УДК 519.6

РАСПАРАЛЛЕЛИВАНИЕ ГЕОМЕТРИЧЕСКИХ МНОГОСЕТОЧНЫХ МЕТОДОВ [1)]

ЧЖОУ ВЭЙСИН[1], П.Д. ТОКТАЛИЕВ[2], С.И. МАРТЫНЕНКО[2]

[1] *Харбинский Политехнический Университет, Харбин, КНР*
[2] *Центральный Институт Авиационного Моторостроения, Москва, Россия*
E-mail Martynenko@ciam.ru

PARALLELIZATION OF GEOMETRIC MULTIGRID METHODS

ZHOU WEIXING[1], P.D. TOKTALIEV[2], S.I. MARTYNENKO[2]

[1] *Harbin Institute of Technology, Harbin, China*
[2] *Central Institute of Aviation Motors, Moscow, Russia*

Аннотация

Показаны трудности построения параллельного многосеточного алгоритма, связанные с увеличением отношения объёма передаваемых данных к объёму вычислений и возможным простоем процессоров на грубых сетках. Предложено использовать естественный параллелизм многосеточной структуры для повышения общей эффективности параллелизма. Приводятся результаты теоретического анализа ускорения и эффективности параллельных оптимизированных и робастных многосеточных методов.

Ключевые слова: краевые задачи, многосеточные методы, распараллеливание вычислений

Summary

Problems of the parallel multigrid algorithm development caused by the relation increase of communication and computation and possible processor idleness on the coarse grids are shown. It is proposed to use natural parallelism of the multigrid structure for increase of the total parallelism efficiency. Results of the theoretical studies of the speed-up and efficiency of the parallel optimized and robust multigrid methods are given.

Key words: boundary value problems, multigrid methods, parallelisation of computation

Введение

В настоящее время многосеточные методы, ставшие доминирующим способом численного решения краевых задач, подразделяются на классические (КММ или optimized multigrid algorithms) и универсальные (robust multigrid algorithms) [1]. Классические методы предназначены для решения с оптимальными усилиями отдельных задач, причём наивысшая скорость сходимости достигается за счёт адаптации проблемно-зависимых компонент алгоритма к решаемой задаче. Универсальные методы предназначены для решения широкого класса задач, но вычислительные усилия лишь близки к оптимальным.

Распараллеливание многосеточных методов затрудняется уменьшением отношения объёма вычислений к объёму передаваемых данных и возможным простоем процессоров на грубых сетках, а так же трудностью распараллеливания прямого метода решения СЛАУ, обычно используемого на самой грубой сетке. В данной работе для эффективного распараллеливания многосеточных методов использованы дополнительные сетки на отдельных сеточных уровнях. Приведены оценки ускорения и эффективности параллелизма и результаты вычислительных экспериментов.

[1)] Работа выполнена при поддержке РФФИ (проект 12-01-00109)

1. Сглаживающая процедура и её распараллеливание

В качестве сглаживающей процедуры использован итерационный метод Якоби с блочным упорядочиванием неизвестных. Сглаживающие итерации сводятся к решению совокупности СЛАУ прямым методом Гаусса, благодаря которому данный сглаживатель может быть применён к решению широкого класса задач, включая седловые [2].

Распараллеливание итерационного метода Якоби основано на декомпозиции исходной области на подобласти [3]. В каждой подобласти значения искомой сеточной функций вычисляются независимо. После завершения сглаживающей итерации на границах подобластей граничные значения пересылаются в соседние процессоры.

