

Лекция 11. Параллельные алгоритмы решения уравнения теплопроводности.

1. Постановка модельной трёхмерной начально-краевой задачи

Рассмотрим проблему решения трехмерного линейного уравнения теплопроводности:

$$\frac{\partial u}{\partial t} = \sum_{\alpha=1}^3 \frac{\partial}{\partial x_{\alpha}} \left(k_{\alpha}(\bar{x}, t) \frac{\partial u}{\partial x_{\alpha}} \right) + f(\bar{x}, t), \quad \bar{x} \in D, \quad t > 0. \quad (1)$$

Здесь $u = u(\bar{x}, t)$ – температура, функция пространственных координат $\bar{x} = (x_1, x_2, x_3)$ и времени t , $k_{\alpha} = k_{\alpha}(\bar{x}, t) > 0$ – коэффициенты теплопроводности, $f(\bar{x}, t)$ – плотность мощности источников/стоков тепла, D – некоторая ограниченная прямоугольная область трехмерного пространства. Уравнение (1) замыкается добавлением некоторых начальных и граничных условий. В качестве начального возьмем однородное распределение температуры в объеме

$$u(\bar{x}, 0) = u_0 = \text{const} > 0. \quad (2)$$

Граничные условия 3-го рода наиболее естественны для задач теплопроводности:

$$(k(\bar{x}, t) \nabla u, \vec{\sigma}) = -\eta(u - u_0), \quad \bar{x} \in S. \quad (3)$$

Здесь $\vec{\sigma}$ – вектор внешней нормали к S , η – коэффициент теплопередачи через поверхность S .

В дальнейшем D – трехмерный единичный куб, интервал по времени ограничен значением $T > 0$. В этих предположениях граничные условия (3) можно переписать в виде

$$k_{\alpha} \frac{\partial u}{\partial x_{\alpha}} = +\eta(u - u_0), \quad x_{\alpha} = 0, \quad k_{\alpha} \frac{\partial u}{\partial x_{\alpha}} = -\eta(u - u_0), \quad x_{\alpha} = 1, \quad \alpha = 1, 2, 3. \quad (4)$$

2. Методы численного решения уравнения теплопроводности

Рассмотрим возможные конечно-разностные методы решения задачи (1), (2), (4). Наша цель – предложить экономичный прямой метод или группу методов для решения задачи на МВС с распределенной памятью.

Представим уравнение (1) в следующем виде

$$\frac{\partial u}{\partial t} = Lu + f, \quad Lu \equiv \sum_{\alpha=1}^3 L_{\alpha} u, \quad L_{\alpha} u \equiv \frac{\partial}{\partial x_{\alpha}} \left(k_{\alpha} \frac{\partial u}{\partial x_{\alpha}} \right). \quad (5)$$

Введем сетку $\Omega = \bar{\omega}_{x_1} \times \bar{\omega}_{x_2} \times \bar{\omega}_{x_3} \times \bar{\omega}_t$, в которой $\bar{\omega}_{x_{\alpha}} = \{x_{\alpha, i_{\alpha}} = i_{\alpha} h_{\alpha}, i_{\alpha} = 0, 1, \dots, N_{\alpha}, h_{\alpha} = 1/N_{\alpha}\}$ ($\alpha = 1, 2, 3$) и $\bar{\omega}_t = \{t = t_j = j\tau, j = 0, 1, \dots, N_t, \tau = t_{\max}/N_t\}$. На этой сетке определим разностный аналог искомой функции u_h . Для нахождения функции u_h с помощью интегро-интерполяционного подхода построим семейство разностных схем, называемое схемой с весами:

$$\frac{\hat{y}_h - y_h}{\tau} = \sigma \left[\hat{\Lambda}_h \hat{y}_h + \hat{\phi}_h \right] + (1 - \sigma) \left[\Lambda_h y_h + \phi_h \right]. \quad (6)$$

Здесь использованы общепринятые обозначения. Аппроксимация Λ_h оператора L имеет вид

$$\Lambda_h y_h = \sum_{\alpha=1}^3 \Lambda_{h, \alpha} y_h, \quad \Lambda_{h, \alpha} y_h = \left(\bar{k}_{\alpha} y_{h, x_{\alpha}} \right)_{\bar{x}_{\alpha}}.$$

В семействе схем (6) выделяют обычно явную ($\sigma = 0$) и неявную ($\sigma = 1$) разностные схемы, имеющие первый порядок точности по времени и второй – по пространству, а также симметричную схему Кранка-Николсон ($\sigma = 0.5$), имеющую второй порядок точности и по времени, и по пространству.

