

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ МЕТОДА ЦИКЛИЧЕСКОЙ ПРОГОНКИ

Головашкин Д.Л.¹, Филатов М. В.²

¹ Институт систем обработки изображений РАН

² Самарский государственный аэрокосмический университет

Аннотация

Работа посвящена построению параллельных алгоритмов метода циклической прогонки для решения сеточных уравнений ленточного вида. Рассмотрены два подхода к разбиению сеточной области: линейная и циклическая декомпозиции. Каждый подход применен к построению алгоритмов с использованием правой и встречных циклических прогонок. Произведено аналитическое и экспериментальное сравнение предложенных алгоритмов, выявлены их достоинства и недостатки.

Введение

Математическое моделирование находит все большее применение в различных отраслях науки. Это связано как с развитием математического аппарата в соответствующих отраслях, так и с совершенствованием вычислительной техники. В свою очередь, новые модели и вычислительные ресурсы нуждаются в новых алгоритмах, среди которых наиболее популярными в настоящее время следует признать параллельные.

В частности, многие задачи математической физики с помощью разностного и проекционного методов сводятся к набору сеточных уравнений, записываемых для многомерных областей в виде совокупности систем линейных алгебраических уравнений (СЛАУ) ленточного вида. При этом круговые, сферические и цилиндрические подобласти характеризуются матрицей СЛАУ, содержащей, кроме ленты на центральных диагоналях, ненулевые элементы в верхнем правом и нижнем левом углах. Для решения таких СЛАУ традиционно применяется метод циклической прогонки [1] (различные итерационные методы остаются за рамками настоящей работы).

Задавшись целью синтезировать параллельные алгоритмы этого метода, авторы обратились к параллельным алгоритмам решения ленточных СЛАУ без ненулевых элементов вне ленты, как к примерам решения аналогичной задачи. Из алгоритмов, основанных на декомпозиции области [2], циклической редукции [3] и функциональной декомпозиции [4,5], наиболее эффективными [3] были признаны последние. Их развитию для метода циклической прогонки и посвящена настоящая работа.

1. Параллельный алгоритм для одномерной сеточной области

Иллюстрируя основные идеи работы рассмотрим параллельный алгоритм для одномерной области, основанный на применении встречных циклических прогонок. Не обладая свойством масштабируемости [6] он полезен при составлении алгоритмов для многомерных областей.

Функциональная декомпозиция метода циклической прогонки задается спецификой использования в ней метода обычных встречных прогонок [1].

Произведем разбиение одномерной области данных ω^1 на равные подобласти ω_1^1 и ω_2^1 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_2^1 .

Вычисления по алгоритму разделим на семь шагов: прямой ход прогонок, обмен данными, обратный ход, пересылка дополнительных данных, вычисление коэффициента y_0 [1], пересылка коэффициента и нахождение окончательного решения.

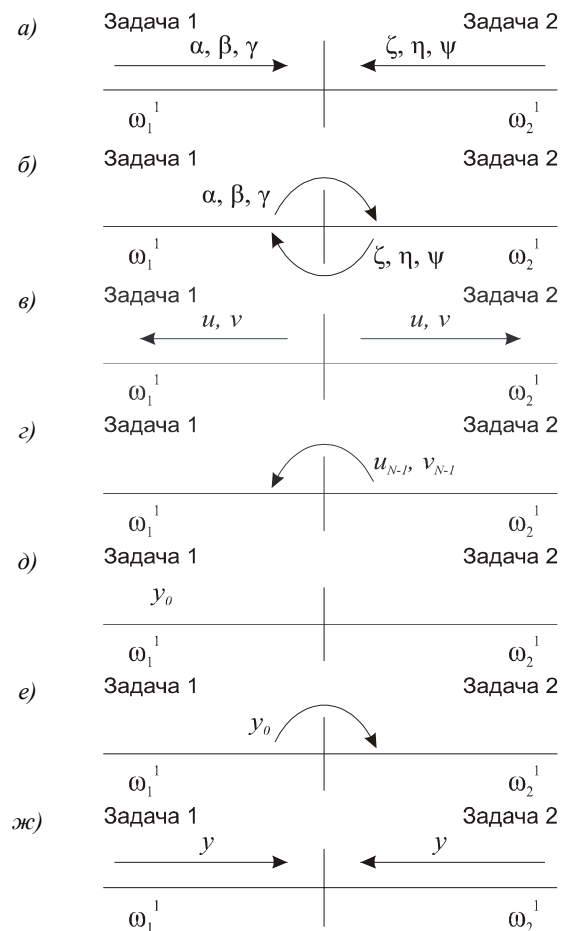


Рис. 1 Этапы вычислений по двухзадачному параллельному алгоритму, реализующему метод встречных циклических прогонок (а-в - встречный ход прогонок, г - обмен крайними коэффициентами, д - вычисление y_0 , е - обмен коэффициентами, ж - вычисление окончательного решения).

Во время прямого хода (рис. 1а) прогоночные коэффициенты вычисляются одновременно с двух сторон по направлению к центру области: первая задача находит α, β, γ , вторая - ζ, η, ψ .

