

# Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества

А.А. Воропинов

Методика ТИМ-2D предназначена для решения задач механики сплошной среды на нерегулярных многоугольных сетках произвольной структуры. В докладе описываются несколько алгоритмов декомпозиции данных для мелкозернистого распараллеливания методики ТИМ-2D, учитывающие разбиение на математические области. Вводится ряд критериев для оценки качества декомпозиции, для использования при мелкозернистом распараллеливании методики ТИМ-2D. Необходимость применения дополнительных критериев к декомпозиции обуславливается счетными алгоритмами, используемыми в методике ТИМ-2D для поддержания качественной расчетной сетки (коррекция сетки и локальные перестройки) и накладываемые этими алгоритмами ограничения на декомпозицию. Представленные критерии и алгоритмы проверяются на ряде тестовых декомпозиций.

## 1. Введение

Методика ТИМ-2D [1] предназначена для расчета нестационарных задач механики сплошной среды на нерегулярных лагранжевых сетках произвольной структуры. Ячейками сетки являются произвольные многоугольники, в узлах сходится произвольное количество ребер.

Для повышения точности проводимых расчетов и более адекватного моделирования сложных задач необходимо производить численное моделирование на сетках с большим количеством ячеек (счетных точек). Проведение расчетов с большим количеством счетных точек требует значительного календарного времени. Один из путей сокращения сроков – проведение расчетов в параллельном режиме счета.

Для методики ТИМ-2D реализуется трехуровневое распараллеливание [2]. 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 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 [3] с использованием интерфейса OpenMP. Эти подходы могут использовать как вместе в различных сочетаниях, так и отдельно при расчете одной задачи. Разнообразие методов и их сочетаний определяет сложность задачи выполнения декомпозиции таким образом, чтобы она подходила для использования в методике ТИМ-2D.

Эффективное исполнение счетных программ методик на параллельных машинах с распределенной памятью требует декомпозиции данных по процессам таким образом, чтобы распределение вычислительной нагрузки по процессам было равномерным, а количество межпроцессорных обменов минимальным.

Для методик, использующих регулярные сетки задача декомпозиции данных по процессам, как правило, решается с помощью разбиения сетки по столбцам или строкам (в трехмерном случае, также может использоваться и разбиение по листам), или с помощью матричного метода. Разбиение сетки по строкам или столбцам осуществляется путем «разрезания» сетки вдоль выделенного сеточного направления. В матричном методе «разрезание» осуществляется сразу по двум сеточным направлениям (в трехмерном случае может осуществляться и по трем сеточным направлениям). Эти методы хорошо себя зарекомендовали, так как достаточно просты для реализации, и в случае, когда расчет различных точек в задаче требует примерно одинакового количества вычислений, позволяют получить хорошую сбалансированность вычислительной нагрузки между процессами. Однако в случае если в разных фрагментах задачи объем вычислений для расчета точек существенно различается, то добиться хорошей сбалансированности загрузки такими методами достаточно сложно.

Использование декомпозиции по строкам, столбцам или матричной декомпозиции для методик использующие нерегулярные сетки не возможно, поскольку в нерегулярной сетке структурно не имеется выделенных сеточных направлений. Поэтому для методик, основанных на нерегулярных сетках, используются геометрические методы декомпозиции данных [4, 5, 6, 7, 8] и методы декомпозиции, основанные на разбиении графов [9, 10, 11], отображающих структуру счетной сетки. Необходимо отметить, что методы, которые применимы для декомпозиции данных для методик на нерегулярных сетках, могут применяться и для методик на регулярных сетках.

Геометрические методы выполняют разбиение сетки, основываясь на координатной информации об ее элементах. Эти методы не всегда позволяют равномерно распределить нагрузку между процессами. Некоторые геометрические методы позволяют уменьшить проблему с разбалансированностью как, например, метод полос и клеток, используемый в методике «Медуза-П» [8].

Недостатками геометрических методов являются: ограничения на форму геометрии; сложности при организации многообластного счета, так как трудно добиться особой обработки точек вдоль границ; также возникают проблемы в случае, если ячейки сетки имеют существенно различный размер в разных частях области. Также к недостаткам этих методов относится то, что они не учитывают информацию о соседстве элементов сетки, поэтому геометрические методы не могут явно привести к минимизации передаваемых между процессами данных. Все это приводит к тому, что использование геометрических методов для декомпозиции данных на достаточно большое количество процессов затруднительно.

К геометрическим методам относятся: метод покоординатного разбиения [12, 13], рекурсивный инерционный метод деления пополам [14], метод заполняющих кривых [7, 15], метод полос и клеток, используемый в методике Медуза-П [8] и другие.

Задача разбиения графа на подграфы оптимальным образом относится к классу *NP*-полных [16]. Поэтому для ее решения как правило используют приближенные алгоритмы двух классов: геометрические и эвристические методы. Эвристические методы обеспечивают более сбалансированное разбиение и передаваемых между процессами данных обычно меньше, чем при применении геометрических методов.

Существует большое количество эвристических методов: уровневое ячеечное разбиение [17], алгоритмы Kernighan-Lin [18] и Fiduccia-Mattheyses [19], алгоритм спектральной бисекции [20, 21], многоуровневые методы разбиения графов [10, 11, 22, 23, 24, 25] и другие.