2. Основные определения и допущения

Следуя [3], введём следующие меры параллелизма:

Определение 1. *Ускорением* ($\bar{\mathsf{S}}$) *и эффективностью* ($\bar{\mathsf{E}}$) *параллельного алгоритма называется величина*

$$\bar{\mathsf{S}} = p\bar{\mathsf{E}} = \frac{T(1)}{T(p)},$$

где $T(1)$ *есть время выполнения алгоритма на одном процессоре, а* $T(p)$ *— время выполнения параллельного алгоритма на системе из* p *процессоров.*

Определение 2. *Ускорением* ($\tilde{\mathsf{S}}$) *и эффективностью* ($\tilde{\mathsf{E}}$) *параллельного алгоритма по отношению к наилучшему последовательному алгоритму называется величина*

$$\tilde{\mathsf{S}} = p\tilde{\mathsf{E}} = \frac{T(1)}{T(p)},$$

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

Определение 3. *Ускорением* (S_l) *и эффективностью* (E_l) *параллельного сглаживания на сетках уровня* l *по отношению к последовательному сглаживанию на тех же сетках называется величина*

$$\mathsf{S}_l = p\mathsf{E}_l = \frac{T_l(1)}{T_l(p)},$$

где $T_l(1)$ *есть время выполнения последовательного сглаживания на одном процессоре, а* $T_l(p)$ *— время выполнения параллельного сглаживания на системе из* p *процессоров.*

Далее для оценки ускорения и эффективности в качестве быстрейшего последовательного алгоритма будет использован V-цикл [4].

Положим, область является d-мерным кубом и построена вычислительная сетка посредством разбиения его рёбер на N_x^0, N_y^0 и N_z^0 частей. Тогда количество узлов данной (самой мелкой) сетки составит $N^0 = (N_x^0+1)(N_y^0+1)(N_z^0+1)$, а шаги сетки суть $h_x^0 = 1/N_x^0$, $h_y^0 = 1/N_y^0$ и $h_z^0 = 1/N_z^0$. При оценке ускорения и эффективности КММ будем полагать, что $N_{xyz}^0 = N_x^0 = N_y^0 = N_z^0 = 2^{L_2^+ + 1} - 1$, т.е. $N^0 = 2^{d(L_2^+ + 1)}$, где L_2^+ есть номер сеточного уровня с самыми грубыми сетками, построенных удвоением шага. Тогда количество узлов сетки уровня l ($0 \leqslant l \leqslant L_2^+$) составит $N_l = N^0 2^{-dl}$. Поскольку вычисления проводят по одним и тем же формулам, то время выполнения сглаживания пропорционально количеству узлов $T_l \sim N_l$ или $T_l = T_0 2^{-dl}$. Заметим, что при выполнении сглаживания на грубых сетках возрастают затраты, связанные с обменами данных, поэтому ожидается, что $\mathsf{E}_0 > \mathsf{E}_1 > \ldots > \mathsf{E}_{L_2^+}$.

3. Распараллеливание V-цикла

Характерная трудность распараллеливания КММ состоит в следующем: количество узлов на грубых сетках уровня l $(l \gg 1)$ составляет $N^0 2^{-dl}$, что может быть меньше используемых процессоров $p > N^0 2^{-dl}$. Поэтому часть процессоров будет простаивать при распараллеливании сглаживающих итераций на грубых сетках.

Положим, что начиная с некоторого уровня l^* $(0 < l^* < L_2^+)$ для выполнения сглаживающих итераций использован один процессор, а $p-1$ процессоров простаивают. Тогда ускорение параллельного V-цикла составит

$$\bar{\mathsf{S}} = \frac{T(1)}{T(p)} = \frac{\sum\limits_{l=0}^{L_2^+} T_l(1)}{\underbrace{\sum\limits_{l=0}^{l^*-1} T_l(p)}_{p>1} + \underbrace{\sum\limits_{l=l^*}^{L_2^+} T_l(p)}_{p=1}} \approx \frac{1}{\dfrac{1-2^{-d}}{p}\sum\limits_{l=0}^{l^*-1}\dfrac{2^{-dl}}{\mathsf{E}_l} + 2^{-dl^*}} < \frac{2^{dl^*}}{\dfrac{2^{dl^*}-1}{p\mathsf{E}_0}+1}.$$

Отсюда следует, что $\bar{\mathsf{S}} < 2^{dl^*}$ при $p \to \infty$. Полученная оценка ускорения параллельного V-цикла является иллюстрацией закона Амдаля, который ограничивает значение ускорения при наличии последовательных компонент алгоритма [3]. Очевидно, что ускорение параллельного W-цикла будет ещё меньше из-за бóльшего объёма вычислений на уровнях с грубыми сетками.

