

УДК 517.968:519.612:004.021

Алгоритмы решения обратных геофизических задач на многопроцессорных вычислительных системах*

Е.Н. Акимова^{1,2}, Д.В. Белоусов^{1,2}, В.Е. Мисиллов¹

¹Институт математики и механики УрО РАН, ул. С. Ковалевской, 16, Екатеринбург, 620990

²Уральский Федеральный университет, ул. Мира, 19, Екатеринбург, 620002

E-mails: aen15@yandex.ru (Акимова Е.Н.), rtfdeamon@mail.ru (Белоусов Д.В.), out.mrscreg@gmail.com (Мисиллов В.Е.)

Акимова Е.Н., Белоусов Д.В., Мисиллов В.Е. Алгоритмы решения обратных геофизических задач на многопроцессорных вычислительных системах // Сиб. журн. вычисл. математики / РАН. Сиб. отд.-ние. — Новосибирск, 2013. — Т. 16, № 2. — С. 107–121.

Для решения обратных задач гравиметрии предложены устойчивые параллельные алгоритмы на основе итерационных методов градиентного типа. Для решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки построены параллельные методы матричной прогонки, квадратного корня и метод сопряженных градиентов с предобуславливателем. Алгоритмы реализованы на многопроцессорных вычислительных системах различного типа: многопроцессорном комплексе МВС–ИММ, графических процессорах NVIDIA и многоядерном процессоре Intel с использованием новых вычислительных технологий. Параллельные алгоритмы встроены в разработанную систему удаленных вычислений “Специализированный Веб-портал решения задач на многопроцессорных вычислителях”. Решены задачи с квазимодельными и реальными данными.

Ключевые слова: обратные задачи гравиметрии, параллельные алгоритмы, прямые и итерационные методы, многопроцессорные вычислительные системы.

Akimova E.N., Belousov D.V., Misilov V.E. Algorithms for solving inverse geophysical problems on parallel computing systems // Siberian J. Num. Math. / Sib. Branch of Russ. Acad. of Sci. — Novosibirsk, 2013. — Vol. 16, № 2. — P. 107–121.

For solving inverse gravimetry problems, efficient stable parallel algorithms based on iterative gradient methods are proposed. For solving systems of linear algebraic equations with block-tridiagonal matrices arising in geoelectrics problems, a parallel matrix sweep algorithm, a square root method, and a conjugate gradient method with preconditioner are proposed. The algorithms are implemented numerically on the MVS-IMM parallel computing system, NVIDIA graphics processors, and the Intel multi-core CPU with the use of new computing technologies. The parallel algorithms are incorporated into a developed system of remote computations “Specialized Web-Portal for Solving Geophysical Problems on Multiprocessor Computers”. Problems with “quasi-model” and real data are solved.

Key words: inverse gravimetry problems, parallel algorithms, direct and iterative methods, parallel computing systems.

*Статья рекомендована к публикации программным комитетом Международной конференции “Параллельные вычислительные технологии 2012”. Работа выполнена при поддержке УрО РАН в рамках Программ фундаментальных исследований Президиума РАН №15 (проект №12-П-1-1023) и №18 (проект №12-П-15-2019).

1. Введение

Важнейшими задачами исследования структуры земной коры являются обратные задачи гравиметрии: задача гравиметрии о нахождении переменной плотности в слое [1] и структурная задача гравиметрии о восстановлении поверхности раздела между средами [2]. Задачи гравиметрии описываются линейными и нелинейными интегральными уравнениями Фредгольма первого рода, т. е. являются существенно некорректными задачами. При разработке методов решения задач используются идеи итеративной регуляризации [3]. После дискретизации с использованием итерационных процессов задачи сводятся к системам линейных алгебраических уравнений (СЛАУ) с плохо обусловленными заполненными матрицами большой размерности (несколько сотен тысяч). Необходимость повышения точности результатов решения задач, в частности использование более мелких сеток, существенно увеличивает время вычислений.

Важнейшими задачами исследования неоднородности земной коры являются задачи электроразведки. Важной задачей электроразведки является задача бокового каротажного зондирования (БКЗ). В результате интерпретации данных каротажа получают значение удельного электрического сопротивления пласта, близкое к истинному. В работе [4] показано, что после использования конечно-разностной аппроксимации двумерная задача БКЗ сводится к решению СЛАУ с блочно-трехдиагональной матрицей большой размерности.

Одним из путей уменьшения времени расчетов и повышения эффективности решения геофизических задач является распараллеливание алгоритмов и использование многопроцессорных вычислительных систем (МВС). В научно-исследовательских институтах и университетах России распространены массивно-параллельные суперкомпьютеры кластерного типа с распределенной памятью. В Институте математики и механики УрО РАН (г. Екатеринбург) установлены МВС–1000/17ЕК, МВС–ИММ и суперкомпьютер “Уран”, которые успешно используются при решении прикладных задач.

В настоящее время в мире для решения прикладных задач наметилась тенденция к использованию в качестве вычислительных систем многоядерных гибридных вычислителей с графическими процессорами (видеокартами). По сравнению с суперкомпьютерами МВС гибридные вычислительные системы на основе графических процессоров представляют собой более дешевую многопроцессорную технику с низким энергопотреблением. В Институте математики и механики УрО РАН установлен гибридный вычислительный кластер на основе видеоускорителей NVIDIA Tesla.

В данной работе для решения линейной обратной задачи гравиметрии о нахождении плотности в слое итерационными методами градиентного типа и нелинейной обратной задачи гравиметрии о восстановлении поверхности раздела между средами с помощью итеративно регуляризованного метода Ньютона, а также для решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки построены эффективные параллельные прямые и итерационные алгоритмы. Алгоритмы реализованы на многопроцессорных вычислительных системах различного типа: многопроцессорном комплексе МВС–ИММ, графических процессорах NVIDIA и многоядерном процессоре Intel. Проведены исследование эффективности и оптимизация параллельных алгоритмов для решения геофизических задач на гибридных вычислительных системах. Параллельные алгоритмы встроены в разработанную систему удаленных вычислений “Специализированный Веб-портал решения задач на многопроцессорных вычислителях”. Решены задачи с квазимодельными и реальными данными.