Основой многоуровневых методов разбиения графов является следующая последовательность действий:

- огрубление графа: построение последовательности уменьшающихся в размере вложенных графов путем многократного стягивания ребер;
- начальная декомпозиция: вершины огрубленных графов распределяются на заданное число подграфов;
- восстановление графа и локальное уточнение декомпозиции: вершины всех графов сформированной последовательности, начиная с имеющих наименьший размер, распределяются по подграфам.

Многоуровневые алгоритмы эффективны по двум причинам: во-первых, алгоритм позволяет скрыть большое количество ребер в огрубленном графе, что уменьшает количество передаваемых между процессами данных; во-вторых, эти алгоритмы работают эффективнее, так как перемещение одной вершины в огрубленном графе эквивалентно перемещению большого числа вершин исходного графа.

Недостатком эвристических алгоритмов является то, что они не учитывают геометрические особенности сетки. Это может приводить к тому, что геометрически граница между подсетками может иметь очень сложный вид, что может осложнять работу алгоритмов работающих с сеткой (в методике ТИМ-2D – дифференцируемые связи и локальные перестройки).

Существует несколько пакетов программ для выполнения декомпозиции различными методами, среди них: MeTiS [26], ParMeTiS [27], Chaco [28], JOSTLE [29], PARTY [30], SCOTCH [31], S-HARP [32], ZOLTAN [33] и другие. Некоторые из этих пакетов доступны для свободного скачивания в сети Интернет, другие являются внутренней разработкой различных институтов.

Одними из известных и общедоступных являются библиотеки семейства MeTiS. Одна из них библиотека параллельных программ ParMeTiS использующая стандарт MPI. В библиотеке реализованы алгоритмы декомпозиции и передекомпозиции неструктурированных графов. Пакет ParMeTiS специально предназначен для задач параллельного численного моделирования, использующих большие неструктурированные сетки. Реализованные в библиотеке алгоритмы дают качественную декомпозицию по количеству передаваемых между процессами данных, и возможность балансировки загрузки процессов. Декомпозиция может выполняться как в последовательном, так и в параллельном режиме счета. В настоящее время пакет ParMeTiS используется в качестве базового при декомпозиции данных для мелкозернистого распараллеливания в методике ТИМ-2D.

Необходимо отметить, что любой метод декомпозиции можно рассматривать как разбиение графа, отображающего структуру сетки, на подграфы. Поэтому рассмотрение декомпозиции как разбиения графа является наиболее универсальным и позволяет единым образом описывать любой из способов декомпозиции (как геометрические, так и эвристические).

Общим недостатком всех приведенных методов декомпозиции является то, что они учитывают ограниченное число критериев. В тоже время, методика может накладывать дополнительные ограничения на получаемые подграфы (точнее на соответствующие им параобласти). В результате разбиение на подграфы, полученное каким-либо методом, может оказаться неудачным для использования в конкретной методике.

Так в методике ТИМ-2D для поддержания качественной расчетной сетки используются два базовых метода. Первый – коррекция лагранжевой сетки путем наложения дифференцируемых связей [34], выполняемая без изменения структуры сетки. Второй – локальные перестройки сетки [35], при котором меняется как структура сетки, так и количество элементов сетки (ячеек, ребер и узлов). Использование этих алгоритмов приводит к наложению ряда дополнительных требований на декомпозицию. В случае если эти требования не учитывать, то сетка может обрабатываться не корректно и как следствие качество сетки может значительно ухудшиться. Такими ограничениями являются, например, минимизация количества особых точек, в которых сходится более двух компактов, «оторванные ячейки» и ряд других.

## **2. Декомпозиция данных в методике ТИМ-2D**

В методике ТИМ-2D используется трехуровневое распараллеливание в модели смешанной памяти. Эти три уровня могут использоваться по отдельности, попарно в различных сочетаниях и все вместе, таким образом, возможно 7 различных режимов распараллеливания. Конкретный режим определяется параметрами вычислительной системы, используемыми приближениями и параметрами задачи.

Учитывая возможность использования различных режимов распараллеливания, вопрос о корректной декомпозиции данных встает особенно остро. Сначала рассмотрим вопрос о декомпозиции данных для каждого из уровней отдельно, а затем для смешанных режимов.

### **2.1 Декомпозиция для распараллеливания по математическим областям**

При распараллеливании по областям максимальное количество узлов, которые можно задействовать равно количеству областей. Разбиение задачи на математические области выполняется на этапе расчета начальных данных исходя из геометрии задачи и начальных данных. В результирующей сетке может очень существенно различаться объем вычислений приходящихся на каждую область из-за различного количества точек, веществ, используемых приближений и счетных алгоритмов. Это приводит к не высокой эффективности распараллеливания при расчете на максимальном количестве узлов. Для того чтобы обойти эту проблему, в рамках методики ТИМ-2D реализована возможность отнесения нескольких математических областей к одному MPI процессу. То есть одним процессом может обсчитываться не одна математическая область, а несколько. Благодаря этой возможности, как правило, удается сбалансировать вычислительную нагрузку, при этом уменьшается количество задействованных процессов, но ускорение остается на том же уровне, что и с использованием максимального количества.

Декомпозиция для распараллеливания по математическим областям производится в автоматическом режиме. Первоначальная декомпозиция выполняется на основе информации о количестве точек в областях. Недостатком такой декомпозиции является то, что она не учитывает реальную вычислительную нагрузку, связанную с областью, так как в разных областях используются различные вещества, используются различные приближения, по-разному работают алгоритмы контроля за качеством счётной сетки (коррекция и локальные перестройки). Это приводит к тому, что хорошо сбалансированная задача с дисбалансом по количеству точек, например, в 5% может считаться с эффективностью 60-70%. Для того чтобы преодолеть эти недостатки в процессе счета используются алгоритмы автоматической передекомпозиции, формирующие новую декомпозицию на основе реально затраченного времени на расчёт каждой математической области.

