

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МЕТОДА ЦИКЛИЧЕСКИХ ВСТРЕЧНЫХ ПРОГОНОВ ДЛЯ ДВУМЕРНОЙ ОБЛАСТИ

© 2008 Л.В. Логанова

Самарский государственный аэрокосмический университет

Работа посвящена построению параллельного алгоритма метода встречных циклических прогонок для решения сеточных уравнений ленточного вида. Рассмотрен вариант линейного разбиения двумерной сеточной области. Произведено сравнение с известными алгоритмами, выявлены недостатки и достоинства приведенного.

Параллельный алгоритм, метод встречных циклических прогонок

Введение

Математическое моделирование получает все большее распространение в различных областях науки и техники, например, в дифракционной оптике [1]. Развитие вычислительных средств привело к необходимости интеграции методов математического моделирования и современных компьютерных технологий.

Моделирование физических процессов зачастую сводится к необходимости решения систем линейных алгебраических уравнений (СЛАУ) вида $Ax = b$, где матрица A имеет ленточную структуру. Причем круговые, сферические и цилиндрические физические подобласти характеризуются матрицей, содержащей кроме ленты на центральных диагоналях ненулевые элементы в верхнем правом и левом нижнем углах. Переход к реальным задачам больших размеров сопряжен с заметным ростом объема вычислений. Именно этот фактор определяет растущую потребность в применении параллельных вычислительных систем для решения больших и очень больших задач и, как следствие, синтез алгоритмов, определяемых такой архитектурой. Для создания параллельных алгоритмов решения вышеуказанных СЛАУ традиционно используются следующие методы: метод прогонки, циклической редукции, декомпозиции. Необходимо обратить внимание на то, что метод прогонки является экономичным, но плохо масштабируемым; метод циклической редукции удовлетворительно масштабируется и вместе с тем имеет повышенный объем коммуникаций, метод

декомпозиции обладает приемлемым объемом пересылок и характеризуется большим объемом вычислений. Данные соображения и тот факт, что ограничение на масштабируемость может быть снято в случае многомерной сеточной области, был определен выбор в пользу применения метода прогонки для решения СЛАУ с матрицей ленточного вида.

В работах [2-4] приведены параллельные алгоритмы данного метода. Наряду с несомненными достоинствами каждый из них имеет и недостатки.

Так применение метода правой прогонки из работы [2] непосредственно для одномерной (1D) сеточной области невозможно. Для случая линейного разбиения одномерной сеточной области возрастает время простоя. Метод встречных прогонок не может быть применен для решения разностных уравнений с циклическими граничными условиями [3], имеет ограничение на число задач (четное) [4].

В данной работе предложен параллельный алгоритм метода встречных циклических прогонок, являющийся развитием алгоритмов из [3], [4]. При этом длительность вычислений по сравнению с алгоритмом из работы [4] не увеличивается и предъявляются относительно низкие требования к объему оперативной памяти вычислительной системы. Целесообразность создания алгоритма данного метода определяется областью его применения:

– при аппроксимации краевых задач для обыкновенных дифференциальных уравнений второго порядка с переменными (периодическими) коэффициентами:

$$\begin{aligned}
 a_i y_{i-1} - c_i y_i + b_i y_{i+1} &= -f_i, \\
 a_i \neq 0, \quad b_i \neq 0, \quad i &= 0, \pm 1, \pm 2, \dots, \\
 a_i &= a_{i+N}, \quad b_i = b_{i+N}, \\
 c_i &= c_{i+N}, \quad f_i = f_{i+N}
 \end{aligned}$$

– реализация разностных схем для уравнений с частными производными в цилиндрических и сферических координатах.

Параллельный алгоритм для одномерной сеточной области

Прежде чем приступить к рассмотрению параллельного алгоритма метода встречных циклических прогонок для многомерных областей, обратимся к параллельному алгоритму данного метода для одномерной сеточной области. Функциональная декомпозиция метода встречных циклических

прогонок определяется спецификой использования в ней алгоритма из [5]. Параллельный алгоритм для одномерной сеточной области подробно изложен в работе [4]. Кратко приведем основные моменты данного подхода.

