. Вершина графа соответствует ячейке сетки, а ребро – соседству между ячейками. Для решения этой задачи реализовано подключение библиотек MeTiS, ParMeTiS и SCOTCH [5]. В качестве исходных данных используется граф, отображающий структуру сетки. Возможно использование различных весов вершин и ребер графа.

В процессе численного моделирования объем вычислений для конкретной ячейки может изменяться (за счет алгоритмов поддержания качества счетной сетки, использования различных уравнений состояний, кинетических моделей: кинетика детонации, кинетика разрушения и т. д.). На обсчет такой ячейки (или всей параобласти) может тратиться больше времени, значит, может появиться несбалансированность, а распределение вычислительной нагрузки по процессам должно быть оптимальным.

Также в процессе численного моделирования по методике ТИМ в задаче могут образовываться струйные и вихревые течения, локально могут утончаться и перебиваться параобласти, параобласти могут претерпевать изломы вблизи параллельной границы, которые могут приводить к увеличению межпроцессных обменов.

Опыт проведения расчетов показал, что основная проблема заключается не в получении изначально плохой декомпозиции, а в постепенном ухудшении качества декомпозиции в процессе проведения расчета из-за работы различных алгоритмов. А также может требоваться разное время для вычисления различных фрагментов задачи. Таким образом, видно, что необходима балансировка вычислительной нагрузки.

Алгоритмы балансировки для сеточных методик можно классифицировать следующим образом:

1. Статическая балансировка. Подход статической балансировки используется для больших задач, которые рассчитываются в несколько заходов. В этом случае в процессе проведения расчета накапливается информация о времени, необходимом на расчет различных ячеек сетки. Данная информация сохраняется вместе с контрольной точкой и используется в следующем запуске для определения весовой функции.

Преимуществом данного подхода является отсутствие необходимости в разработке алгоритмов динамической балансировки, достаточно, чтобы алгоритмы декомпозиции учитывали вес, а необходимая информация сохранялась и считывалась из контрольной точки.

Главным недостатком такого подхода является то, что он плохо подходит для задач, рассчитываемых за небольшое количество запусков. Для таких задач приходится делать специальные короткие запуски для накопления временных засечек. Также существенным недостатком является то, что данный подход не применим для задач, в которых вычислительная нагрузка меняется значительно и довольно быстро в пределах одного запуска. Все эти недостатки ограничивают область применения данного подхода.

2. Квази-динамическая балансировка. Квази-динамическая балансировка во многом является развитием подхода статической балансировки. В этом случае в методике предусматривается механизм оценки текущей декомпозиции с точки зрения сбалансированности. Если текущая декомпозиция не сбалансирована, то производится сохранение контрольной точки и выполнение новой декомпозиции с полной переинициализацией задачи (по этой причине такую операцию также называют передекомпозицией).

Такой подход не требует выполнения остановок и, благодаря этому, решает ряд проблем, присущих статической балансировке. Его можно применять для задач с небольшим количеством запусков, также улучшается ситуация для задач, где существенно меняется вычислительная нагрузка, передекомпозиция при этом может быть сделана несколько раз за один запуск. Важным преимуществом данного подхода является возможность использования существующих алгоритмов сохранения контрольной точки, декомпозиции и инициализации параллельного счета. Также получение новой декомпозиции позволяет рассматривать всю задачу сразу, уходя от локальных операций.

Недостатком данного подхода является необходимость сохранения и чтения контрольной точки, а также построения новой декомпозиции. Эти операции для больших задач часто являются достаточно дорогими, и могут занимать существенное время. По этой причине операция передекомпозиции должна выполняться достаточно редко, чтобы выигрыш в скорости счета превысил накладные расходы. В результате данный подход плохо применим для задач, где вычислительная нагрузка меняется достаточно интенсивно на длительном протяжении расчета.

3. Динамическая балансировка. Динамическая балансировка основана на передаче ячеек между параобластями внутри математической области без выполнения операций сохранения контрольных точек и вообще каких-либо внешних данных. С одной стороны, динамическая балансировка лишена недостатков квази-динамической балансировки, так как она не требует выполнения дорогостоящей операции сохранения и чтения контрольной точки и полной передекомпозиции задачи.

