

Е. Н. Акимова, Д. В. Белоусов

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ РЕШЕНИЯ СЛАУ С БЛОЧНО-ТРЕХДИАГОНАЛЬНЫМИ МАТРИЦАМИ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЯХ

Для решения СЛАУ с блочно-трехдиагональными матрицами предложены и численно реализованы на многопроцессорных вычислителях различного типа параллельный алгоритм матричной прогонки и параллельный метод сопряженных градиентов с предобуславливателем. Проведено сравнение времени счета параллельных алгоритмов на видеоускорителе NVIDIA GeForce, 4-ядерном процессоре Intel Core i5-750 и многопроцессорном вычислительном комплексе МВС-1000 при решении задачи о нахождении распределения потенциала на поверхности земли в проводящей среде. *Параллельные алгоритмы; эффективность и ускорение; суперкомпьютер; видеоускоритель*

ВВЕДЕНИЕ

Системы линейных алгебраических уравнений (СЛАУ) с блочно-трехдиагональными матрицами возникают при решении ряда задач математического моделирования, в частности, при решении задачи диффузии и задач электроразведки.

Важной задачей при моделировании структурных превращений в многокомпонентных сплавах является решение задачи диффузионного массопереноса, когда в каждый момент времени необходимо знать распределение концентраций диффундирующих компонентов. Диффузионный массоперенос описывается системой параболических дифференциальных уравнений в частных производных вида

$$\frac{\partial C_i}{\partial \tau} = \frac{1}{r^q} \frac{\partial}{\partial r} \left(r^q \sum_{j=1}^{N-1} \tilde{D}_{ij} \frac{\partial C_j}{\partial r} \right), \quad (1)$$

где N – число компонентов сплава; r – пространственная координата; C_j – концентрация j -го компонента; \tilde{D}_{ij} – парциальные коэффициенты взаимной диффузии; q – показатель степени, зависящий от симметрии задачи: 0 – для плоской, 1 – для цилиндрической, 2 – для сферической.

При использовании абсолютно устойчивой неявной разностной схемы система дифференциальных уравнений (1) сводится к системе уравнений с блочно-трехдиагональной матри-

цей [1–2]. Для i -го компонента в k -м узле пространственной сетки система уравнений имеет вид

$$\sum_{j=1}^{N-1} [a_{ik}^j C_j(k-1) + c_{ik}^j C_j(k) + b_{ik}^j C_j(k+1)] = d_k^i, \quad (2)$$

где $a_{ik}^j, b_{ik}^j, c_{ik}^j, d_k^i$ – коэффициенты, рассчитываемые в зависимости от модели, $C_j(k)$ – концентрация j -го компонента в k -м узле сетки.

Важнейшими задачами исследования неоднородности земной коры являются задачи электроразведки. Одним из известных методов электроразведки является метод вертикального электрического зондирования (ВЭЗ). На поверхности земли собирают электроразведочную установку, которая состоит из двух питающих и двух приемных электродов. К питающим электродам A и B подключают источник тока, а на приемных электродах M и N измеряют напряженность электрического поля. По результатам выполненных измерений вычисляют кажущееся электрическое сопротивление $\rho_k = K \Delta V_{MN} / I_{AB}$ для неоднородной среды, где K – геометрический коэффициент, зависящий от расстояний между электродами A, B, M и N , ΔV_{MN} – разность потенциалов на приемных электродах M и N , I_{AB} – сила тока, протекающего в питающей линии.

Для модели среды с плоско-параллельными горизонтальными границами, когда слои земной коры залегают горизонтально и сопротивление среды зависит только от глубины $\rho = \rho(z)$, кажущееся сопротивление $\rho = \rho(r)$, где $r = AO$ – расстояние от центра приемной цепи до питающего электрода. Задача интерпретации результатов ВЭЗ заключается в определении $\rho(z)$, дающего «электрический разрез» среды по из-

Контактная информация: (343)375-34-46

Статья рекомендована к публикации программным комитетом Международной научной конференции "Параллельные вычислительные технологии 2011".

