УДК 681.3:518.5

СХЕМА РАСПАРАЛЛЕЛИВАНИЯ ОПЕРАЦИЙ РЕШЕНИЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ МНОГОМЕРНОЙ СКАЛЯРНОЙ ПРОГОНКИ

Г. А. Тарнавский[1], **С. И. Шпак**[1]

Для решения систем алгебраических уравнений, проводимого методом многомерной скалярной прогонки и составляющего основу неявного метода расщепления уравнений Эйлера и Навье–Стокса, предлагается ряд схем распараллеливания операций, позволяющих перенести алгоритмы, эффективно использовавшиеся для широкого класса научных и прикладных задач аэродинамики, в область новых компьютерных технологий с использованием высокопроизводительных многопроцессорных вычислительных систем. Рассмотренные схемы могут быть непосредственно реализованы на многопроцессорных вычислительных системах с общей памятью. Работа выполнена при поддержке Российского фонда фундаментальных исследований (гранты № 00–07–90297 и 99–01–00514).

Введение. Настоящая работа представляет собой комплексный теоретический анализ и ориентирована на создание новых современных вычислительных технологий и методов параллельного программирования, направленных на повышение эффективности решения фундаментальных научных и прикладных проблем в области численного моделирования задач аэродинамики и физической газовой динамики, теоретических вопросов проектирования летательных аппаратов, связанных с большим объемом вычислений.

Предлагаемая разработка параллельных алгоритмов является одним из аспектов создания информационно-вычислительной системы “Поток-3” для развития, сохранения и распространения разработанных за длительный период результатов научной деятельности в области вычислительной аэродинамики (методы, алгоритмы, программы и расчетные данные), для хранения, пополнения и систематизации накопленной информации, обеспечения непрерывности научного прогресса и его ускорения в данной области за счет преемственности осуществленных разработок. Данная система ориентирована на современные многопроцессорные ЭВМ и предназначена для информационно-вычислительной поддержки исследований физики и механики процессов обтекания тел различной конфигурации сверхзвуковым и гиперзвуковым потоком вязкого теплопроводного газа с учетом его реальных свойств, в том числе для изучения отрывных течений над поверхностью тела, а также течений в ближнем и дальнем следе за ним, в широком диапазоне определяющих параметров и высот полета. Фундаментальные научные проблемы (от особенностей распространения ударных волн в воздухе до физико-химических процессов, происходящих в газовых средах) и прикладные задачи проектирования летательных аппаратов различного назначения, расчетов их движения в атмосферах Земли и планет Солнечной системы приводят к сложным системам интегро-дифференциальных уравнений, предъявляющим исключительно высокие требования к компьютерным ресурсам (быстродействию и памяти).

Постановка задачи. Весьма часто используемым сегментом в алгоритмах задач механики сплошной среды (в частности, в области аэродинамики и физической газовой динамики) являются различные процедуры прогонки для решения систем алгебраических уравнений — дискретных представлений дифференциальных уравнений. Операции выполнения этих процедур в ряде методов [1] интегрирования сложных нелинейных систем дифференциальных уравнений Эйлера, Навье–Стокса, Барнетта зачастую являются доминирующими (или, по крайней мере, существенными) с точки зрения затрат вычислительных ресурсов. Вследствие этого одним из направлений продвижения методов, отработанных на широких классах научных и прикладных задач, в область новых компьютерных технологий с использованием параллельных вычислительных систем и повышения эффективности этих методов является разработка соответствующих параллельных алгоритмов реализации процедур прогонки, существенно снижающих общее (астрономическое) время вычислительного процесса, что сделает возможным решение задач нового уровня, ранее недоступных по чисто техническим причинам.

Рассмотрим вопросы реорганизации трехмерной процедуры прогонки (на трехточечном по каждому координатному направлению шаблоне) из последовательного в последовательно-параллельный. Отметим, что приводимая ниже методология распараллеливания носит достаточно идеализированный характер и может быть, по-видимому, непосредственно реализована на вычислительных параллельных системах только с общей памятью. Использование ее на системах с распределенной памятью возможно при специальной организации обменов данными, однако эти технические проблемы требуют особого рассмотрения и лежат за рамками настоящей статьи.

[1] Институт теоретической и прикладной механики СО РАН, 630007 Новосибирск, e-mail: shpak@itam.nsc.ru

Постановка задачи имеет следующий вид: необходимо найти решение $F_{i,j,k}$ системы алгебраических уравнений

$$\begin{aligned} &AF_{i-1,j,k} + BF_{i,j,k} + CF_{i+1,j,k} + D = 0,\\ &AF_{i,j-1,k} + BF_{i,j,k} + CF_{i,j+1,k} + D = 0,\\ &AF_{i,j,k-1} + BF_{i,j,k} + CF_{i,j,k+1} + D = 0. \end{aligned} \tag{1}$$

Здесь $A = \|A_{i,j,k}\|$, $B = \|B_{i,j,k}\|$, $C = \|C_{i,j,k}\|$, $D = \|D_{i,j,k}\|$ — известные матричные коэффициенты, заданные в области R, представляющей собой в отнормированной системе координат (подробнее см. [1]) прямоугольный параллелепипед с упорядоченной структурой расчетной сетки

$$R \in \{X_i, Y_j, Z_k\}, \quad i = 0, \ldots, I+1, \quad j = 0, \ldots, J+1, \quad k = 0, \ldots, K+1,$$

где i, j, k — целочисленные индексы, нумерующие узлы вычислительного поля по его X, Y и Z координатным направлениям с краевыми условиями первого или второго рода, т.е. заданными на гранях параллелепипеда значениями F, $\triangle F/\triangle x$, $\triangle F/\triangle y$, $\triangle F/\triangle z$ или их комбинациями типа $\hat{F} = aF + b\triangle F/\triangle x$, которые могут быть представлены в символическом виде для каждой из шести граней следующим образом:

$$\{\hat{F}_{0,j,k}; \hat{F}_{I+1,j,k}; \hat{F}_{i,J+1,k}; \hat{F}_{i,j,0}; \hat{F}_{i,j,K+1}\} = \varphi_\beta = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6\}. \tag{2}$$

Здесь φ_ℓ, $\ell = 1, \ldots, 6$, — заданные (каждая на своей грани) функции. Аэрофизический смысл F и φ_b, конкретный вид их записи и переход от непрерывной задачи к дискретной подробно изложен в [1].