Сложностью при разработке таких алгоритмов стала «плавающая вычислительная нагрузка». Связано это с тем, что во время расчета вычислительная нагрузка меняется от шага к шагу главным образом в алгоритмах, работающих со счетной сеткой (коррекция и перестройки). При этом на соседних шагах время расчета области может различаться на 30-40% и более. С другой стороны, в процессе вычислений в математических областях, в результате работы алгоритмов локальных перестроек сетки постепенно изменяется количество счетных точек, что также изменяет время расчета области (на последующих шагах). Чтобы преодолеть эти трудности, декомпозиция выполняется на основе не общего времени счета с самого старта задачи и не на основе времени счета одного шага, а на основе времени счета последних 100 шагов. Такой подход позволяет с одной стороны сгладить влияние времени выполнения одного шага, с другой стороны учитывает реальное количество точек в расчете на момент выполнения декомпозиции (за 100 шагов количество точек меняется не значительно).

При сохранении разреза вызывается процедура, формирующая наилучшую декомпозицию для текущего количества процессов. В случае если сформированная декомпозиция оказывается лучше текущей и дает выигрыш в эффективности более чем на 10%, то производится передекомпозиция. При этом полностью освобождается вся память и зачитывается только что сохраненный разрез. Кроме этого формируются декомпозиции на количество процессов от 2 до  $N$ , где  $N$  - количество областей в задаче. При этом выдаются декомпозиции дающие ускорение выше, чем текущая. В случае необходимости пользователю рекомендуется изменить количество задействованных процессов для ускорения расчета.

## **2.2 Распределение вычислительной нагрузки между процессами при распараллеливании в модели общей памяти**

При распараллеливании в модели общей памяти каждый счетный блок ТИМ-2D распараллеливается не зависимо. Всего в настоящее время для расчета задач газовой динамики и упругопластичности выделено 4 счетных блока: расчет узловых величин, расчет ячейечных величин, коррекция счетной сетки, локальные перестройки сетки. В каждом счетном блоке организована одна параллельная область OpenMP.

При распараллеливании в модели общей памяти понятие декомпозиции не вполне корректно, так как все данные являются общими для нитей (вычислительных ядер одного узла). При этом между нитями разделяются не данные, а итерации счетных циклов, каждая из которых может выполняться независимо. Такая организация возможна благодаря использованию в методике ТИМ-2D нерегулярных сеток и как следствие организация всех счетных алгоритмов без привязки к структуре счетной сетки. То есть каждая точка рассчитывается полностью не зависимо от расчета других точек.

Такая схема распараллеливания позволяет использовать распараллеливание в модели общей памяти совместно с распараллеливанием в модели распределенной памяти (как по математическим областям, так и мелкозернистое).

## **2.3 Декомпозиция для мелкозернистого распараллеливания**

Программы, выполняющие декомпозицию, такие как ParMeTiS [27], в качестве исходных данных используют граф, отражающий связи между элементами сетки. При выполнении де-

композиции каждая вершина графа относится к одному из компактов, а некоторые ребра графа «разрезаются».

В методике ТИМ-2D декомпозиция осуществляется по ячейкам, так как ячейки являются основными счетными элементами.

Подход с разрезанием графа отображающего структуру сетки на подграфы является одним из самых универсальных. При таком подходе структура сетки представляется в виде графа и решается задача о разрезании этого графа на подграфы с условием равенства подграфов по суммарному весу (далее декомпозиция графа сетки). В качестве весов используется вычислительная нагрузка на расчет точки.

Так как в методике ТИМ-2D декомпозиция осуществляется по ячейкам, то в качестве вершин графа используются ячейки, ребра графа соответствуют соседству между ячейками.

Все это делает программы разрезания графа сетки на подграфы наиболее привлекательными для декомпозиции данных по процессам.

## **2.4 Декомпозиция в режиме счета с несколькими уровнями распараллеливания**

При использовании нескольких уровней распараллеливания вопрос с декомпозицией встает особенно остро.

### *2.4.1 Прозрачность декомпозиции для OpenMP распараллеливания в модели смешанной памяти*

Для распараллеливания в модели смешанной памяти распараллеливание с использованием OpenMP никаким модификациям не подвергается. Это происходит благодаря тому, что OpenMP распараллеливание осуществляется внутри математической области (распараллеливаются итерации счетных циклов по ячейкам или по узлам). В результате при распараллеливании по областям для OpenMP «не чувствует» этого распараллеливания (области относящиеся к другим процессам текущим процессом не обчитываются и соответственно OpenMP для них не используется). При использовании мелкозернистого распараллеливания (совместно с распараллеливанием по областям) каждая параобласть в структурах данных оформляется как область и поэтому OpenMP распараллеливание работает с ней точно также как и с обычной математической областью.

### *2.4.2 Мелкозернистое распараллеливание совместно с распараллеливанием по областям*

В режиме мелкозернистого распараллеливания на верхний уровень распараллеливания возлагается задача формирования параобластей из математической области и дальнейшее управления расчетом полученных параобластей. Это включает в себя сбор общей информации о ходе расчета, выработка счетного шага, сбор интегральной информации, задание параметров расчета области и так далее. Помимо этого, на верхнем уровне распараллеливания выполняется расчет контактных границ математической области (в случае если допускается их фрагментация).