Работа выполнена в рамках Программы фундаментальных исследований Президиума РАН № 14 при поддержке УрО РАН, проект 09-П-1-1003.

вестным значениям $\rho(r)$ (ось z направлена вниз, ось r направлена вправо).

Задача ВЭЗ о нахождении потенциала $V_0(r, z=0)$ на поверхности земли сводится к решению двумерного уравнения Лапласа в цилиндрической системе координат [3]

$$\Delta V \equiv \frac{\partial^2 V}{\partial r^2} + \frac{1}{r} \frac{\partial V}{\partial r} + \frac{\partial^2 V}{\partial z^2} = 0 \quad (3)$$

с граничными условиями непрерывности потенциала и непрерывности нормальной составляющей к границам плотности тока:

$$V_0 \Big|_{z=l} = V_1 \Big|_{z=l}, \quad \frac{1}{\rho_0} \frac{\partial V_0}{\partial z} \Big|_{z=l} = \frac{1}{\rho_1} \frac{\partial V_1}{\partial z} \Big|_{z=l} -$$

условия на горизонтальных прямых $z=l$, $\frac{\partial V_0}{\partial z} \Big|_{z=0} = 0$ – сверху, $\frac{\partial V_0}{\partial r} \Big|_{r=0} = 0$ – слева, $V_0 = 0$ – справа и снизу.

Здесь

$$\rho = \frac{2\pi r^2}{I} \cdot \left| \frac{\partial V}{\partial r} \right|. \quad (4)$$

После использования конечно-разностной аппроксимации краевая задача (3)–(4) сводится к решению СЛАУ с блочно-трехдиагональной матрицей (рис. 1).

Другой важной задачей электроразведки является задача бокового каротажного зондирования (БКЗ), предусматривающая использование приборов однотипных зондов разной длины при измерении потенциала электрического поля. В результате интерпретации данных каротажа получают значение удельного электрического сопротивления пласта, близкое к истинному.

В работе [4] показано, что после использования конечно-разностной аппроксимации задача БКЗ сводится к решению системы линейных алгебраических уравнений с блочно-трехдиагональной матрицей следующего вида:

$$AV \equiv -\frac{1}{r} \left(\bar{r} a V_{\bar{r}} \right)_r^{\wedge} - \left(b V_{\bar{z}} \right)_z^{\wedge} = F, \quad (5)$$

где A – матрица размерности $(N_r \times 2N_z) \times (N_r \times 2N_z)$, имеющая блоки размерности $N_r \times N_r$ (рис. 1); V и F – векторы размерности $N_r \times 2N_z$; N_r – число точек по r ; $2 \times N_z$ – число точек по z ; $\sigma = 1/\rho$ – коэффициент электропроводности;

$$a_{ij} = \sigma \left(r_i - \frac{h_i^{(r)}}{2}, z_j + \frac{h_j^{(z)}}{2} \right),$$

$$b_{ij} = \sigma \left(r_i + \frac{h_i^{(r)}}{2}, z_j - \frac{h_j^{(z)}}{2} \right),$$

$$(V)_{r,ij}^{\wedge} = (V_{ij} - V_{i-1,j}) / h_i^{(r)},$$

$$(V)_{r,ij}^{\wedge} = (V_{i+1,j} - V_{i,j}) / h_i^{(r)},$$

$$(V)_{z,ij}^{\wedge} = (V_{ij} - V_{i,j-1}) / h_i^{(z)},$$

$$(V)_{z,ij}^{\wedge} = (V_{i,j+1} - V_{i,j}) / h_i^{(z)},$$

$$h_i^{(r)} = (h_i^{(r)} + h_{i+1}^{(r)}) / 2, \quad h_i^{(r)} = r_i - r_{i-1}, \quad i = 1, \dots, N_r,$$

$$h_j^{(z)} = (h_j^{(z)} + h_{j+1}^{(z)}) / 2, \quad h_j^{(z)} = z_j - z_{j-1},$$