Обмен коэффициентами (рис. 1б) необходим для запуска обратного хода, при этом первая задача передает второй тройку коэффициентов $\alpha_{g+1}, \beta_{g+1}, \gamma_{g+1}$, получая $\zeta_{g+1}, \eta_{g+1}, \psi_{g+1}$.

Обратный ход прогонки (рис. 1в) реализуется обеими задачами одновременно.

На следующем шаге (рис. 1г) вторая задача пересылает в первую коэффициенты u_{N-I} и v_{N-I} [1].

Пятый шаг (рис. 1д) характеризуется вычислением y_0 первой задачей (вторая задача простаивает).

Затем полученное значение пересылается второй задаче (рис. 1е).

Вычисление окончательного решения производится обеими задачами одновременно (рис. 1ж) на последнем шаге алгоритма.

Ускорение [3] приведенного двухзадачного алгоритма оценивается величиной

$$S = \frac{14N\tau_{ap}}{(7N+9)\tau_{ap} + 4\tau_n},$$

где $14N\tau_{ap}$ – длительность вычислений по последовательному алгоритму (τ_{ap} – время производства одной арифметической операции с плавающей точкой), $(7N+9)\tau_{ap}$ – длительность арифметических операций для одной задачи параллельного алгоритма, τ_n – время пересылки одного пакета по сети. При пакетной модели передачи данных время передачи разного объема информации остается константой до тех пор, пока объем передаваемого массива не превысит размер пакета. Здесь это условие соблюдается, в дальнейшем также будем полагать его верным.

2. Параллельные алгоритмы для двумерной области, основанные на методе правой циклической прогонки

В отличие от предыдущего случая, на двумерной сеточной области возможно построение параллельных алгоритмов на основе метода правой циклической прогонки, как это было предложено в [4] для ленточных СЛАУ без ненулевых элементов вне ленты.

2.1 Алгоритм правой циклической прогонки с линейным разбиением сеточной области

Записывая двухзадачный алгоритм, разобьем область ω^2 так (рис. 2), что вычисления циклических прогонок по столбцам области не будут требовать пересылки данных, которая останется необходимой только при работе со строками.

Первый этап вычислений по алгоритму состоит из $I+1$ шагов (рис. 2а-2г), где I – это число строк сеточной области. На первом шаге первая задача начинает прямой ход прогонки для первой строки (рис. 2а), вторая задача простаивает в ожидании прогоночных коэффициентов этой строки области.

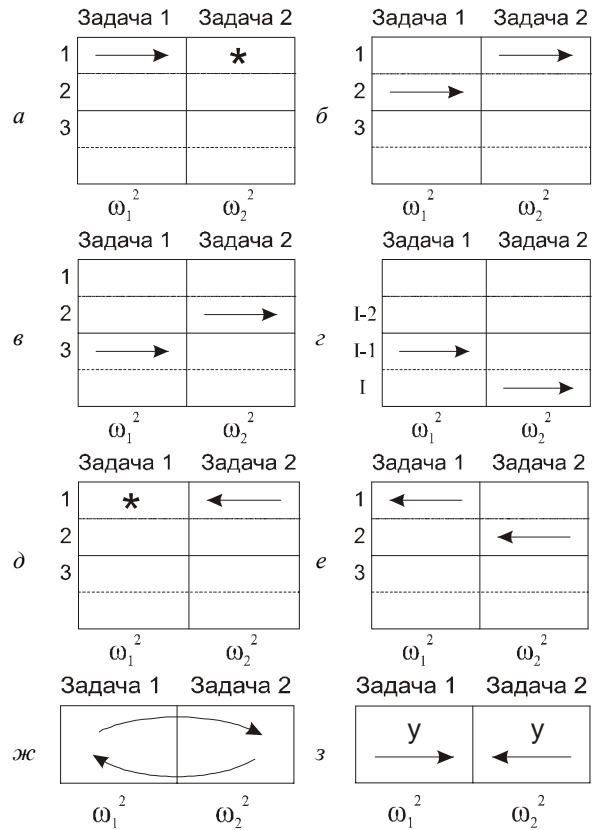


Рис. 2 Этапы вычислений по двухзадачному параллельному алгоритму, реализующему метод правой циклической прогонки на двумерной области с линейным разбиением (а-е – начальные шаги, ж – обмен крайними коэффициентами и вычисление y_0 , з – обмен коэффициентами y_0 и вычисление окончательного решения). Звездочками обозначены простые задач.

Второй шаг обе задачи производят одновременно (рис. 2б), осуществляя прямой ход прогонки для разных строк: вторая задача для первой, первая для второй (функциональная декомпозиция: одновременно решаются разные СЛАУ). Аналогично происходит перебор строк области вплоть до шага I , когда вторая задача производит прямой ход для строки для строки $I-1$, первая для I . На $I+1$ шаге первая задача простаивает, вторая завершает прямой ход правой циклической прогонки для последней строки области.

На втором этапе (рис. 2д, е) вычислений аналогично выполняется обратный ход прогонки для всех строк области. Теперь на первом шаге простаивает первая задача алгоритма, на последнем – вторая.