2. Методы решения обратных задач гравиметрии

2.1. Задача о нахождении плотности в слое

Рассматривается задача о нахождении переменной плотности $\sigma = \sigma(x, y)$ в слое $\Pi = \{(x, y, z) \in R^3 : (x, y) \in D, H_1(x, y) \leq z \leq H_2(x, y)\}$ по гравитационным данным, измеренным на площади $D = \{(x, y) \in R^2 : a \leq x \leq b, c \leq y \leq d\}$ земной поверхности (H_1, H_2 — константы для горизонтального слоя). Используется априорная информация об отсутствии аномалий плотности вне слоя с границами $H_1 = H_1(x, y)$ и $H_2 = H_2(x, y)$ такими, что $H_1 < H_2 \quad \forall (x, y)$, и выполняется условие $H_i(x, y) \xrightarrow[x, y \rightarrow \pm\infty]{} h_i = \text{const.}$

При этом предполагается, что распределение плотности $\sigma = \sigma(x, y)$ внутри слоя не зависит от z (ось z направлена вниз). Данное предположение обеспечивает единственность решения для латерально неоднородной плотности. Если плотность зависит от z , то в самой общей постановке существует континуум решений.

Задача нахождения неизвестной плотности $\sigma(x, y)$ сводится к решению линейного двумерного интегрального уравнения Фредгольма первого рода

$$A\sigma \equiv f \iint_{a \ c}^{b \ d} \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + H_1^2(x', y')]^{1/2}} - \frac{1}{[(x-x')^2 + (y-y')^2 + H_2^2(x', y')]^{1/2}} \right\} \sigma(x', y') dx' dy' = \Delta g(x, y), \quad (1)$$

где f — гравитационная постоянная, $\Delta g(x, y)$ — гравитационный эффект, порождаемый источниками в горизонтальном или криволинейном слое.

Предварительная обработка гравитационных данных, связанная с выделением аномального поля, выполняется по методике, предложенной П.С. Мартышко и И.Л. Пруткиным [5].

После дискретизации уравнения на сетке, где задана $\Delta g(x, y)$, и аппроксимации интегрального оператора по квадратурным формулам задача (1) сводится к решению системы линейных алгебраических уравнений (СЛАУ) либо с симметричной положительно определенной матрицей (горизонтальный слой), либо с несимметричной матрицей (криволинейный слой). Так как уравнение (1) относится к классу некорректно поставленных задач, то СЛАУ, возникающее в результате дискретизации уравнения, является плохо обусловленной и преобразуется к виду (схема Лаврентьева):

$$(A + \alpha E)z = b, \quad (2)$$

где α — параметр регуляризации.

В случае криволинейного слоя исходная матрица СЛАУ несимметрична, поэтому система предварительно преобразуется к виду (схема Тихонова):

$$(A^T A + \alpha' E)z = A^T b, \quad (3)$$

где A^T — транспонированная матрица, α' — параметр регуляризации.

Для решения систем уравнений (2) и (3) используются итерационные методы градиентного типа: итеративно регуляризованный метод простой итерации (МПИ):

$$z^{k+1} = z^k - \frac{1}{\lambda_{\max}} [(A + \alpha E)z^k - b], \quad (4)$$

где λ_{\max} — максимальное собственное значение матрицы $A + \alpha E$ (симметричный случай); метод минимальных невязок (ММН):

$$z^{k+1} = z^k - \frac{(A(Az^k - b), Az^k - b)}{\|A(Az^k - b)\|^2} (Az^k - b); \quad (5)$$

метод минимальной ошибки (ММО):

$$z^{k+1} = z^k - \frac{\|Az^k - b\|^2}{\|A^\top(Az^k - b)\|^2} A^\top(Az^k - b) \quad (6)$$

и метод наискорейшего спуска (МНС):

$$z^{k+1} = z^k - \frac{\|A^\top Az^k - A^\top b\|^2}{\|A(A^\top Az^k - A^\top b)\|^2} A^\top(Az^k - b). \quad (7)$$

Для методов (5)–(7) в регуляризованном варианте матрица A заменяется на $A + \alpha E$. Условием останова итерационных процессов является $\|Az^k - b\|/\|b\| < \varepsilon$ — останов по невязке.

Для решения линейной обратной задачи гравиметрии о восстановлении плотности в слое устойчивые итерационные методы градиентного типа численно реализованы на многопроцессорном комплексе МВС–ИММ с помощью библиотеки MPI [6] и графических процессорах NVIDIA с помощью технологии CUDA [7].

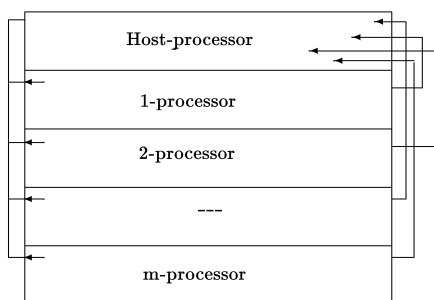


Рис. 1. Схема распределения данных по процессорам

m процессоров умножает свою часть строк транспонированной матрицы A^\top на всю матрицу A . Host-процессор (ведущий) отвечает за пересылки данных и также вычисляет свою часть вектора решения.

Отметим, что при реализации параллельных итерационных методов на МВС–ИММ для решения задачи гравиметрии матрица СЛАУ формируется и хранится в памяти каждого процессора по частям, что увеличивает эффективность распараллеливания.

Распараллеливание и численная реализация итеративно регуляризованного МПИ на видеоускорителях NVIDIA при решении линейной задачи гравиметрии описаны в работе [8]. В данной работе эти принципы распространяются на распараллеливание и реализацию на видеоускорителях NVIDIA итерационных методов ММН, ММО и МНС.

Для оптимизации работы с памятью при вычислениях используются два приема.

