

Послойное разделение конечно-элементных сеток для мультиядерных архитектур *

А.К. Новиков, Н.К. Пиминова, С.П. Копысов

Институт механики УрО РАН

Рассматриваются алгоритмы разделения неструктурированной сетки, которые исключают конфликты при выполняемой в общей памяти параллельной сборке конечно-элементных векторов и матриц. Вводится отношение соседства между ячейками сетки, учитывающее связность конечно-элементных степеней свободы. С использованием отношения соседства определяются слои сетки, из которых формируются блоки ячеек сетки для параллельных процессов/нитей. Предложенные алгоритмы сравниваются с известными подходами на квазиструктурированных и существенно неструктурированных сетках по числу возможных конфликтов при параллельной конечно-элементной сборке и сбалансированности полученных блоков ячеек сетки.

Ключевые слова: разделение неструктурированных сеток, слои сетки, общая память, параллельные конечно-элементные схемы, мультиядерные процессоры

1. Введение

В параллельных вычислениях, основными целями разделения данных являются: равномерное распределение данных, минимизация коммуникаций, локализация доступа к данным [1, 2].

На обеспечение сбалансированности вычислительной нагрузки ориентированы геометрические алгоритмы разделения сеток, такие как рекурсивные координатная [3] и инерциальная бисекции [4], кривые заполняющие пространство [5].

Минимизацию коммуникаций обеспечивают алгоритмы разделения графов и гиперграфов [6] расчетной сетки (многоуровневое разделение k-way [7], спектральная бисекция [8], комбинированный графовый и геометрический подходы [9]) за счет минимизации границы получаемых подобластей сетки. Задача разделения графа расчетной сетки формулируется следующим образом: пусть задан граф расчетной сетки $G = (V, E)$ и $|V| = n$, тогда V разделяется на k подмножеств V_1, V_2, \dots, V_k , таких что $V_i \cap V_j = \emptyset$ при $i \neq j$, $|V_i| \approx n/k$ и $\bigcup_i V_i = V$, а число ребер из E , которые содержат вершины из разных подмножеств минимизируется. Под вершинами такого графа могут пониматься, как узлы, так и ячейки расчетной сетки.

При параллельных вычислениях на мультиядерных процессорах, происходит интенсивное использование общей памяти, которое накладывает дополнительные ограничения на разделения, связанные с локализацией и синхронизацией доступа к сеточным данным. При использовании общей памяти, узким местом параллельных алгоритмов численных методов являются операции сборки (суммирования). В методе конечных элементов такими операциями являются: сборка глобальных матриц и/или векторов (в поэлементных схемах) из локальных матриц и векторов соответственно, численное интегрирование локальных матриц и векторов [10]. В ходе операции сборки суммируются компоненты вектора или элементы матрицы, соответствующие вершинам общим для нескольких конечных элементов. В параллельном случае, одновременное суммирование компонент вектора или элементов матрицы из нескольких процессов/потоков приводит к конфликтам в общей памяти, а в результате — к ошибкам вычислений.

Для исключения таких ошибок, применяются различные подходы: атомарные операции

*Работа выполнена при частичной финансовой поддержке РФФИ (гранты 14-01-00055-а, 16-01-00129-а).

и критические секции; выделяются области памяти для хранения общих данных; изменяется схема сборки с поэлементной на узловую; применяются различные варианты раскрашивания сетки.

Раскрашивание (раскраска) графов традиционно применяется для переупорядочения неизвестных при параллельном решении систем алгебраических уравнений [11]. Применительно к ячейкам сетки, алгоритмы раскрашивания позволяют получать множества из несвязанных между собой ячеек сетки [12], но имеют NP-сложность, так как зависят не только от числа вершин графа, но и от их степеней (валентности) и выбора начальной вершины. Для того, чтобы получить равномерное распределение ячеек сетки по параллельным процессам/потокам после раскрашивания ячеек необходима процедура балансировки, а также переупорядочение сетки для локализации доступа к памяти.

Ранее, в [13] были предложены варианты послойного разделения расчетной сетки на подмножества ячеек, при котором в ходе операции суммирования параллельные вычислительные процессы не обращаются одновременно к конечным элементам, содержащим общие вершины. Варианты такого разделения основаны на разбиении расчетной сетки на слои ячеек (конечных элементов) толщиной в одну ячейку и последующем объединении этих слоев в блоки конечных элементов, назначенных одному потоку или ядру процессора. Слои из конечных элементов определяются, используя введенное отношения соседства ячеек. Алгоритм формирования слоёв использует связность ячеек по узлам и имеет NP-сложность.

В данной работе исследуются алгоритмы разделения неструктурированных сеток с ячейками произвольной формы на слои из ячеек и варианты объединения полученных слоев в блоки конечных элементов. Построенные разделения сетки сравниваются с известными алгоритмами на квазиструктурированных и существенно неструктурированных сетках по числу возможных конфликтов при сборке и сбалансированности.

