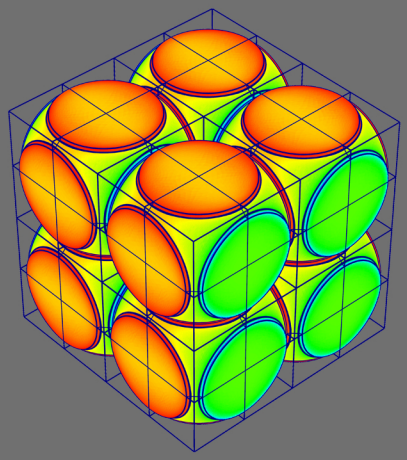
ISSN 2309-3684



Математическое моделирование и численные методы

Мартыненко С. И. О построении параллельных многосеточных алгоритмов. Математическое моделирование и численные методы, 2015, №2 (6), с. 105-120

Источник: https://mmcm.bmstu.ru/articles/51/

Параметры загрузки:
IP: 216.73.216.68
28.11.2026 05:54:02

УДК 519.63

О построении параллельных многосеточных алгоритмов

© С.И. Мартыненко

ФГУП «Центральный институт авиационного моторостроения
имени П.И. Баранова», Москва, 111116, Россия

Рассмотрены основные направления развития параллельных классических многосеточных алгоритмов и их характерные недостатки. На примере универсальной многосеточной технологии показана возможность эффективного распараллеливания сглаживающих итераций на уровнях с грубыми сетками; многосеточная структура использована для построения гибридного многосеточного метода. Приведены оценки ускорения и эффективности различных параллельных многосеточных алгоритмов, а также результаты вычислительных экспериментов.

***Ключевые слова**: многосеточные методы, краевые задачи, параллелизм.*

Введение. Краевые задачи для дифференциальных уравнений в частных производных являются составной частью многих математических моделей, поэтому проблема создания высокоэффективных методов численного решения краевых задач еще долго будет актуальна, несмотря на гигантский прогресс вычислительной техники [1, 2].

Многосеточные методы, берущие начало с работ выдающихся отечественных математиков Р.П. Федоренко и Н.С. Бахвалова, в настоящее время стали доминирующими алгоритмами для численного решения краевых задач и используются в подавляющем большинстве современных программных продуктов. Однако у классических многосеточных методов (КММ) есть определенные недостатки, связанные с необходимостью оптимального выбора отдельных компонент для конкретного приложения. Кроме того, до сих пор не удавалось построить высокоэффективный параллельный многосеточный алгоритм [3].

Инженерная практика зачастую связана с необходимостью рассмотрения различных вариантов конструкции или схем организации рабочего процесса. Для решения подобных задач была разработана универсальная многосеточная технология (УМТ) — вариант геометрических многосеточных методов с минимальным количеством проблемно-зависимых компонент [4−8]. Одной из особенностей УМТ является используемая иерархия сеток — так называемая многосеточная структура. Несмотря на то что УМТ несколько уступает КММ в вычислительной эффективности, общие затраты на адаптацию алгоритма к решаемой задаче и решение сеточных уравнений у УМТ оказываются ниже для отдельных приложений, в то время как КММ остаются лучшими методами решения крупных серий однотипных

задач. Поэтому УМТ и КММ, скорее, дополняют друг друга, нежели являются конкурентами.

В различных проблемных областях существует ряд задач, решение которых требует чрезмерного времени даже с привлечением самых совершенных процессоров и численных методов, поэтому неуклонно возрастает интерес к параллельным алгоритмам. Практически сразу были отмечены характерные трудности построения параллельного варианта КММ, связанные со снижением эффективности параллелизма на грубых сетках из-за уменьшения отношения объема вычислительной работы к объему пересылаемых данных. Выработано четыре направления повышения эффективности параллельных КММ, однако преодолеть «барьер грубых сеток» не удалось. Основная проблема состоит в равномерном распределении вычислительной работы между процессорами параллельного компьютера и минимизации числа обменов данными.

Совершенно иначе с точки зрения параллельных вычислений выглядит УМТ. Если внимательно присмотреться, то УМТ является односеточным алгоритмом, который позволяет использовать основополагающую идею Р.П. Федоренко для ускорения сходимости отдельных итерационных методов (сглаживателей). Фактически УМТ позволяет уменьшить объем вычислительной работы, требуемой для решения широкого класса линейных (нелинейных) краевых задач методом Зейделя с блочным упорядочиванием неизвестных, до $O(N_0 \ln N_0)$ арифметических операций, где N_0 количество узлов сетки. Причем столь радикальное уменьшение вычислений по сравнению с традиционным односеточным вариантом метода Зейделя не требует применения каких-либо проблемно-зависимых компонент в конструкции УМТ [8]. Естественно, что отсутствие грубых сеток в УМТ сразу сняло проблему снижения эффективности параллелизма. Данное обстоятельство было отмечено в работе [9], а соответствующие оценки ускорения и эффективности параллельной УМТ приведены в [10, 11].

Целью данной работы является получение оценок уменьшения времени решения краевых задач в сравнении не только с последовательной УМТ, как в [9−11], но и с последовательным *V*-циклом. Кроме того, предложен комбинированный подход к построению параллельных КММ, основанный на использовании многосеточной структуры для сглаживания на грубых сетках.

