

ЭФФЕКТИВНОСТЬ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЧИСЛЕННОГО РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ ДЛЯ УРАВНЕНИЙ С ЧАСТНЫМИ ПРОИЗВОДНЫМИ ПРИ ИХ РЕАЛИЗАЦИИ НА SIMD КОМПЬЮТЕРАХ

Фельдман Л.П., Труб И.И.

Кафедра ПМиИ ДонГТУ

E-mail: feldman@pmi.donetsk.ua trub@pmi.donetsk.ua

Abstract

Feldman L., Trub I. About the effectiveness of parallel numerical algorithms for boundary partial equation problems and implementation on SIMD-supercomputers. This article contains the description of parallelism of some basic numerical methods for partial equations of parabolic, elliptic and hyperbolic types. Multiprocessor configurations for implementation of those methods are proposed. Evaluation of main indexes of parallel algorithms, such as width, speedup and others, are derived.

Введение

Современный этап развития вычислительной техники характеризуется интенсивными исследованиями и разработками в области архитектуры параллельных вычислительных систем и параллельного программирования. Наибольшая необходимость в параллельных вычислениях возникает при решении краевых задач для уравнений с частными производными, требующих огромных объемов вычислений. Характерной чертой всех новых ВС является возможность одновременного или параллельного использования для обработки большого числа процессоров. Их создание определило один из важнейших путей повышения скорости решения сложных и трудоемких задач.

Эффективность использования многопроцессорных систем снижается не только из-за ограниченных возможностей их внутренних коммуникационных сетей. Высокая производительность ВС будет достигнута лишь в том случае, если все процессоры или большая их часть будет постоянно загружена. Но алгоритм решения задачи в силу его собственной структуры может просто не обладать возможностью обеспечить постоянную загрузку большого числа процессов независимо от устройства коммуникационной сети. Имеется много численных методов, которые нельзя эффективно реализовать ни на каких многопроцессорных системах. Таким образом, использование многопроцессорных систем должно обязательно сопровождаться согласованием структуры численных методов и архитектуры ВС.

Ниже рассматриваются алгоритмы ориентированные на SIMD системы, которые могут наиболее полно использовать потенциальный параллелизм, имеющийся в разностных схемах, аппроксимирующих краевые задачи для уравнений в частных производных. Наибольший эффект от параллельного процессора SIMD, по видимому, достигается в тех случаях, когда он применяется для выполнения матричных вычислений и решения дифференциальных уравнений разностными методами. Известно, что многие алгоритмы в этих областях требуют одних и тех же преобразований, повторяющихся много раз над элементами данных.

Наиболее распространенными критериями оценки эффективности параллельных алгоритмов являются ускорение и эффективность. Пусть N - число параметров задачи и $T_p(N)$ - время выполнения параллельного

алгоритма на ВС с $p > 1$ процессорами, а $T_1(N)$ - время работы "наилучшего" последовательного алгоритма. Тогда

$$S_p(N) = \frac{T_1(N)}{T_p(N)} \leq p$$

называется ускорением, а

$$E_p(N) = \frac{S_p(N)}{p} \leq 1 \quad (1)$$

эффективностью параллельного алгоритма. Одной из целей при разработке параллельных алгоритмов является достижение наибольшего ускорения: в идеальном случае $S_p = p$. Однако максимальное ускорение можно получить только для задач, по существу тривиальных. Главные факторы, обуславливающие меньшие значения ускорения, это отсутствие максимального параллелизма в алгоритме и затраты времени на обмен данными между процессорами. Переходя к рассмотрению численных алгоритмов решения задач математической физики будем оценивать их учитывая оба этих фактора. Сначала рассмотрим алгоритмы, обладающие наибольшим параллелизмом, а затем оценим затраты времени на обмены при их реализации в SIMD - системе для модели MasPar - 1216.

1. Алгоритмы решения одномерных параболических уравнений.