Алгоритм решения задачи (1)–(2) структурно состоит из трех этапов, каждый из которых, в свою очередь, подразделяется на два цикла.

I этап: X-координатное направление. Первый цикл: прямой ход прогонки, т.е. вычисление прогоночных коэффициентов $\alpha^{(1)}$ и $\beta^{(1)}$ по формулам

$$\alpha^{(1)}_{i+1,j,k} = -\frac{C_{i,j,k}}{\alpha^{(1)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}, \quad \beta^{(1)}_{i+1,j,k} = -\frac{\beta^{(1)}_{i,j,k}A_{i,j,k} + D_{i,j,k}}{\alpha^{(1)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}. \tag{3}$$

Последовательность изменения индекса i строго детерминирована: $i = 0, 1, 2, \ldots, I-1$; граничные значения $\alpha^{(1)}_{0,j,k}$ и $\beta^{(1)}_{0,j,k}$ — известны (получены из (2)).

Второй цикл: обратный ход прогонки, т.е. вычисление значений функции $F^{(1)}$) по формуле

$$F^{(1)}_{i,j,k} = \alpha^{(1)}_{i+1,j,k}F^{(1)}_{i+1,j,k} + \beta^{(1)}_{i+1,j,k}. \tag{4}$$

Последовательность изменения индекса: $i = I, I-1, \ldots, 1$; $F^{(1)}_{i+1,j,k}$ — известны (получены из (2)). Заметим, что последовательность изменения индексов j и k в (3)–(4) в диапазонах своего изменения $j \in [1, J]$, $k \in [1, K]$ может быть произвольной: или от меньших значений к большим (традиционно принятый в программировании порядок), или от больших значений к меньшим, или даже неупорядоченной. Это исключительно важное обстоятельство будет использовано ниже для организации параллельного счета.

II этап: Y-координатное направление. Первый цикл: прямой ход прогонки, т.е. вычисление прогоночных коэффициентов $\alpha^{(2)}$ и $\beta^{(2)}$ по формулам

$$\alpha^{(2)}_{i,j+1,k} = -\frac{C_{i,j,k}}{\alpha^{(2)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}, \quad \beta^{(2)}_{i,j+1,k} = -\frac{\beta^{(2)}_{i,j,k}A_{i,j,k} + D_{i,j,k}}{\alpha^{(2)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}. \tag{5}$$

Последовательность изменения индекса j строго детерминирована: $j = 0, 1, 2, \ldots, J-1$; граничные значения $\alpha^{(2)}_{i,0,k}$ и $\beta^{(2)}_{i,0,k}$ — известны (получены из (2)).

Второй цикл: обратный ход прогонки, т.е. вычисление значений функции $F^{(2)}$) по формуле

$$F^{(2)}_{i,j,k} = \alpha^{(2)}_{i,j+1,k}F^{(2)}_{i,j+1,k} + \beta^{(2)}_{i,j+1,k}. \tag{6}$$

Последовательность изменения индекса j: $j = J, J-1, \ldots, 1$; значения $F^{(2)}_{i,J+1,k}$ известны из (2). Последовательность изменения индексов $i \in [1, I]$ и $k \in [1, K]$ в (5)–(6) произвольна (см. замечание к I этапу).

III этап: Z-координатное направление. Первый цикл: прямой ход прогонки, т.е. вычисление прогоночных коэффициентов $\alpha^{(3)}$ и $\beta^{(3)}$) по формулам

$$\alpha^{(3)}_{i,j,k+1} = -\frac{C_{i,j,k}}{\alpha^{(3)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}, \quad \beta^{(3)}_{i,j,k+1} = -\frac{\beta^{(3)}_{i,j,k}A_{i,j,k} + D_{i,j,k}}{\alpha^{(3)}_{i,j,k}A_{i,j,k} + B_{i,j,k}}. \tag{7}$$

Последовательность изменения индекса k: $k = 0, 1, 2, \ldots, K-1$; граничные значения $\alpha^{(3)}_{i,j,0}$ и $\beta^{(3)}_{i,j,0}$ известны из (2).

Второй цикл: обратный ход прогонки, т.е. вычисление значений функции $F^{(3)}$ по формуле

$$F^{(3)}_{i,j,k} = \alpha^{(3)}_{i,j,k+1} F^{(3)}_{i,j,k+1} + \beta^{(3)}_{i,j,k+1}. \tag{8}$$

Последовательность изменения индекса k: $k = K, K-1, \ldots, 1$; значения $F^{(3)}_{i,j,K+1}$ известны из (2). Последовательность изменения индексов $i \in [1, I]$ и $j \in [1, J]$ в (7)–(8) произвольна (см. замечание к I этапу).

Следует сделать несколько важных замечаний к приведенному выше краткому описанию процедуры многомерной прогонки (3)–(8).

1. Окончательным результатом решения (1)–(2) методом (3)–(8) есть значения $F \equiv F^{(3)}$, при этом этапы I–III (X, Y, Z -координатные прогонки) являются, говоря вообще в теоретическом плане, перестановочными, однако при решении конкретных аэрогазодинамических задач порядок их следования, т.е. последовательность вычисления, существенно влияет (иногда весьма значительно) на затраты компьютерных ресурсов (см. [1]).

