

УДК 519.6

## МЕТОД ТРЕХУРОВНЕВОГО РАСПАРАЛЛЕЛИВАНИЯ МЕТОДИКИ ТИМ-2D

А. А. Воропинов, С. С. Соколов  
(РФЯЦ-ВНИИЭФ, г. Саров)

Методика ТИМ-2D предназначена для решения задач механики сплошной среды на неструктурированных многоугольных лагранжевых сетках произвольного вида. В методике используется трехуровневое распараллеливание. На первом и втором уровнях, при распараллеливании счета по математическим областям и внутри математических областей по параобластям, используется модель распределенной памяти и интерфейс MPI. На третьем уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с помощью интерфейса OpenMP. Уровни распараллеливания могут использоваться как по отдельности, так и в различных сочетаниях при решении одной задачи.

*Ключевые слова:* методика ТИМ-2D, неструктурированные сетки, распараллеливание, модель смешанной памяти.

### Введение

Методика ТИМ-2D [1] предназначена для решения двумерных нестационарных задач механики сплошной среды на неструктурированных многоугольных лагранжевых сетках произвольного вида. Ячейками сетки являются несамопересекающиеся многоугольники, в узлах может сходиться любое число ребер. Для расчета задач газодинамики и упругопластичности используются явные конечно-разностные схемы, для моделирования диффузионных приближений — неявные конечно-разностные схемы. Кинематические величины отнесены к узлам разностной сетки, термодинамические величины — к ячейкам. Решение уравнений механики сплошных сред осуществляется в цилиндрической или плоской декартовой системе координат.

Для скорейшего получения результатов с необходимой точностью используются различные методы распараллеливания, позволяющие рассчитывать задачу одновременно на большом количестве процессорных ядер. В настоящее время наиболее распространенными прикладными стандартами для распараллеливания являются интерфейс передачи сообщений MPI, предназначенный для модели распределенной памяти, и интерфейс OpenMP для модели общей памяти.

При решении сложных двумерных задач начальную геометрию системы часто приходится разбивать на математические области (далее для краткости *матобласти*). Такое разбиение иногда необходимо для более точного описания взаимодействия тел на выделенной линии скольжения. Методика ТИМ-2D позволяет проводить расчеты в многообластной постановке. При этом в каждой матобласти используется независимая расчетная сетка. В начале каждого счетного шага между матобластями решается задача контактного взаимодействия, далее положение и скорости границ выступают в качестве навязанных граничных условий при расчете матобластей. Сами матобласти рассчитываются независимо друг от друга. Это дает первый ресурс для распараллеливания методики ТИМ-2D — по матобластям, который достаточно легко использовать.

Второй ресурс связан с независимым расчетом всех внутренних элементов матобластей, что является следствием неструктурированности сетки. В модели распределенной памяти он реализуется как мелкозернистое распараллеливание с пространственным разделением области на фрагменты-параобласти (декомпозиция по пространству).

Если проводить независимый расчет элементов сетки в модели общей памяти, то естественно воспользоваться третьим ресурсом — распараллеливанием итераций счетных циклов (которые, как правило, осуществляются по элементам сетки). Такой подход не требует выполнения декомпозиции по пространству, что позволяет применить его в сочетании с другими.

Таким образом, в методике ТИМ-2D имеется три основных ресурса, использование которых привело соответственно к трем уровням распараллеливания. Первые два основаны на пространственном разбиении задачи на фрагменты (по матобластям и внутри матобластей по параобластям) и применении модели распределенной памяти. Третий — распараллеливание итераций счетных циклов в модели общей памяти.

Данная статья посвящена вопросам одновременного использования уровней распараллеливания при расчете одной задачи.

## 1. Общие принципы распараллеливания

Подробное описание алгоритмов распараллеливания в методике ТИМ-2D приводилось ранее в работах [2—8]. Здесь укажем лишь общие принципы, положенные в их основу.

**Декомпозиция данных.** Эффективное использование счетных программ методик на параллельных ЭВМ требует выполнения декомпозиции таким образом, чтобы ядра вычислительной системы были загружены равномерно, а взаимодействия между ними были минимальными.