Численное решение одномерной параболической задачи

$$\left. \begin{aligned} \frac{\partial u}{\partial t} &= a^2 \frac{\partial^2 u}{\partial x^2} + f(x, t), \quad t > 0; \quad 0 < x < 1; \\ u(x, 0) &= \varphi(x); \quad u(0, t) = \psi(t); \quad u(1, t) = \xi(t) \end{aligned} \right\} \quad (2)$$

по явной разностной схеме приводит к следующим расчетным формулам, позволяющим получить решение последовательно для очередного временного слоя

$$\left. \begin{aligned} v_n^0 &= \varphi_n; \quad n = \overline{0, N+1}; \\ v_n^{k+1} &= \sigma^2 v_{n-1}^k + (1 - 2\sigma^2) v_n^k + \sigma^2 v_{n+1}^k + \tau f_n^k, \quad n = \overline{1, N}, \end{aligned} \right\} \quad (3)$$

где $v_0^k = \psi^k$, $v_{N+1}^k = \xi^k$, $\sigma^2 = a^2 \tau / h^2$; τ, h - шаги сетки. Очевидно, что этот алгоритм на каждом временном шаге обладает максимальным параллелизмом: $S_N(N) = N$. Кроме того он очень удобен для реализации в SIMD- процессорах, т.к. на каждом временном шаге он требует лишь двух обменов данными, которые выполняются между соседними процессорными элементами (ПЭ). Для реализации этого алгоритма требуется N процессоров, его эффективность

$$S_N(N) = N / (1 + 2t_u/t_f),$$

где t_u - время параллельного обмена в SIMD структуре, t_f время вычисления значения v_n^{k+1} по формуле (3) на однопроцессорной ЭВМ. Однако применение этого алгоритма ограничивается тем, что явная разностная схема является устойчивой лишь для $\tau \leq h^2 / (2a^2)$.

Аппроксимация задачи (2) чисто неявной разностной схемой приводит к следующей трехдиагональной системе уравнений

$$\left. \begin{aligned} v_n^0 &= \varphi_n; \\ v_0^{k+1} &= \psi^{k+1}, \\ \sigma^2 v_{n-1}^{k+1} - (1 + 2\sigma^2) v_n^{k+1} + \sigma^2 v_{n+1}^{k+1} &= -v_n^k - \tau f_n^k, \quad n = \overline{1, N}, \\ v_N^{k+1} &= \xi^{k+1}. \end{aligned} \right\} \quad (4)$$

На последовательных ЭВМ систему (4) решают обычно методом прогонки, который является последовательным и его реализация на параллельных ВС неэффективна. Для решения трехдиагональных систем линейных уравнений имеется ряд эффективных параллельных алгоритмов [J. Evans, p. 313]: LU-разложение Стоуна, алгоритм рекуррентного произведения, метод параллельной циклической редукции, краткое рассмотрение последнего из них проводится ниже [R. W. Hockney, p. 277].

Написав три прилежащих уравнения (4) для случая переменных коэффициентов опустим в них для краткости верхние индексы и приняв для удобства $N-1=2^q$, а коэффициенты и правую часть обозначив через $\alpha_n, \beta_n, \gamma_n, g_n$ соответственно, получим

$$\left. \begin{aligned} \beta_{n-1} v_{n-2} - \alpha_{n-1} v_{n-1} + \gamma_{n-1} v_n &= g_{n-1} \\ \beta_n v_{n-1} - \alpha_n v_n + \gamma_n v_{n+1} &= g_n \\ \beta_{n+1} v_n - \alpha_{n+1} v_{n+1} + \gamma_{n+1} v_{n+2} &= g_{n+1}, \quad n = \overline{2, N-1}. \end{aligned} \right\} \quad (5)$$

Если в каждой такой группе из трех уравнений исключить неизвестные v_{n-1} и v_{n+1} , а затем вновь сгруппировать по три соседних, то получим

$$\left. \begin{aligned} \beta_{n-2}^{(1)} v_{n-4} - \alpha_{n-2}^{(1)} v_{n-2} + \gamma_{n-2}^{(1)} v_n &= g_{n-2}^{(1)}, \\ \beta_n^{(1)} v_{n-2} - \alpha_n^{(1)} v_n + \gamma_n^{(1)} v_{n+2} &= g_n^{(1)}, \\ \beta_{n+2}^{(1)} v_n - \alpha_{n+2}^{(1)} v_{n+2} + \gamma_{n+2}^{(1)} v_{n+4} &= g_{n+2}^{(1)}, \quad n = \overline{2, N-1}, \end{aligned} \right\} \quad (5^1)$$