2. Вообще говоря, матричные коэффициенты A, B, C и D сами зависят от функции F:

$$A = A(F), \quad B = B(F), \quad C = C(F), \quad D = D(F), \tag{9}$$

вследствие чего могут быть использованы следующие модификации прогоночной процедуры (3)–(8):

— "явная", когда на всех трех этапах в правых частях (3)–(8) в качестве (9) берутся значения A, B, C и D от некоторого начального $F^{(0)}$:

$$A = A(F^{(0)}), \quad B = B(F^{(0)}), \quad C = C(F^{(0)}), \quad D = D(F^{(0)}); \tag{10}$$

— "корректирующая", с отслеживанием получаемых в процессе значений $F^{(1)} u \;\; F^{(2)}$ для использования на втором и третьем этапах соответственно ($n = 2, 3$):

$$A = A(F^{(n-1)}), \quad B = B(F^{(n-1)}), \quad C = C(F^{(n-1)}), \quad D = D(F^{(n-1)}); \tag{11}$$

— "схема с дальней памятью", эффективная для подавления возможных коротковолновых осцилляций решения ($n = 2, 3$):

$$\begin{aligned} A^{(n)} &= A(F^{(n-1)}/2 + F^{(n-2)}/2), \\ B^{(n)} &= B(F^{(n-1)}/2 + F^{(n-2)}/2), \\ C^{(n)} &= C(F^{(n-1)}/2 + F^{(n-2)}/2), \\ D^{(n)} &= D(F^{(n-1)}/2 + F^{(n-2)}/2). \end{aligned} \tag{12}$$

Отметим, что в (12) могут быть применены и несимметричные схемы с весами типа "$(4-1)/3$" и т.п.

Следует особо подчеркнуть важное для последующего распараллеливания обстоятельство, что в реальных задачах массивы коэффициентов A, B, C и D зависят от значений F лишь в ближайших узлах расчетного шаблона:

$$\begin{aligned} A_{i,j,k} &= A(F_{i,j,k}, F_{i\pm1,j,k}, F_{i,j\pm1,k}, F_{i,j,k\pm1}), \\ B_{i,j,k} &= B(F_{i,j,k}, F_{i\pm1,j,k}, F_{i,j\pm1,k}, F_{i,j,k\pm1}), \\ C_{i,j,k} &= C(F_{i,j,k}, F_{i\pm1,j,k}, F_{i,j\pm1,k}, F_{i,j,k\pm1}), \\ D_{i,j,k} &= D(F_{i,j,k}, F_{i\pm1,j,k}, F_{i,j\pm1,k}, F_{i,j,k\pm1}). \end{aligned} \tag{13}$$

3. Кратко заметим, что в аэродинамике функция F является не скалярной, а векторной с размерностью, как правило, 5 (плотность, три компоненты скорости, энергия (давление или температура)). В задачах с учетом физико-химических процессов размерность F существенно повышается, однако фактором, облегчающим компьютерные расчеты, является высокая разреженность блочно-диагональных матриц A, B, C и D.

Таким образом, алгебраическая задача (1)–(2), решение которой обеспечивает алгоритм последовательной трехмерной прогонки (4)–(13), требует высоких затрат компьютерных ресурсов, а их снижение является одной из главных технологических (существуют, наряду с этим, и физико-математические) проблем вычислительной аэродинамики.

Здесь следует отметить два пути. Первый, традиционный в рамках последовательного программирования, связан с классическими подходами оптимизации как алгоритмов решения разностных уравнений в целом [2], так и самого метода прогонки и его отдельных внутренних элементов (см., например, [3]), а также с привлечением некоторых специальных средств программирования с использованием метода "крупноформатных таблиц" [4].