В методике ТИМ-2D для распараллеливания в модели распределенной памяти используется декомпозиция по пространству. Такой подход позволяет единым образом выполнять декомпозицию как с разбиением матобласти на параобласти, так и без такого разбиения. При применении трех уровней распараллеливания декомпозиция производится в два прохода. Сначала выполняется декомпозиция каждой области на большое количество фрагментов. Затем строится макрограф, вершины которого ассоциируются с полученными фрагментами, и он разрезается на необходимое количество частей, соответствующих параобластям. Непосредственно задача о разрезании графа на подграфы решается при помощи алгоритмов библиотек ParMeTiS или SCOTCH [9]. Отметим, что при декомпозиции небольшие матобласти могут вообще не раз-

деляться на параобласти и рассчитываются без деления.

В модели общей памяти между нитями OpenMP распределяются итерации счетных циклов — как правило, это циклы по счетным элементам сетки, таким как ячейки и узлы. Благодаря отсутствию декомпозиции данных не возникает конфликтов между распараллеливанием в общей и распределенной памяти (подробнее см. разд. 2).

**Уровень распараллеливания по областям.** На верхнем уровне распараллеливание осуществляется по матобластям в модели распределенной памяти. Поскольку расчет матобластей полностью независим, взаимодействие между процессами осуществляется только на этапе расчета контактного взаимодействия между ними. При этом сама контактная граница между двумя областями рассчитывается на одном из двух процессов, к которым отнесены области, формирующие эту границу. Для взаимодействия между процессами используется обмен сообщениями MPI.

**Уровень мелкозернистого распараллеливания.** В методике ТИМ-2D деление на области производится, исходя из начальных данных, таких как параметры геометрии и распределение веществ. При этом не преследуется цель достижения высокой эффективности распараллеливания. В разных областях может быть задано различное количество точек, используются разные вещества и приближения. Это приводит к тому, что распараллеливание удастся выполнить для небольшого количества процессоров. Дополнительное распараллеливание в модели общей памяти позволяет увеличить количество задействованных ядер почти на порядок. Тем не менее в реальных расчетах их число не превышает нескольких десятков. Отсюда следует, что такого распараллеливания недостаточно и необходимо использование мелкозернистого распараллеливания.

Основной вопрос, который возникает при мелкозернистом распараллеливании, — это способ расчета узлов сетки, которые окружают ячейки, рассчитываемые в разных параобластях (параграничные узлы). В работе [7] для методики ТИМ-2D рассмотрены два варианта — с наложением параобластей в один слой ячеек и без наложения, с расчетом контактного взаимодействия между параобластями. В данной работе

при проведении тестовых расчетов использовался режим без наложения.

### Уровень распараллеливания на общей памяти.

Методика ТИМ-2D использует неструктурированные многоугольные сетки произвольного вида. Поэтому все алгоритмы методики *поточечные*, так что результат расчета не зависит от последовательности обработки элементов сетки. Отсюда следует, что для счетных циклов внутри области, по ячейкам или узлам, каждая итерация может быть выполнена независимо от других; итерации могут выполняться одновременно разными процессорными ядрами, но всегда будет получаться одинаковый результат. Связанный с этим ресурс распараллеливания реализован с использованием модели общей памяти с помощью программного интерфейса OpenMP, который является кроссплатформенным стандартом.

## 2. Согласование уровней распараллеливания

Отдельной проблемой оказалось согласование алгоритмов трех уровней распараллеливания между собой. Наиболее удобно рассматривать решение этой проблемы с точки зрения мелкозернистого распараллеливания, так как оно представляет собой средний уровень и для него вопрос согласования стоял наиболее остро.

**Использование смешанной модели памяти.** Первый вопрос заключается в использовании смешанной модели памяти и одновременном выполнении алгоритмов распараллеливания, реализованных с помощью средств MPI и OpenMP. При распараллеливании в модели смешанной памяти на каждом вычислительном узле выполняется один процесс. Процесс осуществляет цикл по счетным областям задачи, но области, отнесенные к другим узлам, пропускает. Счетной областью может быть как матобласть, так и параобласть, в зависимости от режима. Получая нумерацию, принятую в стандарте MPI (от 0 до  $N - 1$ , где  $N$  — общее количество используемых вычислительных узлов для расчета задачи), эти процессы выступают в качестве MPI-процессов, они управляют расчетом задачи в рамках вычислительного узла, производят MPI-обмены. В рамках каждого вычислительного узла MPI-процесс независимо от других узлов порождает параллельные нити OpenMP для прове-

дения расчетов областей, отнесенных к данному вычислительному узлу (и только этих областей). Таким образом, OpenMP-распараллеливание не зависит от распараллеливания в модели распределенной памяти.