Для вычислительных экспериментов применена система с симметричной мультипроцессорной обработкой, или SMP-система (Symmetric MultiProcessors). В этой архитектуре все процессоры имеют равноправный доступ ко всему пространству оперативной памяти и ввода-вывода, поэтому SMP-архитектура называется симметричной. Ее интерфейсы доступа к пространству ввода-вывода и опе-

ративной памяти, система управления кэш-памятью, системное программное обеспечение и т. п. построены таким образом, чтобы обеспечить согласованный доступ к разделяемым ресурсам. Память обеспечивает передачу сообщений между процессорами, при этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти. Последнее обстоятельство позволяет очень эффективно обмениваться данными с другими вычислительными устройствами [12−14].

Параллельные классические многосеточные алгоритмы. Традиционно попытки построить параллельный многосеточный алгоритм основаны на распараллеливании отдельных компонент (операторы переходов и сглаживающая процедура). Положим, что в качестве сглаживающей процедуры выбран метод Якоби с блочным упорядочиванием неизвестных. Для реализации метода Якоби исходную сетку декомпозируют с «перехлестом» на блоки, с каждым блоком сетки ассоциирован блок неизвестных. Далее с помощью некоторого (внутреннего) итерационного метода независимо вычисляют новые значения неизвестных, образующих блок. Как правило, сетку декомпозируют на блоки приблизительно одинаковой величины, чтобы добиться сбалансированной загрузки процессоров, после чего осуществляют обмен данными между процессорами, т. е. обмен значениями искомых сеточных функций в узлах на границах блоков. Принципиальным и неустранимым недостатком данного подхода является уменьшение отношения объема вычислительной работы, выполненной каждым процессором, к объему пересылаемых данных при уменьшении количества узлов сетки. Более того, возможен случай, когда количество узлов самых грубых сеток будет меньше количества процессоров, т. е. часть процессоров будет простаивать при выполнении сглаживающих итераций.

Сначала рассмотрим влияние потерь, связанных с обменом данными и возможным простоем процессоров, на эффективность параллельного многосеточного алгоритма. Введем следующие количественные характеристики параллельных алгоритмов [15].

Определение 1. Ускорением $\overline{S}$ и эффективностью $\overline{\varepsilon}$ параллельного алгоритма называют величину

$$\overline{S} = p\overline{\varepsilon} = \frac{T(1)}{T(p)}, \qquad (1)$$

где $T(1)$ — время выполнения алгоритма на одном процессоре; $T(p)$ — время выполнения параллельного алгоритма на системе из p процессоров.

Определение 2. Ускорением $\tilde{S}$ и эффективностью $\tilde{\varepsilon}$ параллельного алгоритма по отношению к наилучшему последовательному алгоритму называют величину

$$\tilde{S} = p\tilde{\mathcal{E}} = \frac{T(1)}{T(p)}, \qquad (2)$$

где $T(1)$ — время выполнения быстрейшего последовательного алгоритма на одном процессоре; $T(p)$ — время выполнения параллельного алгоритма на системе из p процессоров.

*Определение 3***.** Ускорением S_l и эффективностью $\mathcal{E}_l$ параллельного сглаживания на сетках уровня l по отношению к последовательному сглаживанию на тех же сетках называют величину

$$S_l = p\mathcal{E}_l = \frac{T_l(1)}{T_l(p)}, \qquad (3)$$

где $T_l(1)$ — время выполнения последовательного сглаживания (и оператора перехода) на одном процессоре; $T_l(p)$ — время выполнения параллельного сглаживания (и оператора перехода) на системе из p процессоров.

Ускорение $\overline{S}$ и эффективность $\overline{\mathcal{E}}$ (1) представляют теоретический интерес и характеризуют параллелизм конкретного алгоритма. Например, метод Якоби обладает почти полным параллелизмом, т. е. ожидается, что $\overline{S} \to p$ и $\overline{\mathcal{E}} \to 1$ [15].

Ускорение $\tilde{S}$ и эффективность $\tilde{\mathcal{E}}$ (2) представляют практический интерес и характеризуют уменьшение времени решения задачи по сравнению со временем расчета с использованием наиболее быстрого последовательного алгоритма. Например, метод Якоби обладает очень медленной скоростью сходимости, поэтому ожидается, что в сравнении с *V*-циклом $\tilde{S} \to 0$ и $\tilde{\mathcal{E}} \to 0$ при использовании достаточно мелких сеток. Подобные методы распараллеливать не имеет смысла, поскольку они даже в параллельном исполнении уступают наилучшим последовательным алгоритмам.

Ускорение S_l и эффективность $\mathcal{E}_l$ (3) представляют интерес для оценки параллелизма многосеточных алгоритмов. Выполняя серию вычислительных экспериментов, нетрудно для каждого сглаживателя эмпирически установить зависимость S_l и $\mathcal{E}_l$ от числа узлов сетки N^l и используемых процессоров p. Тогда ускорение и эффективность параллельного многосеточного алгоритма однозначно определяются значениями $S_l(N^l, p)$ и $\mathcal{E}_l(N^l, p)$.

Предположим, что начиная с уровня l^* резко возрастают потери, связанные с обменами данных между процессорами. Простейший выход состоит в последовательном выполнении сглаживающих ите-

раций на грубых сетках уровней $l > l^*$. Получим оценки ускорения и эффективности параллельных многосеточных итераций *V*-цикла при условии, что на сетках уровней $l < l^*$ вычисления проводят на p процессорах, на сетках $l \ge l^*$ – только на одном, а остальные $p-1$ процессоров простаивают.

