

Math-Net.Ru

Общероссийский математический портал

А. В. Бабаков, О. С. Бацуков, О. М. Белоцерковский,
Л. Н. Столяров, О возможностях достижения высокой про-
изводительности на векторно-конвейерных ЭВМ при реше-
нии задач математической физики,
Ж. вычисл. матем. и матем. физ., 1986, том 26, но-
мер 4, 601–613

<https://www.mathnet.ru/zvmmf8194>

Использование Общероссийского математического портала Math-Net.Ru подра-
зумевает, что вы прочитали и согласны с пользовательским соглашением
<https://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.14.88

22 апреля 2025 г., 12:23:22



УДК 519.63

**О ВОЗМОЖНОСТЯХ ДОСТИЖЕНИЯ ВЫСОКОЙ
ПРОИЗВОДИТЕЛЬНОСТИ НА ВЕКТОРНО-КОНВЕЙЕРНЫХ ЭВМ
ПРИ РЕШЕНИИ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ**

**БАБАКОВ А. В., БАЦУКОВ О. С., БЕЛОЦЕРКОВСКИЙ О. М.,
СТОЛЯРОВ Л. Н.**

(Москва)

Рассматриваются вопросы анализа алгоритмов решения задач математической физики для их программирования на векторно-конвейерных ЭВМ. Предлагаемая методика позволяет согласовать алгоритм с архитектурой машины и получить программу большой эффективности. Анализ проводится на примере разностной схемы метода потоков, используемой для численного решения задач механики сплошных сред. Приводятся оценки быстродействия полученных трансляций с языка высокого уровня и с использованием предлагаемой методики.

Введение

В настоящее время при численном решении уравнений математической физики существует насущная необходимость в использовании ЭВМ, которые обладают сверхбольшим быстродействием и сверхбольшой оперативной памятью. Так, в таблице приводятся требования к производительности и оперативной памяти ЭВМ, обеспечивающие расчет стационарного обтекания элементов самолета с получением результата через 15 мин (по оценкам [1]).

| Рассчитываемый объект | Быстродействие * | Объем памяти (число слов) | Тип ЭВМ |
|-----------------------|-------------------|---------------------------|-------------------|
| Профиль | 1 | 10 ⁵ | CDC-7600 |
| Фюзеляж | 10 | 10 ⁷ | ILLIAC-IV |
| Лопатки турбины | 10 ² | 10 ⁷ | CRAY-I, CYBER-205 |
| Крыло | 5·10 ² | 10 ⁸ | NAS |
| Винт вертолета | 8·10 ² | 10 ⁸ | NAS |
| Самолет | 10 ³ | 10 ⁹ | ? |

* Быстродействие — в миллионах операций с плавающей запятой в секунду (Mflops).

Такое увеличение быстродействия и наращивание оперативной памяти, как показывает практика разработки машин за последнее десятилетие, возможно только за счет усложнения их архитектуры. При этом возникает проблема разработки алгоритмов и программ, согласованных с особенностями машины. Цена потерь от несогласованности может быть очень высокой, и тем выше, чем больше быстродействие и архитектурная сложность машины.

В настоящей статье предлагается один из возможных подходов к решению этой проблемы. Процедура согласования дается в виде технологии анализа алгоритма и планирования вычислений и гарантирует получение программ, эффективных по быстродействию и использованию памяти.

Для иллюстрации возможностей предлагаемого подхода выбран класс алгоритмов консервативного метода потоков [2]–[4] и дана его реализация на векторно-конвейерной ЭВМ типа CRAY.

§ 1. Алгоритм метода потоков

Метод потоков основан на разностной аппроксимации законов сохранения в виде поверхностных интегралов от векторов плотностей потоков аддитивных характеристик среды для каждого элементарного объекта-ячейки Ω , естественно возникающего при введении вычислительной сетки:

$$(1.1) \quad \frac{\partial F}{\partial t} = - \oint_{S_{\Omega}} Q_F ds.$$

Здесь S_{Ω} – боковая поверхность Ω ; F есть M, X, Y, Z – соответственно, масса и составляющие импульса, а E – полная энергия газа в объеме Ω ; Q_F – соответствующий вектор плотности потока.

Значения плотностей распределения перечисленных аддитивных характеристик среды в некоторой характерной точке объема Ω определим следующим образом:

$$\rho = \frac{M_{\Omega}}{\Omega}, \quad \xi = \frac{X_{\Omega}}{\Omega}, \quad \eta = \frac{Y_{\Omega}}{\Omega}, \quad \zeta = \frac{Z_{\Omega}}{\Omega}, \quad \varepsilon = \frac{E_{\Omega}}{\Omega}.$$

Переменные поля – соответствующие проекции u, v, w вектора скорости \mathbf{v} и удельная внутренняя энергия e – определяются плотностями распределения, а давление P – уравнением состояния:

$$u = \frac{\xi}{\rho}, \quad v = \frac{\eta}{\rho}, \quad w = \frac{\zeta}{\rho}, \quad e = \frac{\varepsilon}{\rho} - \frac{u^2 + v^2 + w^2}{2}, \quad P = (\gamma - 1)\rho e$$

(γ – отношение удельных теплоемкостей).