Для мелкозернистого распараллеливания совместно с распараллеливанием по областям возникает вопрос об учете разделения геометрии задачи на математические области.

Первая проблема возникает из-за того, что математические области в ТИМ-2D являются независимыми объектами уровня задачи (при решении задач газовой динамики и упругопластичности). То есть в каждой области используется собственная сплошная нумерация элементов сетки начинающаяся с 1. В разных областях одной задачи количество точек может варьироваться от нескольких сотен до сотен тысяч. При этом ячейки каждой области разделяются на компакты не зависимо от других областей, но при выполнении декомпозиции необходимо получить сбалансированную нагрузку на процессы.

Решается эта проблема следующим образом. Каждую область делиться на большое количество компактов – примерно на порядок больше чем это необходимо (конкретная цифра тут не требуется, необходимо чтобы получаемые компакты были достаточно мелкими). В настоящее время на этом этапе формируются компакты из примерно 1000 ячеек, области меньше 5000

ячеек на компакты не разбиваются. Полученные маленькие компакты по всем математическим областям собираются вместе и представляются в свою очередь в виде графа уже по всей задаче. То есть формируется макро граф, вершинами которого являются небольшие компакты и маленькие области, полученные при декомпозиции областей. Производится повторная декомпозиция уже на заданное количество процессов, в результате которой получается итоговая декомпозиция. Таким образом, декомпозиция производится за два захода и называется двухпроходной.

Вторая проблема состоит в способе расчета контактного взаимодействия между математическими областями [36]. Для расчета контактного взаимодействия необходима информация обо всех граничных точках вдоль линии раздела двух математических областей, которые разделяет эта граница. Эту проблему можно решать несколькими путями.

1. При разбиении на компакты не учитывать границы между математическими областями, то есть разбивать математические области на компакты полностью независимо друг от друга. Этот подход имеет недостаток – после расчета временного шага будет необходимо выполнять сбор информации о границе на головном процессе математической области. Это в свою очередь может приводить к снижению эффективности распараллеливания, так как головному процессу математической области необходимо выполнить достаточно большое количество обменов до и после расчета границы.

С другой стороны для сильно вытянутых математических областей, не удастся выполнить декомпозицию каким-либо другим образом. Для таких областей приходится либо выполнять декомпозицию данным способом, либо представлять область одним компактом, то есть не использовать для этой области мелкозернистое распараллеливание. Это в свою очередь может существенно ограничить максимальное количество используемых процессов, так как накладывает ограничение на вес компактов в других областях (их вес<sup>1</sup> должен быть не меньше, чем вес рассматриваемой области). То есть эта область ограничивает снизу вес всех параобластей.

2. Второй способ заключается в формировании одного компакта для ячеек около всех границ математической области. В этот компакт включаются один или несколько слоев ячеек, лежащих вдоль всех границ математической области. Количество слоев вдоль границ может быть различным и составлять от 1 до 10 (конкретные требования по количеству слоев определяется алгоритмами поддержания счетной сетки). Этот компакт относится к головному процессу соответствующей области. Расчет контактных границ при этом производится по тем же алгоритмам что и при распараллеливании по областям. Декомпозиция при этом осуществляется для внутренних ячеек области не вошедших в компакт вдоль границ.

Недостатком этого подхода является то, что количество ячеек вдоль границ может быть достаточно велико, особенно в случае использования нескольких слоев. В этом случае приграничная параобласть становится очень большой и также ограничивает вес всех компактов задачи снизу. Другой недостаток этого подхода – это большое количество соседних компактов для приграничного компакта и как следствие большое количество обменов.

Преимуществом этого подхода является то, что он не требует выполнения дополнительных обменов при расчете контактных границ, поскольку при такой декомпозиции не происходит фрагментации границ.

3. Третий способ заключается в формировании пары компактов со стороны каждой области для одной границы. Здесь также может использоваться не один, а несколько слоев ячеек вдоль границы. Сформированная пара компактов относится к одному и тому же процессу. В этом случае для расчета контактной границы вообще не нужно выполнять никаких обменов. Расчет контактной границы при этом производится по той же схеме, что и в последовательном режиме счета. Декомпозиция осуществляется для внутренних ячеек области не вошедших в компакты вдоль границ.

Недостатком этого подхода является то, что при формировании компактов вдоль границ около угловой точки области возникает наложение по ячейкам между приграничными компактами этой области. В случае формирования компактов вдоль одного слоя наложение происходит для одной ячейки, для двух слоев 3-4 ячейки и так далее. Возникновение такого

---

<sup>1</sup> Под «весом» понимается вычислительная нагрузка.

наложения требует дополнительной обработки в программах распараллеливания, так как при декомпозиции каждая ячейка относится только к одному компакту. Кроме того такая декомпозиция не решает полностью проблемы с расчетом особых точек, в которых сходятся несколько границ, и с выполнением операции перехода точек с одной границы на другую. По этим причинам было решено данный подход не использовать.

### 2.4.3 Схемы проведения двухпроходной декомпозиции

В зависимости от способа учета контактных границ между математическими областями можно предложить следующие схемы проведения двухпроходной декомпозиции.