2. Послойное разделение конечно-элементных сеток

Конечно-элементные сетки относятся к классу неструктурированных сеток. По определению неструктурированной сетки отношение соседства между узлами или ячейками такой сетки не определяется приращением номера сеточного объекта (узла или ячейки). Для конечно-элементных схем МКЭ в [13] было введено следующее отношение соседства:

$$\omega_k \in Adj(\omega_j), \text{ если } V(\omega_j) \cap V(\omega_k) \neq \emptyset, \quad (1)$$

где ω_j, ω_k — ячейки сетки Ω ; $Adj(\cdot)$ — оператор смежности; $V(\omega_j)$ и $V(\omega_k)$ — соответствующие ячейкам ω_j и ω_k подмножества узлов сетки. Отношение (1) более точно отражает зависимости по данным в операции поэлементной композиции (сборки), чем дуальный граф конечно-элементной сетки или граф, вершинами которого являются узлы сетки.

Важно отметить, что при таком отношении соседства у ячейки будет большое число соседей m_e , даже для квазиструктурированной трехмерной сетки, как показано на рисунке 1.

Таким образом, алгоритм послойного разделения конечно-элементной сетки состоит из двух шагов: 1) формирования слоев из ячеек сетки; 2) объединения слоев сетки в подобласти. Остановимся на этих шагах подробнее.

2.1. Формирование слоев сетки на основе отношения соседства

При формировании слоев сетки на основе (1) используется принадлежность узлов конечным элементам. Как правило, неструктурированные сетки хранятся таким образом, что ячейка содержит узел сетки, а не узел принадлежит ячейке. Принадлежность узлов конечным элементам определяется в двух циклах по ячейкам сетки за $O(2 \cdot m \cdot n_e)$ операций, где m — число конечных элементов; n_e — число узлов в конечном элементе.

Алгоритм формирования слоев сетки [13] представим в следующем виде:

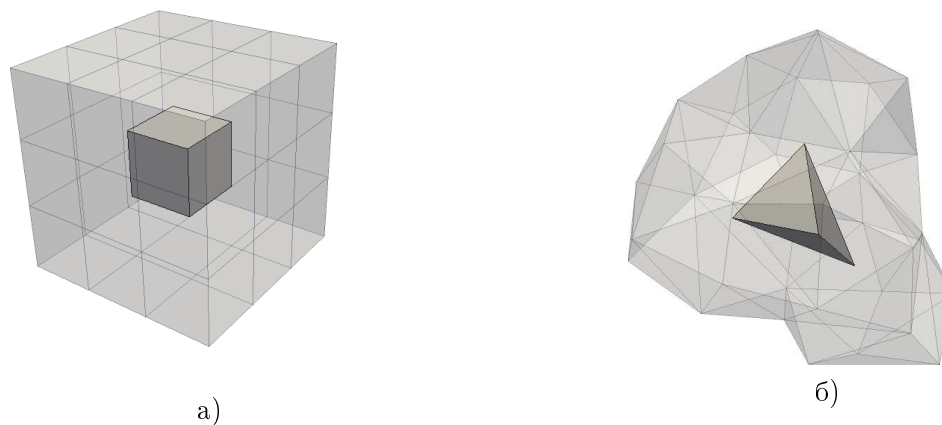


Рис. 1: Соседние ячейки: а) квазиструктурированная сетка, $m_e = 26$; б) существенно неструктурированная сетка, $m_e = 119$.

1. Задается начальный слой $s_1 = \{\omega_j^{(i)}\}, j = 1, 2, \dots, m_1$, здесь m_1 — число ячеек в s_1 .
2. Слои $s_i, i = 2, 3, \dots, n_s$ определяются из выражения

$$s_i = \begin{cases} Adj(s_{i-1}), & i = 2; \\ Adj(s_{i-1}) \setminus s_{i-2}, & i > 2, \end{cases}$$

где $s_i = \{\omega_j^{(i)}\}$ — множество ячеек (конечных элементов) в слое i ; $\omega_j^{(i)}$ — ячейка с номером j в слое i ; $Adj(s_i) = \bigcup_{j=1}^{m_i} Adj(\omega_j^{(i)})$ — множество ячеек соседних с s_i ; m_i — число ячеек в s_i . Этот алгоритм также применяется к выделению подмножеств несвязанных ячеек внутри слоя и к поиску разрывных слоёв.

Если для поэлементных схем МКЭ наличие разрывных слоёв не существенно, то для метода дополнения Шура [14] необходимо обеспечить, не только неразрывность слоев, но и соседство по граням ячеек. Поиск ячеек, имеющих одну общую вершину, и определение разрывных подслоев осуществляются на основе соседства по граням ячеек, при определении которого используется построенное ранее соседство по узлам сетки (1). Для каждой ячейки рассматриваются соседи по вершине и исключаются те из них, с которыми число общих вершин не равно числу вершин на одной грани n_f (трем — для четырехгранной ячейки и четырем — для шестигранной).