$$j = -N_z + 1, \dots, N_z.$$

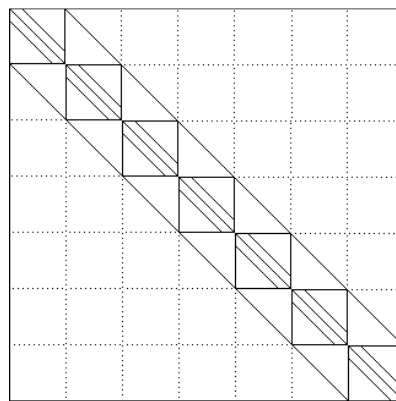


Рис. 1. Вид матрицы СЛАУ

Целью данной работы является разработка и реализация алгоритмов решения СЛАУ с блочно-трехдиагональными матрицами на многопроцессорных вычислителях различного типа: на видеоускорителе NVIDIA GeForce GTX 285, 4-ядерном процессоре Intel Core I5-750 и многопроцессорном вычислительном комплексе МВС-1000 и сравнение времени счета параллельных алгоритмов при решении модельной задачи.

1. ПАРАЛЛЕЛЬНЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ

Рассмотрим систему уравнений с блочно-трехдиагональными матрицами

$$\begin{cases} C_0 \bar{Y}_0 - B_0 \bar{Y}_1 = \bar{F}_0, & i=0, \\ -A_i \bar{Y}_{i-1} + C_i \bar{Y}_i - B_i \bar{Y}_{i+1} = \bar{F}_i, & i=1, \dots, N-1, \\ -A_N \bar{Y}_{N-1} + C_N \bar{Y}_N = \bar{F}_N, & i=N, \end{cases} \quad (6)$$

где \bar{Y}_i – искомые векторы размерности n , \bar{F}_i – заданные векторы размерности n , A_i, B_i, C_i – квадратные матрицы порядка n .

Для решения СЛАУ (6) предлагается использовать параллельный прямой метод матричной прогонки [5] и параллельный итерационный метод сопряженных градиентов с предобуславливателем в случае решения СЛАУ с симметричной положительно-определенной матрицей.

Ранее параллельный метод матричной прогонки, реализованный на многопроцессорном вычислительном комплексе МВС-1000, эффективно использовался при решении модельной задачи диффузии о насыщении плоской металлической пластины тремя компонентами с различной диффузионной подвижностью, а также при моделировании эволюции выделений в двухфазном многокомпонентном сплаве [6–7].

1.1. Прямой метод решения задачи

Опишем параллельный алгоритм матричной прогонки для решения СЛАУ с блочно-трехдиагональными матрицами, к которым сводятся разностные задачи для эллиптических уравнений второго порядка с переменными коэффициентами в области P . Будем предполагать, что область P представляет собой прямоугольник.

При построении параллельного алгоритма исходную область P разобьем на L подобластей (рис. 2) вертикальными линиями так, что $N = L \times M$.

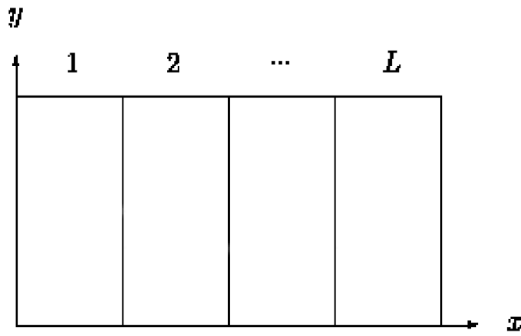


Рис. 2. Разбиение области на подобласти

В качестве параметрических неизвестных выберем векторы $\bar{Y}_K, K = 0, M, \dots, N$, связывающие неизвестные на сетке по вертикали. Относительно \bar{Y}_K строится редуцированная система уравнений. Для этого в подобластях, определяемых интервалами $(K, K + M)$, рассматриваются задачи:

$$\begin{cases} -A_i \bar{U}_{i-1}^1 + C_i \bar{U}_i^1 - B_i \bar{U}_{i+1}^1 = 0, \bar{U}_K^1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \bar{U}_{K+M}^1 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \\ \dots \\ -A_i \bar{U}_{i-1}^n + C_i \bar{U}_i^n - B_i \bar{U}_{i+1}^n = 0, \bar{U}_K^n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \bar{U}_{K+M}^n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}; \end{cases} \quad (7)$$