Явная схема ($\sigma = 0$) наиболее проста в реализации и экономична. Это означает, что при использовании явной схемы число арифметических действий Q необходимое для нахождения решения на очередном временном слое пропорционально количеству узлов пространственной сетки $N = \prod_{\alpha=1}^3 (N_{\alpha} + 1)$. Явная схема не является абсолютно устойчивой. При ее использовании необходимо

соблюдать условие

$$\tau < 0.5 \bar{h}^2 / (3k_0), \quad \bar{h} = \min_{\alpha=1,2,3} h_{\alpha}, \quad k_0 = \max_{x \in D, t \in [0, T], \alpha=1,2,3} k_{\alpha}(\bar{x}, t). \quad (7)$$

Неявная схема ($\sigma = 1$) является более трудоемкой в реализации и в общем случае не экономична. Однако она абсолютно устойчива, то есть вычисления по ней будут корректными при любых $\tau > 0$ и

$h_\alpha > 0$. Однако есть условие асимптотической устойчивости $\tau < 0.5\hbar / \sqrt{k_0}$. При нахождении решения на очередном временном слое в зависимости от выбранного метода (прямого или итерационного) и вида функции $k_\alpha(\bar{x}, t)$ может потребоваться от $O(N \log_2 N)$ до $O(N^{7/3})$ арифметических действий. Отметим также, что при выполнении условия (7) точность решений, полученных по явной и неявной схемам с одинаковым шагом по времени, практически совпадает.

Прямые методы: одна (иногда очень сложная) процедура дает решение с машинной точностью.

Варианты: метод Гаусса, ленточный метод Гаусса, метод Фурье (простое и быстрое преобразования Фурье), комбинированный метод: ленточный метод Гаусса по одной (прогонка) или двум координатам + метод Фурье по двум или одной координатам.

Итерационные методы: одна простая процедура применяется многократно и дает решение с заданной точностью. **Варианты:** двухслойные и многослойные, диагональные, треугольные, факторизованные (см. Лекцию 12). Итерационные методы редко используются для решения линейной задачи.

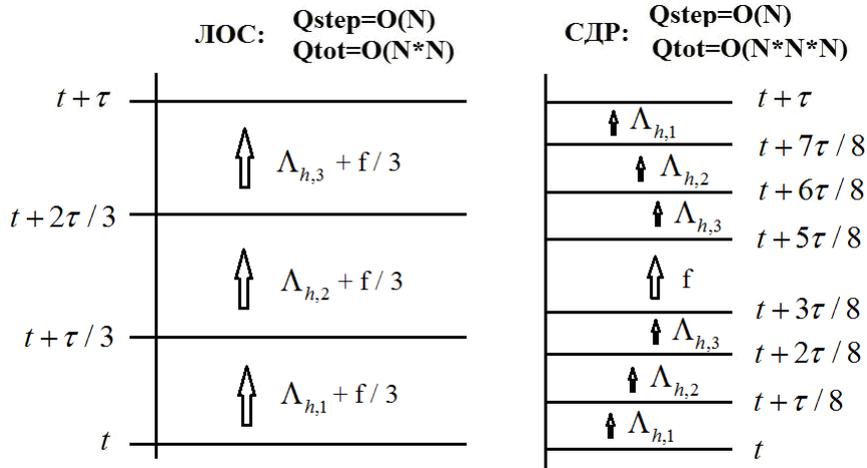
Симметричная схема ($\sigma = 0.5$) по трудоемкости реализации эквивалентна неявной схеме, а по условию устойчивости аналогична явной схеме: $\tau < \hbar^2 / (3k_0)$. Ее преимущество – второй порядок точности по времени.

На практике используются либо явная схема, как наиболее простая в реализации, либо **экономичные неявные схемы**, основанные на расщеплении пространственного оператора по координатам x_α . Например, **локально-одномерные схемы** (ЛОС), предложенные А.А. Самарским. Особенность ЛОС – свойство суммарной аппроксимации уравнения исходной задачи и абсолютная устойчивость. Вариант **неявной ЛОС**:

$$\frac{y_h^{j+\alpha/3} - y_h^{j+(\alpha-1)/3}}{\tau} = \Lambda_{h,\alpha}^{j+\alpha/3} y_h^{j+\alpha/3} + \frac{1}{3} f_h^{j+1}, \quad \alpha = 1, 2, 3. \quad (8)$$

Здесь введены неполные индексные обозначения и промежуточные слои по времени $t_{j+\alpha/3} = t_j + \alpha\tau/3$.