1. Для сеток не очень большой размерности (СЛАУ 12100×12100), когда данные входят в память видеокарты, матрица A порядка n и вектор z размерности n расширяются до размерности M и дополняются нулями таким образом, чтобы M было кратно числу блоков. Размер блока BLOCK_SIZE (threads) выбирается кратным 16, поскольку в одном блоке группируются до 512 потоков. Тогда количество

Распараллеливание итерационных методов градиентного типа основано на разбиении матрицы A горизонтальными полосами на m блоков, а вектора решения z и вектора правой части b СЛАУ на m частей так, что $n = m \cdot L$, где n — размерность системы уравнений, m — число процессоров, L — число строк матрицы в блоке (рис. 1). На текущей итерации каждый из m процессоров вычисляет свою часть вектора решения. В случае умножения матрицы A на вектор z каждый из m процессоров умножает свою часть строк матрицы A на вектор z . В случае матричного умножения $A^\top A$ каждый из

блоков вычисляется по формуле: $\text{blocks} = M/\text{BLOCK_SIZE}$. Вычисления производятся без выгрузки данных в память Host-процессора. Данные находятся только в памяти видеокарты.

2. Для сеток довольно большой размерности (СЛАУ 40000×40000), когда данные не входят в память видеокарты, наилучшим по быстродействию оказывается метод вычисления элементов матрицы A “на лету”, т.е. вычисление значения элемента матрицы происходит в момент обращения к этому элементу без сохранения его в память видеокарты. Это позволяет существенно снизить количество обращений к памяти видеокарты и заметно ускорить процесс вычислений по сравнению с хранением матрицы A в памяти Host-процессора и порционной загрузкой в видеоускоритель для вычислений.

2.2. Задача о нахождении поверхности раздела между средами

Рассматривается трехмерная структурная обратная задача гравиметрии о восстановлении поверхности раздела между средами по известному скачку плотности и гравитационному полю, измеренному на некоторой площади земной поверхности.

Предполагается, что нижнее полупространство состоит из двух слоев постоянной плотности, разделенных искомой поверхностью S . В предположении, что гравитационная аномалия создана отклонением искомой поверхности S от горизонтальной плоскости $z = H$ (ось z направлена вниз), в декартовой системе координат функция $z = z(x, y)$, описывающая искомую поверхность раздела, удовлетворяет нелинейному двумерному интегральному уравнению Фредгольма первого рода

$$A[z] \equiv f \Delta \sigma \iint_{a \ c}^{b \ d} \left\{ \frac{1}{[(x - x')^2 + (y - y')^2 + z^2(x', y')]^{1/2}} - \frac{1}{[(x - x')^2 + (y - y')^2 + H^2]^{1/2}} \right\} dx' dy' = G(x, y), \quad (8)$$

где f — гравитационная постоянная, $\Delta \sigma$ — скачок плотности на границе раздела сред, $G(x, y)$ — аномальное гравитационное поле, $z = H$ — асимптотическая плоскость для данной границы раздела. Задача гравиметрии (8) является существенно некорректной задачей.

После дискретизации уравнения (8) на сетке $n = M \times N$, где задана $G(x, y)$, и аппроксимации интегрального оператора по квадратурным формулам имеем систему нелинейных уравнений:

$$A_n[z] = F_n. \quad (9)$$

Для решения системы уравнений (9) используется итеративно регуляризованный метод Ньютона [9]:

$$z^{k+1} = z^k - [A'_n(z^k) + \alpha_k I]^{-1} [A_n(z^k) + \alpha_k z^k - F_n]. \quad (10)$$

Здесь $A_n(z^k)$ и F_n — конечномерные аппроксимации интегрального оператора и правой части в уравнении (8), $A'_n(z^k)$ — производная оператора A в точке z^k , I — единичный оператор, α_k — последовательность положительных параметров регуляризации.

Нахождение очередного приближения метода Ньютона z^{k+1} по найденному z^k сводится к решению СЛАУ:

$$A_n^k z^{k+1} = F_n^k, \quad (11)$$

где $A_n^k = A_n'(z^k) + \alpha_k I$ — плохо обусловленная несимметричная заполненная $n \times n$ матрица, $F_n^k = A_n^k z^k - (A_n(z^k) + \alpha_k z^k - F_n)$ — вектор размерности n .

Предварительно система уравнений (11) приводится к виду

$$B^k z^{k+1} \equiv [(A_n^k)^\top A_n^k + \alpha_k' I] z^{k+1} = (A_n^k)^\top F_n^k \equiv b, \quad (12)$$

где $(A_n^k)^\top$ — транспонированная матрица, α_k' — параметры регуляризации.

На каждом шаге метода Ньютона для решения СЛАУ (12) с симметричной положительно определенной матрицей используются итерационные методы градиентного типа (4)–(7) и метод сопряженных градиентов (МСГ) в регуляризованном варианте

$$z^{k+1} = z^k - \gamma_k (B^k z^k - b) + \beta_k (z^k - z^{k-1}), \quad (13)$$

где γ_k и β_k вычисляются по известным формулам [10]. Условием останова итерационного процесса МСГ является следующее: $\|Bz^k - b\|/\|b\| < \varepsilon$.

Численная реализация и распараллеливание метода Ньютона с использованием градиентных методов для решения нелинейной обратной задачи гравиметрии о нахождении поверхности раздела между средами выполнены на МВС–ИММ. В ближайшее время предполагается реализация метода Ньютона на графических процессорах NVIDIA.

3. Параллельные алгоритмы решения СЛАУ с блочно-трехдиагональными матрицами

Как упоминалось во введении, задача БКЗ после конечно-разностной аппроксимации сводится к решению СЛАУ с блочно-трехдиагональной матрицей, представленной на рис. 2.

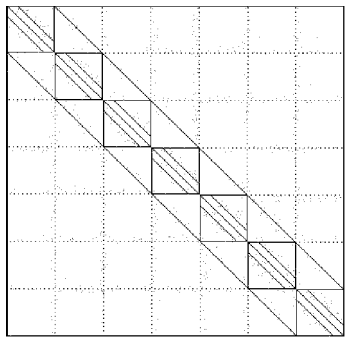


Рис. 2. Вид матрицы СЛАУ

Для решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки используются параллельные алгоритмы матричной прогонки (ПАМП), квадратного корня (ПАКК) и метод сопряженных градиентов с предобуславливателем (ПМСГ) в случае решения СЛАУ с симметричной положительно определенной матрицей. Параллельные алгоритмы реализованы на графических процессорах NVIDIA с помощью технологии CUDA и многоядерном процессоре Intel с помощью технологии OpenMP [6].