где

$$\beta_n^{(1)} = \frac{\beta_{n-1} \beta_n}{\alpha_{n-1}}, \quad \alpha_n^{(1)} = \alpha_n + \frac{\beta_n \gamma_{n-1}}{\alpha_{n-1}} + \frac{\beta_{n+1} \gamma_n}{\alpha_{n+1}}, \quad \gamma_n^{(1)} = \frac{\gamma_n \gamma_{n+1}}{\alpha_{n+1}}, \quad g_n^{(1)} = \frac{\beta_n g_{n-1}}{\alpha_{n-1}} + g_n + \frac{\gamma_n g_{n+1}}{\alpha_{n+1}}.$$

Представленный способ выполнения циклической редукции применялся ранее на последовательных компьютерах, т.к. имеет наименьшее число скалярных арифметических операций и, следовательно, является наилучшим вариантом последовательного алгоритма. Для удобства обозначим вектор $\Pi_n = (\alpha_n, \beta_n, \gamma_n, g_n)$, чтобы указать все значения, вычисляемые по уравнениям (5¹). Через $q-1$ шагов исключений останется одно уравнение с одной неизвестной

$$-\alpha_{N/2}^{(q-1)} v_{N/2} = g_{N/2}^{(q-1)}.$$

Обычно эту процедуру называют прямым ходом. Для его выполнения на последовательном процессоре требуется время

$$T_d(1) = t_f(N - \log_2 N - 1),$$

где t_f — время вычисления коэффициентов α, β, γ, g на последовательном компьютере на очередном шаге исключения. Для параллельного выполнения этой процедуры требуется $N/2$ процессоров при этом затрачивается время.

$$T_d(N/2) = t_f(q-1) = t_f(\log_2 N - 1).$$

Кроме того, на каждом шаге активный процессор должен передать двум, связанным с ним на этом шаге процессорам, значения четырех компонент вектора Π . На это требуется время

$$T_u = 8t_u (\log_2 N - 1),$$

где t_u - время параллельного обмена в SIMD структуре.

Обратный ход состоит в вычислении значения неизвестной из уравнения

$$v_n = -(g_n^{(l-1)} - \beta_n^{(l-1)} v_{n-2^{(l-1)}} - \gamma_n^{(l-1)} v_{n+2^{(l-1)}}) / \alpha_n^{(l-1)},$$

для $l=q, q-1, \dots, 1$, где $n=2^{(l-1)}$ с шагом $2^{(l-1)}$ до $N-2^{(l-1)}$ и $v_0 = v_N = 0$.

На одном компьютере для его выполнения потребуется время

$$T_r(1) = t_g (N-1),$$

где t_g - время вычисления неизвестной из уравнения в обратном ходе. Для параллельного выполнения обратного хода потребуется $N/2$ процессоров при этом затрачивается время

$$T_r(N/2) = t_g \log_2 N.$$

Здесь также надо учесть время передачи данных на каждом шаге, которое составит для обратного хода величину

$$T_v = 4t_u \log_2 N.$$

Таким образом, ускорение параллельного алгоритма SERICR равно

$$S_{N/2}(N) = \frac{T_d(1) + T_r(1)}{T_d(N/2) + T_u + T_r(N/2) + T_v} = \frac{(t_r + t_g)(N-1) - t_r \log_2 N}{(t_r + t_g + 12t_u) \log_2 N - t_r - t_u}.$$

Если на SIMD системе на каждом шаге параллельно на $N-1$ процессоре вычислить коэффициенты уравнений (5¹), а затем передать соответствующим процессорам их значения, то процесс решения системы уравнений (5) потребует $q+1$ шаг. В (5¹) следует положить $v_j=0$, для всех $j < 0$ и $j > N+1$, чтобы иметь возможность параллельного исключения неизвестных в каждой группе. На каждом шаге, кроме последнего, каждый процессор должен передать четыре величины. Очевидно, что процесс будет рекурсивно продолжаться до тех пор, пока после $q = \log_2(N-1)$ шагов каждое уравнение будет содержать только одну неизвестную v_n . На последнем $q+1$ шаге все неизвестные могут быть найдены также параллельно. Таким образом, обратный ход состоит здесь из одного шага. Ускорение этого варианта алгоритма циклической редукции, называемого PARACR, равно

$$S_N(N) \approx \frac{N(t_r + t_g) - t_g \log_2 N}{(t_r + 8t_u) \log_2 N} = O\left(\frac{N}{\log_2 N}\right),$$

т.е. больше, чем в алгоритме SERICR

2. Алгоритмы параллельного решения многомерных параболических краевых задач

Для решения многомерных параболических краевых задач обычно используют явные и неявные методы. Явные разностные схемы для многомерных параболических задач, так же как и в одномерном случае, обладают естественным максимально возможным параллелизмом. Однако, из-за условной устойчивости таких разностных схем их использование не всегда возможно.