Определение разрывных слоев с точки зрения соседства по грани происходит следующим образом. Берется произвольная ячейка в текущем слое s_i и начинается поиск в ширину по соседним ячейкам. По окончании поиска проверяется все ли ячейки попали в множество достижимых вершин. Если нет, то все попавшие ячейки в это множество ячейки помещаются в подслой s_j^* и начинается новый поиск. Таким образом формируется множество подслоев $\{s_j^*\}$ для разрывного слоя.

Поиск висячих ячеек и их присоединение к слою выполняется исходя из информации о разрывности слоя. Если текущий слой s_i — разрывный, то есть число подслоев составляющих его более единицы, тогда определяется его максимальный по числу ячеек подслой. Максимальный подслой полагается текущим, то есть s_i , а подслои с меньшим числом ячеек объединяются с предыдущим слоем s_{i-1} .

Примеры полученных слоев для квазиструктурированной (четвертая часть мембраны) и существенно неструктурированной сетки (рамная конструкция) представлены на рисунках 2, а и 2, б соответственно.



Рис. 2: Слои для неструктурированных сеток: а) из шестигранных ячеек, $m = 31744$, $n_s = 64$; б) из четырехгранных ячеек, $m = 485843$, $n_s = 137$

В таблице 1 приведены максимальные m_{max} и минимальные m_{min} числа ячеек в слоях $s_i, i = 1, 2, \dots, n_s$ и число слоев n_s для квазиструктурированной сетки, подобной представленной на рисунке 2,а, и существенно неструктурированной сетки (рисунок 2,б). Вариант (x_{min}, y, z) означает, что хотя бы одна вершина в ячейке, принадлежащей слою s_1 , имеет абсциссу равную x_{min} . Из представленных в таблице результатов видно, что выбор начального слоя существенно влияет на сбалансированность разделения сетки по слоям.

Таблица 1: Характеристики слоёв в зависимости от выбора начального слоя s_1

Параметры	Вариант		
	(x_{min}, y, z)	(x, y_{min}, z)	(x, y, z_{min})
Квазиструктурированная сетка, $m = 507904$			
m_{min}	1056	1568	15872
m_{max}	6112	5120	15872
n_s	71	28	32
Существенно неструктурированная сетка, $m = 485843$			
m_{min}	9	17	14
m_{max}	9202	46635	12457
n_s	137	47	140

В случае квазиструктурированной сетки (вариант (x, y, z_{min})) получены тридцать два идеально сбалансированных слоя: $m_{max} = m_{min} = 15872$ ячеек. Для существенно неструктурированной сетки (вариант (x, y_{min}, z)) максимальное число ячеек в слое — 46635 при $m = 485843$, а минимальное — 17 ячеек, число слоев $n_s = 47$.

В соответствии с заданным отношением соседства, число слоев n_s ограничивается сверху числом ячеек в направлении диаметра расчетной области или максимальным числом ячеек по координатным направлениям. Например, при указанном выше выборе начального слоя s_1 и отношении соседства (1), для сетки $10 \times 10 \times 100$ шестигранных ячеек $n_s \leq 100$, а в случае неструктурированной сетки с тем же числом тетраэдральных ячеек — $n_s \leq 50$.

Далее, полученные слои и их порядок используются при формировании подобластей/блоков расчетной сетки.

2.2. Объединение слоёв неструктурированной сетки

Рассматривалось несколько вариантов формирования под областей $\Omega_i, i = 1, 2, \dots, n_\Omega$ из слоёв $s_j, j = 1, 2, \dots, n_s$ неструктурированной сетки Ω : блочный, по чётности номеров слоёв и их модификации.

Блочный вариант («block») имеет следующий вид:

1. Объединение слоёв $\Omega = \bigcup_{i=1}^{n_s} s_i$.
2. Разделение Ω на под области Ω_i , такие что $|\Omega_i| \approx m/n_\Omega$, $i = 1, 2, \dots, n_\Omega$, где n_Ω — число подмножеств ячеек или под областей сетки (рисунок 3).