$$\begin{cases} -A_i \bar{V}_{i-1}^1 + C_i \bar{V}_i^1 - B_i \bar{V}_{i+1}^1 = 0, \bar{V}_K^1 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \bar{V}_{K+M}^1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \\ \dots \\ -A_i \bar{V}_{i-1}^n + C_i \bar{V}_i^n - B_i \bar{V}_{i+1}^n = 0, \bar{V}_K^n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \bar{V}_{K+M}^n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}; \end{cases} \quad (8)$$

$$-A_i \bar{W}_{i-1} + C_i \bar{W}_i - B_i \bar{W}_{i+1} = \bar{F}_i, \bar{W}_K = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \bar{W}_{K+M} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad (9)$$

где $i = K + 1, \dots, K + M - 1$.

Утверждение 1. Если $\bar{U}_i^1, \dots, \bar{U}_i^n$ – решения задач (7), $\bar{V}_i^1, \dots, \bar{V}_i^n$ – решения задач (8), \bar{W}_i – решения задачи (9), а \bar{Y}_i – решения исходной задачи (6) на $(K, K + M)$, тогда

$$\bar{Y}_i = (\bar{U}_i^1 \bar{U}_i^2 \dots \bar{U}_i^n) \bar{Y}_K + (\bar{V}_i^1 \bar{V}_i^2 \dots \bar{V}_i^n) \bar{Y}_{K+M} + \bar{W}_i. \quad (10)$$

После подстановки соотношений (10) в исходную систему (6) в точках $K = 0, M, \dots, N$, получим систему уравнений относительно параметров \bar{Y}_K . Эта система уравнений по своей структуре аналогична (6), имеет меньшую размерность и следующий вид:

$$\begin{cases} [C_0 - B_0 U_1] \bar{Y}_0 - [B_0 V_1] \bar{Y}_M = \bar{F}_0 + B_0 \bar{W}_1, K = 0; \\ -[A_K U_{K-1}] \bar{Y}_{K-M} + [C_K - A_K V_{K-1} - B_K U_{K+1}] \bar{Y}_K - \\ - [B_K V_{K+1}] \bar{Y}_{K+M} = \bar{F}_K + A_K \bar{W}_{K-1} + B_K \bar{W}_{K+1}, \\ K = M, 2M, \dots, N - M; \\ -[A_N U_{N-1}] \bar{Y}_{N-M} + [C_N - A_N V_{N-1}] \bar{Y}_N = \\ = \bar{F}_N + A_N \bar{W}_{N-1}, K = N, \end{cases} \quad (11)$$

где U_K и V_K – квадратные матрицы порядка n .

Задача (11) решается классическим алгоритмом матричной прогонки [8] на одном процессоре, задачи (7)–(9) решаются независимо в L подобластях на L процессорах.

Матрицы U_i, V_i и вектор \bar{W}_i на интервалах $(K, K + M)$ вычисляются независимо на L процессорах по следующим формулам.

а) Прямой ход прогонки:

$$\begin{aligned} \alpha_{K+1} &= C_{K+1}^{-1} B_{K+1}, \beta_{K+1} = C_{K+1}^{-1} A_{K+1}, \\ \bar{Y}_{K+1} &= C_{K+1}^{-1} \bar{F}_{K+1}, \alpha_i = [C_i - A_i \alpha_{i-1}]^{-1} B_i, \\ \beta_i &= [C_i - A_i \alpha_{i-1}]^{-1} A_i \beta_{i-1}, \\ \bar{y}_i &= [C_i - A_i \alpha_{i-1}]^{-1} [\bar{F}_i + A_i \bar{y}_{i-1}], \\ i &= K + 2, \dots, K + M - 1. \end{aligned} \quad (12)$$

б) Обратный ход прогонки:

$$\begin{aligned} U_{K+M-1} &= \beta_{K+M-1}, V_{K+M-1} = \alpha_{K+M-1}, \\ \bar{W}_{K+M-1} &= \bar{y}_{K+M-1}, U_i = \alpha_i U_{i+1} + \beta_i, V_i = \alpha_i V_{i+1}, \\ W_i &= \alpha_i \bar{W}_{i+1} + \bar{y}_i, i = K + M - 2, \dots, K + 1. \end{aligned} \quad (13)$$

После вычисления параметров Y_K остальные искомые неизвестные находятся по формуле

(10) также независимо на каждом из L интервалов на L процессорах.