При численном решении системы уравнений (1.1) для временной производной в левой части используется аппроксимация первого порядка, а поверхностные интегралы правых частей вычисляются по некоторой квадратурной формуле со вторым порядком аппроксимации:

$$(1.2) \quad \frac{F^{n+1} - F^n}{\tau} = -L_h \left(\oint_{S_{\Omega}} Q_F ds \right).$$

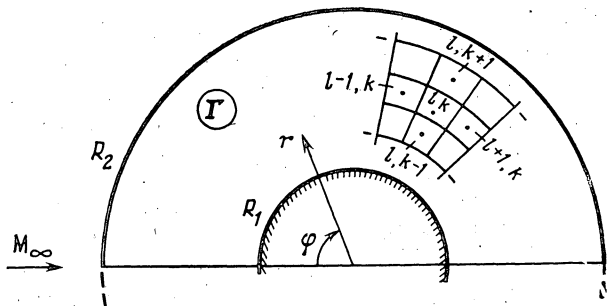
В качестве примера использования метода потоков рассмотрим двумерную задачу о течении сжимаемого невязкого газа около цилиндра. Система координат и область интегрирования представлены на фиг. 1.

Преобразованием

$$(1.3) \quad r = R_1 + (R_2 - R_1) \frac{e^{\alpha z} - 1}{e^{\alpha} - 1}, \quad 0 \leq z \leq 1, \quad \alpha \geq 0,$$

по направлению r вводится неравномерная, экспоненциально сжатая к телу сетка (α – параметр, характеризующий степень неравномерности сетки). Здесь R_1 – радиус цилиндра, R_2 – радиус внешней границы области интегрирования.

При аппроксимации векторов плотностей потоков на границах S_{Ω} объемов-ячеек Ω_{lh} существенным является следующее: плотности распределения аддитивных характеристик среды вычисляются несимметричным образом, а именно экстраполяцией по направлению конвективного переноса, в то время как давление и скорости переноса определяются по симметрич-



Фиг. 1

ным формулам; например, для границы $S_{l+1/2, k}$ имеем

$$(1.4a) \quad \xi_{l+1/2, k} = \begin{cases} 1.5\xi_{lk} - 0.5\xi_{l-1, k}, & (V_N)_{l+1/2, k} \geq 0, \\ 1.5\xi_{l+1, k} - 0.5\xi_{l+2, k}, & (V_N)_{l+1/2, k} < 0, \end{cases}$$

$$(1.4b) \quad P_{l+1/2, k} = \frac{P_{l, k} + P_{l+1, k}}{2}, \quad u_{l+1/2, k} = \frac{u_{lk} + u_{l+1, k}}{2}, \quad v_{l+1/2, k} = \frac{v_{lk} + v_{l+1, k}}{2},$$

$$(1.4в) \quad (V_N)_{l+1/2, k} = u_{l+1/2, k} \sin \varphi_{l+1/2} + v_{l+1/2, k} \cos \varphi_{l+1/2}.$$

Здесь $P_{l+1/2, k}$ — давление на границе $S_{l+1/2, k}$, $(V_N)_{l+1/2, k}$ — нормальная к границе $S_{l+1/2, k}$ компонента скорости.

На границах области интегрирования ставятся следующие граничные условия: на поверхности обтекаемого тела — условия непротекания, на внешней границе при $-\varphi_0 \leq \varphi \leq \varphi_0$ — условия в невозмущенном потоке, на остальной части внешней границы — условия $\partial f / \partial N = 0$, где f — любая переменная поля или проекция вектора плотности потока на нормаль N .

Отметим, что метод потоков обладает свойством консервативности по каждой аддитивной характеристике среды, для которой выписан закон сохранения, причем консервативность имеет место как для каждой ячейки разностной сетки (т. е. локально), так и для всей расчетной области. Действительно, вычисление векторов плотностей потоков производится на отдельных участках поверхности S_a , являющихся границами между двумя соседними ячейками. Перенос соответствующей характеристики среды через эту границу не может привести к потере ее или к увеличению (с точностью до ошибок округления в последних разрядах машинного слова).

§ 2. Анализ алгоритма. Планирование вычислений

Перейдем к анализу численного алгоритма и проектированию программы для векторно-конвейерной машины. В самых общих чертах технология согласования алгоритма с машиной содержит описание в одинаковых терминах особенностей архитектуры машины и алгоритма, процедуру их согласования и планирования вычислений.

Типичным представителем векторных ЭВМ конвейерного типа является CRAY-I, архитектура ее стала классической для машин этого класса.

Остановимся на таких особенностях архитектуры машины, которые необходимо знать и учитывать программисту, работающему в области численного решения задач математической физики (более подробно архитектура CRAY-I изложена в [5]).

Первая особенность состоит в том, что основу машины составляют функциональные устройства, способные независимо друг от друга (парал-

тельно) выполнять векторные операции: сложение, умножение, вычисление обратной величины с плавающей запятой, логические операции и др.

Векторные операции в функциональных устройствах выполняются последовательно над элементами векторов. Последовательная обработка происходит по так называемому конвейерному принципу. Каждое функциональное векторное устройство аппаратно выполнено из нескольких последовательных блоков, обеспечивающих соответствующие уровни обработки. На некотором уровне информация преобразуется и для очередного этапа обработки поступает на следующий уровень. Например, на CRAY-I сложение с плавающей запятой разделено на 6 уровней. Положим, если элементы $a_1 \in a$ и $b_1 \in b$ обрабатываются на 2-м уровне, то в этот же момент на 1-м уровне обрабатываются элементы $a_2 \in a$ и $b_2 \in b$. Передача информации с уровня на уровень и совмещение по времени обработки на разных уровнях напоминают работу конвейера. Преобразование информации на всех уровнях происходит синхронно за определенное время t , которое называется тактом машины (для CRAY-I такт $t=12.5$ нс). Очевидно, что результат для первых элементов вектора появится через время $\tau=mt$, где m — число уровней устройства, а время обработки всего вектора длиной n составит $T(n)=\tau+nt$, $n \leq 64$. Величина τ называется временем «разгона» конвейера.