Третий этап заключается в обмене коэффициентами u и v (для каждой строки) между задачами (рис. 2ж) с последующим вычислением задачей 1 коэффициента y_0 (своего для каждой строки). Найденное значение y_0 пересылается второй задаче.

Четвертый этап – нахождение искомых сеточных функций (рис. 2з). Все вычисления независимы, простои исключены.

Обратим внимание на последнюю задачу. На первом и втором этапах она простаивает. Общая длительность простоев составляет $5N\tau_{ap}$, где N –

размерность сеточной области по горизонтали. Длительность простоев при обратном ходе прогонки составляет $2N\tau_{ap}$.

Простой также присущи третьему этапу при вычислении коэффициента y_0 , их длительность - $6I\tau_{ap}$. Простой, связанные с коммуникациями характеризуются величиной $(3I+1)\tau_n$ - общим временем приема и передачи задачей коэффициентов на первом, втором, третьем и четвертом этапах.

С учетом этого ускорение двухзадачного алгоритма оценивается величиной

$$S = \frac{28IN\tau_{ap}}{14IN\tau_{ap} + 6I\tau_{ap} + 6N\tau_{ap} + (3I+1)\tau_n}.$$

Распространим идею предлагаемого подхода на произвольное количество задач L с линейным разбиением сеточной области по одному направлению. На первом этапе многозадачного алгоритма в течении $I+L-1$ шагов производится прямой ход прогонки. На первом шаге задача 1 работает со строкой 1 (остальные задачи простаивают), на втором задача 1 работает со своей частью строки 2, задача 2 с частью строки 1 (простаивают задачи 3,...,L). Шаг i ($L \leq i \leq L$) характеризуется отсутствием простоев; причем задача l ($1 \leq l \leq L$) производит прямой ход правой циклической прогонки для строки $i-l+1$ подобласти ω_l^2 . С шага $I+1$ начинают простаивать задачи с меньшими номерами.

На втором этапе вычислений аналогично выполняется обратный ход прогонки для всех строк области. Так, на шаге i ($L \leq i \leq L$) задача l ($1 \leq l \leq L$) производит обратный ход правой циклической прогонки для строки $i-L+l$ подобласти ω_l^2 . Первые шаги этапа характеризуются простоями задач с меньшими номерами, последние - с большими.

Третий этап заключается в обмене коэффициентами u и v (каждой строки) между задачами 1 и L с последующим вычислением первой задачей коэффициента y_0 (своего для каждой строки). Задачи 2,...,L-1 простаивают до получения y_0 .

На четвертом этапе в течении $L-1$ шагов производится передача вектора значений y_0 и вычисление y . На первом шаге задача 1 передает вектор значений y_0 задаче 2 и начинает вычисление y . На втором шаге задача 2 передает вектор значений y_0 задаче 3 и начинает вычисление y . На шаге $L-1$ задача $L-1$ передает задаче L вектор значений y_0 и обе задачи начинают вычисление y . Для данного алгоритма характерна топология коммуникаций - кольцо.

Изучая длительность простоев L - задачного алгоритма, обратим внимание на последнюю задачу. На первом этапе она простаивает первые $L-1$ шагов, после которых работает без остановок. Длительность этих шагов составляет

$$8N \frac{L-1}{L} \tau_{ap} + (L-2)\tau_n,$$

где N - размерность сеточной области по горизонтали; $(L-2)\tau_n$ - время пересылок, в которых последняя задача не принимает участия; $8N\tau_{ap}$ - длительность прямого хода прогонки.

На втором этапе задача L простаивает заключительные $L-1$ шагов. Время этого простоя:

$$4N \frac{L-1}{L} \tau_{ap} + (L-2)\tau_n,$$

где $4N\tau_{ap}$ - длительность обратного хода прогонки.

На четвертом этапе задача L простаивает первые $L-2$ шагов. Время этого простоя:

$$(L-2)\tau_n.$$

Простой также присущи третьему этапу при вычислении коэффициента y_0 , их длительность - $6I\tau_{ap}$. С учетом этого ускорение L - задачного алгоритма оценивается величиной

$$S = \frac{28IN\tau_{ap}}{\frac{28IN}{L}\tau_{ap} + 6I\tau_{ap} + 12N \frac{L-1}{L} \tau_{ap} + 3(L-2)\tau_n + (3I+1)\tau_n},$$

где $(3I+1)\tau_n$ - общее время приема и передачи задачей коэффициентов на первом, втором, третьем и четвертом этапах.

2.2 Алгоритм правой циклической прогонки с циклическим разбиением сеточной области

Стремясь избежать простоев, циклически разобьем ω^2 на блоки в шахматном порядке (рис. 3), ведя далее их нумерацию по блочным строкам и столбцам. Пусть задаче 1 принадлежат блоки на главной диагонали блочной области, задаче 2 - блоки, циклически смещенные на одну позицию влево от главной диагонали, задаче L - циклически смещенные на $L-1$ позицию влево.