Схема параллельного алгоритма имеет вид:

(12) \rightarrow (13) \rightarrow (11) \rightarrow (10).

Утверждение 2. Если для исходной системы (6) выполняются достаточные условия устойчивости метода матричной прогонки по А. А. Самарскому [8]

$$\begin{aligned} \|C_0^{-1}B_0\| \leq 1, \quad \|C_N^{-1}A_N\| \leq 1, \\ \|C_i^{-1}A_i\| + \|C_i^{-1}B_i\| \leq 1, \quad i=1, \dots, N-1, \end{aligned}$$

причем хотя бы одно из неравенств – строгое, то эти же условия достаточны и для устойчивости метода матричной прогонки при решении системы уравнений (11) относительно параметров \bar{Y}_k (см. [5]).

1.2. Итерационный метод решения задачи

Одним из быстрых итерационных методов решения СЛАУ с симметричной положительно-определенной матрицей является метод сопряженных градиентов (МСГ) [9].

Введение предобуславливания применяется с целью ускорения сходимости итерационного процесса и состоит в том, что исходная СЛАУ

$$Ax = b \quad (14)$$

заменяется на систему уравнений

$$C^{-1}Ax = C^{-1}b, \quad (15)$$

для которой итерационный метод (в нашем случае МСГ) сходится существенно быстрее.

Пусть $C = C^* > 0$. Предположим, что предобуславливатель представлен в виде $C = B^*B$, где B – невырожденная квадратная матрица. Умножим слева обе части системы (15) на B и положим $y = Bx$. В результате придем к эквивалентной системе уравнений

$$\tilde{A}y = \tilde{b}, \quad (16)$$

где $\tilde{A} = (B^*)^{-1}AB^{-1}$, $\tilde{b} = (B^*)^{-1}b$.

Условием выбора предобуславливателя C является следующее:

$$\begin{aligned} \text{cond}(\tilde{A}) \ll \text{cond}(A), \\ \text{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \quad \text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}, \end{aligned} \quad (17)$$

где $\text{cond}(A)$ и $\text{cond}(\tilde{A})$ – числа обусловленности матриц A и \tilde{A} ; λ_{\max} , $\tilde{\lambda}_{\max}$ и λ_{\min} , $\tilde{\lambda}_{\min}$ – наибольшее и наименьшее собственные значения матриц A и \tilde{A} , соответственно.

Для системы уравнений (15) метод сопряженных градиентов с предобуславливателем C имеет следующий вид [10]:

$$\begin{aligned} r^0 = b - Ax^0, \quad p^0 = C^{-1}r^0, \quad z^0 = p^0, \\ x^{k+1} = x^k + \alpha_k p^k, \quad \alpha_k = \frac{(r^k, z^k)}{(Ap^k, p^k)}, \\ r^{k+1} = r^k - \alpha_k Ap^k, \quad z^{k+1} = C^{-1}r^{k+1}, \\ p^{k+1} = z^{k+1} + \beta_k p^k, \quad \beta_k = \frac{(r^{k+1}, z^{k+1})}{(r^k, z^k)}. \end{aligned} \quad (18)$$

Условием останова итерационного процесса является

$$\frac{\|Ax - b\|}{\|b\|} < \varepsilon. \quad (19)$$

Распараллеливание итерационного метода сопряженных градиентов основано на разбиении матрицы A горизонтальными полосами на m блоков, а вектора решения x и вектора правой части b СЛАУ на m частей так, что $n = m \times L$, где n – размерность системы уравнений, m – число процессоров, L – число строк матрицы в блоке (рис. 3).