Произведем разбиение одномерной области данных ω_1 на равные подобласти ω_1^1 и ω_1^2 между двумя задачами параллельного алгоритма. Пусть первая задача вычисляет прогоночные коэффициенты $\alpha_i, \beta_i, \gamma_i$ и u_i при $1 \leq i \leq g$, вторая ζ_i, η_i, ψ_i и v_i при $g+1 \leq i \leq N$, где g – номер последнего узла подобласти ω_1^1 , N – номер последнего узла областей ω_1 и ω_1^2 .

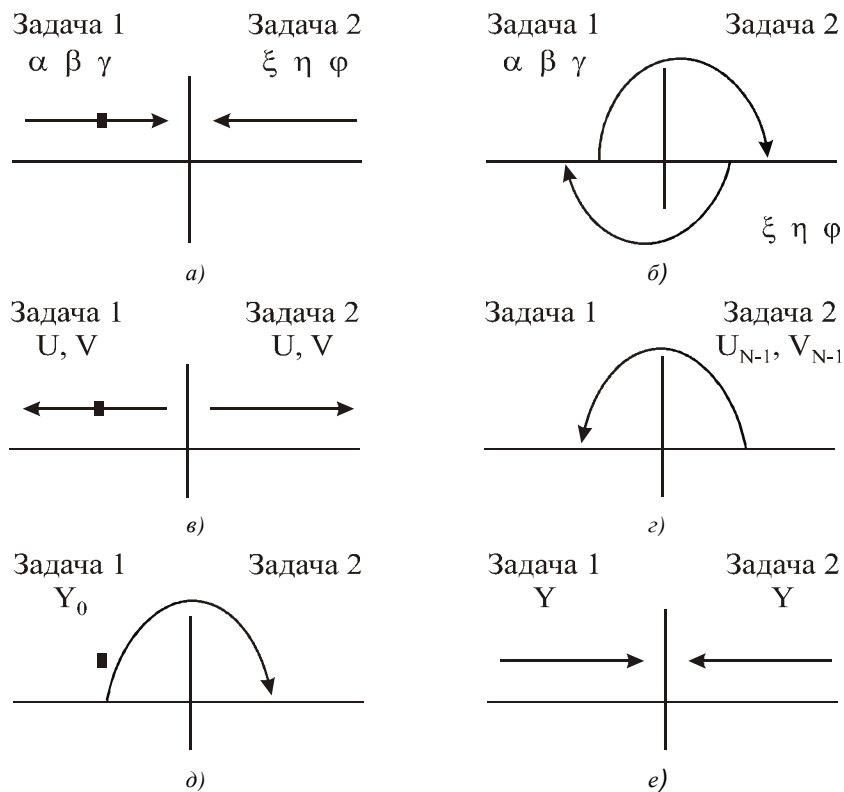


Рис. 1. Этапы вычислений по двухзадачному параллельному алгоритму, реализующему метод встречных прогонок на одномерной сеточной области

Вычисления по алгоритму включают: вычисление коэффициентов $\alpha, \beta, \gamma, \zeta, \eta, \phi$ (прямой ход прогонок, рис. 1а), обмен данными (крайними коэффициентами $\alpha_{g+1}, \beta_{g+1}, \gamma_{g+1}, \zeta_{g+1}, \eta_{g+1}, \phi_{g+1}$, рис. 1б), вычисления зна-

чений сеточных функций U, V (обратный ход, рис. 1в), пересылка дополнительных данных U_{N-1}, V_{N-1} (рис. 1г), вычисление u_0 , пересылка u_0 (рис. 1д) и нахождение окончательного решения (рис. 1е). При этом обе за-

дачи производят вычисления одновременно, за исключением вычисления y_0 . Простой при данном подходе отсутствуют, однако алгоритм не масштабируется.