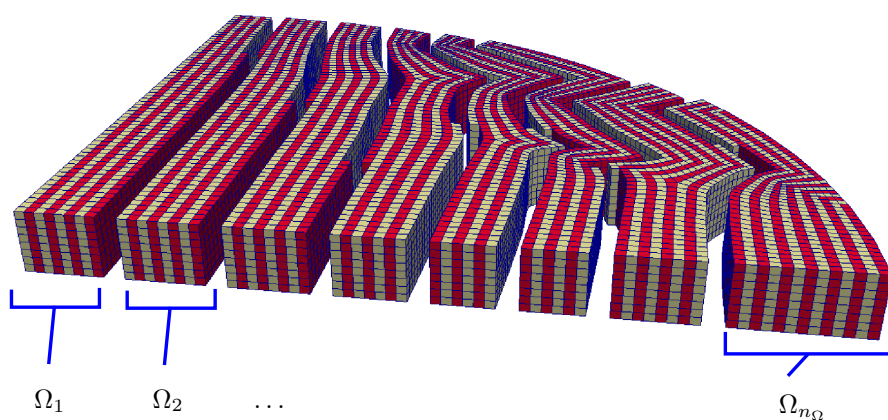


Рис. 3: Под области, полученные при блочном варианте построения

Как видно из рисунка 3, в блочном варианте слои неструктурированной сетки разделены на части, принадлежащие двум соседним под областям. Например, под область Ω_1 включает семь слоёв s_1, s_2, \dots, s_7 и большую часть слоя s_8 . Благодаря делению слоёв получаются равные по числу ячеек под области и соответственно достигается сбалансированность нагрузки на параллельные вычислительные процессы в дальнейшем. Вместе с тем, выделение частей слоёв требует контроля за отношением соседства. При блочном разделении необходимо не менее двух слоёв в под области для исключения ошибок вычислений, связанных с конфликтами при параллельном доступе к сеточным данным. В противном случае необходимо задавать порядок ячеек внутри слоя, например, по координатным направлениям.

Объединение слоёв по чётности их номеров происходит следующим образом. Сначала в под области $\Omega_i, i = 1, 2, \dots, n_\Omega$ объединяются все слои с нечётными номерами, а затем с чётными (рисунок 4,а). В под область Ω_i помещаются слои (нечётные или чётные) с номерами $i, i + n_\Omega, i + 2n_\Omega, \dots$, например, под область Ω_1 на рисунке 4,б.

Оценим полученные разделения сеток с этой точки зрения сбалансированного распределения вычислительной нагрузки. В таблицах 2,3 представлены максимальное m_{max} и минимальное m_{min} число ячеек в под областях, полученных при разделении Ω на n_Ω под областей. Для сравнения приводятся результаты по многоуровневому алгоритму разделения графов k-way из пакета METIS [7].

Объединение слоёв по чётности представлено в виде двух вариантов: 1) «even/odd» — когда чётные («even») слои и нечётные («odd») слои сетки будут находиться при конечно-

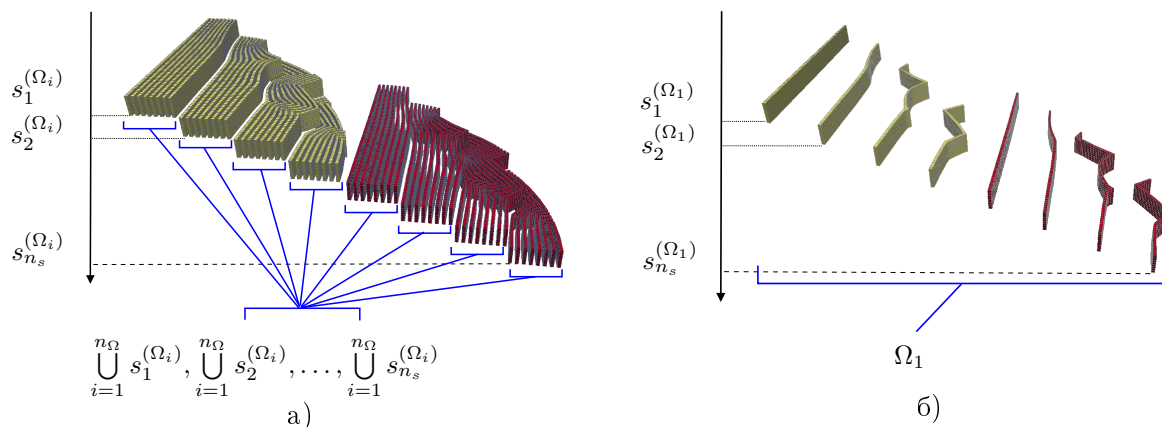


Рис. 4: Подмножества ячеек, сформированные по чётности номеров слоёв и распределение слоёв по подобластям Ω_i : а) слои, принадлежащие восьми подобластям ($n_\Omega = 8$), ячейки которых будут одновременно участвовать в параллельных вычислениях; б) последовательность слоёв в подобласти Ω_1

элементных вычислениях в разных параллельных областях OpenMP; 2) «even+odd» — чётные и нечётные слои будут находиться в одной параллельной области OpenMP (см. рисунок 4,б). Как видно из приведенных результатов (см. таблицы 2,3), объединение целых слоёв по чётности приводит к наименее сбалансированному распределению ячеек по подобластям сетки.