Однако динамическая балансировка не лишена недостатков, ограничивающих ее применение. В частности, довольно часто при разбалансировке наиболее и наименее загруженные параобласти принадлежат различным математическим областям. По смыслу динамической балансировки передача ячеек в этом случае напрямую невозможна, так как между параобластями отсутствует коммуникация. Т.е. возможны ситуации, когда динамическая балансировка не может напрямую решить проблему разбалансировки или потребуются множество операций передачи ячеек. Например, в слу-

чае отсутствия прямого взаимодействия между перегруженными и недогруженными по вычислительной нагрузке параобластями одной математической области может использоваться передача ячеек через параобласти с удовлетворительной загруженностью.

Другой проблемой для динамической балансировки может являться ситуация, когда вычислительная нагрузка меняется очень резко. Такая ситуация возможна, например, когда на старте задачи первоначальная декомпозиция делается без использования весовых функций, а затем в процессе счета определяется время на расчет ячеек. В результате может оказаться, что первоначальная декомпозиция крайне неудачна, вплоть до того, что некоторые процессы должны сменить набор ячеек полностью. В этом случае объем передаваемых данных между параобластями внутри математической области чрезвычайно большой, и оказывается, что гораздо быстрее выполнить передекомпозицию.

Данная работа посвящена алгоритмам динамической балансировки.

Результаты работы алгоритмов динамической балансировки за несколько шагов представлены на рис. 1, 2.

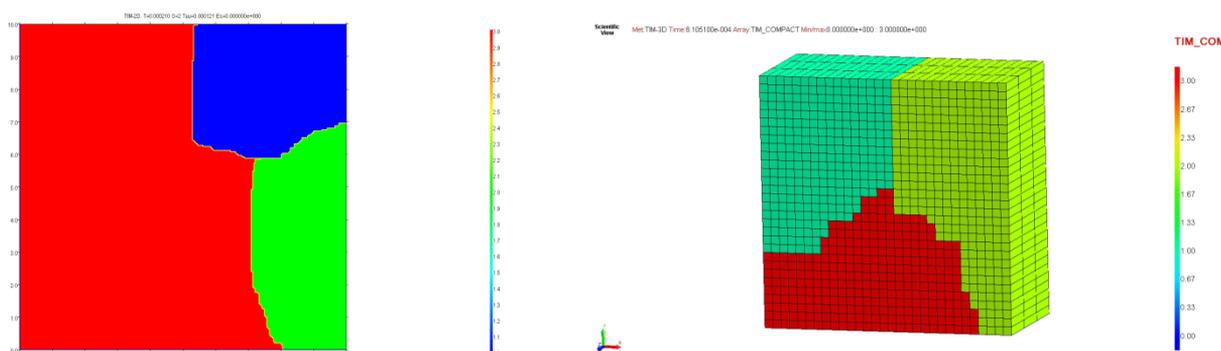


Рис. 1. Начальные декомпозиции

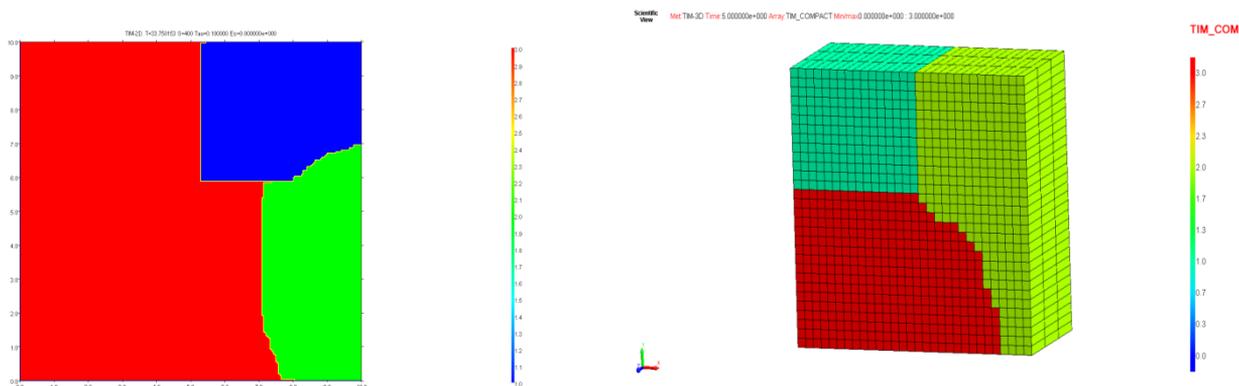


Рис. 2. Полученные декомпозиции через несколько шагов в результате работы алгоритмов динамической балансировки