	1	2	3	4	...	L
1	ω_1^2	ω_L^2	ω_{L-1}^2	ω_{L-2}^2	...	ω_2^2
2	ω_2^2	ω_1^2	ω_L^2	ω_{L-1}^2	...	ω_3^2
3	ω_3^2	ω_2^2	ω_1^2	ω_L^2	...	ω_4^2
4	ω_4^2	ω_3^2	ω_2^2	ω_1^2	...	ω_5^2
...
L	ω_L^2	ω_{L-1}^2	ω_{L-2}^2	ω_{L-3}^2	...	ω_1^2

Рис. 3 Разбиение двумерной области в случае L -задачного алгоритма правой циклической прогонки

Двухзадачный вариант алгоритма характеризуется циклическим разбиением, представленным на рис. 4. На первом этапе такого алгоритма первая задача производит прямой ход прогонки для всех строк блока

(1, 1), вторая задача для (2, 1). На втором шаге она работает с блоком (1, 2) (предварительно получив необходимые прогоночные коэффициенты от другой задачи), вторая задача закончит прямой ход прогонки для второй блочной строки (блок (2, 2)).

	1	2
1	ω_1^2	ω_2^2
2	ω_2^2	ω_1^2

Рис. 4. Разбиение двумерной области в случае двухзадачного алгоритма правой циклической прогонки

Обратный ход прогонки (второй этап) задачи проведут с теми же блоками в обратной последовательности: первая задача начнет с (2,2), завершит с (1,1); вторая начнет с (1,2), завершит с (2,1).

Третий этап вычислений характеризуется обменом коэффициентами u и v между задачами 1 и 2 с последующим вычислением коэффициентов y_0 . А именно: первая задача получает из второй для блока (1,2) крайние значения u и v , вторая из первой - для блока (2,2) крайние значения u и v . Затем в каждой задаче вычисляется y_0 .

Четвертый этап алгоритма. Первая задача производит вычисление y для всех строк блока (1, 1), вторая задача для (2, 1). На втором шаге она работает с блоком (1, 2) (предварительно получив вектор коэффициентов y_0 от задачи 2), вторая задача с блоком (2, 2) (предварительно получив вектор коэффициентов y_0 от задачи 1).

Циклическая прогонка по блочным столбцам выполняется аналогично.

Исследуя длительность пересылок, найдем $6\tau_n$ как общее время пересылок на первом, втором и четвертом этапах (по строкам и столбцам), $2\tau_n$ - время пересылки коэффициента y_0 .

Ускорение алгоритма

$$S = \frac{28IN\tau_{ap}}{14IN\tau_{ap} + 8\tau_n}.$$

Идея циклического разбиения легко обобщается на случай L задач. На первом этапе алгоритма (прямой ход по блочным строкам) задача l ($1 \leq l \leq L$) начинает вычисления с блока ($l, 1$) и продолжает их, циклически смещаясь по своей диагонали на одну блочную позицию вниз и влево. Обратный ход производится по тем же подобластям с возвращением к блоку ($l, 1$) в конце второго этапа алгоритма.

Третий этап вычислений для задачи l характеризуется обменом коэффициентов u и v с задачей $l-1$ (первая задача обменивается с последней), последующим вычислением y_0 и их пересылкой. А именно: первая задача получает из последней крайние значения u и v для блока ($1, L$), вторая из $L-1$ крайние значения u и v для блока ($2, L-1$). Затем в каждой за-

даче вычисляется y_0 . Далее производится одновременное нахождение искомых сеточных функций обеими задачами алгоритма.

На следующем этапе алгоритма задача l начинает вычисления y с блока ($l, 1$) и продолжает их в течение L шагов, циклически смещаясь по своей диагонали на одну блочную позицию вниз и влево, при этом пересылая вектор коэффициентов y_0 задаче $l-1$ (в случае когда $l=1$ пересылаем задаче L).

Для данного алгоритма характерна топология коммуникаций – кольцо.

Хотя предложенный алгоритм в отличие от предыдущего (пункт 2.1) лишен простоев, длительность пересылок возросла в два раза.

Получим $6(L-1)$ как общий объем пересылок на первом, втором и четвертом этапах (по строкам и столбцам), $2\tau_n$ - время пересылки коэффициента y_0 . Учитывая, что пересылки пакетные, найдем ускорение равным:

$$S = \frac{28IN\tau_{ap}}{\frac{28IN}{L}\tau_{ap} + 6(L-1)\tau_n + 2\tau_n}$$

3. Параллельные алгоритмы для двумерной области, основанные на методе встречных циклических прогонок

Сравнение методов правой и встречных прогонок при составлении параллельных алгоритмов для решения ленточных СЛАУ без ненулевых элементов вне ленты показывает [5] превосходство метода встречных прогонок, что побудило авторов к записи алгоритмов на его основе.

3.1 Алгоритм встречных циклических прогонок с линейным разбиением сеточной области

Составляя четырехзадачный алгоритм (двухзадачный случай аналогичен алгоритму из пункта 1) осуществим линейное разбиение области ω^2 , как показано на рис. 5а.