Ускорение приведенного двухзадачного алгоритма оценивается величиной

$$S = \frac{C_1 MN \tau_a}{\frac{1}{2} C_1 MN \tau_a + 3N \tau_k},$$

где M – размерность СЛАУ, $C_1 M \tau_a$ – длительность расчета по последовательному алгоритму [5], τ_a – время выполнения одной

операции с плавающей точкой, τ_k – длительность одной коммуникации между задачами алгоритма в рамках пакетной модели. При пакетной модели передачи [6] данных время передачи некоторого объема данных остается неизменным, если этот объем не превышает размера пакета.

Состоятельность данного алгоритма подтверждается результатами вычислительных экспериментов, которые приведены на рис. 2.

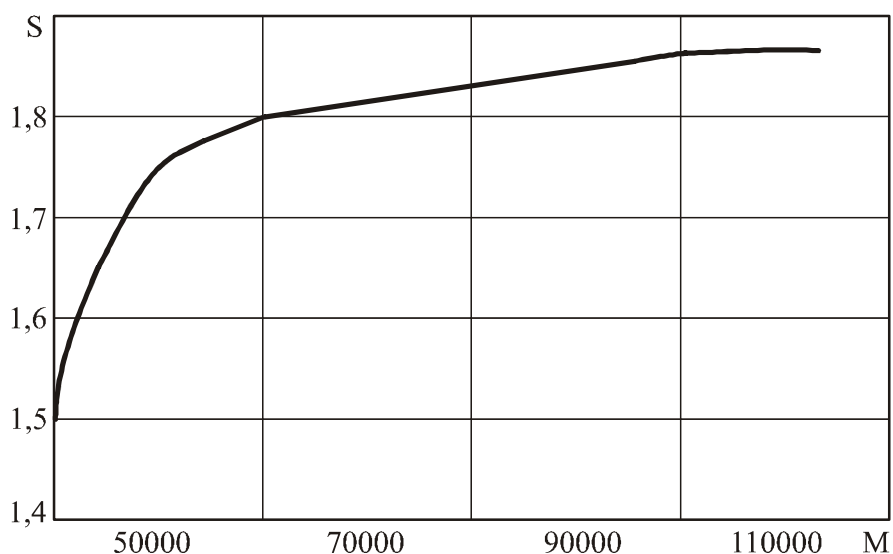


Рис. 2. Зависимость ускорения параллельного вычислительного процесса от размера сеточной области

Полученное ускорение для исследуемого алгоритма соответствует закону Амдала, согласно которому оно не должно превышать 2.

Например, для всех размерностей сеточной области, начиная с $M=50000$, ускорение предложенного алгоритма превышает 1,5.

Параллельный алгоритм для двумерной сеточной области, основанный на методе встречных циклических прогонок с линейным разбиением

Для двумерной области ω_2 ограничение на масштабируемость снимается. В работе [6] приведен параллельный алгоритм метода встречных прогонок для ω_2 . Здесь он модифицирован на случай СЛАУ с трехдиагональной матрицей и ненулевыми элементами

в верхнем правом и левом нижнем углах. Если алгоритм состоит из двух задач, то каждая СЛАУ в продольном направлении (прогонки по строкам) решается одновременно обеими задачами. Вычисления в поперечном направлении (прогонки по столбцам) являются независимыми и не сопровождаются коммуникациями. Пусть размер сеточной области ω_2 будет $N \times M$ узлов. Линейное разбиение выполним по M . Тогда ускорение алгоритма составит

$$S = \frac{2C_1 MN \tau_a}{C_1 MN \tau_a + 6N \tau_k}.$$

Для алгоритма из 4 задач производится линейное разбиение сеточной области, представленное на рис. 3. Первая и последняя за-

дачи начинают прямой ход без принятия прогоночных коэффициентов, остальные задачи перед прямым ходом принимают прогоночные коэффициенты, выполнив необходимые вычисления, передают их. Задачи под номерами $p/2$ и $p/2+1$ (p – число задач) начинают обратный ход принятием прогоночных коэффициентов друг от друга. Вычислив значения сеточных функций, передают их

задачам $p/2-1$ и $p/2+2$. Остальные задачи перед обратным ходом принимают значения сеточных функций, выполнив необходимые вычисления, передают их. Кроме того, задача 4 по завершении обратного хода передает значения сеточных функций задаче 1, которая вычисляет y_0 и пересылает его остальным задачам. После чего производится вычисление решения.