Схема (8) имеет первый порядок точности по времени и второй по пространству. Она экономична и может быть реализована с помощью алгоритма прогонки на каждом промежуточном этапе по времени.



Если нужна повышенная точность по времени и экономичность, то используют **неявные схемы двуциклического расщепления на основе схемы Кранка-Николсон**, которые также являются схемами суммарной аппроксимации, но имеют второй порядок точности по времени. Вариант такой схемы:

$$\frac{y_h^{j+\alpha/8} - y_h^{j+(\alpha-1)/8}}{\tau} = \frac{1}{4} \Lambda_{h,\alpha}^{j+1/2} \left(y_h^{j+\alpha/8} + y_h^{j+(\alpha-1)/8} \right), \quad \alpha = 1, 2, 3,$$

$$\frac{y_h^{j+5/8} - y_h^{j+3/8}}{\tau} = f_h^{j+1/2}, \quad (9)$$

$$\frac{y_h^{j+(9-\alpha)/8} - y_h^{j+(8-\alpha)/8}}{\tau} = \frac{1}{4} \Lambda_{h,\alpha}^{j+1/2} \left(y_h^{j+(9-\alpha)/8} + y_h^{j+(8-\alpha)/8} \right), \quad \alpha = 3, 2, 1.$$

Здесь $t_{j+\beta/8} = t_j + \beta\tau/8$, $\beta = 1, \dots, 8$. Схема (9) абсолютно устойчива в норме L_2 (но не в норме C), экономична и реализуется методом прогонки.

Итог: имеем три схемы решения задачи на МВС: явная (6), неявная ЛОС (8) и схема двуциклического расщепления (9). Выбор обусловлен требованием экономичности вычислений, которое является основным при решении многомерных задач. Для неявной схемы (6) в случае переменных коэффициентов экономичный прямой или итерационный метод пока не найден.

Замечания:

- 1) Для расчета стационарных решений предпочтительна неявная ЛОС, в которой шаг по времени ограничен только условием асимптотической устойчивости ($\tau \sim \hbar$).
- 2) Для расчета псевдопериодических или существенно нестационарных решений подходит схема (9), как более точная и позволяющая также брать шаг $\tau \sim \hbar$. При этом условии схема (9) имеет точность $O(\hbar^2)$, а неявная ЛОС – $O(\hbar)$.
- 3) Явная схема наиболее проста в параллельной реализации и является эталоном эффективности.

3. Параллельные алгоритмы решения уравнения теплопроводности

Для реализации выбранных разностных схем на МВС необходимо применить один из подходов распараллеливания соответствующего “скалярного” (однопроцессорного) численного алгоритма. Наиболее эффективным для решения многомерных задач механики сплошной среды на МВС с распределенной памятью считается **принцип геометрического параллелизма**, который предполагает проведение декомпозиции расчетной области на равные (по числу узлов сетки) подобласти соответственно числу процессоров (вычислителей).

В нашем случае, учитывая геометрические свойства (топологию) исходной расчетной области (а именно, ее прямоугольность), можно применить декомпозицию трех типов: **линейную, квадратичную и кубическую**. Первая предполагает разбиение области по одной из трех координат, вторая – по двум координатам, третья – по трем (примеры различных типов разбиения показаны на рис. 1). В соответствии с типами декомпозиции возникают три различных алгоритма распараллеливания, а также три типа топологии межпроцессорных связей, которые условно называют “линейка”, “квадратная решетка” и “кубическая решетка” соответственно типу декомпозиции.

Параллельная реализация явной схемы (6). Рассмотрим сначала расчетные формулы в последовательном алгоритме реализации явной схемы:

$$y_{h,i_1,i_2,i_3}^{j+1} = y_{h,i_1,i_2,i_3}^j + \tau \left(\Lambda_{h,i_1,i_2,i_3}^j y_{h,i_1,i_2,i_3}^j + f_{h,i_1,i_2,i_3}^j \right), \quad (i_1, i_2, i_3) \in I. \quad (10)$$

Здесь $I = \{(i_1, i_2, i_3), i_1 = 0, \dots, N_1, i_2 = 0, \dots, N_2, i_3 = 0, \dots, N_3\} \equiv I_1 \times I_2 \times I_3$ – индексное множество узлов пространственной сетки, которое предлагается разбивать на подмножества (домены), обрабатываемые отдельными вычислителями.