Первый этап состоит из $I+1$ шага прямого хода встречных прогонок (рис. 5а-5в). На первом шаге задачи 1 и 4 начинают встречный ход для первой строки области (рис. 5а), задачи 2 и 3 простаивают в ожидании прогоночных коэффициентов этой строки. Второй шаг все задачи производят вместе (рис. 5б), осуществляя прямой ход встречных прогонок для разных строк: вторая и третья задача - для первой строки, первая и четвертая - для второй. Шаг I связан с прямым ходом прогонок для последней (исполняют задачи 1 и 4) и предпоследней (исполняют задачи 2 и 3) строки. Шаг $I+1$ завершают задачи 2 и 3, задачи 1 и 4 простаивают.

Второй этап алгоритма (рис. 5г, д) связан с производством обратного хода встречных прогонок. Начиная с последней строки, области задачи 2 и 3 поднимаются по строкам вверх, простаивая последний шаг этапа. Задачи 1, 4 вступают в вычисления по второму этапу с задержкой на один шаг, вызванной ожиданием прогоночных коэффициентов для строки I .

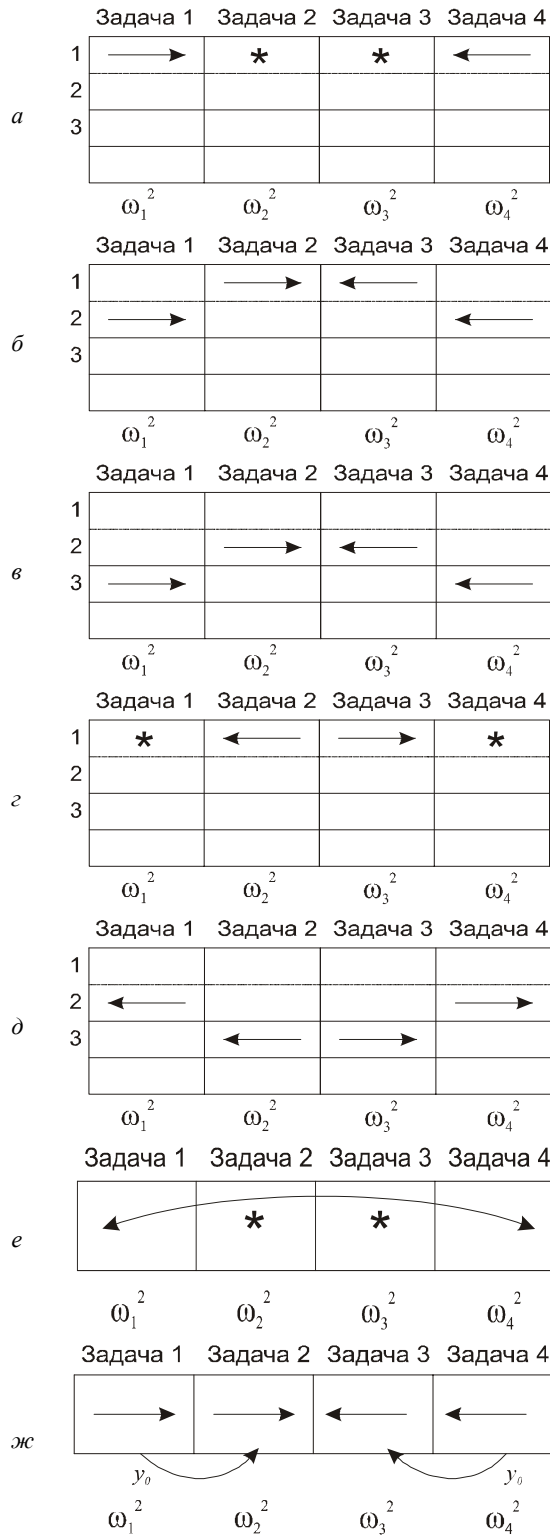


Рис. 5. Этапы вычислений по четырехзадачному параллельному алгоритму, реализующему метод встречных циклических прогонок на двумерной области с линейным разбиением (а-д - начальные шаги алгоритма, е - обмен крайними коэффициентами u и v и вычисление y_0 , ж - обмен коэффициентами y_0 и вычисление окончательного решения). Звездочками обозначены простои задач

Третий этап алгоритма связан с обменом коэффициентами u и v между первой и четвертой задачами (рис. 5е) и последующим вычислением первой

и четвертой задачей коэффициента y_0 , своего для каждой строки (при этом остальные задачи простаивают).

Четвертый этап – рассылка найденных значений y_0 из крайних задач и вычисление значений искомых сеточных функций (рис. 5ж).

Обратим внимание на последнюю задачу. На первом и четвертом этапах она простаивает. Общая длительность простоев составляет $5N\tau_{ap}/2$

Длительность простоев при обратном ходе прогонки составляет $N\tau_{ap}$. Длительность вычислений на четвертом этапе $N\tau_{ap}/2$.