Таблица 2: Сбалансированность подобластей квазиструктурированной сетки

n_Ω	Варианты разделения, m_{max}/m_{min}				
	METIS	«block»	«even/odd»		«even+odd»
			«even»	«odd»	
8	63503/	63488/	31744/	31744/	63488
	42964	63488	31744	31744	63488
16	32343/	31744/	17824/	17888/	35712/
	31507	31744	13920	13856	27776
32	16299/	15872	10144/	10208/	20352/
	15408	15872	6144	6144	12288
60	8692/	8465	6048/	6112	12160/
	1876	8465	1632	1568	3200

Основываясь на полученных результатах по сбалансированности, для объединения чётных или нечётных слоёв («even/odd») был применен подход, обеспечивающий сбалансированное распределению ячеек по подобластям сетки. В этом случае, слои s_1, s_3, s_5, \dots объединялись в подобласть $\Omega^{(even)}$, а слои s_2, s_4, s_6, \dots — в подобласть $\Omega^{(odd)}$. Далее $\Omega^{(even)}$ разделялась на n_Ω подобластей $\Omega_i^{(even)}$, таких что $|\Omega_i^{(even)}| \approx |\Omega^{(even)}|/n_\Omega$ и соответственно $\Omega^{(odd)}$ на n_Ω подобластей $\Omega_i^{(odd)}$, для которых $|\Omega_i^{(odd)}| \approx |\Omega^{(odd)}|/n_\Omega$.

Таблица 3: Сбалансированность подобластей существенно неструктурированной сетки

n_{Ω}	Варианты разделения, m_{max}/m_{min}				
	METIS	«block»	«even/odd»		«even+odd»
			«even»	«odd»	
8	61163/	60731/	34171/	33838/	68009/
	60428	60726	27448	17842	55398
16	31180/	30366/	17803/	17781/	35419/
	29479	30353	11506	11477	23087
32	15638/	15183/	13334/	12965/	25729/
	14739	15170	4469	4209	8678
60	8340/	8098/	9202/	8839/	17605/
	7861	8061	1877	1982	3859

На рисунках 5,6 представлены полученные подобласти сетки для $n_{\Omega} = 8$. Как видно из этих рисунков, послойное разделение в отличие от k-way позволяет упорядочить связи подобластей (порядок цветов на рисунке 5,а и рисунке 6).

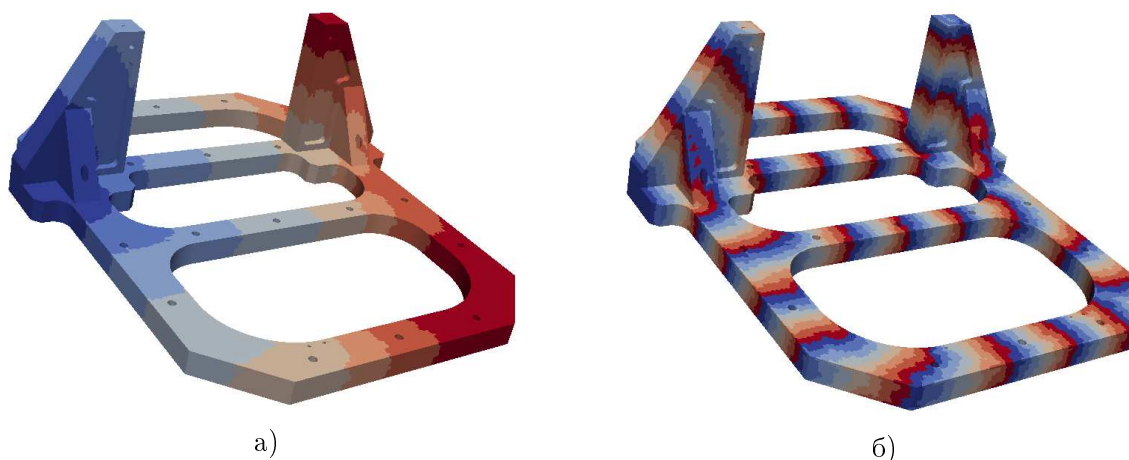


Рис. 5: Разделение существенно неструктурированной сетки на восемь подобластей, $n_{\Omega} = 8$: а) при блочном объединении; б) при объединении по четности.

Кроме того, в послойных вариантах узлы могут принадлежать не более чем двум подобластям сетки, что не гарантирует, например, многоуровневый алгоритм из библиотеки METIS при больших значениях n_{Ω} на адаптивных неструктурированных сетках. Для многосвязной области разрывные подобласти были получены, как при послойном разделении сетки, так и при многоуровневом. Как отмечено ранее, в параллельных поэлементных схемах метода конечных элементов разрывные подобласти сетки не влияют на вычислительные устойчивости алгоритмов, но замедляют доступ к памяти. Во многих алгоритмах необходимо или объединить части разрывных подобластей с сохранением общего числа подобластей, или рассматривать эти части, как независимые подобласти с увеличением их общего числа. Предложенное упорядочение слоёв и подобластей упрощает процедуру объединения с

соседними подобластями, в которой участвует меньшее число соседних подобластей, чем в [9].

В таблице 4 приведены данные о возможных конфликтах при сборке вектора для нескольких вариантов разделения сетки. В качестве оценки возможных конфликтов при параллельной реализации операции конечно-элементной композиции рассмотрено число ячеек, к узлам которых происходит обращение из разных подобластей сетки.