В настоящее время предложен ряд подходов к повышению ускорения и эффективности параллельных КММ. Например предлагается считать l^* уровнем с самыми грубыми сетками, а качестве сглаживателя использовать итерационный метод, который обладает высокой скоростью сходимости на достаточно мелких сетках [5]. Другим подходом является использование нескольких сеток на одном уровне для повышения ускорения и эффективности параллельных многосеточных методов [1]. Далее будет показано, что именно это направление является наиболее перспективным.

4. Распараллеливание универсальной многосеточной технологии

В [2] представлен вариант геометрических многосеточных методов с минимальным количеством проблемно-зависимых компонент, который получил название «Универсальная Многосеточная Технология» (УМТ). В УМТ каждая вычислительная сетка является совокупностью 3^d $(d = 2, 3)$ более грубых сеток, т.е. уровень l образован 3^{dl} сетками. Поэтому для распараллеливания УМТ необходимо использовать компьютер с $p = 3^{dk}$ процессорами, где параметр $k = 1, 2, \ldots$ получил название глубины распараллеливания. Схема многосеточного цикла УМТ показана на Рис. 1, причём на грубых сетках уровня

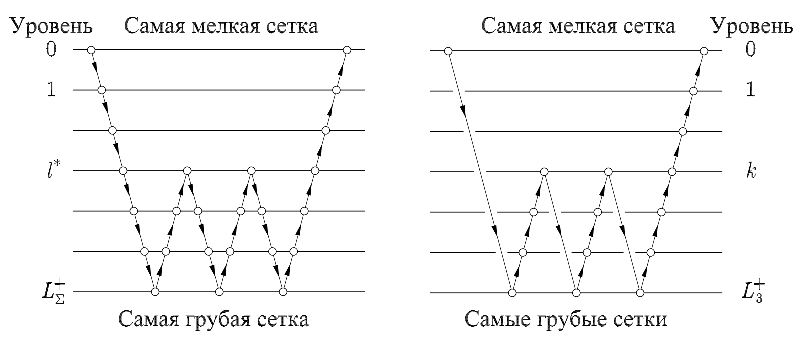

Рис. 1: Многосеточные циклы с $q^* = 3$ для параллельных вычислений: комбинированный V-цикл (слева) и пилообразный цикл УМТ (справа)

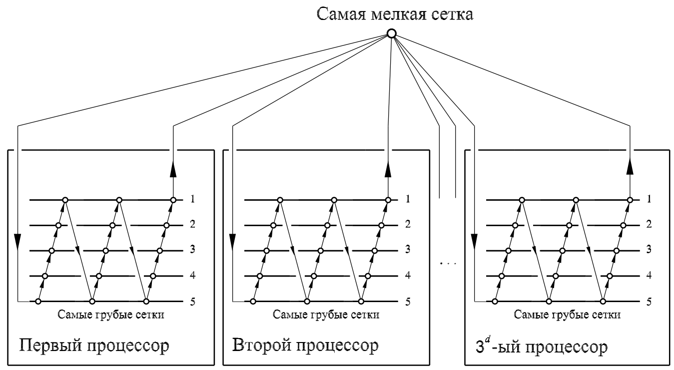

Рис. 2: Сглаживание на грубых сетках при $k = 1$ и $q^* = 2$

$k \leqslant l \leqslant L_3^+$ возможно выполнение дополнительных многосеточных итераций $q^* > 1$.