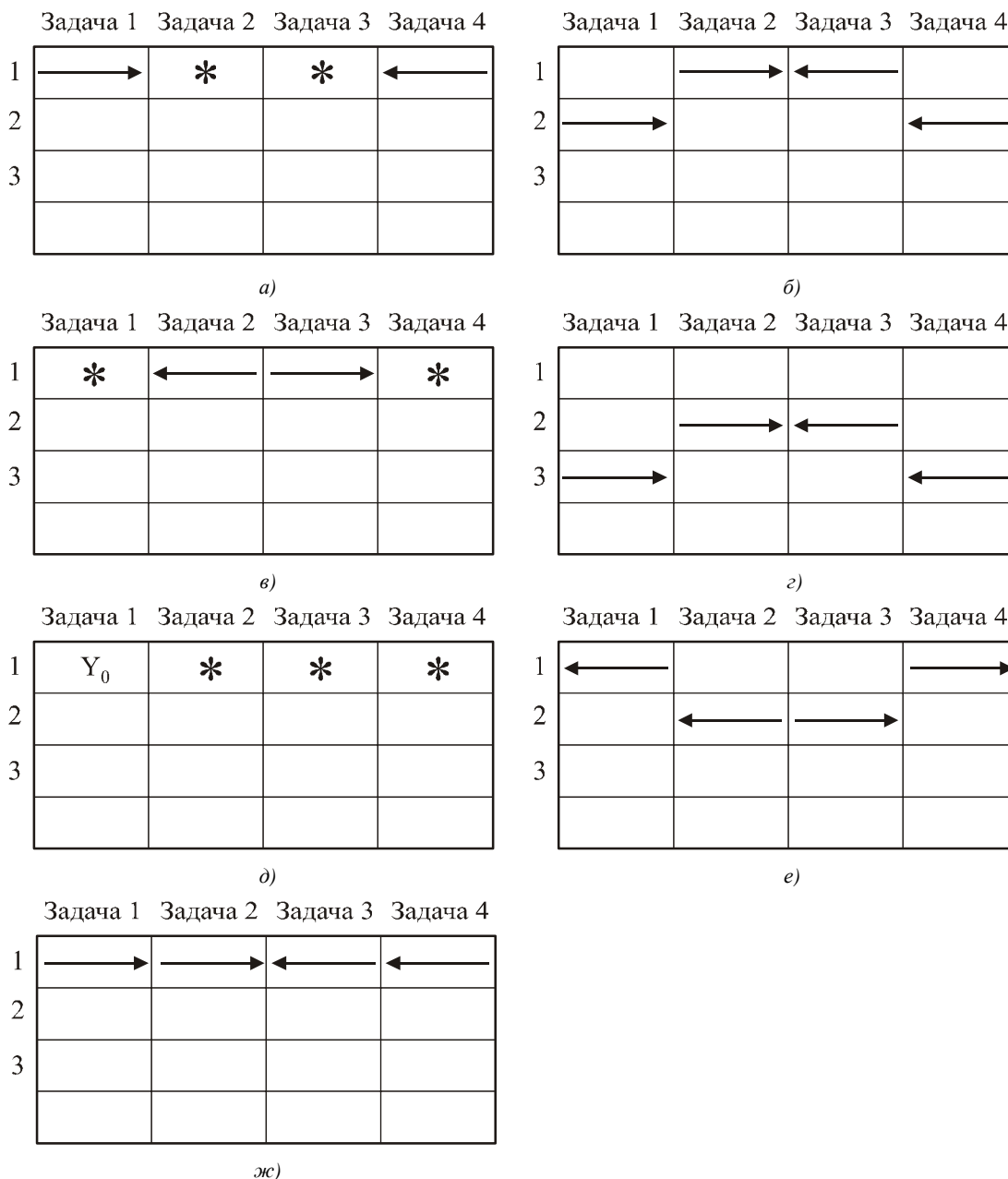


Рис. 3. Этапы вычислений по четырехзадачному параллельному алгоритму, реализующему метод встречных прогонок на двумерной сеточной области