3.1. Параллельный алгоритм матричной прогонки

Рассмотрим систему уравнений с блочно-трехдиагональной матрицей общего вида:

$$\begin{cases} C_0 \bar{Y}_0 - B_0 \bar{Y}_1 = \bar{F}_0, & i = 0, \\ -A_i \bar{Y}_{i-1} + C_i \bar{Y}_i - B_i \bar{Y}_{i+1} = \bar{F}_i, & i = 1, 2, \dots, N-1, \\ -A_N \bar{Y}_{N-1} + C_N \bar{Y}_N = \bar{F}_N, & i = N, \end{cases} \quad (14)$$

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 .

В работе [11] для решения СЛАУ (14) предложен параллельный алгоритм матричной прогонки. Основная идея параллельного алгоритма заключается в следующем. Исходную область P (прямоугольник) разобьем на L подобластей вертикальными линиями так, что $N = L \times M$. 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 меньшей размерности по сравнению с исходной, которая решается классическим методом матричной прогонки [12]. После нахождения векторов-параметров \bar{Y}_K остальные искомые неизвестные выражаются через параметрические неизвестные и находятся в каждой подобласти L независимо.

3.2. Метод сопряженных градиентов с предобуславливателем

Для решения СЛАУ (14) можно использовать эффективный метод сопряженных градиентов с предобуславливателем. Введение предобуславливания применяется с целью ускорения сходимости итерационного процесса и состоит в том, что исходная система уравнений $Ax = b$ заменяется на систему:

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

для которой итерационный метод сходится существенно быстрее.

Условием выбора предобуславливателя C является условие

$$\text{cond}(\tilde{A}) \ll \text{cond}(A), \quad \text{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \quad \text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}, \quad (16)$$

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

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

$$\begin{aligned} r^0 &= b - Ax^0, & p^0 &= C^{-1}r^0, & z^0 &= p^0, \\ x^{k+1} &= x^k + \alpha_k p^k, & \alpha_k &= \frac{(r^k, z^k)}{(Ap^k, p^k)}, & r^{k+1} &= r^k - \alpha_k Ap^k, \\ z^{k+1} &= C^{-1}r^{k+1}, & p^{k+1} &= z^{k+1} + \beta_k p^k, & \beta_k &= \frac{(r^{k+1}, z^{k+1})}{(r^k, z^k)}. \end{aligned} \quad (17)$$

Здесь предобуславливатель выбирается путем неполного LU -разложения матрицы A . Условием останова итерационного процесса МСГ с предобуславливателем является “останов по невязке”.

Распараллеливание МСГ аналогично распараллеливанию других итерационных методов градиентного типа (см. рис. 1).

3.3. Метод квадратного корня

Одним из быстрых методов решения СЛАУ с симметричной положительно определенной матрицей является метод квадратного корня [10]. Метод основан на разложении симметричной матрицы A в произведение $A = S^T S$, где S — верхняя треугольная матрица с положительными элементами на главной диагонали, S^T — транспонированная матрица. Метод состоит в последовательном решении двух систем уравнений с треугольными матрицами:

$$S^T y = b, \quad Sz = y. \quad (18)$$

Решения систем уравнений (18) находятся по рекуррентным формулам:

$$\begin{cases} y_1 = b_1/s_{11}, & y_i = \left(b_i - \sum_{k=1}^{i-1} s_{ki}y_k \right) / s_{ii}, & i = 2, 3, \dots, n; \\ z_n = y_n/s_{nn}, & z_i = \left(y_i - \sum_{k=i+1}^n s_{ik}z_k \right) / s_{ii}, & i = n-1, n-2, \dots, 1. \end{cases} \quad (19)$$

Основная идея распараллеливания метода квадратного корня для многопроцессорного вычислителя с общей памятью основана на параллельном вычислении элементов s_{ij} , $j = i, \dots, n$, каждой i -й строки матрицы S . Строка с номером i разбивается на m частей так, что $n - i = m \times L_i$, где i — номер строки, n — размерность системы уравнений, m — число процессоров, L_i — число элементов строки, вычисляемых каждым процессором (рис. 3).

1-й процессор	2-й процессор	...	m-й процессор
---------------	---------------	-----	---------------

Рис. 3. Разбиение i -й строки по процессорам

4. Специализированный Веб-портал решения задач на многопроцессорных вычислителях

Вы зашли как 40150
Выход

[Общая информация о сервере](#)

[Интерфейс пользователя](#)

[Виды задач и методы решения](#)

[Новая задача](#)

[Запущенные задачи](#)

[Контактная информация](#)

Инструкция пользователя:

- Выберите метод и нажмите кнопку "Описание" для ознакомления с методом
- Нажмите кнопку "Запуск" для введения входных данных и запуска задачи

Задача	Метод	Доступные вычислители	Описание	Запуск
Выделение аномального поля	Предварительная обработка	МВС-НММ	Описание	Запуск
Задача Дирихле	Метод Гаусса-Зейделя	МВС-НММ GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Задача Дирихле	Метод разделения переменных	МВС-НММ	Описание	Запуск
Линейная задача гравиметрии	Метод простой итерации	МВС-НММ GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Линейная задача гравиметрии	Метод наискорейшего спуска	МВС-НММ GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Линейная задача гравиметрии	Метод минимальной ошибки	МВС-НММ GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Линейная задача гравиметрии	Метод минимальных невязок	МВС-НММ GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Нелинейная задача гравиметрии	Метод Ньютона	МВС-НММ	Описание	Запуск
Решение СЛАУ для задач электроразведки	Метод сопряженных градиентов с предобуславливателем	GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Решение СЛАУ для задач электроразведки	Метод матричной прогонки	GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск
Решение СЛАУ для задач электроразведки	Метод квадратного корня	GPU NVIDIA GeForce GTX 480 GPU NVIDIA Tesla S2050	Описание	Запуск

Рис. 4. Специализированный Веб-портал

Параллельные алгоритмы решения обратной задачи гравиметрии о восстановлении плотности в слое и структурной задачи гравиметрии о восстановлении поверхности раздела между средами, а также параллельные алгоритмы решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки встроены в разработанную систему удаленных вычислений “Специализированный Веб-портал решения задач на многопроцессорных вычислителях”, установленный в Отделе некорректных задач анализа и приложений ИММ УрО РАН (рис. 4).