Вторая особенность заключается в том, что функциональные устройства работают непосредственно не с оперативной памятью, а со специальными векторными регистрами. Машина имеет небольшое число векторных регистров: в CRAY-I их всего 8. Обмен между регистрами и памятью реализуется функциональным устройством обращения к памяти.

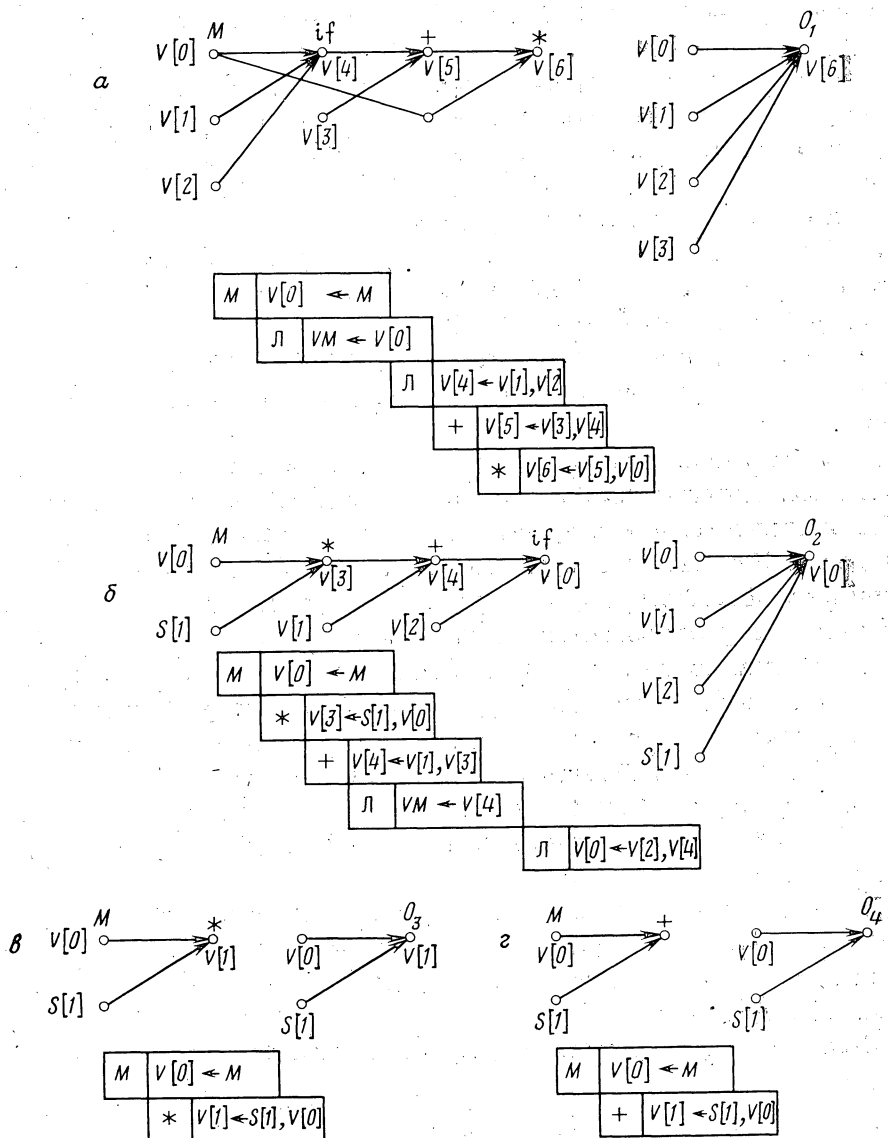
Третья особенность заключается в том, что в машине имеется возможность «сцеплять» функциональные устройства так, что результат с одного устройства через определенный регистр поступает в качестве операнда на другое. В таких «связках» несколько векторных операций выполняются практически параллельно.

Перечисленные особенности векторных ЭВМ конвейерного типа, в сравнении с последовательными машинами, налагают дополнительные требования на программиста при проектировании программ. Эффективную параллельную работу функциональных устройств и их «зацепление» должен организовать сам программист, согласовывая численный алгоритм с архитектурой ЭВМ. Для решения задачи согласования необходимо архитектурные свойства алгоритма и машины описать на одном и том же языке.

Архитектурные свойства алгоритма описываются в виде так называемых информационных структур алгоритмов (см. [6]—[8]). Информационная структура (и. с.) алгоритма, по сути дела, представляет собой отношение зависимости переменных в системе формул, которые задают численный алгоритм. И. с. алгоритма может быть представлена в виде направленного графа без петель и циклов. Каждой вершине отнесено имя переменной и операция, при помощи которой вычисляется данная переменная.

Архитектурные свойства машины также могут быть описаны при помощи графов и. с. Назовем операционными шаблонами набор и. с. элементарных алгоритмов, которые могут быть реализованы аппаратно путем сцепления функциональных устройств. Длина этих цепочек для ЭВМ CRAY-I не более 7 (по числу функциональных устройств).

Для решения задач вычислительной математики на ЭВМ типа CRAY



Фиг. 2

выделен специальный набор операционных шаблонов, некоторые из них показаны на фиг. 2.