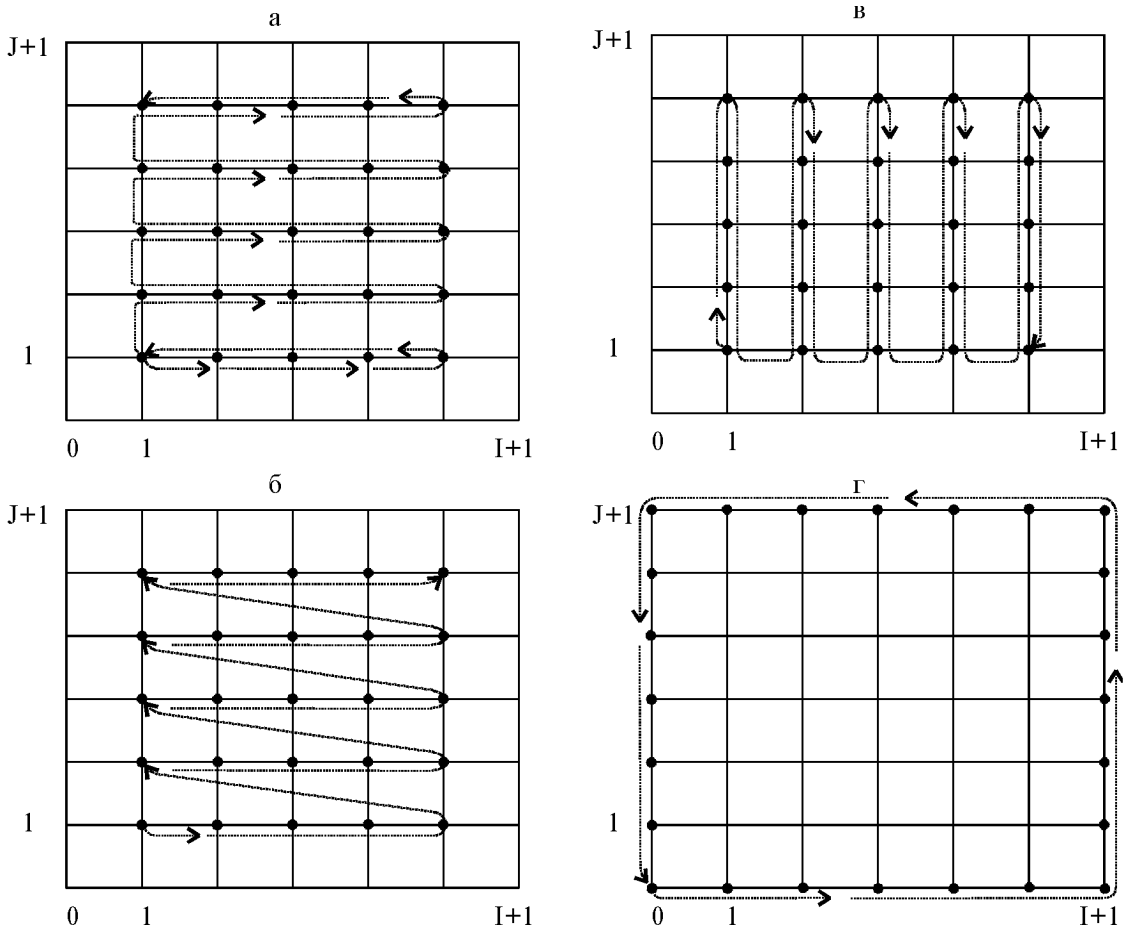

Рис. 1

Второй путь связан с применением многопроцессорных компьютеров с параллельной архитектурой. Этот путь позволяет принципиально, на порядки, снизить затраты компьютерных ресурсов, необходимых для решения сложных задач, однако надо отметить недостаточную разработанность операционных сред и в особенности языков программирования, созданных на базе традиционных языков. Так, в [5] проведена предварительная оценка возможностей языка Fortran в редакции HPF (High-Performance Fortran), принятой в качестве средства для решения задач вычислительной аэродинамики. Заметим, эта работа представляет собой часть более обширного исследования возможностей создания программного обеспечения, включая применение параллельных алгоритмов [6], используемых NASA для комплексного анализа и проектирования летательных аппаратов. В [7–9] проводится общий анализ возможностей новых технологий математического моделирования с широкой областью приложений, в частности, к задачам аэродинамики. В [10] рассматриваются вопросы организации векторизованного процесса вычислений при реализации операции двумерной свертки, элементы которой могут быть использованы и при параллелизации процедуры прогонки, исследуемой в настоящей работе.

Перейдем к анализу алгоритма прогонки (3)–(8) и определим некоторые ключевые моменты его распараллеливания. Для наглядности анализ будет проводиться на двумерной плоскости, распараллеливание операций по третьему измерению проводится аналогично.

Рис. 1 иллюстрирует последовательную схему нахождения $F_{i,j}$ в узлах вычислительной сетки $\{R\} \Rightarrow \{i \times j\}$; $i \in [0, I+1]$; $j \in [0, J+1]$. В начальный момент известны значения всех матричных коэффициентов $A_{i,j}$, $B_{i,j}$, $C_{i,j}$, $D_{i,j}$ во всех узлах R и значения F на ее границах, т.е. известны $F_{0,j}$, $F_{I+1,j}$, $\forall j \in [1, J]$ и $F_{i,0}$, $F_{i,J+1}$ $\forall i \in [1, I]$. Арифметические операции I этапа (3)–(4), начинаясь, например, в точке (1, 1), последовательно проходят узлы $\{i \times j\} \in \{[1, I] \times [1, J]\}$ по маршруту, показанному на рис. 1 а пунктирной линией, при этом движение слева направо реализует прямой ход прогонки (3), а справа налево — обратный (4). Рис. 1 б иллюстрирует промежуточный, при необходимости, этап вычислений значений $A^{(1)}_{i,j}$, $B^{(1)}_{i,j}$, $C^{(1)}_{i,j}$ и $D^{(1)}_{i,j}$ по (12)–(13), если используется алгоритм с корректировкой. Последовательный ход операций показан на рис. 1 б пунктирной линией. Далее выполняются операции II этапа (5)–(6) с последовательным обходом узлов сетки “снизу–вверх”, реализуя алгоритм прямого хода прогонки (5), и “сверху–вниз”, выполняя алгоритм (6) обратного хода. Еще

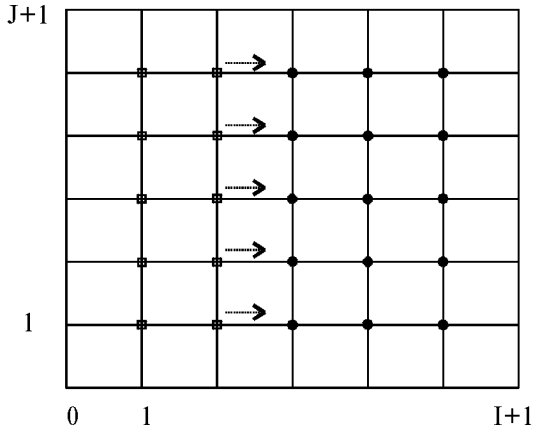

Рис. 2

раз подчеркнем важнейшее для распараллеливания прогонки обстоятельство, что на этом этапе строго определен только порядок изменения индекса j (сначала движение "снизу–вверх", а затем "сверху–вниз"), порядок же изменения индекса i произволен (здесь показано для определенности движение "слева–направо", но вполне допускается и движение "справа–налево").