1. При разбиении на компакты граница между областями не учитывается. Макро граф в таком режиме состоит из несвязанных подграфов, каждый из которых соответствует одной математической области. Далее выполняется декомпозиция этого макрографа на заданное количество процессов. В результате получается декомпозиция, причем к одному процессу могут быть отнесены компакты от разных областей не соседние между собой.

Более сложный вариант этого подхода – введение фиктивных вершин макрографа, эти вершины соединяются ребрами с теми вершинами макрографа которые соответствуют приграничным компактам. В этом случае макрограф не будет распадаться на независимые куски. В этом случае в декомпозиции не будут возникать компакты из разных частей задачи (например, пара компактов с двух сторон от границы, но не обязательно лежащие друг напротив друга относительно этой границы). Для такого варианта можно использовать очень большие веса для ребер макрографа сходящихся в фиктивных вершинах и/или для ребер соединяющие между собой соседние вершины, лежащие вдоль границ. Возможность использовать веса для ребер имеется в ряде многоуровневых методов декомпозиции. Чем больше вес ребра, тем меньше вероятность что он будет «разрезан» при разбиении на подграфы, то есть уменьшается вероятность разнесения по разным компактам вершин соединенных этим ребром. В этом случае после проведения декомпозиции компакты рядом с границами оказываются вытянутыми вдоль границ. Это позволяет в значительной степени уменьшить количество обменов для расчета контактного взаимодействия. Опыт использования такого алгоритма показывает, что увеличения весов ребер между приграничными вершинами графа позволяет примерно втрое сократить количество компактов вдоль границ.

2. Формирование одного компакта для ячеек около всех границ области. Макрограф в этом режиме строится следующим образом. Вершины макрографа, соответствующие приграничным компактам, соединяются ребрами согласно соседству между областями через границу.

Упрощенный подход для такого режима заключается в расчете всех приграничных компактов одним процессом (например, головным процессом задачи). В этом случае для расчета границ не понадобится никаких обменов. Недостатком здесь является то, что при большом количестве областей процесс, рассчитывающий приграничные компакты будет перегружен. Это может ограничивать количество используемых процессов.

3. Формирование пары компактов со стороны каждой границы. В этом режиме в макрографе каждая пара компактов вдоль границы представляется в виде одной вершины. Это позволяет гарантировать, что при любой декомпозиции каждая такая пара будет отнесена к одному процессу, то есть для расчета этой границы обменов требоваться не будет. К плюсам этого подхода можно отнести то, что в этом режиме, по сравнению с режимом 2, слабее ограничения на вычислительную нагрузку, накладываемые размером приграничных компактов. Недостатки следуют из недостатков формирования компактов вдоль границ и возникающего между компактами наложения по ячейкам.

В рамках методики ТИМ-2D в настоящее время используются подходы 1 и 2. Подход 2 для декомпозиции на небольшое количество процессов (десятки) и подход 1 для нескольких сотен процессов.

## 3. Критерии оценки качества декомпозиции

Задача рационального разбиения графа на определенное количество частей, содержащее равное количество вершин при условии минимизации числа разрезанных ребер и равномерного

распределения их по всем подграфам, наиболее близкой к которой является задача о разрезании графов [16], относится к классу *NP*-полных. Поэтому эта задача решается, как правило, приближенно при помощи различных геометрических или эвристических алгоритмов. Общим недостатком алгоритмов декомпозиции является то, что они учитывают ограниченное число критериев, накладываемых на подграфы, в результате, разбиение может оказаться неудачным, для использования в методике ТИМ-2D.

В данной работе предложен первый вариант списка критериев для методики ТИМ-2D. В дальнейшем он, по-видимому, будет пополняться. Например, ряд дополнительных ограничений могут быть введены из-за требований, накладываемых локальными перестройками сетки путем дробления и объединения ячеек. В настоящее время не даются количественные характеристики для получения приемлемой декомпозиции для вводимых критериев. Их доработка проводится при реализации мелкозернистого распараллеливания для счетных блоков при тестировании и выявлении не удачных декомпозиций.

### 3.1 Критерии для достижения более высокой эффективности распараллеливания

Для декомпозиции данных в режиме мелкозернистого распараллеливания в методике ТИМ-2D в качестве базовой используется библиотека ParMeTiS [27]. Некоторые из декомпозиций полученных пакетом ParMeTiS использовались в качестве тестовых.

Алгоритмы в библиотеке ParMeTiS при построении декомпозиции учитывают два обстоятельства: равенство подграфов по суммарному весу и минимальность весов разрезанных ребер (ребро называется разрезанным, если вершины которые оно соединяет отнесены к разным подграфам). Сочетание этих двух критериев не гарантирует получение оптимальной и даже приемлемой декомпозиции. Кроме того декомпозиция данных из-за разбиения геометрии задачи на несколько областей должна выполняться в два этапа (смотри пункт 2.4), при этом могут нарушаться и эти два условия.

Список критериев, используемы для методики ТИМ-2D с точки зрения получения высокой эффективности распараллеливания, следующий.