Оценки ускорения и эффективности параллельных классических многосеточных алгоритмов получены при следующих предположениях:

1) число узлов самой мелкой сетки задано в виде

$$N_0 = 2^{d(L_2^+ + 1)},$$

где $d = 2, 3$ — размерность задачи; L_2^+ — номер уровня с самыми грубыми сетками, построенными посредством удвоения шага. Тогда количество узлов сеток уровня l

$$N^l = N_0 2^{-dl},\ l = 0, 1, \ldots, L_2^+;$$

2) поскольку вычисления проводят по одним и тем же формулам, то время выполнения сглаживающих итераций считаем пропорциональным числу узлов сетки:

$$T_l(1) = CN^l \Rightarrow T_l(1) = T_0(1)2^{-dl},$$

где C — некоторая константа, зависящая от сглаживателя. Здесь предполагается, что на всех сетках выполнено одинаковое количество сглаживающих итераций, а объем вычислений, необходимый для операторов переходов, включен в вычислительную работу сглаживающей итерации;

3) обмен данными между процессорами приведет к тому, что $S_l > S_{l+1}$ и $\mathcal{E}_l > \mathcal{E}_{l+1}$ так как $N^l > N^{l+1}$, $l = 0, 1, \ldots, L^+ - 1$, где L^+ — номер уровня с самыми грубыми сетками. Время выполнения последовательной многосеточной итерации *V*-цикла

$$T(1) = \sum_{l=0}^{L_2^+} T_l(1) = T_0(1) \sum_{l=0}^{L_2^+} 2^{-dl} \approx \frac{T_0(1)}{1 - 2^{-d}}.$$

Время выполнения параллельной многосеточной итерации *V*-цикла

$$T(p) = \sum_{l=0}^{l^*-1} T_l(p) + \sum_{l=l^*}^{L_2^+} T_l(1),$$

где первая сумма есть время выполнения параллельных сглаживающих итераций на p процессорах, а вторая сумма — время выполнения последовательных сглаживающих итераций на уровнях $l^* \le l \le L_2^+$. Из определения 3 (3) следует, что

$$T_l(p) = \frac{1}{p}\frac{T_l(1)}{\mathcal{E}_l} = \frac{1}{p}\frac{2^{-dl}}{\mathcal{E}_l}T_0(1),$$

тогда время выполнения параллельной многосеточной итерации можно оценить как

$$T(p) = T_0(1)\left(\frac{1}{p}\sum_{l=0}^{l^*-1}\frac{2^{-dl}}{\mathcal{E}_l} + \frac{2^{-dl^*}}{1-2^{-d}}\right).$$

Согласно (2), ускорение и эффективность параллельного *V*-цикла

$$\tilde{S} = p\tilde{\mathcal{E}} = \frac{T(1)}{T(p)} \approx \frac{1}{\frac{1-2^{-d}}{p}\sum_{l=0}^{l^*-1}\frac{2^{-dl}}{\mathcal{E}_l} + 2^{-dl^*}}.$$

Более грубая оценка может быть получена в предположении, что $\mathcal{E}_l < \mathcal{E}_0$:

$$\tilde{S} = p\tilde{\mathcal{E}} < \frac{1}{\frac{1}{p\mathcal{E}_0} + 2^{-dl^*}}, \qquad (4)$$

откуда следует $\tilde{S} = p\tilde{\mathcal{E}} < 2^{dl^*}$ при $p \to +\infty$, т. е. последовательное сглаживание на уровнях $l^* \le l \le L_2^+$ приводит к тому, что ускорение $\tilde{S}$ перестает зависеть от количества процессоров при достаточно большом p. Хотя объем вычислительной работы, необходимой для выполнения сглаживающих итераций на уровнях с грубыми сетками, невелик по сравнению с аналогичным объемом на самой мелкой сетке, последовательное сглаживание не позволит построить высокоэффективные параллельные многосеточные алгоритмы.

Иногда поступают следующим образом: сетку уровня $l^* - 1$ считают самой грубой, хотя возможно построение более грубых сеток. На сетке уровня $l^* - 1$ в качестве сглаживателя используют некоторый итерационный метод, обладающий достаточно высокой скоростью сходимости на достаточно мелких сетках и высоким параллелизмом. В работе [16] использованы явные чебышевские итерации для построения сглаживающей процедуры. Построение сглаживателя с указанными свойствами возможно лишь с привлечением дополнительный информации о решаемой задаче, т. е. высокий параллелизм достигается за счет снижения универсальности.

Таким образом, попытка построить параллельный многосеточный алгоритм только на основе распараллеливания компонент обречена на неудачу из-за уменьшения отношения объема вычислитель-

ной работы к объему пересылаемых данных и возможному простою процессоров на грубых сетках. Поэтому многосеточные итерации *W*-цикла, связанные с бóльшим объемом вычислений на грубых сетках, будут менее эффективны в параллельном исполнении, чем многосеточные итерации *V*-цикла. Очевидно, что необходим новый «источник параллелизма», не связанный со сглаживающей процедурой.