При линейном разбиении области по одной из координат (например, по x_2 , см. рис. 1а) k -ый процессор вычисляет решение на верхнем слое по времени в диапазоне узлов:

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = 0, \dots, N_1, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = 0, \dots, N_3\} \equiv I_1 \times I_2^{(k)} \times I_3.$$

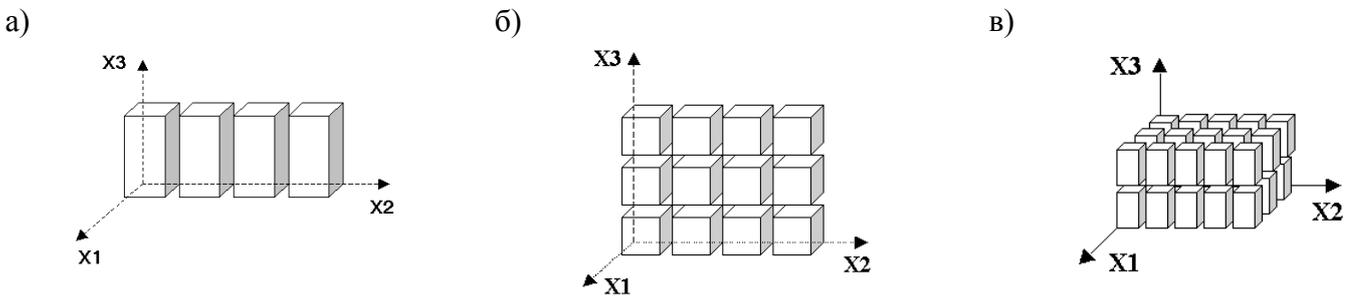


Рис. 1. Разбиение расчетной области по одной (а), двум (б) и трем координатам (в).

Для получения равномерной загрузки процессоров количества точек $N^{(k)} = (N_1 + 1)(i_{21}^{(k)} - i_{20}^{(k)} + 1)(N_3 + 1)$ в диапазонах $I^{(k)}$ должны быть примерно одинаковы. Расчетные формулы для вычисления решения на верхнем слое:

$$y_{h,i_1,i_2,i_3}^{j+1} = y_{h,i_1,i_2,i_3}^j + \tau \left(\Lambda_{h,i_1,i_2,i_3}^j y_{h,i_1,i_2,i_3}^j + f_{h,i_1,i_2,i_3}^j \right), \quad (i_1, i_2, i_3) \in I^{(k)}. \quad (11)$$

В случае “**линейного**” разбиения области в формуле (11) для процессора k участвуют значения функции y_h в точках $(i_1, i_{20}^{(k)} - 1, i_3) \equiv (i_1, i_{21}^{(k-1)}, i_3)$ и $(i_1, i_{21}^{(k)} + 1, i_3) \equiv (i_1, i_{20}^{(k+1)}, i_3)$, лежащие в области данных

процессоров $(k-1)$ и $(k+1)$. Перед началом очередного шага вычислений по формулам (11) на процессоре k необходимо получить требующиеся значения y_h от соседних процессоров. Для этого лучше всего произвести один групповой обмен данными для всех индексов i_1, i_3 . Обмен легко организуется в рамках линейной топологии. Количество передаваемых данных одним процессором равно $M^{(k)} = 4(N_1 + 1)(N_3 + 1)$ (обмен слева и справа в обе стороны) и не зависит от числа процессоров. В случае равного количества узлов сетки по различным направлениям ($N_1 = N_2 = N_3 = \sqrt[3]{N}$) величина $M^{(k)} = 4N^{2/3}$. Оценка ускорения:

$$Q_1 = C_1 N, \quad Q_p = C_1 \frac{N}{p} + C_0(p)M, \quad S_p = \frac{Q_1}{Q_p} = \frac{p}{1 + pC_0(p)M / C_1 N} = \frac{p}{1 + 4pC_0(p)N^{-1/3} / C_1}.$$

В случае "**квадратичного**" разбиения области, пример которого показан на рис. 1б, формулы (11) остаются в силе, но изменяются диапазон индексов $I^{(k)}$ и количество точек $N^{(k)}$, обрабатываемых процессором k :