Так, например, для двухмерной параболической задачи

$$\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f(x, y, t) \quad (6)$$

с соответствующими начальными и краевыми условиями разностные уравнения для явной разностной схемы имеют вид

$$v_{m,n}^0 = \varphi_{m,n},$$

(6¹)

$$v_{m,n}^{k+1} = s^2(v_{m-1,n}^k + v_{m+1,n}^k) + (1 - 4s^2)v_{m,n}^k + s^2(v_{m,n-1}^k + v_{m,n+1}^k) + \tau f_{m,n}^k$$

где $\sigma^2 = a^2\tau/h^2$.

Схема параллельного решения неявного PARACR-метода для одномерной параболической задачи.

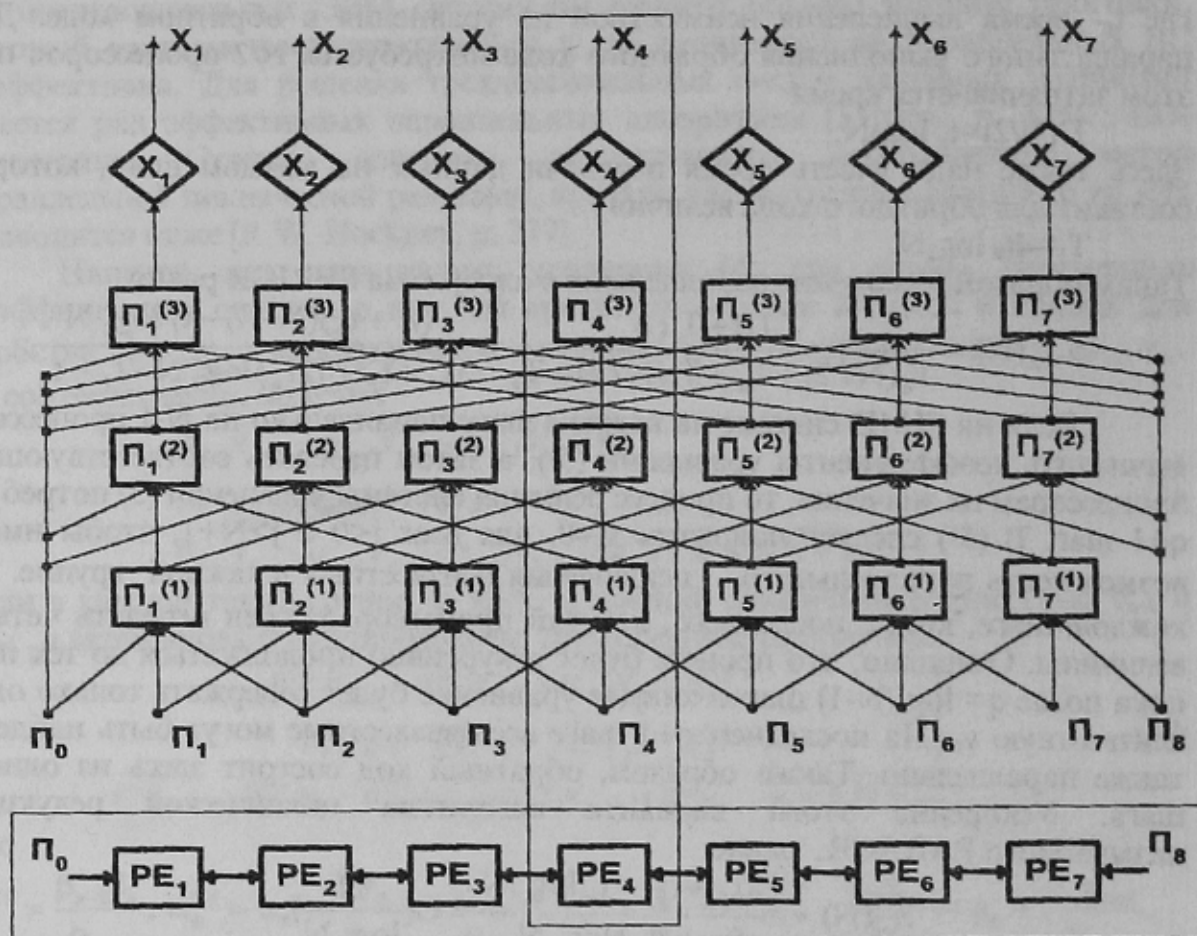


Рис. 1

Если решается краевая задача в квадрате, то при наличии N^2 процессоров она имеет максимальное распараллеливание. Ускорение равно