Параллельная универсальная многосеточная технология. С точки зрения распараллеливания вычислений УМТ обладает следующими привлекательными свойствами:

1) грубые сетки каждого уровня не имеют общих точек, т. е.

$$G_n^l \cap G_m^l = \varnothing, \quad n \neq m, \quad l = 1, 2, \ldots, L_3^+,$$

поэтому сглаживающие итерации на этих сетках можно проводить параллельно, независимо от используемой сглаживающей процедуры;

2) отсутствие общих узлов и граней уменьшает обмен данными между процессорами;

3) фиксированное число сеток на каждом уровне позволяет заранее предсказать необходимое число процессоров для распараллеливания УМТ;

4) почти одинаковое число точек на каждой сетке уровня l позволяет добиться равномерной загрузки процессоров.

Каждый уровень l состоит из 3^{dl} сеток ($d = 2, 3, l = 0, 1, \ldots, L_3^+$), поэтому число процессоров должно составить $p = 3^{d\kappa}$ ($\kappa = 1, \ldots, L_3^+$) для их равномерной загрузки. В дальнейшем величину κ будем называть *глубиной распараллеливания* УМТ. Значение $\kappa = 0$ ($\Rightarrow p = 1$) соответствует последовательной УМТ.

Рассмотрим простейший случай, когда $\kappa = 1$. Тогда число процессоров p, необходимых для реализации параллельной УМТ, составит $p = 3^d$, $d = 2, 3$. Распределение сеток уровней l ($l \geq 1$) между процессорами при $\kappa = 1$ показано на рис. 1. Вычисление поправки на каждой сетке первого уровня и ее подсетках осуществляется независимо на отдельном процессоре. Нетрудно видеть, что УМТ на уровнях с грубыми сетками ($l \geq 1$) обладает почти полным параллелизмом независимо от используемой сглаживающей процедуры. Сглаживающие итерации на самой мелкой сетке ($l = 0$) распараллеливают традиционным образом.

Предположим, что на всех уровнях выполнено одинаковое количество одних и тех же сглаживающих итераций. Поскольку общее число узлов на всех сетках одного уровня равно числу узлов на самой мелкой сетке, то время, необходимое для выполнения сглаживающих итераций, будет одинаковым:

$$T_l(1) = \text{const}, \quad l = 0, 1, \ldots, L_3^+.$$

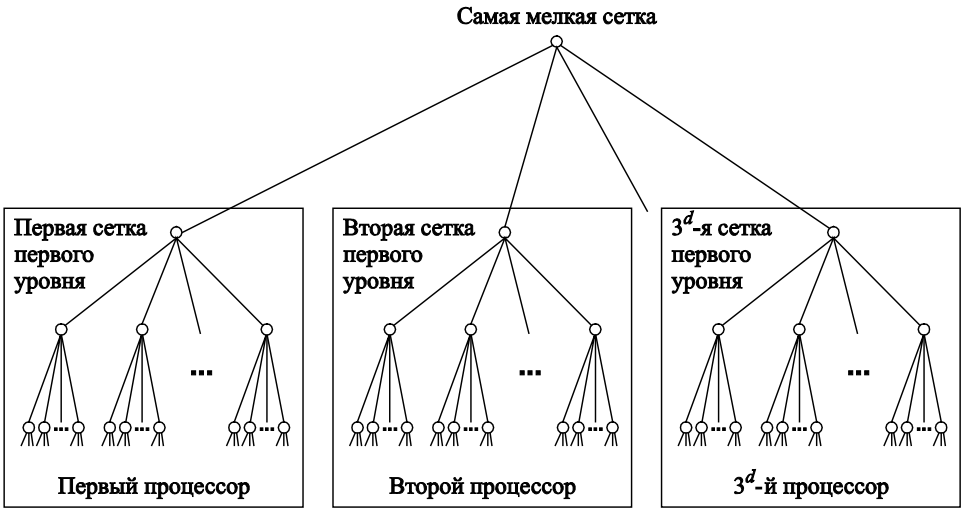


Рис. 1. Распределение грубых сеток между процессорами при $\kappa = 1$

На уровнях с грубыми сетками ($l \geq \kappa$) возможно выполнение дополнительных многосеточных итераций q^*, чтобы увеличить отношение объема вычислительной работы к объему пересылаемых данных. Схема многосеточной итерации при $q^* = 2$ показана на рис. 2.

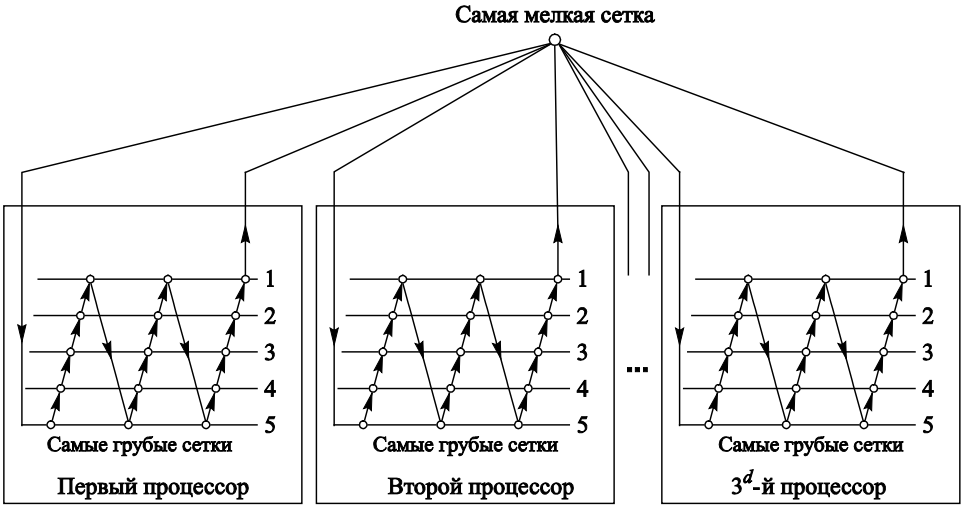


Рис. 2. Сглаживание на грубых сетках при $\kappa = 1$ и $q^* = 2$

Время выполнения последовательной многосеточной итерации УМТ

$$T(1) = T_0(1) + q^* \sum_{l=1}^{L_3^+} T_l(1) = T_0(1)\left(1 + q^* L_3^+\right).$$