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = 0, \dots, N_1, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = i_{30}^{(k)}, \dots, i_{31}^{(k)}\} \equiv I_1 \times I_2^{(k)} \times I_3^{(k)},$$

$$N^{(k)} = (N_1 + 1)(i_{21}^{(k)} - i_{20}^{(k)} + 1)(i_{31}^{(k)} - i_{30}^{(k)} + 1).$$

Соответственно изменяется и топология обменов и межпроцессорных связей, которые образуют теперь решетку. Для определения соседей процессора k , с которыми ему придется обмениваться данными, можно ввести двойную нумерацию процессоров и установить взаимно однозначное соответствие исходных и двойных процессорных номеров. В данном случае можно использовать соотношения:

$$k = p_3 k_2 + k_3, \quad k_2 = 0, \dots, p_2 - 1, \quad k_3 = 0, \dots, p_3 - 1, \quad k_2 = \lfloor k / p_3 \rfloor, \quad k_3 = k \bmod p_3, \quad k = 0, \dots, p - 1, \quad p = p_2 p_3.$$

Здесь p – общее число процессоров, образующих решетку $p_2 \times p_3$. В этих обозначениях процессор $k \leftrightarrow (k_2, k_3)$ перед началом очередного шага вычислений будет обмениваться недостающими данными с четырьмя соседями $k^{left} \leftrightarrow (k_2 - 1, k_3)$, $k^{right} \leftrightarrow (k_2 + 1, k_3)$, $k^{down} \leftrightarrow (k_2, k_3 - 1)$, $k^{up} \leftrightarrow (k_2, k_3 + 1)$ в рамках топологии "решетка". Количество передаваемых данных $M^{(k)} = 4(N_1 + 1)((N_2 + 1) / p_2 + (N_3 + 1) / p_3)$ теперь зависит от числа процессоров. При равном числе узлов сетки по различным направлениям и квадратной решетке процессоров ($p_2 = p_3 = \sqrt{p}$) величина $M^{(k)} = 8N^{2/3} p^{-1/2}$. Оценка ускорения:

$$S_p = \frac{p}{1 + pC_0(p)M / C_1 N} = \frac{p}{1 + 8p^{1/2}C_0(p)N^{-1/3} / C_1}.$$

В случае "**кубического**" разбиения области (см. рис. 1в) получим расчетные формулы (11), в которых

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = i_{10}^{(k)}, \dots, i_{11}^{(k)}, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = i_{30}^{(k)}, \dots, i_{31}^{(k)}\} \equiv I_1^{(k)} \times I_2^{(k)} \times I_3^{(k)},$$

$$N^{(k)} = (i_{11}^{(k)} - i_{10}^{(k)} + 1)(i_{21}^{(k)} - i_{20}^{(k)} + 1)(i_{31}^{(k)} - i_{30}^{(k)} + 1).$$

Для определения соседей и обмена с ними данными вводится новая тройная нумерация процессоров:

$$k = p_2 p_3 k_1 + p_3 k_2 + k_3, \quad k_1 = 0, \dots, p_1 - 1, \quad k_2 = 0, \dots, p_2 - 1, \quad k_3 = 0, \dots, p_3 - 1,$$

$$k_1 = \lfloor k / (p_2 p_3) \rfloor, \quad k_2 = \lfloor (k - p_2 p_3 k_1) / p_3 \rfloor, \quad k_3 = (k - p_2 p_3 k_1) \bmod p_3, \quad k = 0, \dots, p - 1, \quad p = p_1 p_2 p_3.$$

В соответствии с ней каждый процессор в кубической решетке имеет в общем случае 6 соседей, с которыми и обменивается необходимыми данными в начале каждого шага вычислений. Количество передаваемых данных равно

$$M^{(k)} = 4(N_1 + 1)(N_2 + 1) / (p_1 p_2) + 4(N_1 + 1)(N_3 + 1) / (p_1 p_3) + 4(N_2 + 1)(N_3 + 1) / (p_2 p_3).$$

При $N_\alpha = \sqrt[3]{N}$ и $p_\alpha = \sqrt[3]{p}$ величина $M^{(k)} = 12N^{2/3} p^{-2/3}$, ускорение

$$S_p = \frac{p}{1 + pC_0(p)M / C_1 N} = \frac{p}{1 + 12p^{1/3}C_0(p)N^{-1/3} / C_1}.$$

Теоретическая эффективность рассмотренных трех алгоритмов равна 100%. Реальная эффективность алгоритма зависит от схемы обменов и числа процессоров. При прочих равных условиях использование линейной топологии при большом числе процессоров будет менее эффективно, чем использование топологий типа "решетка", поскольку у последних размер передаваемых данных снижается с ростом числа процессоров.