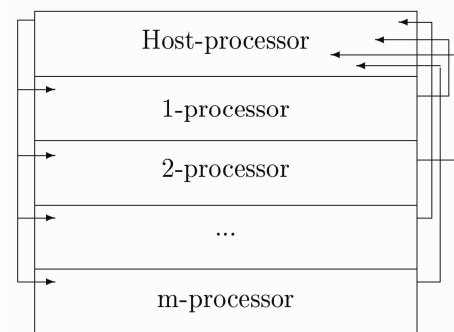


Рис. 3. Схема распределения данных по процессорам

На каждой итерации каждый из m процессоров вычисляет свою часть вектора решения. В случае умножения матрицы A на вектор x каждый из m процессоров умножает свою часть строк матрицы A на вектор x . Host-процессор отвечает за пересылки данных и также вычисляет свою часть вектора решения.

2. РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

Параллельный алгоритм матричной прогонки (ПАМП) и параллельный метод сопряженных градиентов с предобуславливателем (ПМСГ) реализованы на следующих современных многопроцессорных вычислительных системах: многопроцессорном вычислительном

комплексе МВС-1000/64 с помощью технологии MPI [11], 4-ядерном процессоре Intel Core I5-750 (CPU) на языке C++ в среде разработки «Visual Studio» (распараллеливание по потокам данных) и видеоускорителе GeForce GTX 285 (GPU) с помощью технологии CUDA [12].

Многопроцессорный вычислительный комплекс МВС-1000 – российский массивно-параллельный суперкомпьютер кластерного типа с распределенной памятью, установленный в Институте математики и механики УрО РАН (г. Екатеринбург).

МВС-1000/64 состоит из 14 2-процессорных 2-ядерных модулей AMD Opteron 64 bit (2.6 ГГц), интерфейса GbitEthernet и 112 Гб оперативной памяти.

Технические характеристики вычислительной платформы Intel Core I5-750 с видеоускорителем NVIDIA GeForce GTX 285 приводятся в табл. 1.

Таблица 1

Технические характеристики вычислительной платформы

CPU	4-ядер. Intel Core I5-750
Частота проц. (ГГц)	2.66
Оператив. память (Гб)	8
Разрядность ОС (Бит)	64
GPU	GeForce GTX 285
Количество проц. ядер	240
Частота ядра (МГц)	648
Частота проц. (МГц)	1476
Колич. видеопамяти (Мб)	1024

С помощью параллельного алгоритма матричной прогонки и предобусловленного метода сопряженных градиентов решена модельная задача о нахождении распределения потенциала в проводящей среде с известным решением (рис. 4).

Исходные данные и модельное решение задачи предоставлены лабораторией скважинной геофизики Института нефтегазовой геологии и геофизики СО РАН (г. Новосибирск).

После дискретизации задача сводится к решению СЛАУ с плохо обусловленной симметричной положительно-определенной блочно-трехдиагональной матрицей вида (6) размерности 76136×76136 с квадратными блоками порядка 248 (см. рис. 1).

Приближенное решение задачи сравнивалось с модельным решением с помощью вычисления относительной погрешности

$$\sigma = \frac{\|\bar{Y}^M - \bar{Y}^H\|}{\|\bar{Y}^M\|}, \tag{20}$$

где \bar{Y}^M – модельное решение задачи, \bar{Y}^H – приближенное решение задачи.

U-потенциал (решение СЛАУ)

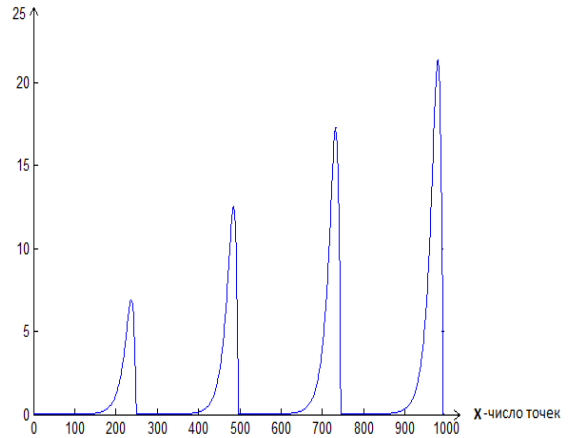


Рис. 4. Решение модельной задачи

Условие (20) выбиралось в качестве критерия останова итерационного ПМСГ при решении модельной задачи.