- Равенство подграфов по количеству вершин подграфа или, если граф взвешенный, то по суммарному весу вершин. Этот критерий отвечает за сбалансированность вычислительной нагрузки между процессами. Этот критерий контролируется в ParMeTiS.
- Процентное отношение разрезанных ребер в графе. Объем передаваемых между процессами данных определяется количеством разрезанных ребер. Уменьшение относительного количества разрезанных ребер позволяет сократить объем передаваемых данных и уменьшить простои процессов. Этот критерий также контролируется в ParMeTiS.
- Количество разрезанных ребер между парой подграфов – количество ребер соединяющих вершины каждого подграфа с вершинами других подграфов. Количество разрезанных ребер между парой подграфов отражает время, необходимое для пересылки информации от одного процесса другому. В этом критерии оценивается объем данных для пересылки между каждой парой процессов, рассчитывающих соседние параобласти. Желательно чтобы эти объемы были приблизительно равны между собой. Однако основное назначение этого критерия – выявление «коротких» границ, которые являются не желательными для алгоритмов поддержания качества сетки.
- Количество соседних подграфов у каждого подграфа. При передаче информации между процессами существенную часть времени занимает установка соединения, которое может превышать само время передачи. Число соседних подграфов определяет количество необходимых соединений. Желательно чтобы количество соседей у каждого подграфа было примерно одинаковым. Критерий позволяет оценить, насколько одинаковым будет время, которое каждый процесс потратит на установку соединений.
- Критерий связности подграфов позволяет в некоторых случаях упростить вычислительные алгоритмы.

Необходимо отметить, что в случае если подграф распадается на достаточно крупные куски, то такая декомпозиция может считаться приемлемой. Плохой декомпозицией может считаться декомпозиция, в которой возникают, например, оторванные ячейки. Таким образом, для несвязного подграфа разбиение можно считать «удачным», если каждая его ком-



понента имеет примерно одинаковый вес, одинаковое количество соседних подграфов и разрезанных ребер. Здесь и далее два подграфа называются соседними, если существует, хотя бы одно ребро, соединяющее вершину одного подграфа с вершиной другого подграфа.

- У вершины количество разрезанных ребер больше, чем неразрезанных. Этот критерий позволяет выявить локальные ошибки алгоритмов декомпозиции, когда одна ячейка отнесена не к тому компакт. Исправление таких локальных ошибок в ряде случаев позволяет улучшить геометрическую форму границы между параобластями.
- Соотношение между суммарным весом вершин и разрезанных ребер подграфа. При распараллеливании вычислений необходимо стремиться к тому, чтобы время, затрачиваемое на обмен информацией, было меньше времени, затрачиваемого на счёт. Время на обмен информацией между процессами складывается из времени, необходимого на установку соединения между процессами, и времени, необходимого непосредственно на передачу информации. Поэтому суммарный вес разрезанных ребер подграфа, который характеризует необходимое время на передачу информации, должен быть намного меньше общего веса вершин подграфа, отражающего время счета соответствующим процессом. Этот критерий контролируется в библиотеке ParMeTiS, однако особая обработка ячеек вдоль границ может приводить к получению неудачной декомпозиции.

### 3.2 Критерии, накладываемые счетными алгоритмами

В методике ТИМ-2D метод дифференцируемых связей используется в качестве основного алгоритма поддержания счетной лагранжевой сетки на протяжении расчета без изменения структуры сетки. В настоящее время в методе дифференцируемых связей реализовано несколько способов поддержания приемлемых многоугольных счетных ячеек для возможности моделирования течений на неструктурированных сетках:

- сетки, содержащие «звездные» ячейки (ячейка называется звездной, если все ее узлы находятся в прямой видимости из ее центра);
- сетки с выпуклыми углами, образованными двумя ребрами ячейки с общим узлом (ячейки – выпуклые многоугольники);
- сетки с произвольными не самопересекающимися многоугольными ячейками (центр ячейки может находиться «вне» ячейки);
- сетки, обладающие какими-либо другими свойствами, наряду с вышеперечисленными: контроль длин ребер и величины углов ячейки, контроль «характерного размера» ячейки, несамопересечение контактных и «свободных» границ и ряд других.

Условия поддержания приемлемой формы счетных ячеек могут применяться в любой комбинации одновременно, в различных сочетаниях в разных частях задачи.

При реализации мелкозернистого распараллеливания для метода дифференцируемых связей возник ряд трудностей с обработкой особых точек (точек, которые окружают ячейки трех или более компактов). Поэтому желательно чтобы таких особых точек было как можно меньше, кроме того необходимо чтобы они находились на достаточном удалении друг от друга. Эти условия нашли отражение в следующих критериях:

- Количество разрезанных ребер между парой подграфов. Этот критерий был введен в предыдущем пункте для отражения равенства объема передаваемых данных между парами процессов. Кроме этого данный критерий позволяет выявлять «короткие» границы между компактами, что является дополнительной сложностью для алгоритмов коррекции счетной сетки.
- Непрерывность линии разделяющей пару подсеток, соответствующих соседним подграфам. Линию образует множество ребер и узлов сетки, которые относятся к ячейкам сетки, принадлежащим двум соседним подсеткам. На каждом разрыве линии возникает две дополнительных особых точки, которые повышают нагрузку на алгоритмы дифференцируемых связей.

### 3.3 Тестирование алгоритмов оценки качества декомпозиции

Ряд декомпозиций, на которых проводилась тестирование алгоритмов оценки качества приведены на рисунке 1.

По результатам тестирования можно сделать вывод о том, что представленные критерии корректно обрабатывают декомпозиции различных типов, как хорошего, так и неудовлетворительного качества. В дальнейшем необходимо их тестирование в сочетании со счетными алгоритмами для выработки конкретных количественных характеристик для различных критериев.