Изначально Веб-портал был предназначен для запуска программ решения задач гравиметрии на многопроцессорном комплексе МВС–1000/17ЕК через Веб-интерфейс [13].

В настоящее время на Веб-портале предусмотрен запуск программ для решения задач на МВС–ИММ, гибридном вычислительном кластере NVIDIA Tesla, установленном в ИММ УрО РАН и гибридной вычислительной системе (ГВС) NVIDIA GeForce, установленной на кафедре ВМ и УМФ ИРИТ-РтФ УрФУ.

МВС–ИММ состоит из 14 2-х процессорных 2-х ядерных модулей AMD Opteron 64 bit (2.6 ГГц), интерфейса GbitEthernet и 112 Гбайт оперативной памяти. Кластер NVIDIA Tesla включает 20 вычислительных узлов, имеющих 8 GPU Tesla S2050, 50 Гбайт ОЗУ и 2 шестиядерных CPU. ГВС представляет собой 4-ядерный процессор Intel Core I7-950 с графическими процессорами NVIDIA GeForce GTX 480.

4.1. Общая характеристика

Специализированный Веб-портал предназначен для запуска программ решения задач гравиметрии (выделение аномального поля, нахождения плотности в слое, восстановления поверхности раздела между средами) и решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки на многопроцессорном комплексе МВС–ИММ, вычислительном кластере NVIDIA Tesla и системе ГВС NVIDIA GeForce. Веб-портал предоставляет возможность пользователю через Веб-интерфейс выбирать тип вычислителя с указанием числа процессорных узлов, вид задачи и метод ее решения, загружать входные данные, получать выходные данные и графическое изображение результатов решения с помощью графических пакетов Surfer и Gnuplot. Для каждой задачи выводится время счета.

Веб-портал состоит из трех основных частей: HTTP-сервер IIS (Internet Information Services — информационные службы Интернета), на котором установлено Веб-приложение; база данных SQL Server 2000, в которой хранятся все задачи пользователей с входными и выходными данными; служба, выполняющая загрузку данных, запуск задач на многопроцессорных вычислителях различных типов, просмотр состояния задачи и загрузку результатов завершившихся задач на Веб-портал.

4.2. Веб-приложение

В настоящее время к исходному Веб-приложению (см. [13]) добавлено следующее.

1. Дополнена система аутентификации. При регистрации на Веб-портале происходит проверка одноименной учетной записи на МВС–ИММ. При ее отсутствии или несовпадении паролей регистрация отменяется. Реализовано SSL-шифрование трафика.
2. Предусмотрена возможность взаимодействия с разными вычислительными устройствами. Веб-портал позволяет запускать методы решения задач на многопроцес-

сорных вычислителях различного типа, поддерживающих связь по ssh, передачу файлы по scp и запуск программ через планировщик mpirun или sbatch.

3. Модифицированы службы, отвечающие за прием и передачу файлов и запуск задач на вычислителях различного типа. В интерфейс запуска задачи добавлен выбор вычислителя, на котором эта задача будет запущена. В интерфейсе просмотра запущенных задач добавлено отображение вычислителя, на котором была запущена выбранная задача. Добавлена возможность удаления запущенной задачи из списка задач.
4. Добавлено управление загруженными файлами данных для вычислений, а именно, возможность запуска другой задачи с этими же данными с целью снижения нагрузки на сервер (перекачки файлов). Расширены наборы параметров задачи. Например, для решения линейных задач гравиметрии имеются две возможности: введение двух констант (границ слоя) для горизонтального слоя и загрузка двух файлов (границ слоя) для криволинейного слоя.
5. Подключено управление созданием изображений: пользователь может выбрать, из каких входных и выходных файлов строить изображения, которые будут видны на странице с конкретной решенной задачей. Для задач гравиметрии строятся трехмерная поверхность и линии уровня с помощью программы Screenshot пакета Surfer.
6. Изменен дизайн страниц, добавлены подробные описания задач и методов. Реализована система управления контентом, а именно, возможность редактирования на сайте описания методов решения задач и добавления новых.

5. Результаты численных экспериментов

5.1. Решение задач гравиметрии с реальными данными

Для восточной части Урала был обработан массив гравитационных данных, измеренный на площади S , имеющей размеры $59.4 \times 144 \text{ км}^2$. Эта площадь пространственно совпадает с зоной Буткинской аномалии векового хода, ограничивающей Урал с востока. Зона представляет собой субмеридиональную аномалию электропроводности земной коры. Данные аномального гравитационного поля предоставлены сотрудниками Института геофизики УрО РАН (г. Екатеринбург).

Для изучения природы аномалии по реальным наблюдаемым данным решены: задача 1 о нахождении плотности в горизонтальном слое между глубинами $H_1 = 10 \text{ км}$ и $H_2 = 20 \text{ км}$ для области S и задача 2 о восстановлении поверхности раздела между средами. При этом шаги сетки $\Delta x = 0.594 \text{ км}$ и $\Delta y = 1.44 \text{ км}$, гравитационная постоянная $f = 6.67 \cdot 10^{-8} \text{ см}^3/\text{г} \cdot \text{с}^2$. Расстояние до асимптотической плоскости составляло $H = 15 \text{ км}$. Скачок плотности принимался равным $\Delta\sigma = 0.2 \text{ г}/\text{см}^3$.

После дискретизации исходных уравнений на сетке задачи свелись к СЛАУ с симметричной (задача 1) и несимметричной (задача 2) матрицами 10000×10000 . Для решения задачи 1 использовался параллельный итеративно регуляризованный ММН с параметром регуляризации $\alpha = 0.001$. Для решения задачи 2 использовался итеративно регуляризованный метод Ньютона, на каждом шаге которого применялся ПМСГ.

Задача 1 решена на МВС-ИММ, NVIDIA Tesla и ГВС GeForce. Задача 2 решена на МВС-ИММ.