Рассмотрим подробнее этапы вычислений для алгоритма из 4 задач, учитывая сказанное выше о пересылках данных. На первом шаге задачи 1,4 начинают прямой ход прогонки для строки 1, задачи 2,3 простаивают (рис. 3а). На втором – задачи 2,3 продолжают прямой ход для строки 1, задачи 1,4 выполняют прямой ход для строки 2 (рис. 3б). Третий шаг характеризуется простоем задач 1,4 и началом обратного хода для строки 1 задачами 2,3 (рис. 3в). Четвертый шаг – прямой ход, выполняются задачи 1,4 для строки 3, задачи 2,3 для строки 2 (рис. 3г). Пятый шаг – все задачи выполняют обратный ход (рис. 3д). После этого прямой и обратный ходы чередуются, пока задачи 2,3 не произведут прямой ход прогонок для последней строки. Каждый раз по завершении обратного хода задачами 1,4, задача 1 получает значения сеточных функций от задачи 4. Задача 1 вычисляет u_0 и пересылает задачам 1,2,3,4, которые в это время простаивают (рис. 3е). Наконец, все задачи выполняют вычисления u_i (рис. 3ж). На заключительном этапе задачи 2,3 выполняют прямой ход, при этом задачи 1,4 простаивают. Обратный ход выполняется всеми задачами. После чего задачи 2,3 простаивают, задачи 1,4 завершают обратный ход прогонок для последней строки сеточной области.

Теоретическая оценка ускорения данного алгоритма:

$$S = \frac{2C_1MN\tau_a}{\frac{1}{2}C_1MN\tau_a + 6N\tau_k + C_1\frac{M}{4}\tau_a}$$

Обобщим данный подход на произвольное число задач p . В течение первых $p/2$ шагов производится только прямой ход прогонки. Задача l ($l \leq p/2$) простаивает $l-1$ шаг, затем выполняет прямой ход прогонки. После этого происходит чередование прямого и обратного хода, причем задача l простаивает $p/2-l$ шагов обратного хода. Далее чередование выполняется без простоев. Причем, когда задача 1 производит прямой ход для i строки, то задача l производит прямой ход для строки $i-l+1$. Когда задача 1 начинает обратный ход для строки i , задача l начинает обратный ход для $i+l-1$ строки. Каждый раз по завершении обратного хода задача 1, получая значения сеточных функций, вычисляет u_0 и пересылает остальным задачам, которые в это время простаивают. Наконец, все задачи выполняют вычисления u_i . Последние шаги связаны с простоями задач с меньшими номерами во время прямого хода ($l \leq p/2$) и с большими номерами во время обратного хода. Аналогично проводятся рассуждения для $l > p/2$. Ускорение такого алгоритма:

$$S = \frac{2C_1MN\tau_a}{\frac{C_1MN\tau_a}{p} + 6N\tau_a + \frac{(p-2)}{2p}C_1\frac{M}{4}\tau_a + (1,5p-2)\tau_k}$$

Заметим, что суммарное время простоев любой задачи одинаково, т.к. оно связано с ожиданием выполнения арифметических операций и коммуникаций между задачами алгоритма.

Состоятельность данного алгоритма подтверждается результатами вычислительных экспериментов для случаев 4 и 6 задач.

Приведенные результаты экспериментов согласуются с законом Амдала. Все вычислительные эксперименты проводились на кластере, состоящем из 6 ЭВМ AMD Sem-

pron (tm) Processor 3000+ (1,8 GHz), связанных локальной сетью Ethernet.