Предварительно находилось число обусловленности исходной матрицы A

$$\text{cond}(A) = \lambda_{\max} / 1,3 \cdot 10^{11},$$

$$\lambda_{\max} = 1,4 \cdot 10^6, \lambda_{\min} = 1,1 \cdot 10^{-5} > 0,$$

где λ_{\max} и λ_{\min} – наибольшее и наименьшее собственные значения исходной матрицы.

В случае решения задачи предобусловленным ПМСГ с целью проверки условия (17) находилось число обусловленности матрицы \tilde{A}

$$\text{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} \approx 4,1 \cdot 10^9 < \text{cond}(A).$$

Для сравнения времени счета решения задачи введем коэффициенты ускорения и эффективности параллельных алгоритмов

$$S_m = T_1 / T_m, E_m = S_m / m, S = T_1 / T_2,$$

где T_m – время выполнения параллельного алгоритма на МВС-1000 либо на многоядерном процессоре с числом процессоров или ядер m ($m > 1$), T_1 – время выполнения последовательного алгоритма на одном процессоре либо на одном ядре, T_2 – время решения задачи на видеоускорителе. T_m представляет собой совокупность чистого времени счета и накладных расходов на межпроцессорные обмены $T_m = T_c + T_o$. Число процессоров m соответствует упомянутому разбиению векторов на m частей и разбиению исходной области на m подобластей.

В общем случае эффективность распараллеливания меняется в пределах $0 < E_m < 1$. В идеальном случае при равномерной и сбалансированной загрузке процессоров и минимальном времени обменов между ними E_m близко к единице, но при решении практических задач она уменьшается за счет накладных расходов.

В табл. 2 и 3 приведены времена счета и коэффициенты ускорения и эффективности решения модельной задачи на 4-ядерном процессоре Intel Core I5-750, видеоускорителе NVIDIA GeForce GTX 285 и многопроцессорном комплексе МВС-1000/64 с помощью параллельного метода сопряженных градиентов с предобуславливателем и параллельного алгоритма матричной прогонки при $\sigma_{\text{ПАМП}} \approx 2 \cdot 10^{-7}$. Использование технологии OpenMP для 4-ядерного процессора Intel Core I5-750 с общей памятью уменьшает время решения задачи.

Таблица 2
Решение задачи методом ПМСГ

Вычислитель	$T_m, \text{с}$	$S_m(S)$	E_m
Intel Core I5-750 (1 ядро)	57 21-OpenMP	—	—
Intel Core I5-750 (2 ядра)	46 16-OpenMP	1,24	0,62
Intel Core I5-750 (4 ядра)	36 14-OpenMP	1,50	0,40
GeForce GTX 285	16	3,56	—
МВС—1000/64 (1 проц.)	120	—	—
МВС—1000/64 (2 проц.)	65	1,85	0,92
МВС—1000/64 (4 проц.)	34	3,53	0,88

Таблица 3
Решение задачи методом ПАМП

Вычислитель	$T_m, \text{с}$	$S_m(S)$	E_m
Intel Core I5-750 (1 ядро)	52 21-OpenMP	—	—
Intel Core I5-750 (2 ядра)	28 18-OpenMP	1,86	0,93
Intel Core I5-750 (4 ядра)	16	3,25	0,81
GeForce GTX 285	10	5,2	—
МВС—1000/64 (1 проц.)	96	—	—
МВС—1000/64 (2 проц.)	60	1,6	0,80
МВС—1000/64 (4 проц.)	31	3,1	0,77

Заметим, что время решения задачи с помощью метода сопряженных градиентов без предобуславливателя на одном ядре Intel Core I5-

750 при $\sigma_{\text{МСГ}} = 10^{-3}$ составило 55 минут, что существенно превышает времена решения задачи, представленные в табл. 2 и табл. 3.