В случае $k = 1$ или $p = 3^d$ ускорение параллельной УМТ можно оценить как

$$\bar{\mathsf{S}} = \frac{T_0(1) + q^* \sum\limits_{l=1}^{L_3^+} T_l(1)}{T_0(p) + q^* \sum\limits_{l=1}^{L_3^+} T_l(p)} \approx 3^d \frac{1 + q^* L_3^+}{\dfrac{1}{\mathsf{E}_0} + q^* \dfrac{L_3^+}{\mathsf{E}_*}}, \tag{1}$$

где E_* есть эффективность распараллеливания сглаживающих итераций на уровнях $l \leqslant k$. Поскольку сглаживание на данных уровнях состоит из решения независимых задач одинакового размера, то $\mathsf{E}_* \approx 1$.

Вычислительный эксперимент состоял из численного решения первой краевой задачи для трёхмерного уравнения Пуассона, которая имела точное решение e^{x+y+z}. Вычислительная сетка состояла из $245^3 = 14706125$ узлов ($L_3^+ = 4$). Распараллеливание вычислений осуществлено при помощи технологии OpenMP [6] с привлечением 27 процессоров ($k = 1$, $d = 3$, $p = 3^{kd} = 27$). На каждой сетке выполнено четыре сглаживающие итерации, на сетках уровней l ($k \leqslant l \leqslant L_3^+$) выполнено две многосеточные итерации $q^* = 2$. Согласно результатам вычислительного эксперимента эффективность распараллеливания сглаживающей процедуры на самой мелкой сетке и на сетках уровней l ($k \leqslant l \leqslant L_3^+$) составила $\mathsf{E}_0 = 0.89$ и $\mathsf{E}_* = 0.95$ соответственно. Эффективность параллельной УМТ составила 0.92, в то время, согласно оценке (1), получим $\bar{\mathsf{S}} = 26.6$ и $\bar{\mathsf{E}} = \bar{\mathsf{S}} 3^{-d} = 0.98$.

Более пессимистичные оценки получаются при сравнении параллельной УМТ с наилучшим последовательным алгоритмом, в качестве которого выбран V-цикл. В случае $k = 1$ нетрудно получить

$$\tilde{\mathsf{S}} = \frac{\sum\limits_{l=0}^{L_2^+} T_l(1)}{T_0(p) + q^* \sum\limits_{l=1}^{L_3^+} T_l(p)} < \frac{p}{1 - 2^{-d}} \frac{1}{\dfrac{1}{\mathsf{E}_0} + q^* L_3^+}. \tag{2}$$

Применительно к выполненному вычислительному эксперименту из данной оценке следует $\tilde{\mathsf{S}} < 3.375$ и $\tilde{\mathsf{E}} < 0.125$, в действительности получено $\tilde{\mathsf{S}} = 2.830$ и $\tilde{\mathsf{E}} = 0.105$. Однако следует помнить, что оценка (2) получена в предположении, что выполнено одинаковое количество многосеточных итераций УМТ и V-цикла. В действительности, вычислительная стоимость многосеточных итераций УМТ выше, чем итераций V-цикла, но их количество в несколько раз меньше [2]. Поэтому значения ускорения $\tilde{\mathsf{S}}$ и эффективности параллелизма $\tilde{\mathsf{E}}$ будут больше, чем следует из (2).

5. Распараллеливание комбинированного V-цикла

Положим, что при распараллеливании V-цикла, начиная с уровня l^*, происходит резкое уменьшение эффективности параллелизма. Тогда имеет смысл продолжить вычисления на грубых сетках уровней l

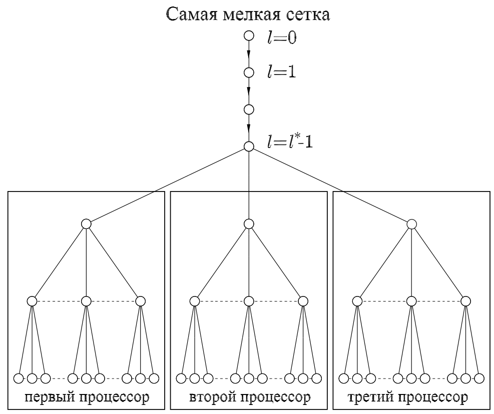

Рис. 3: Комбинированный V-цикл: распределение грубых сеток среди процессоров в одномерном случае