На рис. 5 изображено распределение плотности в слое, восстановленной по аномальному полю для области S .

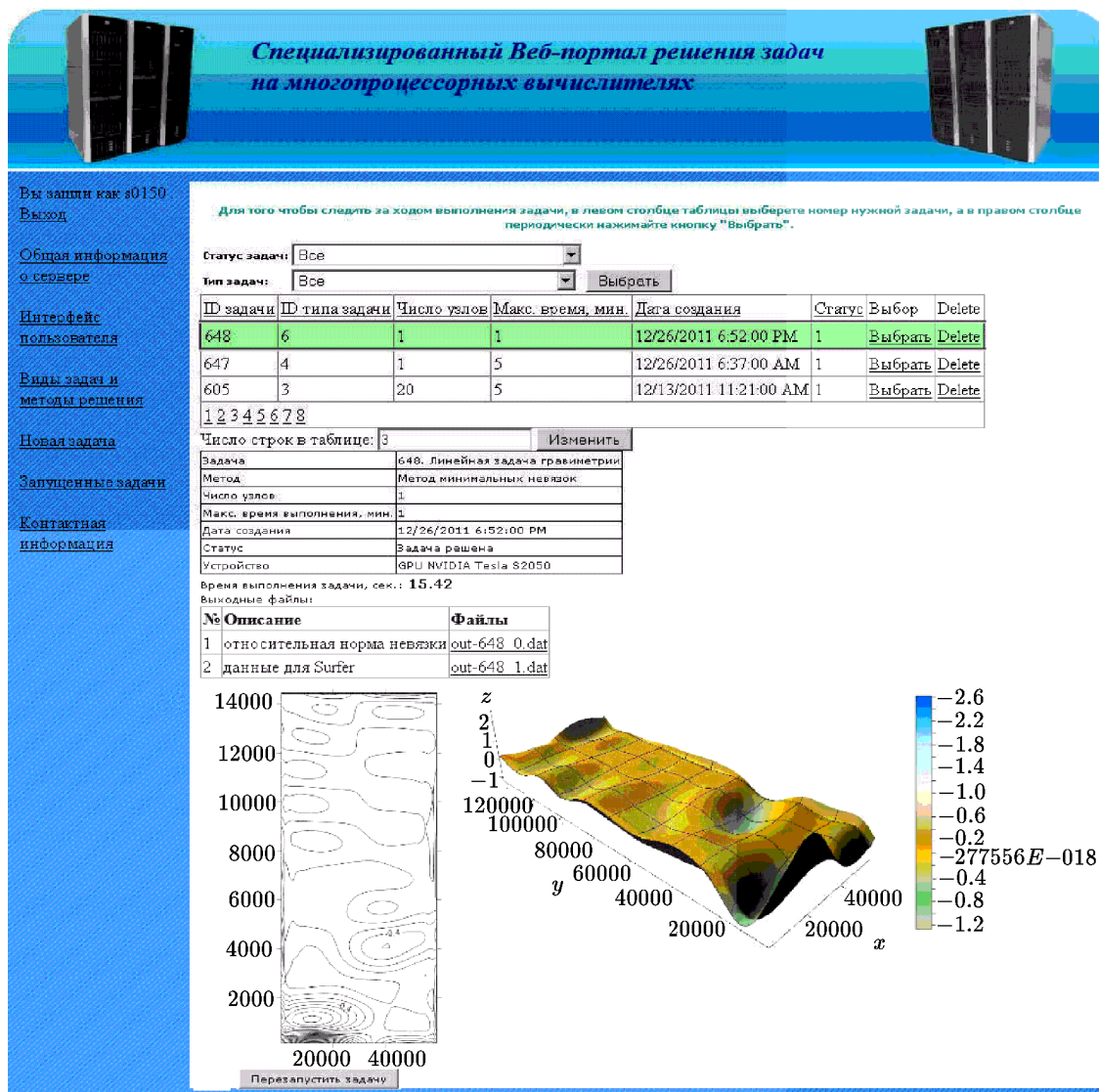


Рис. 5. Решение линейной задачи гравиметрии

На рис. 6 изображена восстановленная поверхность раздела.

Интерпретация результатов проведена сотрудниками ИГФ УрО РАН (г. Екатеринбург). В результате интерпретации выделен протяженный субмеридиональный блок земной коры пониженной плотности ($0.2-0.3 \text{ г/см}^3$) (см. [14]).

В табл. 1 приводятся времена решения линейной задачи гравиметрии на МВС-ИММ и ГВС NVIDIA GeForce GTX 480 при $\|Az^k - b\|/\|b\| \approx 0.011$ (749 итераций). Для решения задачи на МВС-ИММ использовалась технология MPI, для решения задачи на многоядерном процессоре Intel использовалась технология OpenMP, для решения задачи на видеоускорителе GeForce использовалась технология CUDA.

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

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

где T_m — время выполнения параллельного алгоритма на МВС-ИММ либо на многоядерном процессоре с числом процессоров или ядер m ($m > 1$), T_1 — время выполнения