ЗАКЛЮЧЕНИЕ

Для решения СЛАУ с блочно-трехдиагональными матрицами предложены и численно реализованы на многопроцессорных вычислителях различного типа параллельный алгоритм матричной прогонки и параллельный метод сопряженных градиентов с предобуславливателем.

Проведено сравнение времени счета параллельных алгоритмов на видеоускорителе NVIDIA GeForce GTX 285, 4-ядерном процессоре Intel Core I5-750 и многопроцессорном вычислительном комплексе МВС-1000/64 при решении модельной задачи о нахождении распределения потенциала на поверхности земли в проводящей среде.

Использование параллельных методов матричной прогонки и сопряженных градиентов с предобуславливателем позволяет достаточно быстро решать задачи с плохо обусловленными матрицами на многопроцессорных вычислителях различного типа, что позволяет рекомендовать данные методы для решения задач электроразведки.

Авторы выражают признательность за полезные советы и внимание к работе члену-корреспонденту РАН Владимиру Васильевичу Васину.

СПИСОК ЛИТЕРАТУРЫ

1. Crank J. The Mathematics of Diffusion. Oxford: Clarendon press, 1975.
2. Самарский А. А. Теория разностных схем. М.: Наука, 1983.
3. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Наука, 1966.
4. Дашевский Ю. А., Суродина И. В., Эпов М. И. Квазитрехмерное математическое моделирование диаграмм неосесимметричных зондов постоянного тока в анизотропных разрезах // Сиб. ЖИМ. 2002. Т. 5. №3 (11). С. 76–91.
5. Акимова Е. Н. Распараллеливание алгоритма матричной прогонки // Математическое моделирование. 1994. Т. 6. № 9. С. 61–67.
6. Акимова Е. Н., Горбачев И. И., Попов В. В. Решение задач многокомпонентной диффузии с помощью алгоритма матричной прогонки // Мат. моделирование. 2005. Т. 17. № 9. С. 85–92.
7. Горбачев И. И., Попов В. В., Акимова Е. Н. Моделирование диффузионного взаимодействия карбонитридных выделений с аустенитной матрицей с учетом возможности изменения их состава // Физика металлов и металловедение. 2006. Т. 102. № 1. С. 22–32.

8. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений. М.: Наука, 1978.

9. Фаддеев В. К., Фаддеева В. Н. Вычислительные методы линейной алгебры. М.: Гос. издат. физ.-мат. литературы, 1963.

10. Амосов А. А. Циркулянтно предобусловленный метод сопряженных градиентов и его применение для численного решения интегрального уравнения переноса излучения // Современные проблемы математического моделирования: Тр. XI Всероссийск. шк.-семинара. Ростов-на-Дону: Изд-во Ростовск. гос. ун-та, 2005. Выпуск 4. С. 49–65.

11. Руководство пользователя системы MVC-1000 / А. В. Баранов [и др.]. URL: <http://parallel.ru/mvs/user.html> (дата обращения: 15.02.2011).

12. Берилло А. NVIDIA CUDA – неграфические вычисления на графических процессорах. URL: <http://www.ixbt.com/video3/cuda-1.shtml> (дата обращения: 15.02.2011).

ОБ АВТОРАХ

Акимова Елена Николаевна, вед. науч. сотр. отдела некорректных задач анализа и приложений Ин-та математики и механики УрО РАН, проф. каф. вычислительн. методов и уравнений матем. физики Уральск. федерального ун-та, г. Екатеринбург. Дипл. математик (Уральск. гос. ун-т, г. Екатеринбург, 1982). Д-р физ.-мат. наук по матем. моделированию, численным методам и комплексам программ (ЮУрГУ, 2009). Иссл. в обл. параллельных алгоритмов решения прикладных задач.

Белоусов Дмитрий Владимирович, вед. программист отдела вычислительных сетей Ин-та математики и механики УрО РАН, г. Екатеринбург. Дипл. матем.-программист по матем. обеспечению и администрированию инф. систем (УГТУ-УПИ, 2006). Иссл. в обл. параллельных вычислений на многопроцессорных системах.