Декомпозиции, полученные с помощью библиотеки ParMeTiS можно охарактеризовать как хорошо сбалансированные по вычислительной нагрузке, общее количество разрезанных ребер мало по сравнению с общим количеством ребер в графе. Линия между парами подсеток, как правило, непрерывна. Подграфы получаются связными.

К недостаткам декомпозиции можно отнести то, что достаточно часто в декомпозициях возникают «короткие» границы между параобластями. Отклонение от среднего значения количества соседних подграфов достигает 50%. Необходимо отметить, что качество декомпозиции ParMeTiS резко снижается при определенных условиях, например при небольшом количестве вершин в графе. В редких случаях декомпозиции могут оказаться с неодносвязными компактными.

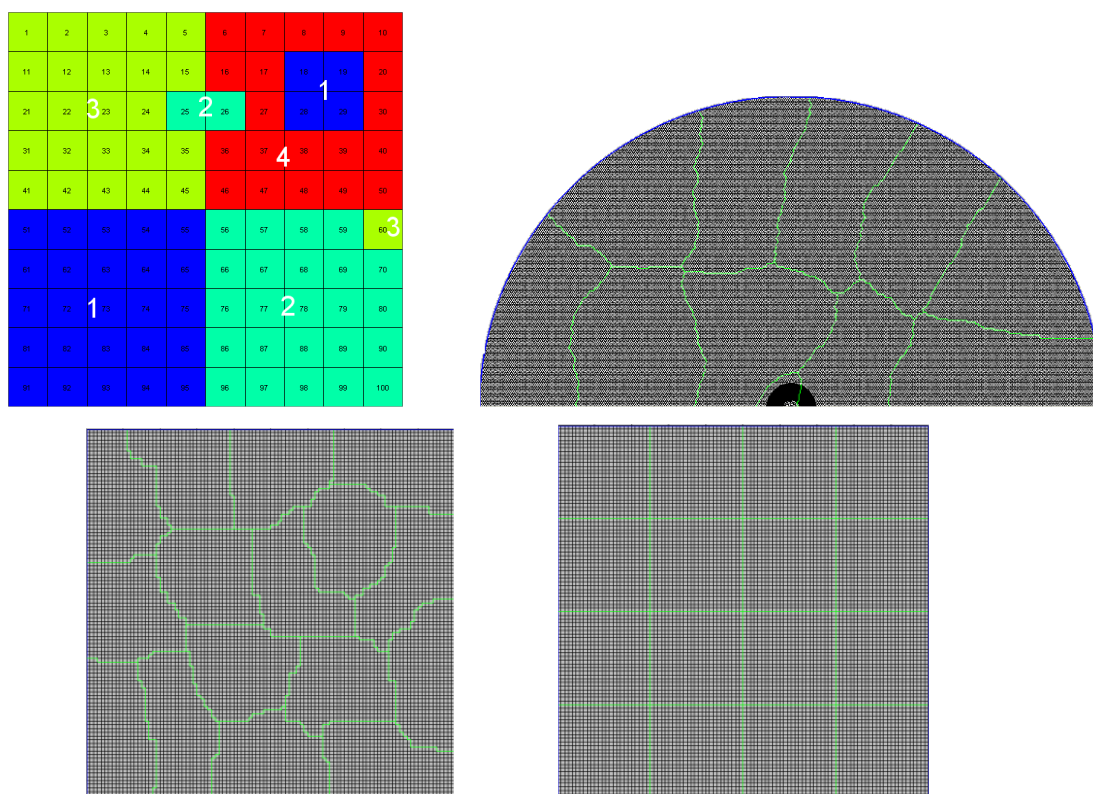


Рис. 1 – Примеры тестовых декомпозиции для оценки критериев качества

## 4. Заключение

В докладе описываются алгоритмы декомпозиции, используемые для методики ТИМ-2D в различных режимах параллельного счета в смешанной модели памяти. Описаны алгоритмы, использующие двухпроходную декомпозицию.

Проведена оценка различных счетных алгоритмов, используемых в ТИМ-2D в части ограничений накладываемых на декомпозицию данных. На основе этих требований сформулирован ряд критериев для оценки качества декомпозиции, который в дальнейшем может пополняться. Разработаны алгоритмы, реализующие эти критерии, которые протестированы на ряде тестовых

вых декомпозиций. Проведено тестирование программы проверки критериев на графах, полученных по четырехугольным и многоугольным (преимущественно шестиугольным) сеткам, которые в настоящее время используются в качестве базовых в ТИМ-2D. При этом использовались декомпозиции, заданные вручную и полученные при помощи пакета ParMeTiS.

В дальнейшем возможна разработка метода учитывающего рассмотренные критерии непосредственно на этапе выполнения декомпозиции.