На этом операции алгоритма (3)–(8) по решению задачи (1)–(2) заканчиваются. Однако, вообще говоря, в реальных задачах аэродинамики эта процедура соответствует какому-то определенному моменту времени (физическому или итерационному). Для других моментов времени она должна быть многократно повторена, вследствие чего существует IV этап решения задачи — этап вычисления матричных коэффициентов A, B, C, D по алгоритму (13) (ход операций которого аналогичен показанному на рис. 1 б) и заполнения граничных узлов сетки соответствующими значениями $F = \varphi_b$ по тому или иному алгоритму реализации (2), который иллюстрируется на рис. 1 г. Следует заметить, что на рис. 1 г показан "круговой" обход границ, однако практически во всех программах границы заполняются для удобства программирования в последовательности "нижняя $\to$ верхняя $\to$ левая $\to$ правая" в направлении или "слева–направо", или "снизу–вверх" с, очевидно, тем же самым количеством выполняемых операций.

Перейдем далее к рассмотрению некоторых способов распараллеливания алгоритма прогонки (3)–(13) в идеализированной постановке с неограниченным числом процессоров и свободной коммутацией между ними (некоторые вопросы размещения, переупорядочивания и маршрутизации данных рассматриваются ниже).

Организуем специальные процессорные блоки P_1, P_2, P_3 и P_4, ориентированные, соответственно, на выполнение операций:

1) I этапа прогонки (3)–(4): блок P_1, состоящий из $2J$ процессоров и структурированный на J сегментов P_{1j} по два процессора в каждом;

2) II этапа прогонки (5)–(6): блок P_2 из $2I$ процессоров, структурированный на I сегментов P_{2i} по два процессора в каждом;

3) пересчета матричных коэффициентов A, B, C, D по формулам (12)–(13): блок P_3 из $4N$ процессоров, где $N = \max(I, J)$, в свою очередь структурированных на N сегментов P_{3n} по четыре процессора в каждом;

4) пересчета граничных значений F: блок P_4 из четырех процессоров.

Подробно рассмотрим и проанализируем выполнение алгоритма (3)–(13) на параллельных структурах.

I стадия. На вычислительной сетке определены значения элементов массива функций $F^{(0)}$ и значения массивов коэффициентов A, B, C, D. Процессорный блок P_1 начинает выполнение первого цикла первого этапа прогонки — вычисление прогоночных коэффициентов $\alpha_{i,j}^{(1)}$ и $\beta_{i,j}^{(1)}$ по схеме (3). При этом, вследствие независимости операций по индексу j (напомним, что для наглядности рассматривается двумерный вариант алгоритма и индекс k опущен) процесс вычислений может быть векторизован: каждый из J сегментов P_{1j} блока P_1 выполняет операции при определенном фиксированном значении индекса j, т.е. схема (3) реализуется одновременно для всех $j \in [1, J]$. При этом один из двух процессоров сегмента P_{1j}, процессор P_{1j1}, проводит вычисление коэффициента $\alpha_{i,j}^{(1)}$, а процессор P_{1j2} — коэффициента $\beta_{i,j}^{(1)}$ на лучах с индексом j. Графически этот цикл представлен на рис. 2, где квадратиками помечены узлы сетки, в которых уже проведены вычисления $\alpha^{(1)}$ и $\beta^{(1)}$

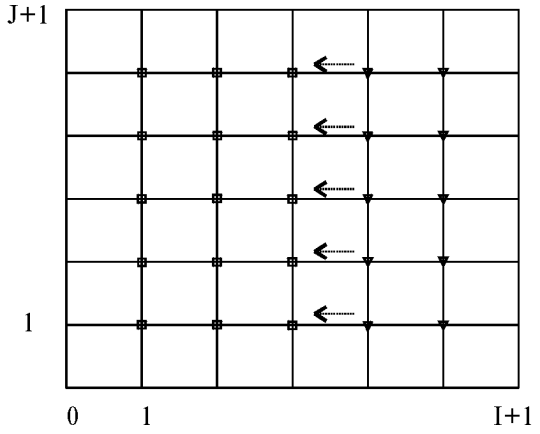

Рис. 3

по (3), а кружочками — узлы, ожидающие проведения операций (3). Возникает "волна счета" W_1, которая визуально выглядит как "волна квадратов", бегущая слева направо от $i = 1$ до $i = I$ одновременно по всей высоте области R, $1 \leqslant j \leqslant J$ (направление одновременного движения показано стрелками). Здесь следует сделать два замечания. Во-первых, идеальная процессорная система, разумеется, обеспечивает синхронность движения волны счета. В реальной системе асинхронность, естественно, будет иметь место, однако в данном цикле операций это не имеет никакого значения — важна лишь фиксация момента, когда последний из процессоров P_{1j} завершит работу. Во-вторых, каждая из операций (3) при вариации i по классификации Флинна относится к типу ОКМД ("одинаковый поток команд–множественный поток данных"), поэтому в сегментах P_{1j} может быть реализован конвейерный процесс вычислений. Таким образом, в целом в качестве блока P_1 целесообразно использовать векторно-конвейерную систему, однако это определяется конкретным типом архитектуры ЭВМ и ее операционной средой.

II стадия. Вычисление прогоночных коэффициентов $\alpha_{i,j}^{(1)}$ и $\beta_{i,j}^{(1)}$ закончено. Начинается выполнение обратного хода прогонки (4) — вычисление значений $F_{i,j}^{(1)}$, при этом задействованы лишь по одному процессору (например, P_{1j1}) каждого сегмента P_{1j} блока P_1, а все J процессоров P_{1j2} освобождаются и переводятся в режим ожидания или могут быть использованы в других процессорных блоках. Аналогично рассмотренному выше в I стадии, вычисление $F_{i,j}^{(1)}$ производится одновременно для всех $j \in [1, J]$ при значениях сначала $i = I$, а затем $i = I - 1$. Процессорные блоки P_2, P_3 и P_4 находятся в режиме ожидания данных. Подчеркнем, что в этой стадии задействованы только два значения i. Окончание ее показано на рис. 3, где перевернутыми треугольниками отмечены узлы, в которых произведен расчет значений $F_{i,j}^{(1)}$. Остальные обозначения аналогичны символам предыдущих рисунков, а стрелками, как и ранее, показано направление движения "волны счета" W_1.