**Согласование мелкозернистого и пообластного распараллеливания.** Алгоритмы пообластного и мелкозернистого распараллеливания используют модель распределенной памяти и декомпозицию по пространству. В связи с этим возникают вопросы, связанные с описанием параобластей в структурах данных и *эффектом фрагментации границ* между матобластями по нескольким параобластям.

*Организация параобластей в структуре данных.* Одно из основных требований при разработке алгоритмов мелкозернистого распараллеливания — чтобы организация параобластей в структуре данных была прозрачной для счетных алгоритмов, т. е. счетный алгоритм должен одинаково "работать" как с матобластями, так и параобластями. Для этого параобласть имеет структуру матобласти, но с присутствием дополнительных величин, описывающих ее как параобласть и делающих привязку к *родительской* матобласти. Кроме того, организуется дополнительная структура, содержащая информацию о глобальной нумерации элементов сетки, информацию для обменов между параобластями (о параграницах) и другие данные.

Параобласти нумеруются в списке матобластей, получая номера вслед за ними. Матобласть, для которой были порождены параобласти, сохраняя свой номер, из счета исключается. Параобласть, как и матобласть, является глобальным объектом уровня задачи. Благодаря такому подходу возможно использование единых алгоритмов для распараллеливания по матобластям и для распределения параобластей по процессам с дальнейшим управлением.

Рассмотрим нумерацию параобластей на примере. Имеются три матобласти; из первой формируются две параобласти, из второй параобластей не формируется, а из третьей формируются три параобласти. В результате параобласти, полученные из первой матобласти, приобретают номера 4, 5, а параобласти из третьей матобласти — номера 6, 7, 8. Общее количество областей в структуре данных устанавливается равным 8. Отметим, что исходные матобласти 1 и 3 сохраняют свои номера, но из счета исключают-

ся. Матобласть 2 рассчитывается под своим номером, параобласть для нее не формируется.

*Контактные границы.* При мелкозернистом распараллеливании многообластных задач может возникать эффект фрагментации границ, когда точки стороны границы распределяются по нескольким параобластям.

В этом случае необходима предварительная сборка границы, требующая дополнительного обмена — пересылки фрагментов границы на управляющий процесс матобласти. После расчета положения и скорости точек границы рассылаются всем процессам, имеющим фрагменты сторон.

В остальном алгоритмы расчета фрагментированных границ остаются такими же, как и при распараллеливании по областям.

**Согласование мелкозернистого распараллеливания и распараллеливания итераций счетных циклов.** Второй и третий уровни распараллеливания логически согласуются достаточно просто, так как в этих режимах используются разные парадигмы. На общей памяти распараллеливаются итерации счетных циклов по ячейкам и узлам. В параобласти вводится локальная нумерация элементов сетки, поэтому для распараллеливания в общей памяти параобласть идентична матобласти и модификаций не требуется. Тем не менее имеются две причины для специальной доработки алгоритмов.

*Разделение счетных циклов.* Для мелкозернистого распараллеливания с наложением необходимо выполнение обменов приграничными элементами сетки между соседними параобластями. При этом используются асинхронные обмены для их совмещения с вычислениями. Чтобы как можно быстрее начать пересылку данных о приграничных элементах сетки при формировании параобластей, им присваиваются номера меньше внутренних. В результате в цикле по ячейкам или узлам эти данные обрабатываются раньше, и можно начать их пересылку соседней параобласти. Однако возникает конфликт. Дело в том, что при распараллеливании счетных циклов нарушается последовательность обработки итераций, т. е., например, в цикле от 1 до 1000 101-я итерация может быть выполнена раньше 100-й. Нарушение последовательности итераций происходит из-за их распределения между нитями OpenMP.