$$S_{N^2}(N^2) = N^2 / (1 + \theta),$$

где $\theta = 4t_v / t_r$ отношение времени обмена в SIMD структуре при вычислении значения по формуле (6¹) ко времени вычисления по этой же формуле на одном процессоре.

Эффективных параллельных прямых методов решения систем неявных разностных уравнений в настоящее время нет.

Большие возможности для распараллеливания дают разностные схемы расщепления [А.А. Самарский с.377]. Для двухмерной параболической задачи (6) локально одномерная схема расщепления имеет вид

$$\left. \begin{aligned} \frac{\tilde{v}_{m,n} - v_{m,n}^k}{\tau} &= a^2 \frac{\tilde{v}_{m-1,n} - 2\tilde{v}_{m,n} + \tilde{v}_{m+1,n}}{h^2} + 0.5f_{m,n}^k, \\ \frac{v_{m,n}^{k+1} - \tilde{v}_{m,n}}{\tau} &= a^2 \frac{v_{m-1,n}^{k+1} - 2v_{m,n}^{k+1} + v_{m+1,n}^{k+1}}{h^2} + 0.5f_{m,n}^{k+1}. \end{aligned} \right\} (7)$$

Здесь решение на каждом временном шаге состоит из двух этапов: нахождения промежуточных значений $\tilde{v}_{m,n}$, а затем вычисления значений на очередном временном шаге $v_{m,n}^{k+1}$. На первом этапе определяются $\tilde{v}_{m,n}$ из системы

$$\sigma^2 \tilde{v}_{m-1,n} - (1 + 2\sigma^2) \tilde{v}_{m,n} + \tilde{v}_{m+1,n} = -v_{m,n-1} - (1 - 2\sigma^2)v_{m,n} - v_{m,n+1} + 0.5\tau f_{m,n}^k, \quad (71)$$

для каждого n . На втором этапе определяются $v_{m,n}^{k+1}$ из системы уравнений

$$\sigma^2 v_{m,n-1}^{k+1} - (1 + 2\sigma^2)v_{m,n}^{k+1} + \sigma^2 v_{m,n+1}^{k+1} = -\sigma^2 \tilde{v}_{m-1,n} - (1 - 2\sigma^2) \tilde{v}_{m,n} - \sigma^2 \tilde{v}_{m+1,n} + 0.5\tau f_{m,n}^k, \quad (72)$$

для каждого m . На каждом из этих этапов разностные уравнения представляют трехдиагональные системы уравнений, которые как и в случае одномерной задачи целесообразно решать методом циклической редукции.

Если для упрощения оценок принять, что область x, y для задачи (6) является квадратом, то при степени параллелизма этого алгоритма равной N^2 наиболее эффективным является метод PARACR, ускорение которого равно

$$S_{N^2}(N^2) = \frac{N\{t_r[N - \log_2 N] + t_g N\}}{(t_r + 8t_u)\log_2 N} = O(N^2 / \log_2 N).$$

Если использовать только N процессоров, то каждую систему уравнений можно решать методом PARACR, используя для этого N процессоров, а затем последовательно решить N таких систем, либо каждую систему решать на одном процессоре методом SERICR, а все N систем параллельно.

1). Каждое из первых уравнений (7) при фиксированном n решается параллельно методом PARACR на N процессорах, а затем последовательно по n решается N таких систем. Аналогично решаются вторые уравнения из (7).

Ускорение PARACR $S_N(N^2) = \frac{\{t_r[N - \log_2 N] + t_g N\}}{(t_r + 8t_u)\log_2 N} = O(N / \log_2 N)$. Эффектив-

ность PARACR $E_N(N^2) \approx (\log_2 N)^{-1}$.

2). Каждое из первых уравнений (7) при фиксированном n решается последовательно по SERICR на одном процессоре, а N систем параллельно на N PE/ Ускорение SERICR $S_N(N^2) = N$. Эффективность SERICR $E_N(N^2) = 1$.