Требования, которые накладываются на балансировку вычислительной нагрузки алгоритмы распараллеливания:

- неизменность графа связей между параобластями (не должно возникать нового соседства между параобластями – так как не должны возникать новые операции обменов между ними, память под которые не выделена);
- минимальность линии раздела между параобластями для уменьшения объема передаваемых данных;
- отсутствие ситуаций, близких к перебитию параобласти для случаев с наложением в несколько слоев (в результате может также меняться граф связей между параобластями).

Таким образом, необходимы критерии, улучшающие качество декомпозиции при динамической балансировке.

Набор критериев для проведения динамической балансировки вычислительной нагрузки состоит из двух частей:

1. Критерии анализа необходимости выполнения операции улучшения качества декомпозиции:

- разбалансированность по вычислительной нагрузке;
- параобласть близка к перебитию (по типу «песочные часы»).

2. Критерии выбора ячеек для переброски:

- ячейка из заданного списка;
- неоптимальное отношение по количеству внутренних и граничных ребер (2D) и внутренних и граничных граней (3D);
- неоптимальное отношение по длине внутренних и граничных ребер (2D) и по площади внутренних и граничных граней (3D);
- ячейка максимального веса;
- ряд критериев, вырабатываемых программами поддержания качества сетки.

Понятно, что данные критерии связаны между собой. Отметим, что разделение критериев на две части упрощает реализацию алгоритмов и согласование работы критериев между собой.

Пользователь может управлять, какими критериями он хочет воспользоваться для улучшения качества декомпозиции и динамической балансировки.

Работа алгоритмов переброски ячеек для динамической балансировки тестировалась на модельной задаче о плоской волне на трех вычислительных узлах. Для тестирования была использована четырехугольная сетка. Такая сетка позволяет не использовать алгоритмы поддержания качества сетки [6] и благодаря этому общий объем вычислений не зависит от вида декомпозиции. Это позволяет корректно оценить влияние алгоритмов переброски на общее время расчета, если принудительно изначально была задана несбалансированная декомпозиция.

При разбалансировке больше 5 % появляется необходимость в использовании алгоритмов динамической балансировки. Тесты прогонялись с использованием двух критериев: критерия разбалансировки вычислительной нагрузки и критерия неоптимального отношения внутренних и внешних ребер по их количеству. После отработки критерия разбалансировки получаем пары номеров параобластей, между которыми необходимо выполнить переброску ячеек.

На рис. 3 представлен фрагмент четырехугольной сетки с разбалансировкой 97 %. Расчет 100 шагов с такой декомпозицией занял 101,4 с.

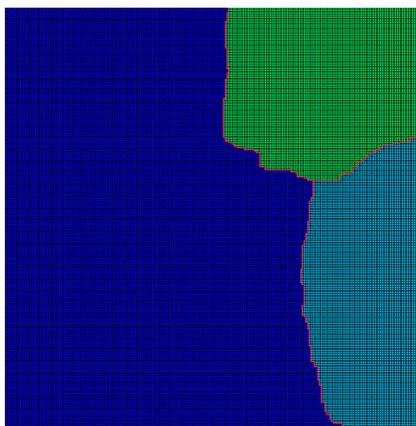


Рис. 3. Фрагмент четырехугольной сетки с разбалансировкой 97 %

На рис. 4 разбалансировка по процессам составляет примерно 5 %. Расчет 100 шагов с такой декомпозицией занял 90,1 с.

Таким образом, для получения разбалансировки менее 5 % необходимо было перебросить 20 % ячеек между параобластями, что привело к ускорению счета на 11 %.