Выходом из положения стало разделение счетных циклов на пары: первый цикл выполняется по параграничным и приграничным элементам сетки, второй — по внутренним элементам. При этом оба цикла заключены в одну параллельную область OpenMP и внутри нее обмены между циклами также выполняются в параллельном режиме.

*Обращение к MPI-процедурам в параллельных областях OpenMP.* Вызов процедур асинхронных обменов производится в параллельных областях OpenMP между выполнением двух счетных циклов: по приграничным и внутренним узлам или ячейкам. При этом несколькими нитями (в рамках одного процесса) одновременно могут вызываться MPI-процедуры для обмена данными о параобластях с соседями. Если на вычислительной системе используется библиотека MPI, не являющаяся *безопасной для нитей* (thread safe), это может приводить к авостам в MPI.

Чтобы избежать одновременного обращения, обеспечив вызов MPI-обменов в данный момент только одной нитью, все обращения к MPI-процедурам заключаются в критическую область OpenMP с одинаковым именем MPI. Такой подход решает проблему, практически не снижая скорости выполнения, при условии, что количество обменов невелико. Кроме того, его использование гарантирует, что и в других блоках подобных проблем не возникнет.

### 3. Различные режимы параллельного счета

Три уровня распараллеливания могут использоваться различным образом: по отдельности, в разных попарных сочетаниях и все три одновременно. Всего вместе с последовательным режимом возможно восемь режимов счета. Выбор конкретного режима определяется задачей, моделируемыми приближениями и характеристиками вычислительной системы.

Последовательный режим, как правило, используется на персональном компьютере для методических расчетов, а также различных обработок результатов. Этот режим часто применяется в начале разработки алгоритмов, а также в процессе устранения проблем, не связанных с распараллеливанием.

Рассмотрим более подробно режимы параллельного счета.

**Режимы счета с одним уровнем распараллеливания.** В эту категорию входят три режима счета.

*Распараллеливание по счетным областям в модели распределенной памяти (крупноблочное распараллеливание).* Этот режим может использоваться для расчета небольших задач на малом количестве ядер (обычно менее 10). Ограничение данного режима распараллеливания обусловлено не только ограниченным количеством матобластей, но и сложностями в балансировке нагрузки, поскольку области отличаются количеством точек, свойствами и типами материалов. Такое распараллеливание применяется, когда использование общей памяти невозможно (например, из-за характеристик вычислительной системы).

*Распараллеливание по параобластям в модели распределенной памяти (мелкозернистое распараллеливание).* В отдельности такой режим может использоваться для расчета однообластных задач; в этом случае уровни управления задачей и областью фактически совмещаются. Из-за малого количества таких задач данный режим применяется редко.

*Распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP.* Использование данного режима ограничено тем, что в современных кластерных системах количество ядер, а следовательно, количество нитей, "работающих" на общей памяти, невелико. По этой причине распараллеливание в модели общей памяти оказывается недостаточным для проведения расчетов с большим числом точек. Данный режим применяется при выполнении некоторых сервисных операций, расчетов с небольшим числом точек (не более 100 тыс. ячеек) и на вычислительных системах, использующих только общую память.

Как видно из описания, каждый из этих режимов имеет существенные ограничения в применении и по отдельности не может быть использован как универсальный. Поэтому основными являются смешанные режимы распараллеливания.

**Режимы счета с использованием двух уровней распараллеливания.** Парное использование методов распараллеливания определяет следующие режимы.

*Смешанное распараллеливание по областям и распараллеливание итераций циклов при помощи OpenMP.* В этом режиме счет задачи осуществляется так же, как в режиме распараллеливания по областям, но на одном вычислительном узле кластера запускается только один MPI-процесс. Каждый MPI-процесс является главной нитью OpenMP на своем вычислительном узле. Такой режим активно используется при проведении расчетов задач средних размеров (от нескольких сотен тысяч до миллиона ячеек).

*Смешанное мелкозернистое распараллеливание и распараллеливание итераций счетных циклов.* Режим работы аналогичен предыдущему, за исключением того, что распараллеливание счета в общей памяти осуществляется внутри параобласти. Этот режим может применяться для больших однообластных задач и из-за малого их количества используется редко.