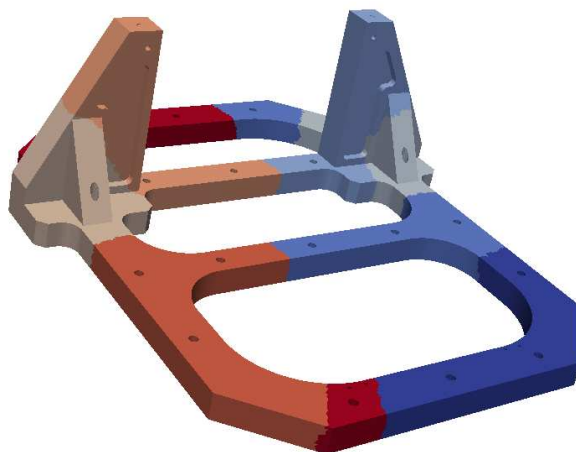


Рис. 6: Разделение, полученное алгоритмом k-way (пакет METIS), $n_{\Omega} = 8$.

В варианте «по построению» используется порядок ячеек, полученный при построении сетки в сеточном генераторе. Сетка делится на заданное число подобластей n_{Ω} с примерно равным числом ячеек в подобластях $\Omega_i \approx m/n_{\Omega}$.

Таблица 4: Число возможных конфликтов при сборке вектора

n_{Ω}	Варианты разделения					
	По построению	METIS	«block»	«even/odd»		«even+odd»
				«even»	«odd»	
Квазиструктурированная сетка, $m = 507904$						
8	0	0	0	0	0	0
16	0	384/0	0	0	0	0
32	0	24/0	0	0	0	0
60	0	60/2	0	0	0	0
Существенно неструктурированная сетка, $m = 485843$						
8	284/88	7/2	0	0	0	0
16	623/198	13/4	0	0	0	0
32	1144/352	60/25	0	0	0	0
60	2425/815	169/71	167/40	0	0	4/3

В числителе приведено число ячеек, к узлам которых происходит обращение из разных подобластей сетки, в знаменателе — число ячеек, в которых обращение из разных подобластей сетки происходит к узлам с одинаковыми локальными (внутри ячейки) номерами. Оценка, представленная в знаменателе, более строгая, исходит из того, что параллельные процессы (потоки) начинаются одновременно, а доступ к каждой ячейке данных происходит за равный отрезок времени из любого параллельного процесса. Если данная оценка отлична от нуля (таблица 4), то при параллельной сборке конечно-элементных векторов происходят ошибки вычислений.

Ошибки вычислений, связанные с конфликтами в общей памяти могут не наблюдаться, вследствие упорядочения ячеек, полученного при построении сетки, как в случае квазиструктурированной сетки (таблица 4, второй столбец).

При разделении существенно неструктурированной сетки по чётности слоёв («even+odd») возникновение конфликтов связано с несбалансированностью слоёв по числу ячеек. Для таких сеток доступ к ячейкам сетки и связанными с ними данными осуществляется в виде последовательности из двух параллельных подобластей OpenMP («even/odd»), отдельно для чётных («even») и нечётных («odd») слоёв.

При блочном разделении конфликты являются следствием разрыва слоёв для обеспечения сбалансированного разделения и отсутствия упорядочения ячеек в слоях. Объединение целых слоев устраняет эти конфликты, но приводит к существенно менее сбалансированному разделению.

Остановимся на затратах предложенных алгоритмов послойного разделения неструктурированной сетки. Наиболее затратной частью является шаг формирования слоёв, который выполнялся для существенно неструктурированной сетки ($m = 485843$) — 0.8–0.9 сек., для квазиструктурированной сетки ($m = 507904$) — 0.39–0.41 сек. Время объединения слоев существенно меньше, так объединение по четности выполнялось за 0.005–0.006 сек., а на блочное объединение потребовалось 0.003–0.004 сек. Сравнение с многоуровневым алгоритмом разделения из пакета METIS показало: 1) предложенные алгоритмы выполняются примерно в три раза быстрее; 2) в отличие от многоуровневого алгоритма время формирования блоков практически не зависит от числа формируемых блоков; 3) упорядочение слоёв, как следствие, подобластей сетки позволяет упорядочить дальнейшие коммуникации между подобластями сетки на этапе её разделения.

Применение послойного разделения позволило достичь 68% эффективности при параллельной конечно-элементной сборке векторов, выполненной на 61-ядерном процессоре Xeon Phi 7110X, 75% эффективности на двух 4-ядерных Xeon E5-2609. Благодаря высокой эффективности сборки вектора-результата удалось получить близкое к линейному ускорение при вычислении матрично-векторного произведения в поэлементной схеме МКЭ на указанных мультиядерных процессорах и неструктурированных сетках.

В данной работе наиболее полно представлен подход, адаптирующий конечно-элементные алгоритмы к парадигме потоковой обработки, характерной для гибридных архитектур на основе массивно-параллельных ускорителей. Подход основывается на разделении неструктурированных сеток, обеспечивающем исключение одновременного доступа к общим данным и сбалансированное распределение вычислительной нагрузки.