Работа вычислительной машины по шаблону иллюстрируется соответствующей временной диаграммой. Длина каждого элемента диаграммы соответствует времени выполнения отдельной векторной операции и состоит из двух частей: времени τ «разгона» конвейера для данной операции и времени получения результата для вектора длиной $1 \leq n \leq 64$. Диаграмма показывает последовательность и совмещение по времени отдельных векторных операций. Элемент диаграммы специфицируется соответствующим знаком операции, номерами векторных $V[i]$ или скалярных $S[i]$ регистров, в которых хранятся операнды и результат операции. Знак if соответствует операции проверки условия, которая выполняется с помощью так называемого регистра векторной маски, VM (см. [5]). Последовательность элементов диаграммы есть программа для выполнения отдельного операционного шаблона. Программа для операционного шаблона выполняется

не только с максимальным быстродействием, но и с минимальным числом регистров.

Теперь задача согласования вычислительного алгоритма с архитектурой машины математически может быть сформулирована как задача отыскания минимального покрытия и. с. алгоритма шаблонами из указанного набора.

Введенный формализм единого описания архитектуры алгоритма и архитектуры машины позволил создать технологию анализа и программирования численных алгоритмов, которая была применена при реализации метода потоков на ЭВМ типа CRAY-I.

Анализ алгоритма заключается в приведении его информационной структуры к некоторому стандартному виду и распадается на несколько этапов.

1. Векторизация алгоритма. И. с. алгоритма приводится к так называемой ярусно-параллельной форме (я. п. ф.). Я. п. ф. представляет собой последовательность ярусов, на которых расположены переменные алгоритма. Каждый ярус образован переменными, которые могут быть вычислены независимо друг от друга. Зависимость определяется между переменными, расположенными только на паре соседних ярусов. На каждом ярусе выделим группы переменных, которые вычисляются единообразно. Будем говорить, что такая группа образует вектор переменных, элементы которого определяются одной и той же функцией, реализуемой на векторных устройствах (векторной функцией).

Заметим, что для большинства численных алгоритмов уравнений математической физики представление и. с. алгоритмов в я. п. ф. является естественным.

Для рассматриваемого алгоритма ярусы его и. с. соответствуют временным слоям. Каждый n -й ярус (слой) содержит четыре векторные переменные ($\rho^n, \xi^n, \eta^n, \varepsilon^n$) и четыре соответствующие векторные функции, которые их определяют.

Взаимосвязь между соседними ярусами показана следующей системой формул:

$$(2.1a) \quad \rho^{n+1} = F_1(\rho^n, \xi^n, \eta^n), \quad \xi^{n+1} = F_2(\rho^n, \xi^n, \eta^n, \varepsilon^n),$$

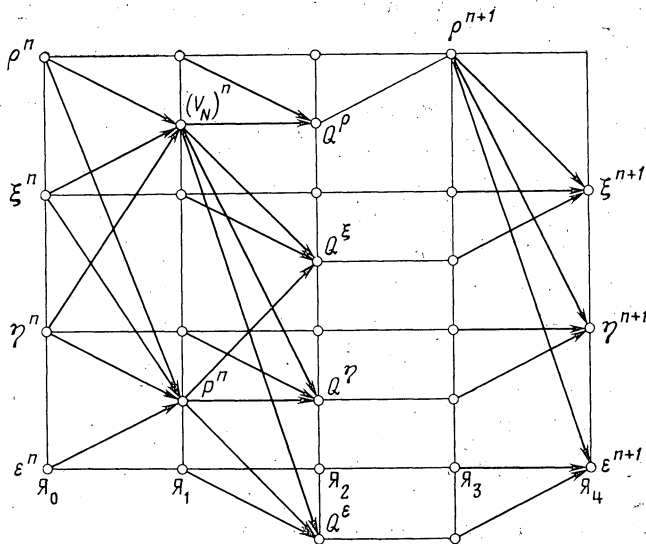
$$(2.1b) \quad \eta^{n+1} = F_3(\rho^n, \xi^n, \eta^n, \varepsilon^n), \quad \varepsilon^{n+1} = F_4(\rho^n, \xi^n, \eta^n, \varepsilon^n),$$

где F_i — векторные функции.

И. с. для системы формул (2.1) показана на фиг. 3. Длина вектора для каждой векторной переменной определяется числом элементарных ячеек Ω в слое. Последовательность выполнения векторных функций, расположенных на одном ярусе, может быть произвольной. Каждая векторная функция выполняется на векторных устройствах машины.

Система векторных функций может обладать следующей особенностью: две различные векторные функции могут содержать общие промежуточные переменные. Такие векторные функции назовем связными по этим переменным. Вычисление связных векторных функций сопряжено с рядом сложностей, которые могут привести к построению неэффективных программ.

Покажем, что в рассмотренной системе (2.1) векторные функции F_1 и F_4 являются связными. Рассмотрим более детально структуру этих функций. Для этого выпишем систему формул, определяющих вычисление неко-



Фиг. 3

торых промежуточных переменных для F_1 :

$$\rho_{ik}^{n+1} = F_1(\rho^n, \xi^n, \eta^n) = \rho_{ik}^n + \frac{\tau}{\Omega_{ik}} (Q_{i-1/2, k}^\rho - Q_{i+1/2, k}^\rho + Q_{i, k-1/2}^\rho - Q_{i, k+1/2}^\rho),$$