*Смешанное по областям и мелкозернистое распараллеливание в модели распределенной памяти.* В этом режиме для каждой матобласти определяется управляющий процесс, который отвечает за ее расчет, а сама матобласть разделяется на некоторое количество параобластей. Такой режим распараллеливания может применяться для расчета многообластных задач, когда необходимо задействовать большое количество процессов. Он используется для вычислительных систем без общей памяти.

**Режим счета с трехуровневым распараллеливанием.** Последний режим параллельного счета заключается в использовании всех трех базовых режимов одновременно. Этот режим наиболее сложный с точки зрения реализации и управления, но в нем полностью решаются проблемы запуска на произвольном количестве ядер. Поэтому режим с трехуровневым распараллеливанием является основным для решения больших задач с несколькими миллионами точек. Работа в таком режиме аналогична работе в смешанном режиме с распараллеливанием по областям и мелкозернистым распараллеливанием, но на одном вычислительном узле запускается только один MPI-процесс, порождающий OpenMP-нити.

Необходимо отметить, что выделение указанных восьми режимов счета является отчасти

условным. В реальных расчетах могут возникнуть ситуации, когда, например, для небольшой области не будет использоваться мелкозернистое распараллеливание. В случае же совсем маленькой области, состоящей из нескольких сотен ячеек, может отключаться и OpenMP-распараллеливание, так как накладные расходы на организацию параллельной области выше выигрыша от расчета этой области в параллельном режиме.

Таким образом, хотя глобально, в масштабах задачи, может быть использован режим со всеми тремя уровнями распараллеливания, для конкретной матобласти уровни мелкозернистого и OpenMP-распараллеливания могут отключаться.

#### 4. Исследование эффективности распараллеливания

В качестве характеристик эффективности распараллеливания использовались следующие функции:  $S_p = \frac{t_1}{t_p}$  — ускорение счета;  $E_p = \frac{t_1}{pt_p} 100\%$  — эффективность распараллеливания, где  $t_1$  — время расчета на одном узле используемой параллельной ЭВМ,  $t_p$  — время счета на  $p$  узлах.

Для исследования эффективности распараллеливания рассматривалась задача об обжатии цилиндра в многообластной постановке. Геометрия задачи  $10 \times 1$  разбивалась на 10 матобластей размером  $1 \times 1$  (рис. 1, а). Между матобластями решается задача контактного взаимодействия, на боковой поверхности цилиндра задано постоянное по времени давление  $P_{гр} = 5$ . Остальные границы являются жесткими стенками. Вещество — идеальный газ с параметрами

$\gamma = 3$ ,  $\rho_0 = 1$ . В каждой области использовалась сетка диаграммы Вороного из 1 млн ячеек со случайной расстановкой центров. Фрагмент сетки приведен на рис. 1, б.

При расчете на одном вычислительном узле использовался только уровень распараллеливания на общей памяти. Время этого расчета принималось в качестве базового для оценки эффективности распараллеливания. При расчете на 10 узлах использовалось два уровня распараллеливания — по областям и с OpenMP. При большем количестве узлов каждая матобласть разбивалась на параобласти: при 20 узлах — на 2, при 40 — на 4 и т. д. Примеры декомпозиций одной области для разного количества процессов приведены на рис. 2 (см. также цветную вкладку).

Расчеты проводились в модели смешанной памяти — по одному MPI-процессу на каждом узле вычислительной системы, в каждом узле использовалось 8 процессорных ядер. Таблица содержит полученные значения ускорения и эффективности распараллеливания. Для каждого режима счета указано количество ядер, а в скобках  $(m \times n)$  число  $m$  означает количество MPI-процессов, а  $n$  — количество OpenMP-нитей.

Для тестирования корректности результатов в различных режимах счета также использовалась газодинамическая задача о полете несферической оболочки [10]. Наблюдается полное совпадение результатов, полученных как в разных режимах распараллеливания, так и в последовательном режиме. Результаты расчетов приведены в [7].