Приведенные оценки свидетельствуют о преимуществах метода SERICR.

3. Параллельные алгоритмы решения эллиптических краевых задач.

Оценки параллельности алгоритмов проведем для модельной задачи, аппроксимирующей уравнение Пуассона Соответствующее разностное уравнение которой имеет вид

$$(v_{m-1,n} + v_{m+1,n} - 4v_{m,n} + v_{m,n-1} + v_{m,n+1}) / h^2 = f_{m,n}, \quad m, n = \overline{0, N}. \quad (8)$$

Эффективных прямых методов решения подобных систем уравнений в настоящее время не имеется. Анализ алгоритмов решения разностных уравнений, проведенный Марчуком Г.И. [с. 279], позволил сделать выбор для реализации на SIMD в системе MasPar следующих параллельных алгоритмов: метода верхней релаксации, метода переменных направлений и релаксационного многосеточного метода [Федоренко Р.П.]. Первые два из

указанных методов широко известны, в отличие от третьего, поэтому кратко изложим идею метода Федоренко. Для уменьшения нормы первоначальной погрешности в N^2 раз этот метод требует $O(N^2)$ арифметических операций. Известно, что наиболее быстросходящийся метод Дугласа-Рекфорда требует $O((\ln N)N^2)$ операций.

Для решения разностных уравнений (8) используем схему установления

$$v_{m,n}^{k+1} = v_{m,n}^k + \tau \left[\frac{v_{m-1,n}^k - 2v_{m,n}^k + v_{m+1,n}^k}{h^2} + \frac{v_{m,n-1}^k - 2v_{m,n}^k + v_{m,n+1}^k}{h^2} - f_{m,n} \right] \quad (9)$$

Процесс установления для рассматриваемой задачи сходится, однако на мелкой сетке сходимость медленная. После выполнения k итераций по формуле (9) найдем поправку $\epsilon_{m,n}$ к приближенному решению $v_{m,n}^k$, которая удовлетворяет уравнению на вдвое более крупной сетке

$$\frac{\epsilon_{m-1,n} + \epsilon_{m+1,n} - 4\epsilon_{m,n} + \epsilon_{m,n-1} + \epsilon_{m,n+1}}{(2h)^2} = \Gamma_{m,n}^k, \quad (10)$$

$$\text{где } \Gamma_{m,n}^k = \frac{v_{m-1,n}^k + v_{m+1,n}^k - 4v_{m,n}^k + v_{m,n-1}^k + v_{m,n+1}^k}{(2h)^2} - f_{m,n}$$

представляет собой невязку, возникающую при подстановке v^k в уравнение (8).

Благодаря более крупной сетке быстрее происходит убывание ошибки. Ускорение сходимости, достигнутое за счет использования укрупненной сетки и итерационного процесса (9), может оказаться недостаточным. Поэтому целесообразно проделать еще одно укрупнение сетки вдвое. При решении задачи на четверо укрупненной сетке снова использовать процесс удвоения и т.д. Затем, сначала с самой крупной сетки интерполируем полученную там последнюю поправку на вдвое мелкую сетку, вносим ее в полученную до этого поправку на предпоследней сетке и делаем несколько итераций, чтобы погасить привнесенную при интерполяции ошибку. Результат этих итераций интерполируем на еще вдвое мелкую сетку и т.д. На последнем шаге после интерполяции и внесения поправки с предшествующей сетки получим приближенное решение. Рассмотренный многосеточный метод легко распараллеливается и эффективно реализуется на SIMD системах.

4. Параллельные алгоритмы решения смешанных гиперболических задач Коши

Для параллельного решения одномерной смешанной гиперболической задачи для уравнения второго порядка были использованы как явные, так и неявные разностные схемы. Эффективность решения неявных разностных схем достигалась за счет применения метода параллельной циклической редукции. Численное решение задач для систем гиперболических уравнений базировалось на методе характеристик. Разработаны параллельные алгоритмы, позволяющие находить как сами характеристики, так и значения искомых функций на соответствующих характеристиках. Основная трудность в решении подобных задач заключается в большом объеме данных, которые надо хранить в ПЭ, т.к. обращение к host ЭВМ для вывода значительно замедляет решение.