где

$$Q_{i+1/2, k}^\rho = \rho_{i+1/2, k}^n (V_N)_{i+1/2, k} S_{i+1/2, k},$$

$$(V_N)_{i+1/2, k}^n = 0.5[(u_{i, k} + u_{i+1, k}^n) \sin \varphi_{i+1/2, k} + (v_{i, k}^n + v_{i+1, k}^n) \cos \varphi_{i+1/2, k}],$$

$$(2.2) \quad \rho_{i+1/2, k}^n = \begin{cases} \rho_1 = 1.5\rho_{ik}^n - 0.5\rho_{i-1, k}^n, & \rho_1 \geq 0 \\ \rho_{ik}^n, & \rho_1 < 0 \end{cases} (V_N)_{i+1/2, k} \geq 0, \\ \begin{cases} \rho_2 = 1.5\rho_{i+1, k}^n - 0.5\rho_{i+2, k}^n, & \rho_2 \geq 0 \\ \rho_{i+1, k}^n, & \rho_2 < 0 \end{cases} (V_N)_{i+1/2, k} < 0.$$

Остальные потоки плотности распределения массы $Q_{i-1/2, k}^\rho, Q_{i, k-1/2}^\rho, Q_{i, k+1/2}^\rho$ на границах Ω вычисляются аналогично.

Вычисление векторной функции F_4 через промежуточные переменные определяется следующей системой формул:

$$\varepsilon_{ik}^{n+1} = F_4(\rho^n, \xi^n, \eta^n, \varepsilon^n) =$$

$$= \varepsilon_{ik}^n + \frac{\tau}{\Omega_{ik}} (Q_{i-1/2, k}^\varepsilon - Q_{i+1/2, k}^\varepsilon + Q_{i, k-1/2}^\varepsilon - Q_{i, k+1/2}^\varepsilon),$$

$$Q_{i+1/2, k}^\varepsilon = (\varepsilon_{i+1/2, k}^n + P_{i+1/2, k}^n) (V_N)_{i+1/2, k} S_{i+1/2, k},$$

$$(2.3) \quad \varepsilon_{i+1/2, k}^n = \begin{cases} \varepsilon_1 = 1.5\varepsilon_{ik}^n - 0.5\varepsilon_{i-1, k}^n, & \varepsilon_1 \geq 0 \\ \varepsilon_{ik}^n, & \varepsilon_1 < 0 \end{cases} (V_N)_{i+1/2, k} \geq 0, \\ \begin{cases} \varepsilon_2 = 1.5\varepsilon_{i+1, k}^n - 0.5\varepsilon_{i+2, k}^n, & \varepsilon_2 \geq 0 \\ \varepsilon_{i+1, k}^n, & \varepsilon_2 < 0 \end{cases} (V_N)_{i+1/2, k} < 0.$$

$$P_{i+1/2, k}^n = 0.5R(\rho_{ik}^n T_{ik}^n + \rho_{i+1, k}^n T_{i+1, k}^n),$$

где R — универсальная газовая постоянная, температура

$$T_{ik}^n = e_{ik}/C_v,$$

где C_v — теплоемкость газа при постоянном объеме.

Из формул (2.2) и (2.3) видно, что функции F_1 и F_2 связаны промежуточными переменными $(V_N)_{l+1/2, k}$ и $(V_N)_{l, k+1/2}$.

Имеются два способа вычисления связанных векторных функций:

- 1) вычислить связную переменную столько раз, во сколько функций она входит; при этом неоправданно увеличивается объем вычислений;
- 2) вычислить связную переменную один раз и хранить ее значение в векторном регистре для дальнейших вычислений; при большом числе связанных переменных не удастся построить программу, рационально использующую регистровую память.

Реализация этих способов на векторно-конвейерной машине неэффективна.

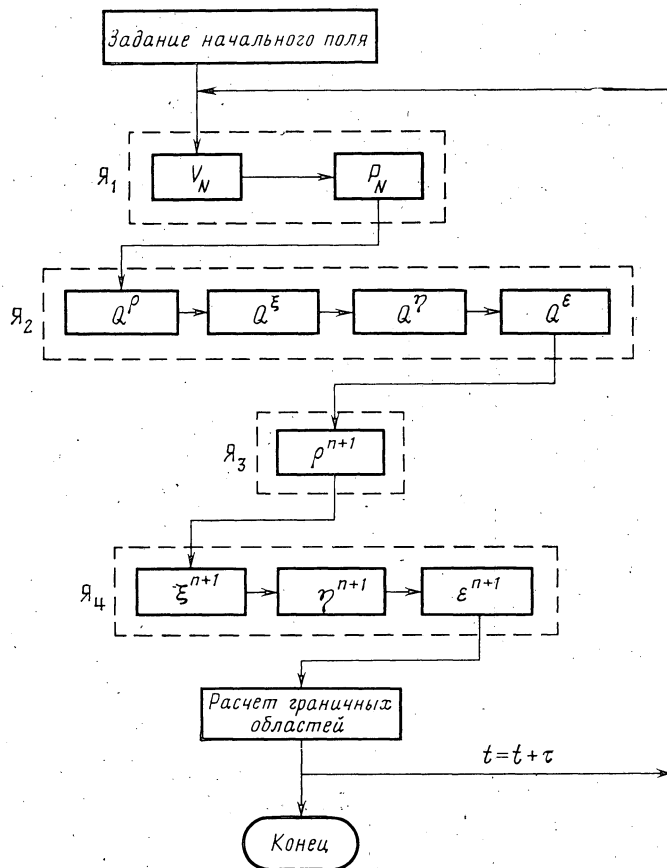
Другой путь решения проблемы связанных векторных функций состоит в разбиении такой функции на несвязные подфункции. Для этих целей необходимо выполнить еще один этап анализа алгоритма.