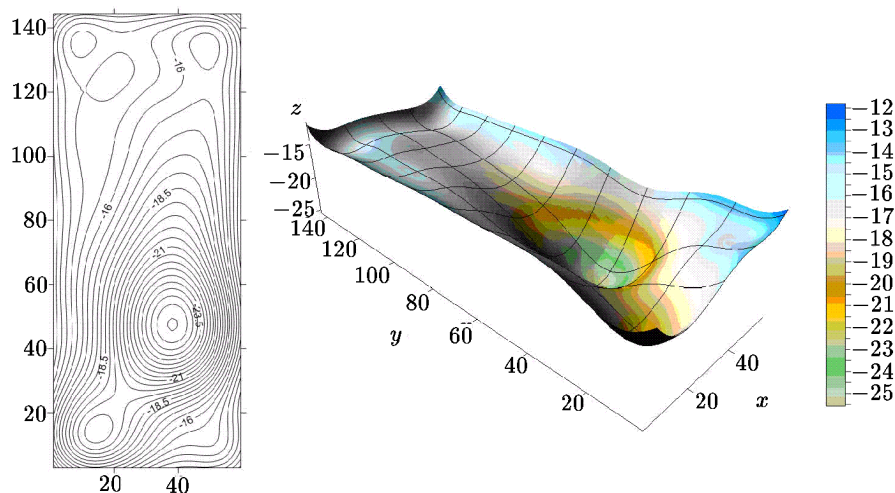


Рис. 6. Решение нелинейной задачи гравиметрии

Таблица 1. Времена решения линейной задачи гравиметрии

Вычислитель	Время T_m , мин.	Ускорение S_m либо S	Эффективность E_m
Intel Core I7 (1 ядро)	7.73	—	—
Intel Core I7 (2 ядра)	4.05	1.91	0.96
Intel Core I7 (4 ядра)	2.1	3.68	0.92
NVIDIA GeForce GTX 480	0.2	38.7	—
МВС-ИММ (1 процессор)	24.33	—	—
МВС-ИММ (2 процессора)	12.18	1.99	0.99
МВС-ИММ (4 процессора)	6.1	3.99	0.99
МВС-ИММ (10 процессоров)	2.46	9.89	0.99
МВС-ИММ (20 процессоров)	1.24	19.6	0.98
МВС-ИММ (50 процессоров)	0.5	48.7	0.97

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: $T_m = T_c + T_o$.

Результаты вычислений показывают, что использование метода Ньютона и итерационных методов градиентного типа при решении обратных задач гравиметрии позволяет получать корректные решения и определять аномальные плотностные параметры изучаемых глубинных зон земной коры. Применение параллельных алгоритмов при решении задач гравиметрии уменьшает время счета.

5.2. Решение задачи о нахождении распределения потенциала

С помощью параллельного алгоритма матричной прогонки, предобусловленного метода сопряженных градиентов и метода квадратного корня решена задача о нахождении распределения потенциала в проводящей среде с известным квазимодельным решением. Исходные данные и 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 размерности 76136×76136 с квадратными блоками порядка 248.

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

$$\sigma = \|\bar{Y}^M - \bar{Y}^{\Pi}\| / \|\bar{Y}^M\|,$$

где \bar{Y}^M — модельное решение задачи, \bar{Y}^{Π} — приближенное решение задачи.

Условие $\sigma < \varepsilon$ выбиралось в качестве критерия останова итерационного ПМСГ. Предварительно находилось число обусловленности исходной матрицы A :

$$\text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \approx 1.3 \cdot 10^{11}, \quad \lambda_{\max} \approx 1.4 \cdot 10^6, \quad \lambda_{\min} \approx 1.1 \cdot 10^{-5} > 0,$$

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

В случае решения задачи предобусловленным ПМСГ с целью проверки условия (16) находилось число обусловленности матрицы \tilde{A} :

$$\text{cond}(\tilde{A}) = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} \approx 4.1 \cdot 10^9 < \text{cond}(A).$$

Задача решена с помощью параллельного метода сопряженных градиентов с предобуславливателем, параллельного алгоритма матричной прогонки и параллельного метода квадратного корня при $\sigma_{\text{ПМСГ}} \approx 10^{-7}$, $\sigma_{\text{ПАМП}} \approx 2 \cdot 10^{-7}$, $\sigma_{\text{ПАКК}} \approx 6 \cdot 10^{-7}$.

На рис. 7 представлено численное решение задачи.

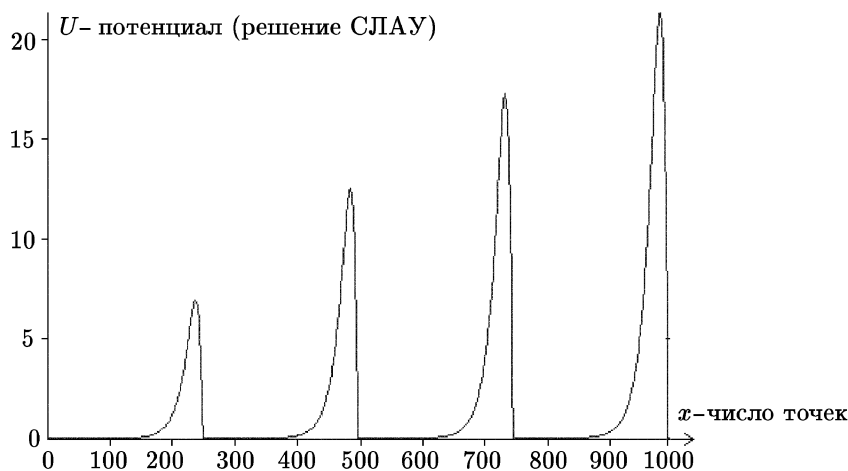


Рис. 7. Численное решение задачи

В табл. 2 приведены времена счета решения задачи на ГВС, установленной в Отделе некорректных задач анализа и приложений ИММ УрО РАН. Система состоит из 4-х ядерного процессора Intel Core I5-750 и видеоускорителя NVIDIA GeForce GTX 285. Предполагается включить данную ГВС в Веб-портал. Заметим, что время решения задачи с помощью МСГ 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.

Вначале распараллеливание методов ПМСГ, ПАМП и ПАКК для многоядерного процессора Intel с общей памятью проводилось средствами создания потоков операционной системы (ОС) с использованием средств разработки Windows API [15]. Для параллельного выполнения блока вычислений программы создавались потоки (Threads), каждый из потоков выполнялся на “логическом процессоре” ОС, вычисляя свою порцию данных.

Таблица 2. Времена решения задачи

Метод	Вычислитель	T_m , с (Windows API)	T_m , с (OpenMP)
ПМСГ	Intel Core I5 (1 ядро)	57	21
ПМСГ	Intel Core I5 (2 ядра)	46	16
ПМСГ	Intel Core I5 (4 ядра)	36	14
ПАМП	Intel Core I5 (1 ядро)	52	21
ПАМП	Intel Core I5 (2 ядра)	28	18
ПАМП	Intel Core I5 (4 ядра)	16	14
ПАМП	NVIDIA GeForce GTX 285	—	10
ПАКК	Intel Core I5 (1 ядро)	12	7.4
ПАКК	Intel Core I5 (2 ядра)	9	4.6
ПАКК	Intel Core I5 (3 ядра)	10	3.8
ПАКК	Intel Core I5 (4 ядра)	12	4.2

В конце каждого блока вычислений производилась барьерная синхронизация потоков. Для оптимизации и уменьшения времени счета была использована технология OpenMP. Средствами библиотеки функций OpenMP с использованием специальных директив компилятора проведено автоматическое распараллеливание циклов. Интервал размера L переменной цикла i разбивался на m частей. Каждый поток процесса вычислял свою p -ю часть данных, где $p = L/m$ (рис. 3).