Сравнение известных и предложенного алгоритмов

На общее время решения задачи влияет много факторов. Главные из них – время выполнения арифметических операций и время взаимодействия с памятью. Не останавливаясь подробно на вопросах, связанных с организацией памяти ЭВМ, отметим, что оптимальным является вариант использования лишь «быстрой» оперативной памяти.

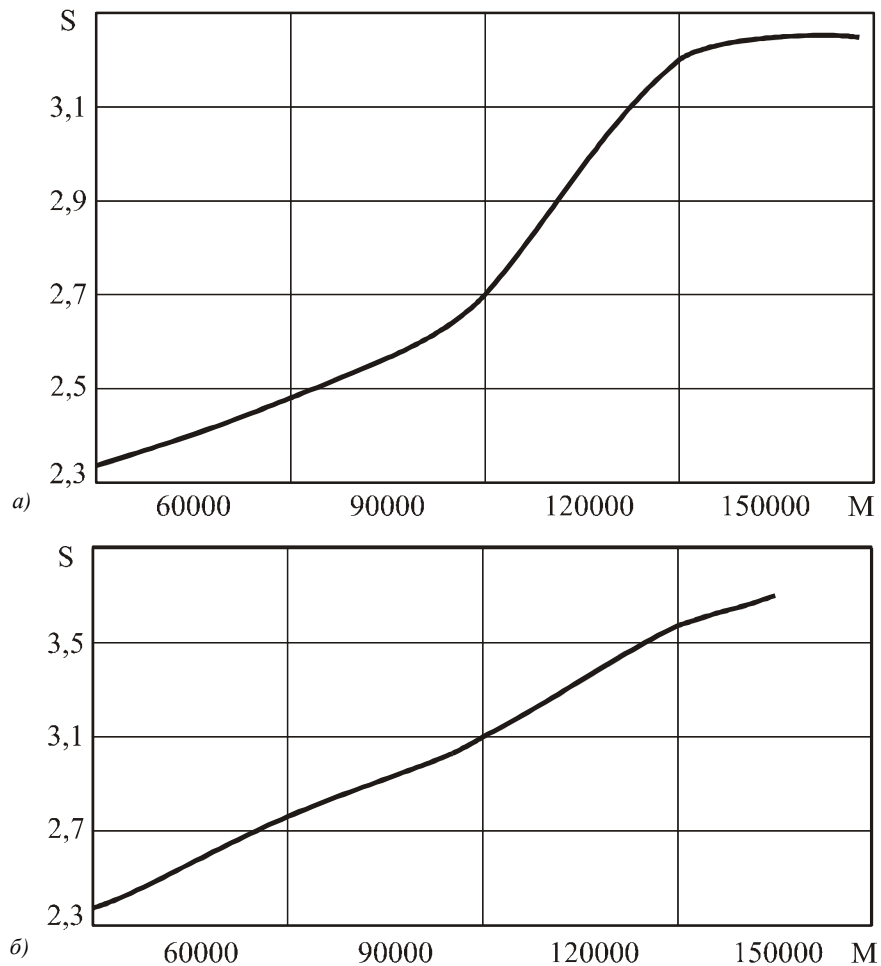


Рис. 4. Зависимость ускорения параллельного вычислительного процесса от размера сеточной области для 4-х задач (а), 6 задач (б)

Традиционно, в силу тенденции к созданию аппаратно-независимых программ понятие размера занимаемой памяти считалось несущественным. Однако известно, что в ряде случаев указанная особенность может становится определяющей [7]. Стремление максимально увеличить размер памяти, необходимой для выполнения программы, может привести к использованию дополнительной «медленной» памяти, которая чаще всего реализуется на жестких дисках. Ее размеры во много раз больше быстрой памяти, но время доступа также во много раз больше.