2. Приведение и.с. алгоритма к я.п.ф. с несвязными векторными функциями. Процедура получения несвязных векторных функций достаточно проста и очевидна. Пусть имеется пара связанных векторных функций. Упорядочим их и.с. таким образом, чтобы одинаковые переменные, принадлежащие разным функциям, находились на одном ярусе. Такую процедуру можно повторить для всех связанных переменных. В результате временной слой «расщепляется» на дополнительные ярусы, на которых будут расположены несвязные векторные функции. На фиг. 3 приведено такое расщепление и.с. векторных функций, определяющих расчет $(n+1)$ -го временного слоя для внутренней области течения газа.

3. Блок-схема программы. Ярусно-параллельная форма и.с. алгоритма с несвязными векторными функциями на ярусах позволяет построить схему программы стандартным образом. На фиг. 4 приведена блок-схема программы, построенная по ярусному разложению структуры алгоритма, приведенному на фиг. 3. Каждый блок схемы на фиг. 4 соответствует программе расчета одной векторной функции. Входные данные для каждого блока берутся из оперативной памяти, результат также помещается в оперативную память. Сама векторная функция рассчитывается только на сверхбыстрых регистрах. Последовательность расчета векторных функций в каждом ярусе может быть произвольной, но последовательность расчета ярусов фиксирована и определяется ярусной формой и.с. алгоритма фиг. 3. Блок за один проход обрабатывает практически параллельно все данные предыдущего временного слоя не более чем для 64 элементарных объемов-ячеек Ω . Для расчета всего временного слоя необходимо повторение этой же программы, но с данными других ячеек; число повторений $N=n/64$, где n — число объемов-ячеек в области интегрирования.

4. Планирование вычисления векторной функции. Быстродействие выполнения программы в основном зависит от хорошего согласования процесса вычисления векторной функции с архитектурой машины. Планирование согласования разбивается на три этапа.

На первом этапе и.с. векторной функции представляется в виде разложения на цепочки связанных между собой операций. Например, и.с. век-

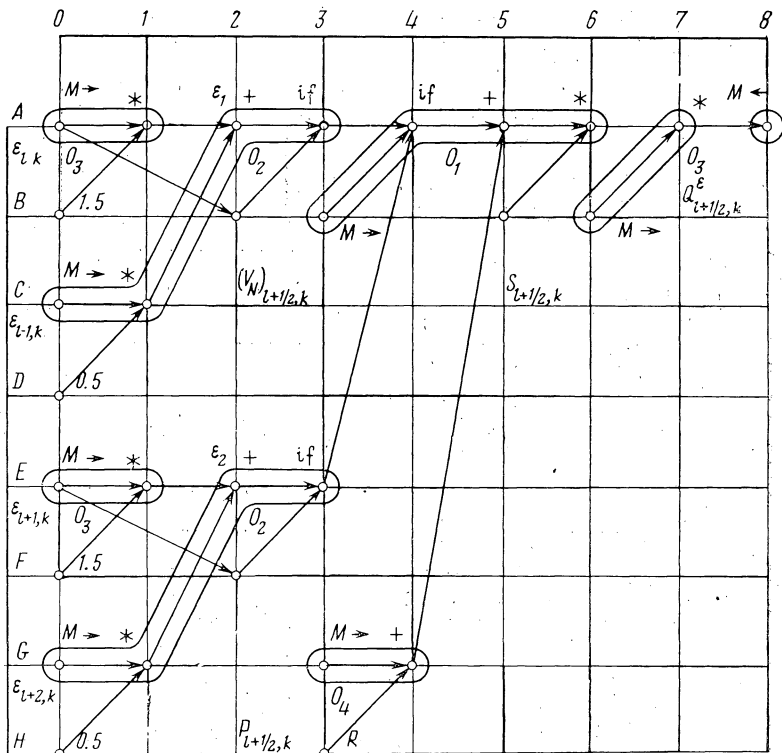


Фиг. 4

торной функции потока энергии Q^* через грань $l+1/2, k$ элементарного объема-ячейки Ω_{lk} , которая задается формулой (2.3), показана на фиг. 5. Как видно из графа и.с., цепочки имеют вид «елочек» и поэтому называются E-разложением структуры алгоритма. Вид E-разложения полностью соответствует виду структур алгоритмов, представленных операционными шаблонами на фиг. 2, и характеризуют эффективные возможности машины.

На втором этапе решается задача покрытия структуры алгоритма операционными шаблонами из заготовленного заранее набора. Простота выполнения этой процедуры очевидна, так как структура алгоритма в виде E-разложения соответствует структуре алгоритма шаблона. Отыскивается вариант покрытия, состоящий из минимального числа операционных шаблонов; он, очевидно, дает максимальное быстродействие. Формально такая процедура может быть построена как перебор всех возможных вариантов покрытия. На фиг. 5 показано оптимальное покрытие для векторной функции (2.3).