Время выполнения параллельной многосеточной итерации УМТ

$$T(p) = T_0(p) + q^* \sum_{l=1}^{L_3^+} T_l(p) \ .$$

С учетом определения 3 (3) получим

$$T_0(p) = \frac{T_0(1)}{p\mathcal{E}_0} \quad \text{и} \quad \sum_{l=1}^{L_3^+} T_l(p) \ = \frac{T_0(1)}{p\mathcal{E}_*} L_3^+,$$

где $\mathcal{E}_*$ — эффективность параллельного сглаживания на сетках уровней $l \geq \text{к}$ по отношению к последовательному сглаживанию. Тогда выражение для ускорения и эффективности параллельной УМТ примет вид

$$\overline{S} = p\overline{\mathcal{E}} = p\frac{1 + q^* L_3^+}{\dfrac{1}{\mathcal{E}_0} + \dfrac{q^* L_3^+}{\mathcal{E}_*}}. \tag{5}$$

Поскольку УМТ на уровнях $l \geq \text{к}$ обладает почти полным параллелизмом, то величина $\mathcal{E}_*$ будет близка к единице. Полагая $\mathcal{E}_* = 1$, с учетом $p = 3^d$ получим следующую оценку:

$$\overline{S} = 3^d\,\overline{\mathcal{E}} < 3^d\,\frac{1 + q^* L_3^+}{\dfrac{1}{\mathcal{E}_0} + q^* L_3^+}, \tag{6}$$

т. е. эффективность параллелизма первой глубины критически зависит от эффективности параллельного сглаживания на самой мелкой сетке $\mathcal{E}_0$. При этом эффективность параллельной УМТ ожидается выше эффективности параллельных сглаживающих итераций на самой мелкой сетке: $\overline{\mathcal{E}} > \mathcal{E}_0$. Кроме того, увеличение количества узлов самой мелкой сетки приводит к повышению эффективности параллелизма: $L_3^+ \to +\infty \Rightarrow \overline{\mathcal{E}} \to 1$.

Аналогично могут быть получены оценки ускорения $\tilde{S}$ и эффективности $\tilde{\mathcal{E}}$ параллельной УМТ по отношению к *V*-циклу:

$$\tilde{S} = p\tilde{\mathcal{E}} < \frac{3^d}{1-2^{-d}} \frac{1}{\dfrac{1}{\mathcal{E}_0} + q^* L_3^+}. \qquad (7)$$

Поскольку УМТ в отличие от КММ не обладает оптимальной скоростью сходимости [8], то ускорение $\tilde{S}$ сравнительно невелико, причём $\tilde{S} \sim \ln^{-1} N^0$ при достаточно больших значениях L_3^+. Следует заметить, что полученная оценка достаточно груба, поскольку получена в предположении, что выполнено одинаковое количество многосеточных итераций *V*-цикла и УМТ. На самом деле для получения численного решения краевых задач при помощи УМТ требуется меньше многосеточных итераций, чем классических, особенно при $q^* > 1$.

Согласно оценкам (6) и (7), запишем максимальные значения ускорения и эффективности параллелизма в виде

$$\bar{S}_{\max} = 3^d \bar{\mathcal{E}}_{\max} = 3^d \frac{1 + q^* L_3^+}{\dfrac{1}{\mathcal{E}_0} + q^* L_3^+} \quad \text{и} \quad \tilde{S}_{\max} = 3^d \tilde{\mathcal{E}}_{\max} = \frac{3^d}{1-2^{-d}} \frac{1}{\dfrac{1}{\mathcal{E}_0} + q^* L_3^+}.$$

Зависимость максимального ускорения и эффективности параллелизма от количества сеточных уровней показана на рис. 3.

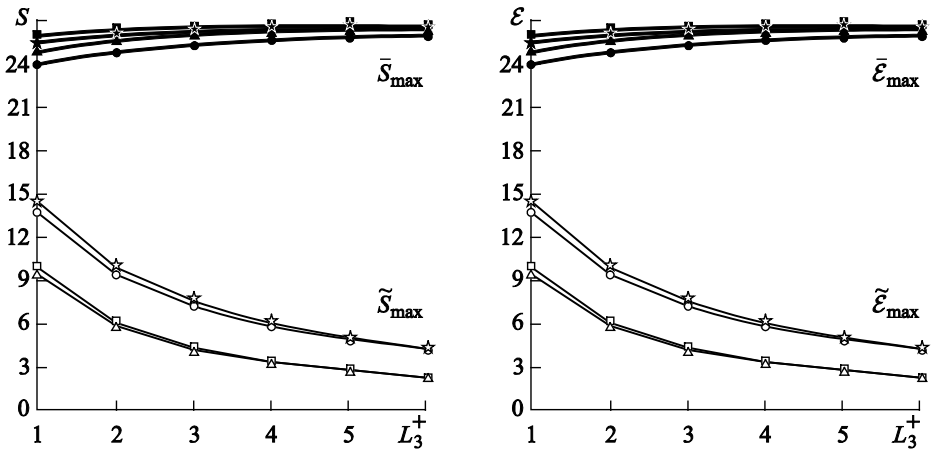


Рис. 3. Зависимость максимального ускорения и эффективности параллелизма от количества сеточных уровней:

$\mathcal{E}_0$ = 0,8, q^* = 1; $\mathcal{E}_0$ = 0,8, q^* = 2; $\mathcal{E}_0$ = 0,9, q^* = 1; $\mathcal{E}_0$ = 0,9, q^* = 2;

$\mathcal{E}_0$ = 0,8, q^* = 1; $\mathcal{E}_0$ = 0,8, q^* = 2; $\mathcal{E}_0$ = 0,9, q^* = 1; $\mathcal{E}_0$ = 0,9, q^* = 2

Для вычислительного эксперимента воспользуемся первой краевой задачей для трехмерного уравнения Пуассона

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=-f(x,y,z)$$

в единичном кубе, которая имеет точное решение $\exp(x+y+z)$. Точное решение однозначно определяет правую часть и граничные условия. Вычислительная сетка состоит из 245^3 = 14 706 125 узлов ($L_3^+=4$). Распараллеливание многосеточных итераций УМТ осуществлено с помощью технологии OpenMP при использовании 27 процессоров ($\kappa=1,\ d=2,3,\ p=3^{\kappa d}=27$) [17]. На каждой сетке выполнено четыре сглаживающие итерации, на сетках уровней l ($\kappa\le l\le L_3^+$) — две многосеточные итерации: $q^*=2$. Согласно результатам вычислительного эксперимента, эффективность распараллеливания сглаживающей процедуры на самой мелкой сетке и на сетках уровней l ($\kappa\le l\le L_3^+$) — $\overline{\varepsilon}_0=0,89$ и $\overline{\varepsilon}_*=0,95$ соответственно. Эффективность параллельной УМТ составила 0,92, в то время как, согласно оценке (6), $\overline{S}=26,6$ и $\overline{\varepsilon}=\overline{S}\cdot 3^{-d}=0,98$. Если сравнивать с последовательным *V*-циклом при одинаковом количестве многосеточных итераций, то эффективность параллельной УМТ составит 0,112, в то время как, согласно оценке (7), $\tilde{\varepsilon}=0,125$. Таким образом, время решения модельной краевой задачи для уравнения Пуассона с помощью параллельной УМТ при использовании 27 процессоров оказывается в 3 раза меньше, чем с помощью последовательного *V*-цикла.

Оценки (4) и (5) показывают, что при достаточно большом числе процессоров p ускорение параллельных КММ составит $\tilde{S}<2^{dl^*}$, а ускорение параллельной УМТ – $\tilde{S}\sim p$, причем независимо от используемой сглаживающей процедуры. Поэтому именно многосеточная структура является тем новым источником параллелизма, который должен быть задействован при распараллеливании сглаживающих итераций на грубых сетках.

Параллельный комбинированный многосеточный алгоритм. Основная идея комбинированного подхода состоит в следующем: от самой мелкой сетки ($l=0$) до сетки уровня l^*-1 основным источником параллелизма многосеточного алгоритма является сглаживающая процедура (как в КММ), а на уровнях $l^*\le L_*^+$ – многосеточная структура (как в УМТ). Используемая иерархия сеток и ее распределение по процессорам в одномерном случае ($d=1$) приведены на рис. 4.

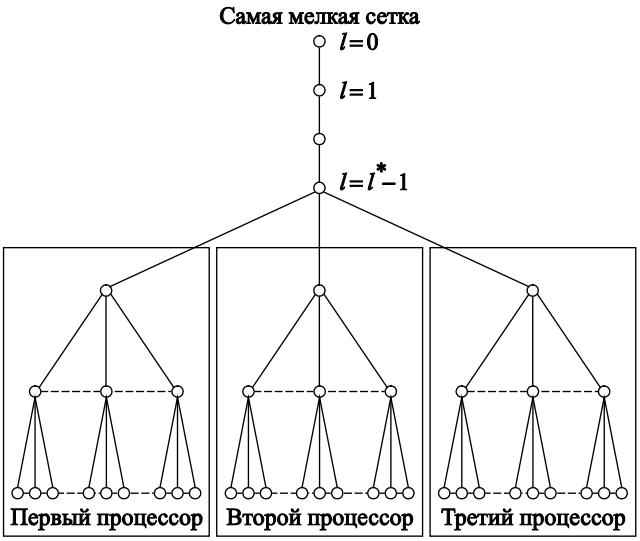


Рис. 4. Комбинированная иерархия сеток и распределение грубых сеток среди процессоров в одномерном случае

Номер уровня с самыми грубыми сетками L_*^+

$$L_*^+ = l^* + L_3^+ = l^* + \left[d\left(L_2^+ - l^* + 2\right)\frac{\lg 2}{\lg 3} - 1 \right],$$

где квадратные скобки означают целую часть числа. Аналогично оценивая время решения задачи, нетрудно получить следующую оценку ускорения и эффективности параллелизма:

$$\tilde{S} = p\tilde{\mathcal{E}} < p\frac{(1-2^{-d})^{-1}}{\sum\limits_{l=0}^{l^*-1}\frac{2^{-dl}}{\mathcal{E}_l} + q^* \cdot 2^{-d(l^*-1)}(L_*^+ - l^* + 1)}. \qquad (8)$$

Число процессоров для достижения наилучшей балансировки нагрузки следует выбирать как в УМТ, т. е. $p = 3^{\kappa d}$. Очевидно, что при $l^* = 1$ (т. е. $L_*^+ = L_3^+$) полученная оценка совпадает с (7). Для практического применения оценки (8) необходимо предварительно эмпирически определить эффективность параллельных сглаживающих итераций $\mathcal{E}_l$.