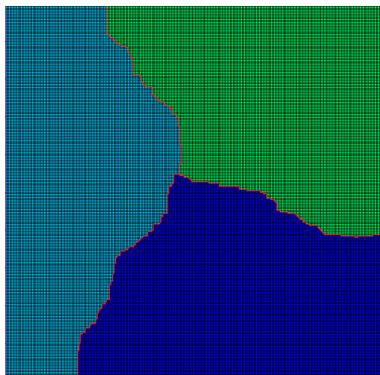


Рис. 4. Фрагмент четырехугольной сетки с разбалансировкой примерно 5 %

На рис. 5 представлено сравнение одного и того же фрагмента четырехугольной сетки до переброски ячеек и после.

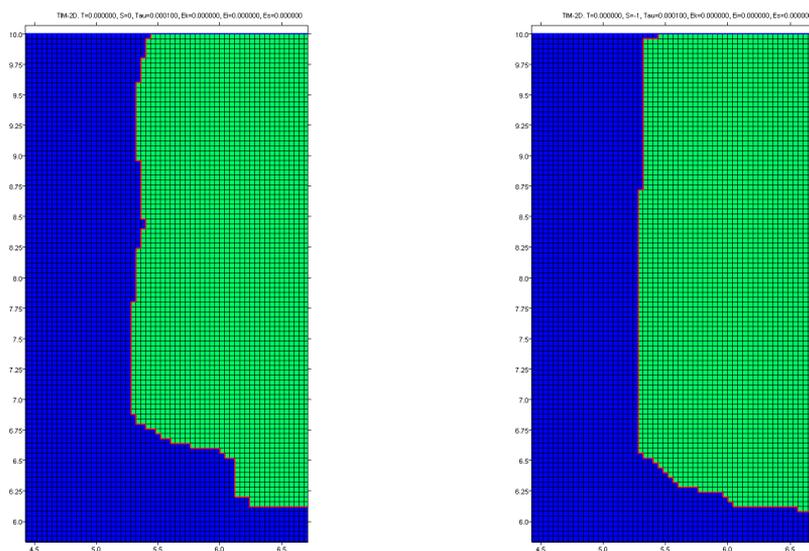


Рис. 5. Фрагмент четырехугольной сетки до переброски ячеек и после

Таким образом, в результате работы алгоритмов получили более ровную границу, т. е. уменьшили длину границы, что привело к уменьшению количества обменов между процессами.

Были проведены тестовые расчеты трехмерной задачи, которая состоит из 5 математических областей, количество ячеек и параобластей в каждой из которых представлено в табл. 1, сетка шестигранная, декомпозиция выполнена с помощью библиотеки MeTiS. Задача считалась на 9 вычислительных узлах по 16 процессорных ядер в каждом.

Таблица 1

Количество ячеек по математическим областям

Номер математической области	Количество ячеек	Количество параобластей	
		Тест 1	Тест 2
1	132505	3	4
2	61392	3	3
3	157560	3	3
4	4648	Параобласти не создавались	
5	533830	7	7
Общее количество ячеек	889935		

Начальная декомпозиция фрагмента трехмерной задачи с заливкой по номеру параобласти представлен на рис. 6.

Scientific View Met:TIM-3D Time:4.009059e-001 Array:Matter Min/max:1.000000e+000 : 5.000000e+000