Простои также присущи третьему этапу при вычислении коэффициента y_0 , их длительность – $6I\tau_{ap} \cdot (3I+1)\tau_n$ – общее время приема и передачи задач коэффициентов на первом, втором, третьем и четвертом этапах. С учетом этого ускорение 4-задачного алгоритма оценивается величиной

$$S = \frac{28IN\tau_{ap}}{\frac{14IN}{2}\tau_{ap} + 6I\tau_{ap} + 3N\tau_{ap} + (3I+1)\tau_n}$$

Распространим идею этого подхода на четное количество задач L . На первом этапе такого алгоритма производится $I+L/2-1$ шагов встречных циклических прогонок. Максимальными простоями на первых шагах этапа характеризуются задачи $L/2$ и $L/2+1$, которые не участвуют в вычислениях вплоть до $L/2$ шага. Затем до шага $I+1$ простоев нет, задача l (пусть $1 \leq l \leq L/2$) производит прямой ход для своей подобласти строки $i-l+1$ (номер шага i меняется от $L/2$ до I). Аналогично проводятся рассуждения для задач $L/2+1 \leq l \leq L$. На заключительных $L/2-1$ шагах в простои вовлекаются задачи $1, L; 2, L-1; \dots, L/2-1, L/2+2$: по паре задач за один шаг.

Вычисления по второму этапу (обратный ход встречных прогонок) выполняются в обратной последовательности шагов.

Третий этап – обмен коэффициентами u и v между первой и последней задачами, с последующим вычислением первой и последней задачей значений y_0 для каждой строки области.

На четвертом этапе в течении $L/2-1$ шагов производится передача вектора значений y_0 и вычисление y . На первом шаге задача 1 передает вектор значений y_0 задаче 2, задача L задаче $L-1$ и начинают вычисление y . На втором шаге задача 2 передает вектор значений y_0 задаче 3, задача $L-1$ – задаче $L-2$ и начинает вычисление y . На шаге $L/2-1$ задача $L/2-1$ передает вектор значений y_0 задаче $L/2$, , задача $L/2+2$ – задаче $L/2+1$ и все четыре задачи участвовавшие на этом шаге в обмене начинают вычисление y .

Для данного алгоритма характерна топология коммуникаций – кольцо.

Изучая длительность простоев, обратимся к анализу работы задачи $L/2$. На первом этапе она простаивает первые $L/2-1$ шагов, длительность которых

$$4N \frac{L-2}{L} \tau_{ap} + \left(\frac{L}{2} - 2\right) \tau_n.$$

Простой исследуемой задачи на втором этапе (заключительные $L/2-1$ шагов) оцениваются как

$$2N \frac{L-2}{L} \tau_{ap} + \left(\frac{L}{2} - 2\right) \tau_n.$$

Простой на третьем этапе связан с вычислением и ожиданием приема y_0 . Длительность такого простоя составит $6I\tau_{ap}$.

На четвертом этапе простаивает $L/2-2$ шагов, длительность которых

$$\left(\frac{L}{2} - 2\right) \tau_n$$

Следовательно, ускорение L -задачного алгоритма оценивается величиной

$$S = \frac{28IN\tau_{ap}}{\frac{28IN}{L}\tau_{ap} + 6I\tau_{ap} + 6N\frac{L-2}{L}\tau_{ap} + 3\left(\frac{L}{2} - 2\right)\tau_n + (3I+1)\tau_n},$$

где $(3I+1)\tau_n$ – общее время приема и передачи задачей коэффициентов на первом, втором, третьем и четвертом этапах.

3.2 Алгоритм встречных циклических прогонок с циклическим разбиением сеточной области

Записывая восьмизадачный алгоритм, разобьем сеточную область ω^2 на блоки в шахматном порядке (рис. 6). В отличие от циклического разбиения, свойственного алгоритму из пункта 2.2, в данном случае сеточная область делится на четверти, дальнейшее разбиение которых происходит аналогично разбиению из пункта 2.2.

	1	2	3	4
1	ω_1^2	ω_4^2	ω_8^2	ω_5^2
2	ω_2^2	ω_1^2	ω_5^2	ω_6^2
3	ω_3^2	ω_2^2	ω_6^2	ω_7^2
4	ω_4^2	ω_3^2	ω_7^2	ω_8^2

Рис. 6. Циклическое разбиение двумерной области для восьмизадачного алгоритма метода встречных циклических прогонок

Вычисления по восьмизадачному алгоритму разобьем на три этапа. На первом этапе такого алгоритма первая задача производит прямой ход прогонки для всех строк блока (1, 1), вторая задача – для (2, 1), третья – для блока (3, 1) и четвертая – для (4, 1). Восьмая производит вычисления над блоком (1, 4), седьмая – над (2, 4), шестая – над (3, 4), пятая – над (4, 4). На втором шаге, задачи обмениваются коэффициентами и продолжают вычисления над следующими блоками: первая – для блока (1, 2), вто-

торая – для (2, 2), третья – для (3, 2), четвертая – для (4, 2), восьмая производит вычисления над блоком (1, 3), седьмая – над (2, 3), шестая – над (3, 3), пятая – над (4, 3). Третий шаг – это обмен коэффициентами между следующими задачами: 4 и 8, 1 и 5, 2 и 6, 3 и 7.

Обратный ход прогонки (второй этап) задачи проведут с теми же блоками, но в обратной последовательности.