На третьем этапе решается задача планирования программы вычисления векторной функции, которая будет состоять из подпрограмм операционных шаблонов. Для этих целей и.с. алгоритма представляется в виде графа, вершины которого интерпретируются соответствующим операционным шаблоном. И.с. в операционных шаблонах (и.с.о.ш) для алгоритма потока энергии приведена на фиг. 6, а. Далее и.с.о.ш. представляется в виде разложения по ярусам для планирования возможного параллельного выполнения нескольких операционных шаблонов. Как видно из фиг. 6, вы-



Фиг. 5

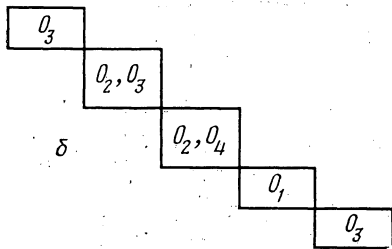
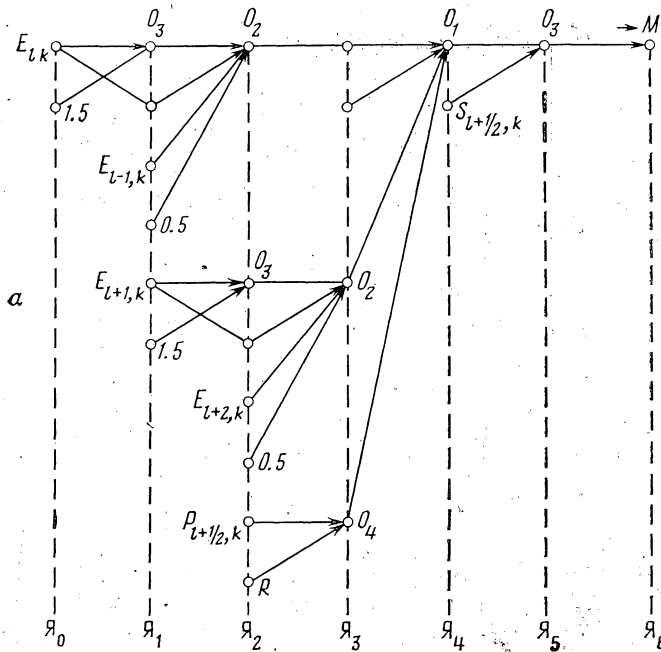
полнение O_2 и O_3 на ярусе $Я_2$ и O_2, O_4 на ярусе $Я_4$ может быть совмещено, что отражено на временной диаграмме фиг. 6, б. Последовательность выполнения операционных шаблонов и их совмещение приведено в детальной диаграмме фиг. 7, состоящей из последовательности машинных команд. По сути дела, она определяет программу вычисления функции.

§ 3. Особенности вычисления векторной функции, содержащей условия

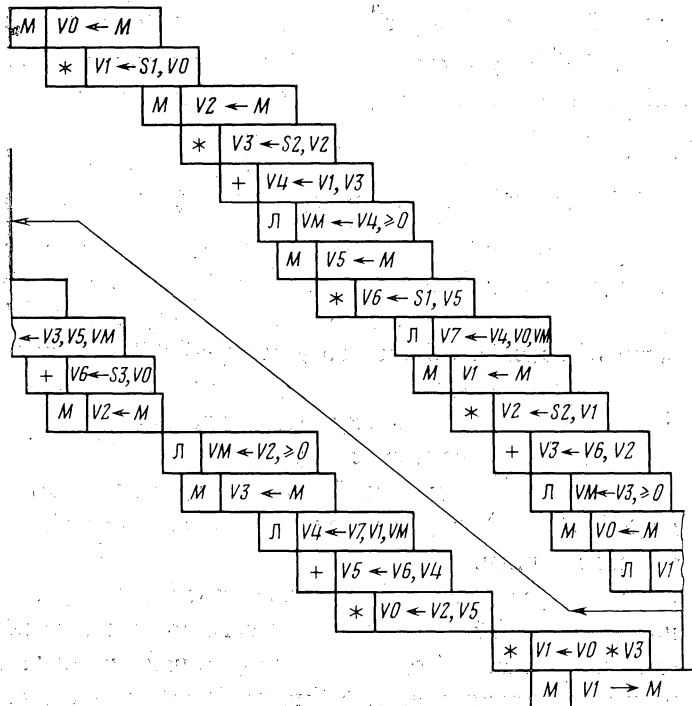
При самых первых применениях ЭВМ CRAY-1 был отмечен один из основных ее недостатков, связанный с невозможностью построения векторных вычислений для векторных функций, выражения которых выбираются в зависимости от условий, определяемых только в процессе вычислений [7]. Например, в нашем случае соотношения для массы и энергии

(2.2) содержат векторные функции с условиями для вычисления $\rho_{l+1/2,k}^n$ и

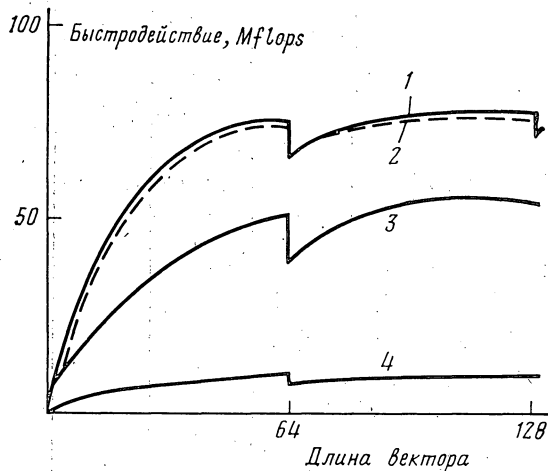
$\epsilon_{l+1/2,k}^n$. Заранее не известно, какие соотношения нужно использовать для вычисления этих физических переменных. В рассматриваемой задаче такие ситуации могут составлять от 15 до 35% случаев в зависимости от моделируемого течения. Причем положение таких областей в поле течения неопределенно. В таких случаях приходится либо отказываться от векторных вычислений, либо создавать сложные программы, объединяющие разрозненные области узлов или отдельные узлы, в которых вычисления однородны. Возможность построения таких программ обсуждается в [9]. Там же отмечается, что существующие трансляторы с языка ФОРТРАН для CRAY-1 не векторизуют циклы, содержащие операторы if.