Параллельная реализация неявной ЛОС (8). Для этого тоже можно использовать три типа разбиения расчетной области и три типа топологии межпроцессорных связей. При **линейном разбиении** области, например, по координате x_2 , получается следующий параллельный алгоритм. Для вычисления решения на новом временном слое каждый процессор последовательно решает три задачи:

$$\left(E - \tau \Lambda_{h,\alpha}^{j+\alpha/3}\right) y_h^{j+\alpha/3} = y_h^{j+(\alpha-1)/3} + \frac{\tau}{3} f_h^{j+1}, \quad \alpha = 1, 2, 3, \quad (12)$$

в своем диапазоне индексов $I^{(k)}$. При этом первый и третий этапы ЛОС можно реализовать независимо от данных на других процессорах и использовать для этого алгоритм скалярной прогонки. На втором этапе ЛОС данные оказываются распределенными между процессорами. Поэтому необходимо использовать параллельный алгоритм, например, алгоритм параллельной прогонки. Он применяется по индексу i_2 для всех фиксированных значений i_1 и i_3 . При этом возникают обмены данными, связанные с формированием массивов коэффициентов "коротких задач" (количество которых равно $N_1 N_3$), а также при передаче значений решения этих задач на все процессоры. Обмены производятся либо в рамках топологии "звезда", либо в рамках топологии "клика". Линейная топология при такой организации обменов фактически не используется, однако она лежит в основе разбиения области. При использовании стандарта коммуникаций MPI лучше всего использовать функцию "коллективное суммирование" (MPI_Reduce или MPI_Allreduce) для сбора информации со всех процессоров. Максимальный объем передаваемых данных с одного процессора $M^{(k)} = O((N_1 + 1)(N_3 + 1)p)$ растет с увеличением p . При равном числе узлов сетки по различным направлениям $M^{(k)} = O(N^{2/3} p)$. Более точно: $M^{(k)} = 8pl$, $l = 1 \div N_b$, $N_b = (N_1 + 1)(N_3 + 1)$. Здесь l – число задач, решаемых одновременно.

В случае **"квадратичного"** разбиения области, например, по координатам x_2, x_3 , первый этап схемы реализуется с помощью скалярной прогонки, второй и третий – с помощью параллельной. Обмены данными между процессорами на каждом параллельном этапе организуются описанным выше способом. Максимальный объем передаваемых данных (максимум по двум параллельным этапам) равен $M^{(k)} = O\left((N_1 + 1) \times \max\left[(N_2 + 1)\frac{p_3}{p_2}, (N_3 + 1)\frac{p_2}{p_3}\right]\right)$ или $M^{(k)} = O(N^{2/3})$ – при равном количестве узлов сетки по различным направлениям и квадратной решетке процессоров.

В случае **"кубического"** разбиения области все этапы схемы реализуются с помощью параллельного алгоритма прогонки. Обмены данных производятся с помощью коллективных процедур на каждом этапе. Максимальный объем передаваемых данных с одного процессора равен

$$M^{(k)} = O\left(\max\left[(N_1 + 1)(N_2 + 1)\frac{p_3}{p_1 p_2}, (N_1 + 1)(N_3 + 1)\frac{p_2}{p_1 p_3}, (N_2 + 1)(N_3 + 1)\frac{p_1}{p_2 p_3}\right]\right).$$

При равном количестве узлов сетки и количестве процессоров по различным направлениям $M^{(k)} = O(N^{2/3} p^{-1/3})$ и убывает с увеличением p .

Эффективность алгоритмов. В скалярном варианте расчет по неявной ЛОС на каждом временном шаге требует $Q_1 = 3(C_1 + C_2)N$ арифм. операций. Здесь C_1 – количество операций при вычислении коэффициентов уравнений в одном узле пространственной сетки, C_2 – количество операций при вычислениях по алгоритму прогонки на каждом этапе схемы, приходящееся на один узел сетки. На практике $C_1 \gg C_2$. При параллельной реализации на системе с p процессорами с учетом коллективных обменов между процессами получаем оценки для $Q_p^{(\alpha)}$ ($\alpha = 1, 2, 3$) рассмотренных выше алгоритмов:

$$Q_p^{(1)} = (3C_1 + 2C_2)\frac{N}{p} + N_1 N_3 \left(3C_2 \frac{N_2}{p} + 2C_2 p\right) + C_0(p)M = \frac{N}{p} \left(3C_1 + 5C_2 + (2C_2 + 8C_0(p))\frac{p^2}{N_2}\right),$$

$$Q_p^{(2)} = \frac{N}{p} \left(3C_1 + 7C_2 + (2C_2 + 8C_0(p))\left(\frac{p_2^2}{N_2} + \frac{p_3^2}{N_3}\right)\right),$$

$$Q_p^{(3)} = \frac{N}{p} \left(3C_1 + 9C_2 + (2C_2 + 8C_0(p)) \left(\frac{p_1^2}{N_1} + \frac{p_2^2}{N_2} + \frac{p_3^2}{N_3} \right) \right).$$

Сравнивая теперь эти величины с Q_1 , в предположениях $C_1 \sim C_2$ (наихудшая ситуация) и $p_\beta^2 \ll \sqrt[3]{N_\beta}$ ($\beta=1,2,3$) получаем оценки для теоретического ускорения $S_p^{(\alpha)} = Q_1 / Q_p^{(\alpha)}$ и теоретической эффективности $E_p^{(\alpha)} = (S_p^{(\alpha)} / p) \cdot 100\%$ приближенно равные $\frac{6p}{8}, \frac{6p}{10}, \frac{6p}{12}$ и 75, 60, 50 % для $\alpha=1,2,3$.

Параллельная реализация схемы двуциклического расщепления (9). В ее основе лежат те же принципы, что и в алгоритме для неявной ЛОС. Только теперь в скалярном и в параллельном случаях для поиска решения на новом временном слое мы решаем семь подзадач (семь этапов схемы (9)), шесть из которых решаются методом прогонки, а седьмой аналогичен этапу явной схемы:

$$\begin{aligned} \left(E - \frac{\tau}{4} \Lambda_{h,\alpha}^{j+1/2} \right) y_h^{j+\alpha/8} &= 2y_h^{j+(\alpha-1)/8}, \quad y_h^{j+\alpha/8} = y_h^{j+\alpha/8} - y_h^{j+(\alpha-1)/8}, \quad \alpha=1,2,3, \\ y_h^{j+5/8} &= y_h^{j+3/8} + \tau f_h^{j+1/2}, \\ \left(E - \frac{\tau}{4} \Lambda_{h,\alpha}^{j+1/2} \right) y_h^{j+\beta/8} &= 2y_h^{j+(\beta-1)/8}, \quad y_h^{j+\beta/8} = y_h^{j+\beta/8} - y_h^{j+(\beta-1)/8}, \quad \beta=9-\alpha, \alpha=3,2,1. \end{aligned} \quad (13)$$

В параллельной реализации формулы (13) относятся к соответствующему индексному пространству $I^{(k)}$. Обмены данными организуются также как и в схеме (8). Для данной схемы $Q_1 = (4C_1 + 6C_2)N$. В параллельных реализациях:

$$\begin{aligned} Q_p^{(1)} &= (4C_1 + 4C_2) \frac{N}{p} + 2C_2 N_1 N_3 \left(3 \frac{N_2}{p} + 2p \right) + C_0(p)M = \frac{N}{p} \left(4C_1 + 10C_2 + (4C_2 + 8C_0(p)) \frac{p^2}{N_2} \right), \\ Q_p^{(2)} &= \frac{N}{p} \left(4C_1 + 14C_2 + (4C_2 + 8C_0(p)) \left(\frac{p_2^2}{N_2} + \frac{p_3^2}{N_3} \right) \right), \\ Q_p^{(3)} &= \frac{N}{p} \left(4C_1 + 18C_2 + (4C_2 + 8C_0(p)) \left(\frac{p_1^2}{N_1} + \frac{p_2^2}{N_2} + \frac{p_3^2}{N_3} \right) \right). \end{aligned}$$

Сравнивая эти величины с Q_1 , в предположениях $C_1 \sim C_2$ и $p_\beta^2 \ll \sqrt[3]{N_\beta}$ ($\beta=1,2,3$) получаем значения теоретических ускорения и эффективности приближенно равные $\frac{8p}{14}, \frac{8p}{18}, \frac{8p}{22}$ и 57,44,36 % для $\alpha=1,2,3$.