Литература

1. Hendrickson B., Devine K. Dynamic load balancing in computational mechanics // Comput. Methods Appl. Mech. Engrg. 2000. Vol. 184. P. 485–500.
2. Копысов С.П., Новиков А.К. Параллельные алгоритмы адаптивного перестроения и разделения неструктурированных сеток // Математическое моделирование. 2002. Т. 14. № 9. С. 91–96.

3. Berger M.J., Bokhari S.H. A partitioning strategy for nonuniform problems on multiprocessors. // IEEE Trans Comput. 1987. Vol. 36. No. 5. P. 570–580.
 4. Farhat C., Lesoinne M. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. // Int. J. Numer. Methods Eng. 1993. Vol. 5. No. 36. P. 745–764.
 5. Pilkington J., Baden S. Partitioning with space-filling curves. // Tech. Rep. CS94-349. Dept. of Computer Science and Engineering, Univ. of California. San Diego. 1994. 49 p.
 6. Fortmeier O., Bucker H.M., Fagginger Auer B.O., Bisseling R.H. A new metric enabling an exact hypergraph model for the communication volume in distributed-memory parallel applications // Parallel Computing. 2013. Vol. 39. P. 319–335.
 7. Karypis G. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 5.1. University of Minnesota, Minneapolis. 2013. 35 p.
 8. Hendrickson B., Leland R. Multidimensional Spectral Load Balancing. // Tech. Rep. SAND93-0074. Sandia National Labs. Albuquerque. 1993. P. 1–13.
 9. Головченко Е.Н., Якобовский М.В. Пакет параллельной декомпозиции больших сеток GridSpiderPar. // Суперкомпьютерные дни в России: Труды межд. конф. (28–29 сен. 2015. Москва). – М.: Изд-во МГУ. 2015. С. 303–315.
 10. Копысов С.П., Кузьмин И.М., Недожогин Н.С., Новиков А.К., Рычков В.Н., Сагдеева Ю.А., Тонков Л.Е. Параллельная реализация конечно-элементных алгоритмов на графических ускорителях в программном комплексе FEStudio // Компьютерные исследования и моделирование. 2014. Т. 6. № 1. С. 79–97.
 11. Пирова А.Ю., Мееров И.Б., Козинев Е.А., Лебедев С.А. Параллельный алгоритм многоуровневого метода вложенных сечений для вычислительных систем с общей памятью // Вычислительные методы и программирование. 2015. Т. 16. С. 407–420.
 12. Komatitsch D., Micha D., Erlebacher G. Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA // J. Parallel Distr. Comp. 2009. Vol. 69. No. 5. P. 451–460.
 13. Копысов С.П., Новиков А.К., Пиминова Н.К. Параллельная композиция в поэлементной схеме метода конечных элементов. Параллельные вычислительные технологии (ПаВТ'2016): Труды межд. науч. конф. (Архангельск, 28 марта – 1 апреля 2016.). Челябинск: Изд. центр ЮУрГУ. 2016. С. 555–560.
 14. Kopysov S.P., Kuzmin I.M., Nedozhogin N.S., Novikov A.K., Sagdeeva Y.A. Hybrid Multi-GPU Solver Based on Schur Complement Method // Lecture Notes in Computer Science. 2013. Vol. 7979. P. 65–79.
-

Layer-by-layer partitioning of finite-element meshes for multi-core architectures

A.K. Novikov, N.K. Piminova, S.P. Kopysov

Institute of Mechanics UB RAS

Algorithms of unstructured mesh partitioning are considered. Its exclude conflicts in the shared memmory for parallel assembling of finite elements. Adjacency ratio between mesh cells is introduced by taking connectivity of finite element degrees of freedom. Mesh layers are defined by using adjency ratio. Its are formed into cells blocks for parallel processes/threads. The built mesh partitions are compared with the known algorithms in case of quasistructured and essential unstructured meshes by the number of potential conflicts and load balance.

Keywords: unstructured meshes, layers of mesh, shared memory, parallel finite element schemes, multi-core processors