($l^* \leqslant l \leqslant L_3^+$) используя дополнительные сетки, как в УМТ. На Рис. 3 показаны грубые сетки уровней l ($l^* \leqslant l \leqslant L_3^+$), а на Рис. 1 – схема комбинированного V-цикла с дополнительными многосеточными итерациями q^*. Тогда справедливы следующие оценки

$$\bar{\mathsf{S}} = p\bar{\mathsf{E}} < p\,\frac{(1-2^{-d})^{-1}}{\sum\limits_{l=0}^{l^*-1}\dfrac{2^{-dl}}{\bar{\mathsf{E}}_l} + q^* 2^{-d(l^*-1)}\left(L_*^+ - l^* + 1\right)},$$

Согласно данной оценке, при комбинированном построении грубых сеток максимальное ускорение параллельного многосеточного алгоритма будет пропорционально количеству процессоров p.

Заключение

Использование дополнительных сеток для вычисления поправки позволяется распараллеливать классические и универсальные многосеточные алгоритмы с высокой эффективностью $\bar{\mathsf{E}}$ по отношению к соответствующему последовательному алгоритму и с существенно меньшей эффективностью $\tilde{\mathsf{E}}$ по отношению к последовательному V-циклу.

ЛИТЕРАТУРА

1. **Trottenberg U., Oosterlee C., Schüller A.** Multigrid. – San Diego: Academic Press, 2001. – 631 p.
2. **Мартыненко С.И.** Универсальная многосеточная технология. – М: ИПМ им. М.В. Келдыша, 2013. – 244 с. URL: http://library.keldysh.ru/prep_vw.asp?pid=3715
3. **Ортега Дж.** Введение в параллельные и векторные методы решения линейных систем. – М: Мир, 1991. – 367 с.
4. **Ольшанский М.А.** Лекции и упражнения по многосеточным методам. – М: ФИЗМАТЛИТ, 2005. – 168 с.
5. **Жуков В.Т., Новикова Н.Д., Феодоритова О.Б.** Параллельный многосеточный метод для разностных эллиптических уравнений. Часть I. Основные элементы алгоритма // Препринты ИПМ им. М.В. Келдыша РАН. – 2012. – № 30. – 32 с. (http://library.keldysh.ru/preprint.asp?id=2012-30)
6. **Антонов А.С.** Параллельное программирование с использованием технологии OpenMP: Учебное пособие. – М: Изд-во МГУ, 2009. – 77 с.

REFERENCES

1. **Trottenberg U., Oosterlee C., Schüller A.** Multigrid. – San Diego: Academic Press, 2001. – 631 p.
2. **Martynenko S.I.** Robust Multigrid Technique [Universal'naja mnogosetochnaja tekhnologija]. – Moscow: Keldysh Institute of Applied Mathematics RAS, 2013. – 244 p. (in Russian)
3. **Ortega J.M.** Introduction to Parallel and Vector Solution of Linear Systems. – New York: Plenum Press, 1988.
4. **Olshanskii M.A.** Lectures and exercises on multigrid methods [Lektcii i uprazhnenija po mnogosetochnym metodam]. – Moscow: FIZMATLIT, 2005. – 168 p. (in Russian)
5. **Zhukov V.T., Novikova N.D., Feodoritova O.B.** Parallel multigrid method for the discrete elliptic equations. Part I. Main components of the algorithm [Parallel'nyi mnogosetochnyi metod dlja raznostnykh ellipticheskikh uravnenii. I. Osnovnye elementy algoritma] // Preprints of Keldysh Institute of Applied Mathematics RAS, 2012. – № 30. – 32 p. (in Russian)
6. **Antonov A.S.** Parallel programming using OpenMP: Manual. [Parallel'noe programmirovanie s ispol'zovaniem tekhnologii OpenMP]. – Moscow: MSU, 2009. – 77 p. (in Russian)