4. Результаты тестовых расчетов

Расчеты проводились по явной схеме (6) и неявной ЛОС (8) на сетках $100^3, 200^3, 300^3$. При этом использовались все три типа разбиения расчетной области. Расчеты проводились на однородной вычислительной системе МВС-1000М, имеющей 768 процессоров.

Расчеты по **явной схеме** выявили высокую эффективность распараллеливания и следующие тенденции.

1) Реальная эффективность тем выше, чем больше узлов расчетной сетки попадает на процессор, так как для подробной сетки загрузка процессоров получается более равномерной.

2) Эффективность при линейном разбиении ниже, чем на квадратной или кубической решетке, что определяется уменьшением накладных расходов при пересылках с ростом числа процессоров на конфигурациях типа “решетка”.

3) На сетке среднего объема преимущества кубической решетки по сравнению с квадратной проявляются при умеренном числе процессоров, так как количество обменов для кубической решетки больше, чем для квадратной, и при большом числе процессоров характеристики МВС заметно ухудшаются. На сетках очень большого объема ситуация меняется в пользу кубической решетки в силу уменьшения размера пересылаемых данных с ростом числа процессоров, которое происходит для этой топологии существенно быстрее.

4) Применение линейной топологии сильно ограничено числом точек по одному из направлений (на практике, число процессоров должно быть $\leq N_\alpha / 2$). Более гибкими оказываются “квадратная” и “кубическая” решетки. Однако при их использовании величины N_α / p_α должны быть примерно

одинаковы. На сетке с одинаковым числом узлов по всем направлениям это требование интерпретируется как уменьшение коэффициента “неквдратности” решетки.

5) При проведении массовых расчетов на сетке фиксированного размера следует определить оптимальные топологию и число процессоров, при которых задача решается максимально быстро и эффективно. Например, для сеток 2 (3) оптимальной конфигурацией оказалась квадратная решетка процессоров $14 \times 14 = 196$ ($19 \times 19 = 361$), дающая ускорение 167.6 (327.2) и эффективность 85.5 (90.6) %.

Расчеты по *неявной ЛОС* позволили сделать следующие выводы.

1) Эффективность расчетов с ростом числа процессоров здесь падает быстрее, чем в случае явной схемы. Это связано с тем, что накладные расходы (отношение суммарного времени обменов данными к общему времени расчета в %), связанные с обменами данных, здесь растут быстрее и могут быть близки к 100% при большом числе процессоров.

2) Ввиду более сложной и объемной процедуры обменов, преимущества кубической решетки по сравнению с “линейкой” и “квадратной” решеткой процессоров проявляются сразу во всем диапазоне процессорных конфигураций (от 1 до 500).

3) При малом и умеренном числе процессоров (1-100) очень сильно проявляется фактор “неквдратности” решетки, который следует учитывать в расчетах.

4) По сравнению с явной схемой эффективность расчетов по неявной ЛОС оказывается в 2-3 раза ниже. Учитывая это, а также различие расчетных формул, получаем, что при одинаковых: размере пространственной сетки, шаге по времени и числе процессоров неявный алгоритм работает в 1.5-3 раза медленнее. Однако если рассчитывается стационарное решение, то шаг в неявной схеме можно взять в N_α раз больший, что в десятки раз окупает ее более низкую эффективность распараллеливания.

При расчетах существенно нестационарных решений следует использовать *схему двуциклического расщепления* (9), где также можно увеличить шаг по времени на несколько порядков при той же точности вычислений, что и в явной схеме.

5. Выводы

1) При расчетах на МВС стационарных и медленно меняющихся со временем нестационарных решений краевых задач для параболических уравнений на грубых сетках следует использовать явные схемы. В этом случае условие устойчивости схемы не является обременительным, а эффективность ее параллельной реализации максимальна.

2) При расчетах на МВС стационарных и слабо нестационарных решений на подробной сетке достаточно использовать неявные ЛОС первого порядка точности по времени. При этом необходимо следить, чтобы выполнялось лишь условие их асимптотической устойчивости.

3) При расчетах на МВС существенно нестационарных решений следует использовать неявные схемы двуциклического расщепления или аналогичные им, имеющие второй порядок точности по времени. При этом шаг по времени должен определяться необходимой точностью решения и удовлетворять условию асимптотической устойчивости.

4) Выбор конкретного разбиения области, топологии и способа межпроцессорных обменов определяется соотношением размеров расчетной и “процессорной” сеток. При этом для фиксированной расчетной сетки всегда существуют оптимальные топология обменов и количество процессоров, при которых достигается скорейшее решение задачи в целом.