III стадия. Операции II стадии выполнения (4) закончились на значении $i = I - 2$ (далее значение i будет последовательно уменьшаться до $i = 1$, что является завершением выполнения алгоритма (4)). При этом на крайнем правом вертикальном луче открылась возможность вычисления матричных коэффициентов $A_{i,j}^{(2)}$, $B_{i,j}^{(2)}$, $C_{i,j}^{(2)}$ и $D_{i,j}^{(2)}$ по "корректирующей схеме" (11) или (12) (при использовании "явной" схемы (10) эта стадия не нужна), поскольку значения этих коэффициентов определяются по (13) только ближайшими значениями $F^{(1)}$. В частности, вследствие того, что значения $F_{I-1,j}^{(1)}$ уже получены, а значения на границе $F_{I+1,j}$ известны всегда, то можно получить все значения $A_{I,j}^{(2)}$, $B_{I,j}^{(2)}$, $C_{I,j}^{(2)}$ и $D_{I,j}^{(2)}$ на вертикальном луче $L \in \{i = I,\ j = 1, \ldots, J\}$. Инициируются последовательно, один за другим, 4-х процессорные (вычисляющие A, B, C и D) сегменты процессорного блока P_{3i} ($i = I, I-1, I-2, \ldots$), т.е. инициализация сегмента P_{3i} происходит через один такт после завершения обработки данных на луче $i-1$ процессорного блока P_1. На карте вычислительной сетки возникает "волна счета" W_2 вычисления коэффициентов $A^{(2)}$, $B^{(2)}$, $C^{(2)}$ и $D^{(2)}$.

IV стадия. Поскольку на части узлов, расположенных вдоль лучей $i = I, i = I-1, \ldots$ коэффициенты $A_{i,j}^{(2)}$, $B_{i,j}^{(2)}$, $C_{i,j}^{(2)}$ и $D_{i,j}^{(2)}$ вычислены, возможно начало реализации II этапа алгоритма — прямого хода прогонки в Y-направлении (т.е. вычисление прогоночных коэффициентов $\alpha_{i,j}^{(2)}$ и $\beta_{i,j}^{(2)}$ по (5)). Поскольку вариации изменения индекса i произвольны, то счет может быть начат со значения $i = I$, затем $i = I - 1$ и т.д. Последовательно инициализируется работа пар процессорных сегментов P_{2i} блока P_2 (i — номер сегмента соответствует номеру

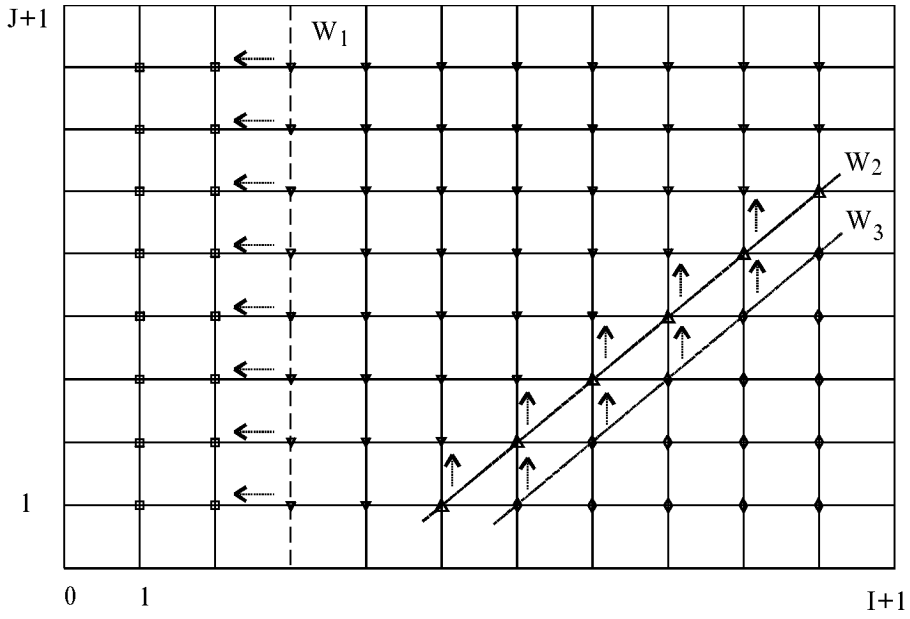

Рис. 4

вертикального луча вычислительной сетки). На карте вычислительной сетки возникает новая "волна счета" W_3, движущаяся вслед за волной W_2. Исключительно важное значение имеет синхронизация счета: диагональная волна W_3, идущая "вверх", не должна "догнать" (и тем более "перегнать") диагональную волну W_2, также идущую "вверх", а волна W_2, в свою очередь, не должна "приближаться" ближе чем на один шаг сетки к волне W_1 с "вертикальным" фронтом, идущей "справа–налево".

Одна из ситуаций IV стадии приведена на рис. 4. Треугольниками отмечены узлы, в которых произведен расчет значений $A^{(2)}_{i,j}$, $B^{(2)}_{i,j}$, $C^{(2)}_{i,j}$ и $D^{(2)}_{i,j}$, ромбиками — узлы, в которых произведен расчет $\alpha^{(2)}_{i,j}$ и $\beta^{(2)}_{i,j}$, остальные символы аналогичны символам на предыдущих рисунках. Стрелками показаны направления движения "волн счета" W_1, W_2 и W_3.