Постановка задачи, решаемой методом характеристик, выглядит следующим образом:

$$A \frac{\partial^2 U}{\partial x^2} + 2B \frac{\partial^2 U}{\partial x \partial y} + C \frac{\partial^2 U}{\partial y^2} = F \left(\frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, U, x, y \right) \text{ с начальными условиями,}$$

заданными на некоторой кривой $U(x, y)$, $P = \frac{\partial U}{\partial x}$, $Q = \frac{\partial U}{\partial y}$. На рис.2 изображена

“треугольная” мультипроцессорная конфигурация, используемая при реализации метода характеристик. $(N-1)$ - число интервалов, на которые разбивается линия Коши. Решение ищется в характеристическом треугольнике. Точки, обведенные овалами, рассчитываются одновременно. Мультипроцессорная конфигурация описана как LIST $[0..N*(N+1)/2-1]$. Для ее

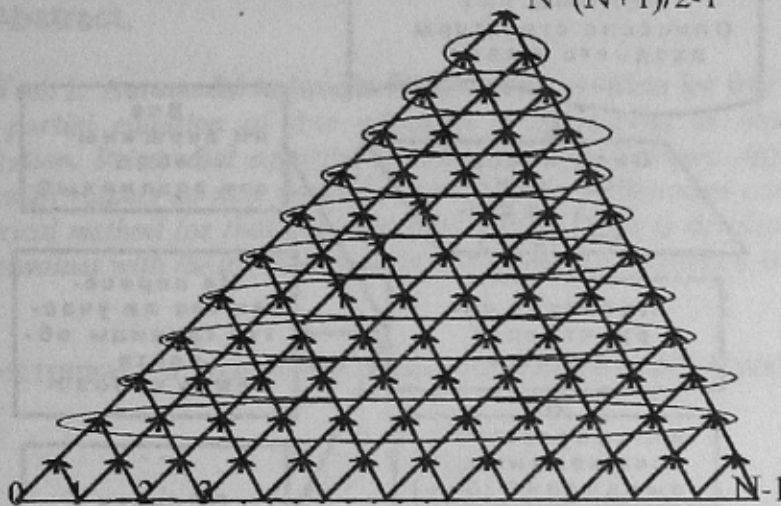


Рис 2. Мультипроцессорная конфигурация для метода характеристик

треугольной интерпретации нужно построить функцию регулярных связей между процессорными элементами $i \rightarrow F(i)$, где i - номер процессорного элемента (стрелки на рис.2). Решение дается формулой (вывод опущен):

$$F(i) = \left\{ N + i - \left[\frac{2 * N + 1 - \sqrt{4 * N^2 + 4 * N + 1 - 8 * i}}{2} \right], \text{ if } \sqrt{4 * N^2 + 4 * N + 1 - 8 * i} - \text{integer} \right.$$

$$F(i) = \left\{ N + i - 1 - \left[\frac{2 * N + 1 - \sqrt{4 * N^2 + 4 * N + 1 - 8 * i}}{2} \right], \text{ if } \sqrt{4 * N^2 + 4 * N + 1 - 8 * i} - \text{integer} \right.$$

В противном случае в конфигурации существуют обе указанные связи.

Для решения двумерных гиперболических уравнений разработаны параллельные алгоритмы, реализующие векторный метод расщепления, являющийся обобщением метода расщепления для параболических уравнений (1, с.330). Гиперболическая краевая задача на плоскости представляется с помощью соответствующих замен функций в виде системы трех уравнений в частных производных первого порядка, матрица которой расщепляется на сумму двух матриц, в каждую из которых входят частные производные только по x или только по y .

5. Библиотека программ для параллельного решения краевых задач.

Библиотека включает процедуры всех численных методов, рассмотренных выше, а также различные вспомогательные программы. Например, проверка корректности постановки задачи, сканирование области, в которой ищется решение, процедуры ввода-вывода, интерпретатор

символьных выражений для ввода функций и другие. Библиотека позволяет выполнить численное решение двумерных параболических и эллиптических краевых задач в области, представляющей собой замкнутую многоугольник, участками границы, которого могут быть отрезки прямых и дуги окружностей. Граничные условия на каждом участке границе задаются в виде условий третьего рода. Реализованы как достаточно простые методы Якоби, Зейделя, верхней релаксации, так и сложные методы-многосеточные, экстраполяция, а также методы Шварца и разделения области. Программы составлены на языке параллельного программирования Parallaxis-3 для ОС Linux, часть вспомогательных процедур написана на C++.