References

1. Hendrickson B., Devine K. Dynamic load balancing in computational mechanics // Comput. Methods Appl. Mech. Engrg. 2000. Vol. 184. P. 485–500.
2. Kopysov S.P., Novikov A.K. Parallel'nye algoritmy adaptivnogo perestroeniya i razdeleniya nestrukturirovannykh setok. [Parallel algorithms of adaptive refinement and partitioning of unstructured grids]. Matematicheskoe modelirovanie. [Mathematical Models and Computer Simulations]. 2002. Vol. 14. No. 9. P. 91–96. (Копысов С.П., Новиков А.К. Параллельные алгоритмы адаптивного перестроения и разделения неструктурированных сеток // Математическое моделирование. 2002. Т. 14. № 9. С. 91–96.)
3. Berger M.J., Bokhari S.H. A partitioning strategy for nonuniform problems on multiprocessors. // IEEE Trans Comput. 1987. Vol. 36. No. 5. P. 570–580.
4. Farhat C., Lesoinne M. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. // Int. J. Numer. Methods Eng. 1993. Vol. 5. No. 36. P. 745–764.
5. Pilkington J., Baden S. Partitioning with space-filling curves. // Tech. Rep. CS94-349. Dept. of Computer Science and Engineering, Univ. of California. San Diego. 1994. 49 p.
6. Fortmeier O., Bucker H.M., Fagginger Auer B.O., Bisseling R.H. A new metric enabling an exact hypergraph model for the communication volume in distributed-memory parallel applications // Parallel Computing. 2013. Vol. 39. P. 319–335.
7. Karypis G. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 5.1. University of Minnesota, Minneapolis. 2013. 35 p.
8. Hendrickson B., Leland R. Multidimensional Spectral Load Balancing. // Tech. Rep. SAND93-0074. Sandia National Labs. Albuquerque. 1993. P. 1–13.
9. Golovchenko E.N., Yakobovskiy M.V. Paket parallel'noy dekompozitsii bol'shikh setok GridSpiderPar [Parallel partitioning tool GridSpiderPar for large mesh decomposition]. Superkomp'yuternye dni v Rossii: Trudy mezhd. konf. (28–29 sen. 2015. Moskva.) [Russian Supercomputing Days: Proceedings of the int. conf. (Sept. 28–29, 2015, Moscow, Russia)]. Moscow. Moscow State University. 2015. P. 303–315. (Головченко Е.Н., Якобовский М.В.)

Пакет параллельной декомпозиции больших сеток GridSpiderPar. // Суперкомпьютерные дни в России: Труды межд. конф. (28–29 сен. 2015. Москва). – М.: Изд-во МГУ. 2015. С. 303–315.)

10. Kopysov S. P., Kuz'min I. M., Nedozhogin N. S., Novikov A. K., Rychkov V. N., Sagdeeva Yu. A., Tonkov L. E. Parallel'naya realizatsiya konechno-elementnykh algoritmov na graficheskikh uskoritelyakh v programmnom komplekse FESstudio [Parallel implementation of a finite-element algorithms on a graphics accelerator in the software package FESstudio]. Komp'yuternye issledovaniya i modelirovaniye [Computer Research and Modeling]. 2014. Vol. 6. No 1. P. 79–97. (Копысов С.П., Кузьмин И.М., Недожогин Н.С., Новиков А.К., Рычков В.Н., Сагдеева Ю.А., Тонков Л.Е. Параллельная реализация конечно-элементных алгоритмов на графических ускорителях в программном комплексе FESstudio // Компьютерные исследования и моделирование. 2014. Т. 6. № 1. С. 79–97.)
11. Pirova A.Yu., Meerov I.B., Kozinov E.A., Lebedev S.A Parallel'nyy algoritm mnogourovnevoogo metoda vlozhennykh secheniy dlya vychislitel'nykh sistem s obshchey pamyat'yu [A parallel multilevel nested dissection algorithm for shared-memory computing systems]. Vychislitel'nye Metody i Programmirovaniye. [Numerical Methods and Programming]. 2015. Vol. 16. P. 407–420. (Пирова А.Ю., Мееров И.Б., Козинев Е.А., Лебедев С.А. Параллельный алгоритм многоуровневого метода вложенных сечений для вычислительных систем с общей памятью // Вычислительные методы и программирование. 2015. Т. 16. С. 407–420.)
12. Komatitsch D., Micha D., Erlebacher G. Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA // J. Parallel Distr. Comp. 2009. Vol. 69. No. 5. P. 451–460.
13. Kopysov S.P., Novikov A.K., Piminova N.K. Parallel'naya kompozitsiya v poelementnoy skheme metoda konechnykh elementov. Parallel'nye vychislitel'nye tekhnologii (PaVT'2016): Trudy mezhd. nauch. konf. (Arkhangel'sk, 28 marta – 1 aprelya 2016.) [Parallel Computational Technologies (PCT'2016): Proceedings of the International Scientific Conference (Arkhangelsk, Russia, March, 28 – April, 1, 2016)]. Chelyabinsk: Publishing of the South Ural State University, 2016. P. 555-560. (Копысов С.П., Новиков А.К., Пиминова Н.К. Параллельная композиция в поэлементной схеме метода конечных элементов. Параллельные вычислительные технологии (ПаВТ'2016): Труды межд. науч. конф. (Архангельск, 28 марта – 1 апреля 2016.). Челябинск: Изд. центр ЮУрГУ. 2016. С. 555–560.)
14. Kopysov S.P., Kuzmin I.M., Nedozhogin N.S., Novikov A.K., Sagdeeva Y.A. Hybrid Multi-GPU Solver Based on Schur Complement Method // Lecture Notes in Computer Science. 2013. Vol. 7979. P. 65–79.