Фиг. 6



Фиг. 7



Фиг. 8

Предложенная в настоящей работе согласующая технология подготовки программ для векторно-конвейерных ЭВМ позволила выбрать иной способ «борьбы» с условными выражениями (if-эффектом). Во всей внутренней области схема вычислений приводится к однородной посредством вычисления всех альтернативных выражений для определения переменной. Отбор необходимых значений согласно условиям также проводится однородно во всей области. Другими словами, при таком подходе становится возможной векторизация многих, вообще говоря, неоднородных алгоритмов. При этом, очевидно, увеличивается количество вычислительных операций. Однако быстродействие снижается лишь на 12% за счет совмещения операций, выполненных по согласующей технологии. На фиг. 5 и с. алгоритма вычисления потока энергии $Q_{i+\frac{1}{2},k}^e$, содержит четыре возможных варианта вычислений переменной $z_{i+\frac{1}{2},k}^n$, которые выполняются на векторных устройствах. Отбор нужных значений производится специальной операцией наложения векторной маски. Половина значений при этом не используется.

§ 4. Реализация и характеристики программы

Для решения задачи обтекания цилиндра совершенным газом была написана программа на автокоде для ЭВМ CRAY-I. Размерность сетки варьировалась вплоть до 130×140 , так что все массивы размещались в оперативной памяти 512 кслов. Объем программы ~3500 операторов.

Характеристики быстродействия показаны на фиг. 8:

кривая 1 — оценочное быстродействие программы, полученной по временным диаграммам для векторных функций;

кривая 2 — фактическое быстродействие программы, которое ниже оценочного на 2% за счет неучета некоторых особенностей работы устройств управления машины.

Представляет интерес сравнение производительности программы, разработанной по согласующей технологии, с программами того же алгоритма, но написанными на модификациях традиционных языков. На фиг. 8 (кривая 4) приведена характеристика быстродействия для программы на языке АЛГОЛ 60 с последующей векторизованной трансляцией на авто-

код CRAY-I. Низкая производительность ее обусловлена тем, что циклы, содержащие операторы *if*, не векторизовались транслятором и выполнялись на скалярных устройствах машины. Далее текст программы на АЛГОЛе 60 был модифицирован для устранения *if*-эффекта способом, который обсуждался выше. Характеристики быстродействия полностью векторизованной программы существенно выше (фиг. 8, кривая 3), но значительно уступают характеристикам программы, написанной по согласующей технологии. При переходе к совместному планированию векторных функций удается достигнуть быстродействия в 130 Mflops.

В заключение можно отметить, что организация программ, согласованных с архитектурой машины, не исчерпывает всех возможностей достижения максимальной производительности машины. Значительный эффект, возможно, даст разработка новых подходов к построению численных методов, ориентированных на высокопроизводительные ЭВМ, например [10], [11].

Литература

1. Петерсон В. Л. Вычислительная аэродинамика.— Тр. Ин-та инж. по электротехн. и радиоэлектронике. М., 1984, т. 72, № 1, с. 83–97.
2. Белоцерковский О. М., Северинов Л. И. Консервативный метод «потоков» и расчет обтекания тела конечных размеров вязким теплопроводным газом.— Ж. вычисл. матем. и матем. физ., 1973, т. 13, № 2, с. 385–397.
3. Бабаков А. В. Применение метода потоков к одной задаче динамики вязкой стратифицированной жидкости.— Ж. вычисл. матем. и матем. физ., 1983, т. 23, № 2, с. 432–439.
4. Белоцерковский О. М., Северинов Л. И., Бабаков А. В. О некоторых применениях консервативного метода потоков.— В кн.: Аэромехан. и газовая динамика. М.: Наука, 1976, с. 124–136.
5. Головкин Б. А. Параллельные вычислительные системы. М.: Наука, 1980.
6. Столяров Л. Н. Диалоговое программирование.— В кн.: Материалы V Всес. семинара по комплексам программ матем. физ. Новосибирск: ИТПМ СО АН СССР, 1978, с. 136–145.
7. Столяров Л. Н. Структурный анализ алгоритмов для эффективной организации вычислений.— В кн.: Вычисл. системы и автоматизация научн. иссл. М.: МФТИ, 1980, с. 23–45.
8. Столяров Л. Н. Алгебра для формального описания информационной структуры пакета.— В кн.: Структура и организация пакетов программ. Тезисы докл. Тбилиси: Мецниереба, 1978, с. 159–160.
9. Уильямсон Д. Л., Шварцраубер П. Н. Модель численного прогноза погоды: перспективы использования супер-ЭВМ CRAY-I.— Тр. Ин-та инж. по электротехн. и радиоэлектронике. М., 1984, т. 72, № 1, с. 69–83.
10. Белоцерковский О. М., Панарин А. И., Щенников В. В. Метод параметрической коррекции разностных схем.— Ж. вычисл. матем. и матем. физ., 1984, т. 24, № 1, с. 65–74.
11. Белоцерковский О. М., Панарин А. И., Щенников В. В. Обобщенные разностные схемы и метод параметрической коррекции разностных схем.— В кн.: Кибернетика и вычисл. техн. Вып. 1. М.: Наука, 1985, с. 99–117.

Поступила в редакцию 16.VII.1985