Тот факт, что операции выполняются только над данными, находящимися в оперативной памяти, приводит к необходимости многократно переносить данные из «медленной» памяти в «быструю» и обратно. Неудачная организация обменов может привес-

ти к тому, что время осуществления обменов между быстрой и медленной памятью будет превосходить время выполнения операций алгоритма в десятки, сотни и даже тысячи раз, что может привести к увеличению времени выполнения вычислений по алгоритму в несколько раз.

Произведем сравнение данного алгоритма и алгоритма, приведенного в работе [4], учитывая вышесказанное. Первые p шагов в этих алгоритмах совпадают. Далее в приведенном алгоритме наблюдается чередование прямого и обратного ходов, что позволяет хранить значения прогоночных коэффициентов и сеточных функций лишь на протяжении p шагов. В алгоритме из работы [4] они сохраняются в течение $2N+p$ шагов. Ясно, что с ростом N возможность увеличения времени выполнения программы возрастает. Таким образом, требования приведен-

ного алгоритма к памяти по объему в $2N/p$ раз меньше по сравнению с алгоритмом из работы [4].

По сравнению с алгоритмами из [2,3] разработанный алгоритм позволил снять ограничения на масштабируемость, возможность использования для решения разностных уравнений с циклическими граничными условиями (по сравнению [6]), при этом скорость вычислений оказалась выше (по сравнению с [4]).

Выводы

Применение функциональной декомпозиции к синтезу параллельных алгоритмов, основанных на методе циклических встречных прогонок для решения сеточных уравнений трехдиагонального вида, приводит к алгоритмам, обладающим высоким ускорением и эффективностью. Для двух задач алгоритм позволяет получить ускорение 1,9; для 4 задач с 1D разбиением – 3,2; для 6 задач с 1D разбиением – 3,6.

Представляется целесообразным развитие приведенного метода и синтеза алгоритма с циклическим разбиением двумерной (2D) области.

Библиографический список

1. Методы компьютерной оптики (Издание второе, исправленное) / под ред. В.А. Соифера – М.: Физматлит, 2003. – 688 с.

2. **Миренков, Н.Н.** Параллельные алгоритмы для решения задач на однородных вычислительных системах [текст] / Н.Н. Миренков // Вычислительные системы – Ново-

сибирск: ИМ СО АН СССР. – 1973. – Вып. 57. – С. 3-32.

3. **Четверушкин, Б.Н.** О возможности реализации квазигидродинамической модели полупроводниковой плазмы на многопроцессорных вычислительных системах [текст] / Л.Ю. Бирюкова, Б.Н. Четверушкин // Математическое моделирование. – 1991. – Т.3, №6. – С. 61-71.

4. **Головашкин, Д.Л.** Параллельные алгоритмы метода циклической прогонки [текст] / Д.Л. Головашкин, М.В. Филатов // Компьютерная оптика. – 2005. – №27. – С. 123-130.

5. **Самарский А.А.** Методы решения сеточных уравнений [текст] / А.А. Самарский, Е.С. Николаев – М.: Наука, 1978. – 561 с.

6. **Головашкин, Д.Л.** Параллельные алгоритмы решения сеточных уравнений трехдиагонального вида, основанного на методе встречных прогонок [текст] / Д.Л. Головашкин // Математическое моделирование. – 2005. – Т. 17, № 11. – С. 118– 128.

7. **Воеводин, В.В.** Параллельные вычисления [текст] / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 608 с.

8. **Голуб, Дж.** Матричные вычисления [текст] / Дж. Голуб, Ван Лоун – М.: Мир, 1999. – 548 с.

9. **Ортега, Джеймс М.** Введение в параллельные и векторные методы решения линейных систем [текст] / Джеймс М. Ортега, перевод с англ. Х.Д. Икрамова, И.Е. Капорина; под ред. Х.Д. Икрамова – М.: Мир, 1991. – 364 с.

Сведения об авторе

Логанова Лилия Владимировна, Самарский государственный аэрокосмический университет имени академика С.П. Королева, старший преподаватель кафедры «Техническая кибернетика», e-mail: tk@smr.ru.