На этой стадии счета задействованы J процессоров блока P_1, $4M$ процессоров блока P_3 и $2(M-1)$ процессоров блока P_2, где M непрерывно увеличивается по мере ухода волны W_1 "влево" и определяется ее положением (текущим значением индекса i, т.е. номером обрабатываемого блоком P_1 вертикального луча сетки): $\forall i < I-2 \Rightarrow M = I-i$, $i = I-3, I-4, \ldots, 1$.

V стадия. Данная стадия возможна при определенном соотношении параметров вычислительной сетки и/или соотношением скоростей расчета $F^{(1)}$, ($A^{(2)}$, $B^{(2)}$, $C^{(2)}$, $D^{(2)}$) и ($\alpha^{(2)}$, $\beta^{(2)}$). Если "волны счета" W_2 и W_3 достигли верхней границы раньше чем волна W_1 — левой, то целесообразна инициализация вычисления значений $F^{(2)}_{i,j}$ по алгоритму (6). На вычислительной карте возникает еще одна "волна счета" W_4, расположенная диагонально ("слева–сверху", "направо–вниз") и двигающаяся "сверху–вниз". Далее, после нескольких тактов алгоритма (6) в вычислительной области, начиная с правого верхнего угла (I, J) (возможно, с учетом (13)), начинают вычисляться коэффициенты $A^{(3)}_{i,j}$, $B^{(3)}_{i,j}$, $C^{(3)}_{i,j}$ и $D^{(3)}_{i,j}$ по алгоритмам (11) или (12). И аналогично связанной паре волн W_2 и W_3, волна W_4 получает спутника W_5 — "волну счета" коэффициентов $A^{(3)}$, $B^{(3)}$, $C^{(3)}$ и $D^{(3)}$ — с той же топологией, что и W_4: диагональное расположение "слева–сверху", "направо–вниз" и движение "сверху–вниз". Важна синхронизация счета — волна W_4 не должна догнать (и тем более обогнать) волну W_5.

В этой стадии задействованы процессорные блоки P_1, P_2 и P_3. Представляет значительную трудность точно провести подсчет количества одновременно инициализированных процессоров этих блоков, поскольку оно существенно зависит от большого ряда факторов конкретной задачи и обрабатываемых данных.

VI стадия. По мере вычисления значений функции $F^{(3)}$ вблизи приграничных областей (или, образно говоря, "накатки" волны W_4 на границы области) становится возможным реализация граничных условий (2) по тем или иным алгоритмам. На вычислительной карте возникает последняя "волна счета" W_6 — счета значений F на границах. W_6 является "одномерной" волной, состоящей из двух участков, бегущих по границе от правого верхнего угла по и против часовой стрелки вслед за "точкой падения" волны W_4 на границу. Важна синхронизация: W_6 не должна догнать (и тем более обогнать) волну W_4.

Задействованы все процессорные блоки P_1, P_2, P_3 и P_4. Как и для стадии V, затруднительно подсчитать точное число инициализированных процессоров этих блоков.

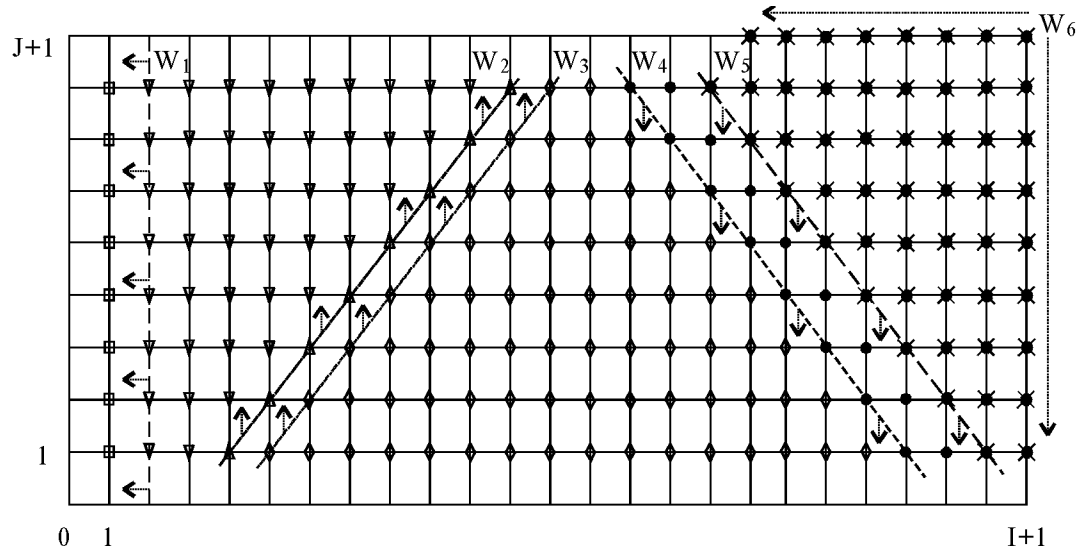

Рис. 5

VI стадия завершает работу всего программного комплекса многомерной (в иллюстрированном случае — двумерной) трехточечной прогонки. Ее топографическая карта приведена на рис. 5. Кружочками обозначены узлы сетки с окончательными значениями F, а перечеркнутыми кружочками — узлы как с окончательными значениями F, так и с новыми значениями коэффициентов A, B, C, D и новыми значениями F на границах, подготовленными для следующего итерационного цикла. Остальные символы аналогичны символам предыдущих рисунков, стрелками показаны направления движения "волн счета" W_1, W_2, W_3, W_4, W_5 и W_6.