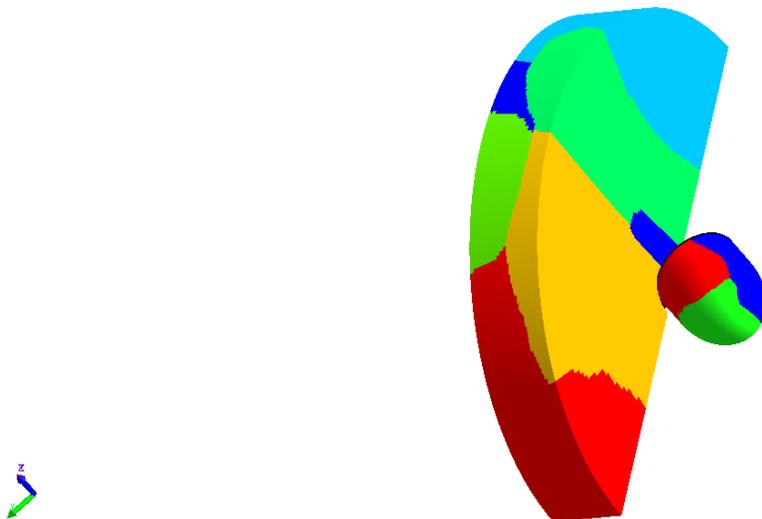


Рис. 6. Начальная декомпозиция задачи

В данном случае использовались критерии разбалансировки вычислительной нагрузки и критерий неоптимального отношения внутренних и внешних граней по их количеству.

В расчетах с включенными алгоритмами динамической балансировки варьировались:

- коэффициент разбалансировки: 1 %, 3 %, 5 % и 10 %;
- максимальное количество перебрасываемых ячеек за шаг: 20, 50 и 100.

Реберный коэффициент равен 0,9 (реберный коэффициент определяет отношение количества разрезанных внешних и внутренних ребер графа в 2D и граней в 3D).

Задача считалась до 200 шагов.

В сводной табл. 2 указаны ускорения по времени при расчетах с алгоритмами динамической балансировки:

Таблица 2

Ускорение по времени в расчетах с алгоритмами балансировки относительно расчетов без них

Максимальное число перебрасываемых ячеек за шаг	Реберный коэффициент	Коэффициент разбалансировки, %	Ускорение, %	
			Тест 1	Тест 2
20	0,9	1	1	-1,7
		3	3	0,6
		5	2,6	-2
		10	2,1	-4
50	0,9	1	3	1
		3	3,7	2,3
		5	-1	-0,6
		10	2,1	-3
100	0,9	1	2,1	-3
		3	2,6	1,7
		5	2,1	2,3
		10	2,3	-2

Алгоритмы динамической балансировки дают дополнительную вычислительную нагрузку. Поэтому при хорошей сбалансированности вычислительной нагрузки между узлами в задаче данные алгоритмы дают замедление, т. к. работают только критерии анализаторы необходимости переброски.

Таким образом, исходя из полученных результатов видно, что для данной задачи оптимальными оказались следующие параметры: коэффициент разбалансировки 3 %, реберный коэффициент 0,9 и количество максимально перебрасываемых ячеек за шаг равно 50.

Работа алгоритмов, реализующих описанные критерии, была продемонстрирована на ряде методических расчетов. Применение алгоритмов динамической балансировки вычислительной нагрузки позволяет ускорить счет и эффективно загрузить процессорное поле, выделенное на задачу. Таким образом, на двумерной модельной задаче без корректировки сетки в результате отработки критериев балансировки было получено ускорение на 11 %, на трехмерной модельной задаче с использованием всего аппарата поддержания качества сетки в многообластной постановке ускорение на 3,7 %.

В дальнейшем планируется провести исследование на серии сложных расчетов для выбора оптимального набора параметров, подходящего для решения широкого класса прикладных задач.

Литература

1. Соколов С. С., Панов А. И., Воропинов А. А. и др. Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 3. С. 37–52.

2. 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 // Там же. 2006. Вып. 4. С. 29–43.

3. Воропинов А. А., Новиков И. Г., Соколов С. С. Распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике ТИМ-2D // Труды РФЯЦ-ВНИИЭФ. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2013. С. 84–99.

4. Воропинов А. А., Новиков И. Г., Соколов С. С. Методы мелкозернистого распараллеливания в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2012. Вып. 3. С. 24–33.

5. Половникова Т. Н., Воропинов А. А. Опыт использования библиотек SCOTCH и MeTiS для декомпозиции неструктурированных сеток в методике ТИМ // XII Международный семинар «Супервычисления и математическое моделирование»: сб. докл. Саров, 2011.

6. Новиков И. Г., Панов А. И., Соколов С. С. Способ коррекции нерегулярной лагранжевой сетки методом наложения дифференцируемых связей // Журнал вычисл. мат. и мат. физ. 2005. Т. 45, № 8. С. 1487–1500.