Третий этап вычислений характеризуется обменом коэффициентами u и v между задачами: 4 и 8, 1 и 5, 2 и 6, 3 и 7; с последующим вычислением y_0 для каждой строки каждого блока во всех задачах.

На четвертом этапе первая задача производит вычисление y для всех строк блока (1, 1), вторая задача – для (2, 1), третья – для блока (3, 1) и четвертая – для (4, 1). Восьмая производит вычисления над блоком (1, 4), седьмая – над (2, 4), шестая – над (3, 4), пятая – над (4, 4). На втором шаге, задачи обмениваются векторами коэффициентов y_0 и продолжают вычисления над следующими блоками, первая – для блока (1, 2), вторая – для (2, 2), третья – для (3, 2), четвертая – для (4, 2), восьмая производит вычисления над блоком (1, 3), седьмая – над (2, 3), шестая – над (3, 3), пятая – над (4, 3).

Исследуя коммуникации алгоритма, отметим, что $12\tau_n$ – общее время пересылок на первом, втором и четвертом этапах (по строкам и столбцам), $2\tau_n$ – время пересылки коэффициента y_0 .

Ускорение алгоритма

$$S = \frac{28IN\tau_{ap}}{\frac{7IN}{2}\tau_{ap} + 14\tau_n}$$

Идея циклического разбиения легко обобщается на случай L задач, когда $\log_2 L$ – целое число. На первом этапе алгоритма (прямой ход по блочным строкам) задача l ($1 \leq l \leq L/2$) начинает вычисления с блока $(l, 1)$ и продолжает их в течение $L/4$ шагов, циклически смещаясь по своей диагонали на одну блочную позицию вниз и влево. Для задачи l ($L/2 < l \leq L$) вычисления начинаются с блока $(l-L/2, L)$ и продолжают в течении $L/4$ шагов, циклически смещаясь по диагонали на одну блочную позицию вниз и вправо. Обратный ход производится по тем же подобластям с возвращением к блоку $(l, 1)$ и (l, L) в конце второго этапа алгоритма.

Третий этап вычислений для задачи l характеризуется обменом коэффициентов u и v с задачей $l-L/2$, с последующим вычислением y_0 . Далее производится одновременное нахождение искомым сеточных функций всеми задачами алгоритма. На этом этапе алгоритма задача l ($1 \leq l \leq L/2$) начинает вычисления y с блока $(l, 1)$ и продолжает их в течение $L/4$ шагов, циклически смещаясь по своей диагонали на одну блочную позицию вниз и влево, при этом пересылая вектор коэффициентов y_0 задаче $l-1$ (в случае когда $l=1$ пересылаем задаче $L/2$). Для задачи l ($L/2 < l \leq L$) вычисления начи-

наются с блока $(l-L/2, L)$ и продолжаются в течении $L/4$ шагов, циклически смещаясь по диагонали на одну блочную позицию вниз и вправо, при этом вектор коэффициентов y_0 пересылается задаче $l-1$ (в случае, когда $l=L/2+1$ пересылаем задаче L).

Для данного алгоритма характерна топология коммуникаций – гиперкуб.

Найдем $3L/2$ как общий объем пересылок на первом, втором и четвертом этапах (по строкам и столбцам), $2\tau_n$ – время пересылки коэффициента y_0 .

Учитывая, что пересылки рассчитывались в рамках пакетной модели коммуникаций, получим искомое ускорение равным

$$S = \frac{28IN\tau_{ap}}{\frac{28IN}{L}\tau_{ap} + \frac{3L}{2}\tau_n + 2\tau_n}.$$

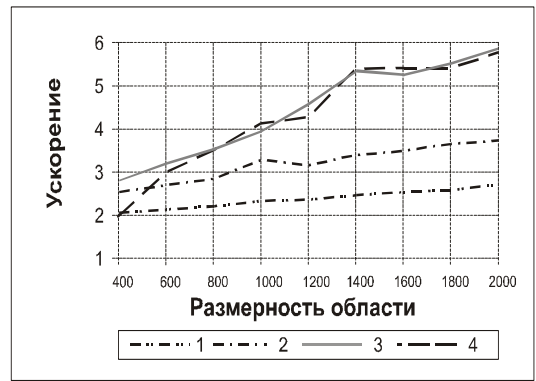
4. Сравнение предложенных алгоритмов

Произведем сравнение четырех подходов к записи алгоритмов для двумерной области.

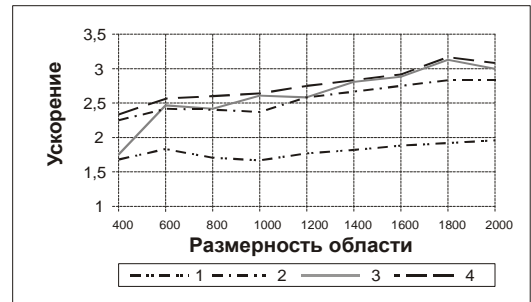
Очевидно, на вытянутых прямоугольных областях ($L \gg M$) предпочтение следует отдавать алгоритмам с одномерным (линейным) разбиением, характеризующимся в этом случае меньшим объемом пересылаемых данных, а для квадратных областей выгоднее алгоритмы с циклическим разбиением. Следовательно, уместно сравнивать алгоритмы внутри данной классификации.