Заключение. Традиционный способ построения параллельных многосеточных алгоритмов, основанный на распараллеливании сглаживающей процедуры и операторов переходов, не позволяет достичь

высокой эффективности параллелизма из-за уменьшения отношения объема вычислительной работы к объему пересылаемых данных и возможного простоя процессоров на уровнях с грубыми сетками.

Многосеточная структура позволяет выполнять сглаживание на уровнях с грубыми сетками параллельно, причем независимо от выбора сглаживающей процедуры.

Высокоэффективный параллельный многосеточный алгоритм может быть построен только с использованием искусственного параллелизма его компонент и естественного параллелизма многосеточной структуры.

Полученные оценки ускорения и эффективности параллелизма (7) и (8) являются достаточно грубыми, поскольку в последовательных и параллельных многосеточных алгоритмах используют многосеточные циклы, которые различаются по количеству многосеточных итераций, необходимых для достижения критерия останова. Тем не менее применение многосеточной структуры для сглаживающих итераций на грубых сетках позволяет получить линейную зависимость ускорения от числа используемых процессоров независимо от вида сглаживателя.

Работа выполнена при финансовой поддержке РФФИ (проект № 12-01-00109).

Автор признателен профессору М.П. Галанину за критическое обсуждение полученных результатов.

ЛИТЕРАТУРА

[1] Власова Е.А., Зарубин В.С., Кувыркин Г.Н. *Приближенные методы математической физики.* Зарубин В.С., Крищенко А.П., ред. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2001, 700 с.

[2] Галанин М.П., Савенков Е.Б. *Методы численного анализа математических моделей*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2010, 591 с.

[3] Trottenberg U., Oosterlee C.W., Schüller A. *Multigrid*. Academic Press, London, 2001.

[4] Мартыненко С.И. Универсальная многосеточная технология для численного решения краевых задач на структурированных сетках. *Вычислительные методы и программирование*, 2000, т. 1, разд. 1, с. 85–104.

[5] Martynenko S.I. Robust multigrid technique for black box software. *Comp. Meth. in Appl. Math.*, 2006, vol. 6, no. 4, pp. 413–435.

[6] Мартыненко С.И. Универсальная многосеточная технология. *Математическое моделирование*, 2009, т. 21, № 9, с. 66–79.

[7] Martynenko S.I. Formalisation of the multigrid computations. *Int. J. of Comp. Science and Math.*, 2013, vol. 4, no. 4, pp. 309−320.

[8] Мартыненко С.И. *Универсальная многосеточная технология*. Москва, ИПМ им. М.В. Келдыша, 2013, 244 с. URL: http://library.keldysh.ru/prep_vw.asp?pid=3715

[9] Мартыненко С.И. Распараллеливание универсальной многосеточной технологии. *Вычислительные методы и программирование*, 2003, т. 4, разд. 1, с. 45–51.

[10] Martynenko S.I. Potentialities of the Robust Multigrid Technique. *Comp. Meth. in Appl. Math.*, 2010, vol. 10, no. 1, pp. 87–94.
[11] Мартыненко С.И. Оценки эффективности распараллеливания универсальной многосеточной технологии. *Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки*, 2011, № 4, с. 63–80.
[12] Воеводин В.В., Воеводин Вл.В. *Параллельные вычисления*. Санкт-Петербург, БХВ-Петербург, 2002, 608 с.
[13] Немнюгин С.А., Стесик О.Л. *Параллельное программирование для многопроцессорных вычислительных систем*. Санкт-Петербург, БХВ-Петербург, 2002, 400 с.
[14] Копысов С.П., Новиков А.К. *Промежуточное программное обеспечение параллельных вычислений*. Ижевск, Изд-во Удмуртского университета, 2012, 140 с.
[15] Ортега Дж. *Введение в параллельные и векторные методы решения линейных систем*. Москва, Мир, 1991, 367 с.
[16] Жуков В.Т., Новикова Н.Д., Феодоритова О.Б. *Параллельный многосеточный метод для разностных эллиптических уравнений. Ч. I: Основные элементы алгоритма*. ИПМ им. М.В. Келдыша РАН, Препринт, 2012, № 30, 32 с.
URL: http://library.keldysh.ru/preprint.asp?id=2012-30
[17] Антонов А.С. *Параллельное программирование с использованием технологии OpenMP*. Москва, Изд-во МГУ, 2009, 77 с.

Статья поступила в редакцию 05.11.2014

Ссылку на эту статью просим оформлять следующим образом:
Мартыненко С.И. О построении параллельных многосеточных алгоритмов. *Математическое моделирование и численные методы*, 2015, № 2(6), с. 105–120.