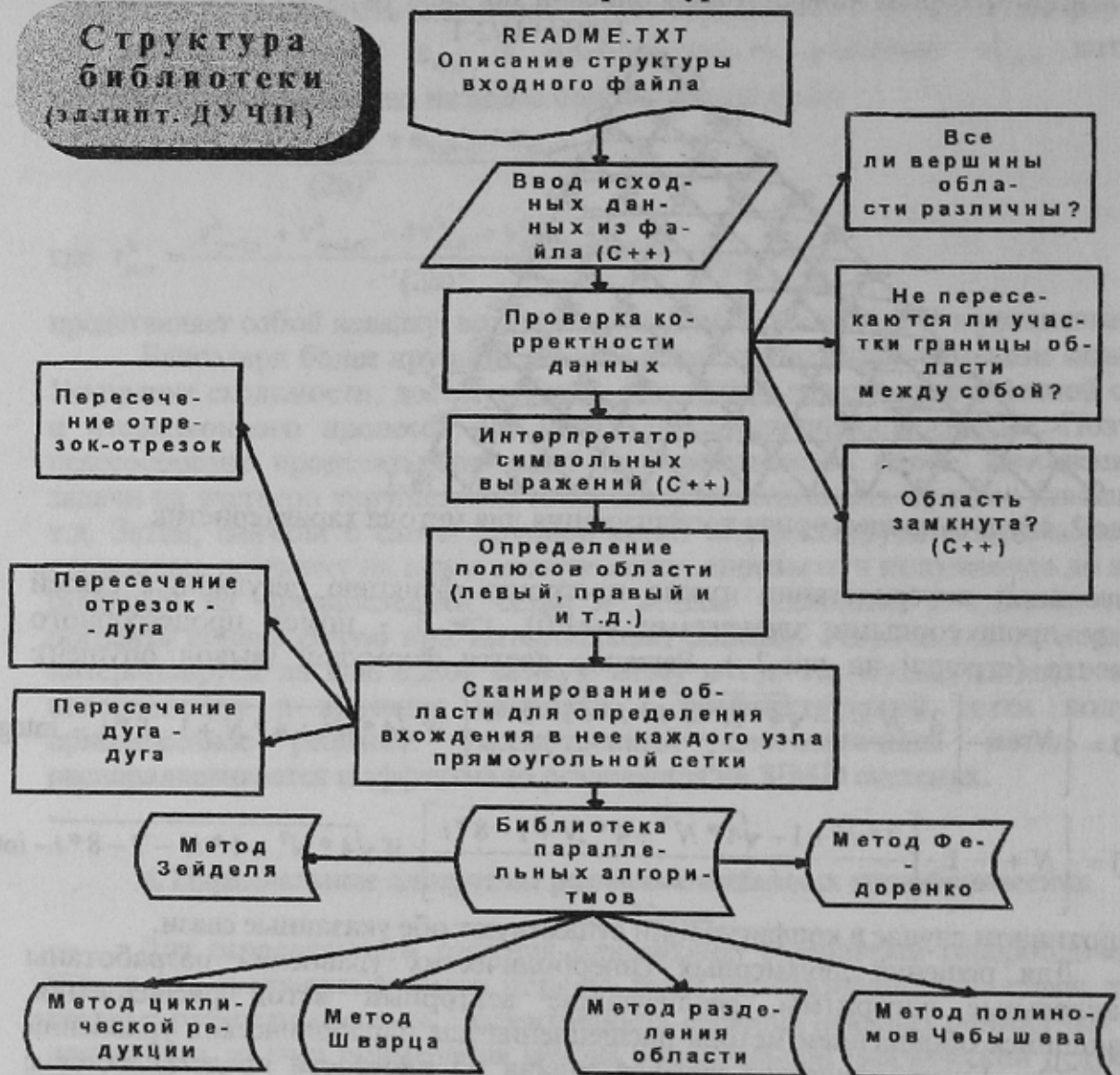


Рис.3 Структура библиотеки

Литература

1. Марчук .Г.И. Методы вычислительной математики. -М.: Наука, 1989.
2. Самарский А.А., Гулин А.В. Численные методы. -М.: Наука, 1989.
3. Parallel processing system. Ed. D.J. Evans.-London, русс.пер.1985.
4. Parallel computers. R.W. Hockney, C.R.Jesshope.- Bristol, русс. пер.1986.
5. J.M. Ortega. Introduction to Parallel and Vektor Solution of Linear System.