Для уменьшения времени решения задачи также использовалась технология NVIDIA CUDA. Существенное отличие выполнения программы с использованием CUDA от выполнения программы с помощью OpenMP состоит в том, что программа на CUDA выполняется “в тысячи потоков”. Такие программы используют массивно-параллельный принцип программирования. При разработке алгоритмов были перенесены на GPU только “ресурсоемкие” векторно-матричные операции. Принцип работы состоял в следующем. Основной поток программы выполнялся на CPU, данные из оперативной памяти загружались на GPU, где проводились расчеты. Далее результаты расчетов выгружались в оперативную память. Очень часто распараллеливание на GPU не является эффективным ввиду образования “узкого горлышка” пропускной способности памяти.

По времени счета наиболее быстрым является метод ПАКК. Время решения СЛАУ 76136×76136 на 4-ядерном процессоре Intel и видеокарте составляет несколько секунд.

Результаты вычислений показывают, что использование параллельных методов ПАМП, ПАКК и ПМСГ с предобуславливателем позволяет достаточно быстро решать задачи с плохо обусловленными матрицами на многопроцессорных вычислителях и можно рекомендовать данные методы для решения задач электроразведки.

6. Заключение

Для решения линейной обратной задачи гравиметрии о нахождении плотности в слое итерационными методами градиентного типа и нелинейной обратной задачи гравиметрии о восстановлении поверхности раздела между средами с помощью итеративно регуляризованного метода Ньютона, а также для решения СЛАУ с блочно-трехдиагональными матрицами применительно к задачам электроразведки построены параллельные прямые и итерационные алгоритмы. Алгоритмы реализованы на многопроцессорных вычислительных системах различного типа: многопроцессорном комплексе МВС–ИММ, графических процессорах NVIDIA и многоядерном процессоре Intel. Параллельные алгоритмы встроены в разработанную систему удаленных вычислений “Специализированный Веб-портал решения задач на многопроцессорных вычислителях”. Решены задачи с квазимодельными и реальными данными.

Результаты вычислений показывают, что использование метода Ньютона и параллельных итерационных методов градиентного типа при решении обратных задач гравиметрии на многопроцессорных системах позволяет эффективно получать корректные решения. Использование параллельных методов матричной прогонки, квадратного корня и сопряженных градиентов с предобуславливателем позволяет с высокой скоростью решать СЛАУ с плохо обусловленными блочно-трехдиагональными матрицами применительно к задачам электроразведки на многопроцессорных вычислителях.

Литература

1. **Мартышко П.С., Кокшаров Д.Е.** Об определении плотности в слоистой среде по гравитационным данным // Геофизический журнал. — 2005. — Т. 27, № 4. — С. 678–684.
2. **Нумеров Б.В.** Интерпретация гравитационных наблюдений в случае одной контактной поверхности // ДАН СССР. — 1930. — № 21. — С. 569–574.
3. **Васин В.В., Еремин И.И.** Операторы и итерационные процессы фейеровского типа. Теория и приложения. — Екатеринбург: Изд-во УрО РАН, 2005.
4. **Дашевский Ю.А., Суродина И.В., Эпов М.И.** Квазитрехмерное математическое моделирование диаграмм неосесимметричных зондов постоянного тока в анизотропных разрезах // Сиб. журн. индустриальной математики — 2002. — Т. 5, № 3 (11). — С. 76–91.
5. **Мартышко П.С., Пруткин И.Л.** Технология разделения источников гравитационного поля по глубине // Геофизический журнал. — 2003. — Т. 25, № 3. — С. 159–168.
6. **Воеводин Вл.В.** Технологии параллельного программирования. — URL: <http://parallel.ru/> (дата обращения: 06.02.2012).
7. **Берилло А.** NVIDIA CUDA — неграфические вычисления на графических процессорах. — URL: <http://www.ixbt.com/video3/cuda-1.shtml> (дата обращения: 06.02.2012).
8. **Акимова Е.Н., Белоусов Д.В.** Распараллеливание алгоритмов решения линейной обратной задачи гравиметрии на МВС–1000 и графических процессорах // Вестник ННГУ Ч. 1. — 2010. — № 5. — С. 193–200.
9. **Vakushinsky A., Goncharky A.** Ill-Posed Problems: Theory and Applications. — London: Kluwer Akad. Publ., 1994.
10. **Фаддеев В.К., Фаддеева В.Н.** Вычислительные методы линейной алгебры. — М.: Гос. изд-во физ.-мат. литературы, 1963.
11. **Акимова Е.Н.** Распараллеливание алгоритма матричной прогонки // Математическое моделирование. — 1994. — Т. 6, № 9. — С. 61–67.
12. **Самарский А.А., Николаев Е.С.** Методы решения сеточных уравнений. — М.: Наука, 1978.
13. **Акимова Е.Н., Гемайдинов Д.В.** Параллельные алгоритмы решения обратной задачи гравиметрии и организация удаленного взаимодействия между МВС–1000 и пользователем // Вычислительные методы и программирование. — 2008. — Т. 9, № 1. — С. 133–144.
14. **Мартышко П.С., Васин В.В., Акимова Е.Н., Пьянков В.А.** О комплексной интерпретации гравитационных и магнитовариационных данных (на примере Башкирского Предураля) // Геофизика. — 2011. — № 4. — С. 30–36.
15. Методика разработки многопоточных приложений: принципы и практическая реализация. — URL: <http://www.rsdn.ru/article/baseserv/RUThreadingMethodology.xml> (дата обращения: 06.02.2012).

*Поступила в редакцию 20 февраля 2012 г.,
в окончательном варианте 9 апреля 2012 г.*