Среди алгоритмов, характеризующихся одномерным разбиением, наибольшим ускорением и эффективностью обладает алгоритм, сформулированный в пункте 3.1 настоящей работы. Будучи равен по объему вычислений и количеству пересылаемых данных с алгоритмом, использующим правую прогонку (пункт 2.1), он характеризуется вдвое меньшей длительностью простоев. К недостаткам обсуждаемого алгоритма из 3.1 можно отнести наличие простоев (впрочем, не играющее существенной роли в приведенном сравнении в силу их кратковременности по сравнению с длительностью арифметических операций и коммуникаций) и требование на четность к числу процессоров вычислительной системы (также несущественное, учитывая, что современные кластеры, как правило, состояются из двухпроцессорных ЭВМ).

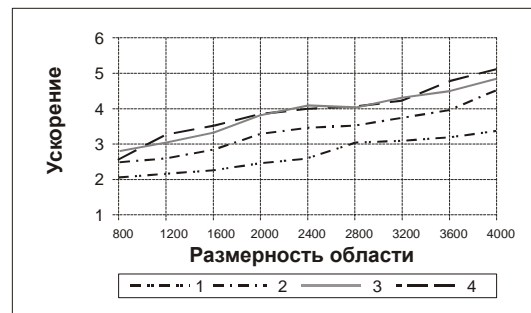
Проводя изучение алгоритмов с циклическим разбиением, для простоты положим $I=N$. Тогда алгоритм из 3.2 будет отличать от алгоритма из 2.2 четверо меньший объем коммуникаций. Недостатками синтезированного алгоритма из 3.2 следует признать ограничение на количество требуемых процессоров (степень двойки) и высокое требование к топологии коммуникаций вычислительной системы (гиперкуб). Однако большинство современных кластеров удовлетворяют таким требованиям.



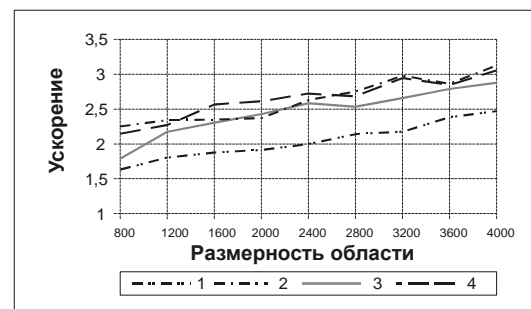
а



б



в



г

Рис. 7. Результаты вычислительных экспериментов: а - для восьми задач на квадратной области, б - для четырех задач на квадратной области, в - для восьми задач на вытянутой прямоугольной области, г - для четырех задач на вытянутой прямоугольной области. Линии на графике: 1 - параллельный алгоритм правой циклической прогонки с линейным разбиением, 2 - встречных циклических прогонок с линейным разбиением, 3 - правой циклической прогонки с циклическим разбиением и 4 - встречных циклических прогонок с циклическим разбиением

Переходя к сравнению параллельных вычислительных процессов, порожденных представленными алгоритмами, отметим, что благодаря отсутствию простоев и меньшему объему коммуникаций (данные передаются блоками), наилучшие результаты на квадратных областях продемонстрировали алгоритмы с циклическим разбиением области. При небольшом количестве задач на сильно вытянутых прямоугольных областях, большую эффективность показывает параллельный алгоритм встречной циклической прогонки с линейным разбиением области.

Заключение

Применение функциональной декомпозиции к синтезу параллельных алгоритмов, основанных на методах правой и встречных прогонок для решения сеточных уравнений трехдиагонального вида, оправдано и приводит при указанных условиях к алгоритмам, обладающим высоким ускорением и эффективностью.

Представляется целесообразным развитие данного подхода к синтезу алгоритмов для решения ленточных систем с большим количеством диагоналей.

Благодарность

Работа выполнена при поддержке Российско-американской программы “Фундаментальные исследования и высшее образование” (BRHE), а также Фонда содействию отечественной науке, гранта Президента РФ (N НШ – 1007.2003.01) и гранта РФФИ (N 04-07-90149).

Литература

1. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.:Наука, 1978. – 561 с.
2. Голуб Дж., Ван Лоун Ч. Матричные вычисления М.:Мир, 1999, 548 с.
3. Ортега Джеймс М. Введение в параллельные и векторные методы решения линейных систем/ Перевод с англ. Х.Д. Икрамова, И.Е. Капорина; под ред. Х.Д. Икрамова. – М.: Мир, 1991. – 364 с.
4. Миренков Н.Н. Параллельные алгоритмы для решения задач на однородных вычислительных системах// Вычислительные системы/ ИМ СО АН СССР. – Новосибирск. –1973. –Вып. 57. –С. 3-32
5. Головашкин Д.Л. Применение метода встречных прогонок для синтеза параллельного алгоритма решения сеточных уравнений трехдиагонального вида// Компьютерная оптика N24 2002 г. стр. 33-39
6. Braunl T. The art of parallel programming. – Prentice Hall International (UK) Limited, 1993. – 378p.