Мартыненко Сергей Иванович родился в 1965 г., окончил МВТУ им. Н.Э. Баумана в 1988 г., специальное вечернее отделение механико-математического факультета МГУ в 1990 г. Д-р физ.-мат. наук, научный сотрудник отдела «Спецдвигатели и химмотология» ЦИАМ им. П.И. Баранова. Автор более 50 работ в области многосеточных методов, вычислительной гидродинамики и конвективного теплообмена. e-mail: Martynenko@ciam.ru

Parallel multigrid algorithms

© S.I. Martynenko

Baranov Central Institute of Aviation Motor Development,
Moscow, 111116, Russia

The paper represents the main directions of development of the parallel classic multigrid algorithms and discusses their disadvantages. The possibility of efficient parallelization of smoothing iterations at the levels of coarse grids is shown using the Robust Multigrid Technique. Then multigrid structure is used for developing hybrid multigrid method. The paper describes estimations of speed-up and efficiency of different parallel multigrid algorithms as well as the results of numerical experiments.

Keywords: *multigrid technique, boundary value problems, parallelism.*

REFERENCES

[1] Vlasova E.A., Zarubin V.S., Kuvyrkin G.N. *Priblizhennye metody matematicheskoi fiziki* [Approximate methods in mathematical physics]. Zarubin V.S., Krischenko A.P., eds. Moscow, BMSTU Publ., 2001, 700 p.

[2] Galanin M.P., Savenkov E.V. *Metody chislennogo analiza matematicheskikh modelei* [Methods of numerical analysis of mathematical models]. Moscow, BMSTU Publ., 2010, 592 p.

[3] Trottenberg U., Oosterlee C.W., Schüller A. *Multigrid*. Academic Press, London, 2001.

[4] Martynenko S.I. Universalnaya mnogosetochnaya tehnologiya dlya chislennogo resheniya kraevyh zadach na strukturirovannyh setkah [Robust multigrid technique for solving boundary value problems on the structured grids]. *Vychislitelnye metody i programmirovanie* [Numerical methods and programming], 2000, vol. 1, section 1, pp. 85–104.

[5] Martynenko S.I. *Computational Methods in Applied Mathematics*, 2006, vol. 6, no. 4, pp. 413–435.

[6] Martynenko S.I. *Matematicheskoe modelirovanie – Mathematical Modeling*, 2009, vol. 21, no. 9, pp. 66–79.

[7] Martynenko S.I. *International Journal of Computer Science and Mathematics*, 2013, vol. 4, no. 4, pp. 309−320.

[8] Martynenko S.I. *Universalnaya mnogosetochnaya tekhnologiya* [Robust multigrid technique]. Moscow, Keldysh Institute of Applied Mathematic of RAS Publ., 2013, 244 p. Available at: http://library.keldysh.ru/prep_vw.asp?pid=3715

[9] Martynenko S.I. Rasparallelivanie universalnoy mnogosetochnoy tehnologii [Deparallelization of the robust multigrid technique]. *Vychislitelnye metody i programmirovanie* [Numerical methods and programming], 2003, vol. 4, section 1, pp. 45–51.

[10] Martynenko S.I. *Computational Methods in Applied Mathematics*, 2010, vol. 10, no. 1, pp. 87–94.

[11] Martynenko S.I. *Vestnic MGTU im. N.E. Baumana. Seria Estestvennye nauki – Herald of the Bauman Moscow State Technical University. Series: Natural Sciences*, 2011, no. 4, pp. 63–80.

[12] Voevodin V.V., Voevodin Vl.V. *Parallelnye vychisleniya* [Parallel computing]. St. Petersburg, BHV-Petersburg Publ., 2002, 608 p.

[13] Nemnyugin S.A., Stesik O.L. *Parallelnoe programmirovanie dlya mnogoprotsessornyh vychislitelnyh sistem* [Parallel programming for the multiprocessor computing systems]. St. Petersburg, BHV-Petersburg Publ., 2002, 400 p.

[14] Kopysov S.P., Novikov A.K. *Promezhutochnoe programmnoe obespechenie parallelnyh vychisleniy* [Intermediate software for the parallel computing]. Izhevsk, Udmurtia University Publ., 2012, 140 p.

[15] Ortega J.M. *Introduction to parallel and vector solution of linear systems*. Plenum Press, New York, 1988. [Russian edition: Ortega Dzh. Vvedenie v parallelnye i vektornye metody resheniya lineinykh sistem. Moscow, Mir Publ., 1991, 367 p.].

[16] Zhukov V.T., Novikova N.D., Feodoritova O.B. *Parallelnyi mnogosetochnyi metod dlya raznostnykh ellipticheskikh uravneniy* [A parallel multigrid method for the elliptic difference equations]. *Part I*: *Osnovnye komponenty algoritma* [Basic components of the algorithm]. Preprint of Keldysh Institute of Applied Mathematic of RAS, 2012, no. 30, 32 p. Available at: http://library.keldysh.ru/preprint.asp?id=2012-30

[17] Antonov A.S. *Parallelnoe programmirovanie s ispolzovaniem tehnologii OpenMP* [Parallel programming using technology OpenMP]. Moscow, MSU Publ., 2009, 77 p.

Martynenko S.I. (b. 1965) graduated from Bauman Moscow Higher Technical School in 1988, from Moscow State University (Mechanical and Mathematical Department) in 1990. Dr. Sci. (Phys. & Math.), researcher of Engines and Chemmotology Department of the State Research Center at Baranov Central Institute of Aviation Motor Development (CIAM). Author of more than 50 publications in the fields of multigrid methods, CFD and convective heat transfer. e-mail: Martynenko@ciam.ru.