#### Заключение

Представлен метод трехуровневого распараллеливания, применяемый в методике ТИМ-2D. На первом уровне осуществляется рас-



Рис. 1. Вид геометрии с разделением на области (а) и фрагмент начальной сетки диаграммы Вороного (б)

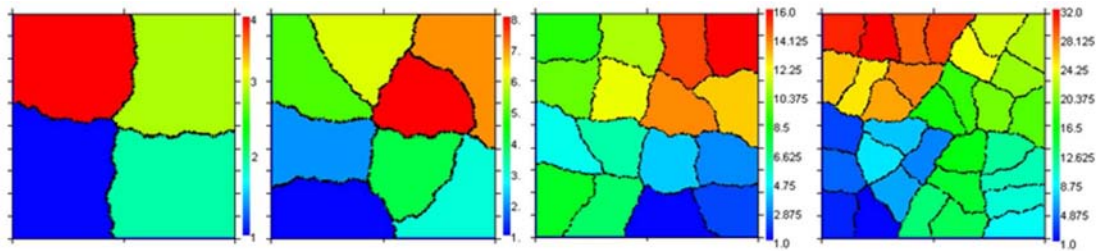


Рис. 2. Вид декомпозиций одной области для 40, 80, 160 и 320 процессов

**Результаты мелкозернистого распараллеливания**

Результирующий параметр	Open MP 8(1×8)	По областям + Open MP 80(10×8)	Трехуровневое распараллеливание			
			160(20×8)	240(30×8)	320(40×8)	400(50×8)
Ускорение	1	8,92	16,08	21,61	23,18	24,5
Эффективность, %	100	89	80	72	58	49

параллеливание счета по математическим областям. Второй уровень — распараллеливание счета внутри математической области по параобластям. На этих двух уровнях используется модель распределенной памяти и интерфейс MPI. На третьем уровне выполняется распараллеливание итераций счетных циклов в модели общей памяти с применением интерфейса OpenMP. Уровни могут использоваться как по отдельности, так и в различных сочетаниях при решении одной задачи. При этом результаты расчетов в последовательном и параллельном режимах полностью совпадают.

Применение метода трехуровневого распараллеливания позволяет гибко настраиваться на конфигурацию вычислительной системы и решать задачи с использованием нескольких сотен процессорных ядер с эффективностью не менее 50 %.

**Список литературы**

1. Соколов С. С., Воропинов А. А., Новиков И. Г. и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29—43.
2. Воропинов А. А. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества // Вестник

- ЮУрГУ. Сер. Математическое моделирование и программирование. 2009. № 37(170). Вып. 4. С. 40—50.
3. Воропинов А. А., Соколов С. С., Новиков И. Г. Распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике ТИМ-2D // Параллельные вычислительные технологии (ПаВТ'2008). Тр. межд. науч. конф. (Санкт-Петербург, 28 января — 1 февраля 2008 г.). Челябинск: Изд-во ЮУрГУ, 2008. С. 69—79.
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. Воропинов А. А. Алгоритмы мелкозернистого распараллеливания в методике ТИМ-2D // Вычислительные методы и программирование. 2009. Т. 10. С. 112—120.
6. Воропинов А. А., Новиков И. Г., Соколов С. С. Методы мелкозернистого распараллеливания в методике ТИМ-2D // Тез. докл. XVIII Всерос. конф. "Теоретические основы и конструирование численных алгоритмов решения задач математической физики", посвященной памяти К. И. Бабенко (Абрау-Дюрсо, 13—17 сентября 2010 г.). М.: ИПМ им. М. В. Келдыша, 2010. С. 19.

7. 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.
8. 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. Т. 8, № 1. С. 134–141.
9. Половникова Т. Н., Воропинов А. А. Опыт использования библиотек SCOTCH и MeTiS для декомпозиции неструктурированных сеток в методике ТИМ // Супервычисления и математическое моделирование. Тр. XII межд. семинара / Под ред. Р. М. Шагалиева. Саров: ФГУП РФЯЦ-ВНИИЭФ, 2011. С. 282–288.
10. Sofronov I. D., Rasskazova V. V., Nesterenko L. V. The use of nonregular nets for solving two-dimensional nonstationary problems in gas dynamics // Numerical Methods in Fluid Dynamics / Ed. by N. N. Yanenko, Ju. I. Shokin. Moscow: Mir Publishers, 1984. С. 82–121.

Статья поступила в редакцию 23.01.13.