Третье координатное измерение. Аналогично рассмотренной выше двумерной задаче может быть проведено распараллеливание операций (7)–(8) прогонки по третьему координатному измерению (индекс k массивов данных). При этом операции прямого хода прогонки (7) могут быть начаты при еще не законченных операциях не только прогонки по второму измерению (5)–(6), но и при возможно незаконченных операциях прогонки по первому измерению (3)–(4). Это определяется постановкой и основными параметрами конкретной физико-математической задачи. К сожалению, визуально-графическая иллюстрация вычислительных стадий и ситуаций, возможных в пространственном случае, в отличие от двумерного весьма затруднительна, а аналитические формулы, применяемые для расчета движения "волн счета" слишком громоздки даже для двух измерений и не могут быть здесь приведены.

Следует кратко остановиться еще на одном аспекте рассматриваемой проблемы. Важным критерием в параллельной обработке является глубина детализации графа операций, однозначно определяющая топологию процессорной сети, и отношение времени коммуникации к времени вычислений. Уменьшение времени связи может обеспечить использование реконфигурируемых (переключаемых) сетей, соответственно уменьшающих число промежуточных процессоров, через которые осуществляется маршрутизация сообщений (в частности, синхронизирующих команд) и данных. Реконфигурируемость сети дает также некоторую степень надежности, так как отказавшие процессоры могут быть обойдены и/или исключены из сети в процессе выполнения программы без ее перезагрузки и проведения счета заново (заметим в скобках, что, вообще говоря, самым простым, однако неэкономичным способом обеспечения контроля за точностью счета, является применение дублирующей процессорной сети той же конфигурации хотя бы на этапе отладки программ и получения устойчивого параллельного счета, исключающего появление специфических ситуаций, приводящих к получению неверных данных; многочисленные примеры этих ситуаций хорошо описаны, например, в [11]).

Поскольку конфигурацию графа операций прогонки можно считать известной, варьируемой в зависимости от параметров конкретной задачи, то процессорная конфигурация является программно управляемой. Топология процессорной сети, отражающая в определенном смысле топологию вычислительной сетки с учетом описанных выше стадий прогонки (от I до VI), также меняется: она может быть или статической, или квазистатической, или динамической. Так, например, на I стадии прогонки топология процессорной сети статична, задана заранее и определяется только числом J (один из параметров расчетной сетки) с одновременным функционированием всех процессоров блока P_1. Однако выполнение следующих стадий определяет динамическое изменение топологии процессорной сети: априори, до проведения счетных операций, нельзя сказать заранее, в какой момент времени, какие процессоры и из каких сегментов и блоков будут начинать или заканчивать функционирование. При этом неизбежно возникновение многочисленных режимов ожидания, в особенности в

трехмерных задачах, и реконфигурация сети, отслеживающая тем или иным способом эти ситуации, может повысить экономичность использования процессоров и уменьшить их число, необходимое для решения задачи (однако следует учесть, что повышение экономичности увеличивает вероятность отказов или неверного счета).

Вообще говоря, в данном конкретном случае операции прогонки являются операциями низшего уровня в общем комплексе программ [12] решения задач аэродинамики. Поэтому время вычислений намного больше, чем время коммуникаций, а накладные расходы на связь, по-видимому, будут незначительны и не будут существенно зависеть от конфигурации; при этом, естественно, не имеет особого значения, какая используется топология и соответственно нет особой необходимости в реконфигурации сети процессоров.

Заключение. Рассмотренная выше методология распараллеливания процедуры многомерной трехточечной скалярной прогонки, в которой использованы конвейерные и древовидные (ветвящиеся) структуры операций, закладывается в отработанный на широком классе задач комплекс алгоритмов и программ [12]. Это позволит существенно снизить требуемые вычислительные ресурсы и достичь высокой общей производительности при решении сложных систем интегро-дифференциальных уравнений задач физической газовой динамики.

СПИСОК ЛИТЕРАТУРЫ

1. *Ковеня В.М., Тарнавский Г.А., Черный С.Г.* Применение метода расщепления в задачах аэродинамики. Новосибирск: Наука, 1990.
2. *Хмельнов Д.Е.* Улучшенные алгоритмы решения разностных и q-разностных уравнений // Программирование. 2000. № 2. 70–78.
3. *Парийский Б.С.* Об экономии памяти ЭВМ и времени счета при дифференциальной прогонке // ЖВМ и МФ. 2000. **40**, № 2. 332–334.
4. *Morishita E.* Spreadsheet fluid dynamics // J. Aircraft and Rockets. 1999. **36**, N 4. 720–723.
5. *Bogucz E.A., Fox G.C., Haupt T., Hawick K.A, Ranka S.* Preliminary evaluation of High-Performance Fortran as a language for computational fluid dynamics // AIAA Paper. 1994. N 2262. 1–13.
6. *Amaladas J.R., Kamath H.* Accuracy assessment of upwind algorithms for steady-state computations // Comp. in Fluids. 1998. **27**, N 8. 941–962.
7. *Angster S., Jayaram S.* An object-oriented, knowledge-based approach to multi-disciplinary, parametic design // AIAA Paper. 1995. N 0323. 1–13.
8. *Ильин В.П.* Вычислительно-информационные технологии математического моделирования // Автометрия. 2000. № 1. 3–16.
9. *Семенов В.А., Крылов П.Б., Морозов С.В., Тарлапан О.А.* Объектно-ориентированная архитектура для приложений научной визуализации и математического моделирования // Программирование. 2000. № 2. 29–40.
10. *Мельник Э.А.* Параллельные алгоритмы двумерной свертки // Автометрия. 2000. № 1. 122–126.
11. *П*рограммирование на параллельных вычислительных системах / Под ред. Р. Бэбба. М.: Мир, 1991.
12. *Лебедева М.К., Медведев А.Е., Тарнавский Г.А.* База данных "ExtFlow2" информационной поддержки численного моделирования задач внешней аэродинамики // Автометрия. 1994. № 5. 76–84.

Поступила в редакцию
01.06.2000