## 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., Соколов С.С. Использование интерфейса OpenMP для распараллеливания методики ТИМ // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2007. Вып. 3. С. 74-82.
- 4 Gilbert J., Miller G., Teng S. Geometric mesh partitioning: Implementation and experiment // Proceedings of International Parallel Processing Symposium, USA. 1995.
- 5 Miller G., Teng S., Thurston W., Vavasis S. Automatic mesh partitioning // Sparse Matrix Computations: Graph Theory Issues and Algorithms. IMA Volumes in Mathematics and its Applications / ed. by A. George, J. Gilbert, J. Liu. Springer-Verlag, 1993.
- 6 Berger M., Bokhari S. Partitioning strategy for nonuniform problems on multiprocessors // IEEE Transactions on Computers, C-36(5): 570-580, 1987.
- 7 Pilkington J., Baden S. Partitioning with spacefilling curves. Technical Report CS94-349, Dept. of Computer Science and Engineering, Univ. of California, 1994.
- 8 Бутнев О.И., Пронин В.А. Алгоритмы декомпозиции нерегулярной сетки с учетом вычислительной нагрузки. // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов 2004. Вып. 4. С. 81-87.
- 9 Schloegel K., Karypis G., Kumar V. Graph Partitioning for High Performance Scientific Simulations. Army HPC Research Center Dept. of Computer Science and Engineering, University of Minnesota; Minneapolis, Minnesota, 1999.
- 10 Karypis G., Kumar V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. University of Minnesota, Department of Computer Science Minneapolis, MN 55455, Technical Report: 95-035, 1998.
- 11 Яковлевский М. В. Обработка сеточных данных на распределенных вычислительных системах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2004. Вып. 2. С. 40 – 53.
- 12 Williams R. Performance of dynamic load balancing algorithms for unstructured mesh calculations // Concurrency: Practice and Experience, № 3. 1991. P. 457-481.
- 13 Hu F., Blake R. Load Balancing for Unstructured Mesh Applications // Parallel and Distributed Computing Practices. №2. 1999.
- 14 Gremban K., Miller G.L., Teng S.-H. Moment of inertia and graph separators // Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. January, 1994. P. 452 - 461.
- 15 Behrens J., Zimmermann J. Parallelizing an Unstructured Grid Generator with a Space-Filling Curve Approach, Euro-Par 2000 // Parallel Processing: 6th International Euro-Par Conference, Munich, Germany, August/ September 2000. Proceedings Volume 1900/2000. P.815.
- 16 Евстигнеев В.А. Применение теории графов в программировании / Под ред. А.П. Ершова. - М.: Наука. 1985.
- 17 George A., Liu J. Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs, NJ, 1981.

- 
- 18 Kernighan B., Lin S. An efficient heuristic procedure for partitioning graphs // *Bell System Technical Journal*, 29. 1970. P. 291-307.
  - 19 Fiduccia C., Mattheyses R. A linear time heuristic for improving network partitions // *In Proc. 19th IEEE Design Automation Conference*. 1982. P. 175-181.
  - 20 Hendrickson B., Leland R. An Improved Spectral Graph Partitioning Algorithm For Mapping Parallel Computations // *SIAM J. Sci. Comput.* 1995, vol.16. № 2. P. 452-469.
  - 21 Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory // *Czechoslovak Mathematical Journal*, 25(100). Praha. 1975. P. 619-633.
  - 22 Bui T., Jones C. A heuristic for reducing fill in sparse matrix factorization // *In 6<sup>th</sup> SIAM Conf. Parallel Processing for Scientific Computing*. 1993. P. 445-452.
  - 23 Gupta A. Fast and effective algorithms for partitioning and sparse matrix reordering // *IBM Journal of Research and Development*, 41 (1/2): 1996. P. 171-183.
  - 24 Hendrickson B., Leland R. A multilevel algorithm for partitioning graphs // *Proceedings Supercomputing '95*. San Diego, CA, 1995.
  - 25 Walshaw C., Cross M., Diekmann R., Schlimback F. Multilevel mesh partitioning for optimizing domain shape. Technical Report 98/IM/38, School of Computing and Mathematical Sciences, University of Greenwich, London, UK. 1998.
  - 26 Karypis G., Kumar V. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 4.0. Technical report, Univ. of Minnesota, Dept. of Computer Sci. and Engr., 1998.
  - 27 Karypis G., Schloegel K., Kumar V. ParMeTiS: Parallel Graph Partitioning and Sparse Matrix Ordering Library, version 3.1. University of Minnesota, Department of Computer Science and Engineering Army HPC Research Center Mineapolis, 2003.
  - 28 Hendrickson B. The CHACO user's guide, version 2.0. Technical Report SAND94-2692, Sandia National Laboratories, USA. 1994.
  - 29 Walshaw C. Parallel JOSTLE user guide. Technical Report User guide Version 1.2.9, University of Greenwich, London, UK, 1998.
  - 30 Preis R., Diekmann R. PARTY – a software library for graph partitioning. Technical report, University of Paderborn, 1997.
  - 31 Pellegrini F., Roman J. SCOTCH: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. HPCN-Europe, Springer LNCS 1067. P 493-498, 1996.
  - 32 Sohn A. S-HARP: A parallel dynamic spectral partitioner. Technical report, Dept. of Computer and Information Science, New Jersey Institute of Technology, 1997.
  - 33 Zoltan: Data-Management Services for Parallel Applications [Electronic resource]. Mode of access: [<http://www.cs.sandia.gov/~kddevin/09.07.2006>].
  - 34 Новиков И.Г., Панов А.И., Соколов С.С. Способ коррекции нерегулярной лагранжевой сетки методом наложения дифференцируемых связей // *Журнал вычислительной математики и математической физики РАН*. 2005. Том 45. №8. С. 1487-1500.
  - 35 Новиков И.Г., Соколов С.С. Локальные перестройки нерегулярной сетки в методике ТИМ-2D // «Молодежь в науке». Сборник докладов шестой научно-технической конференции. г. Саров, 30 октября - 1 ноября 2007 года. Саров: ФГУП "РФЯЦ-ВНИИЭФ", 2007. С. 136 - 144.
  - 36 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.