

УДК 519.6

ПРИМЕНЕНИЕ МНОГОПРОЦЕССОРНЫХ СИСТЕМ ДЛЯ РЕШЕНИЯ ТРЕХМЕРНЫХ ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ ФРЕДГОЛЬМА ПЕРВОГО РОДА ДЛЯ ВЕКТОРНЫХ ФУНКЦИЙ

Д. В. Лукьяненко¹, А. Г. Ягола¹

Рассматриваются особенности численной реализации решения трехмерных интегральных уравнений Фредгольма 1-го рода для векторных функций с применением многопроцессорных систем. Для решения этой некорректной задачи применяется алгоритм, основанный на минимизации функционала Тихонова. В качестве метода минимизации используется метод сопряженных градиентов. Выбор параметра регуляризации осуществляется в соответствии с обобщенным принципом невязки. Предлагается схема распараллеливания задачи, показывается эффективность данного подхода на примере задачи восстановления параметров намагничности. Работа выполнена при поддержке РФФИ (коды проектов 08-01-00160-а и 10-01-91150-ГФЕН_а). Тестовые расчеты выполнялись с использованием ресурсов суперкомпьютерного комплекса Московского государственного университета.

Ключевые слова: трехмерные интегральные уравнения Фредгольма 1-го рода, метод сопряженных градиентов, метод регуляризации Тихонова, параллельные алгоритмы.

1. Введение. При решении многих современных прикладных задач часто возникает необходимость восстанавливать характеристики исследуемых объектов в пространстве, при этом эти характеристики могут являться векторными функциями. Это приводит к необходимости решать трехмерные интегральные уравнения Фредгольма 1-го рода, что зачастую невозможно сделать с использованием обычных компьютеров. В таких случаях обычно используются различные упрощения и допущения, которые понижают размерность решаемой задачи, но при этом приводят к существенным ошибкам в восстанавливаемых значениях либо дают ограниченную информацию об исследуемом объекте. Предлагается решать такие многомерные задачи в самой общей постановке, в которой необходимо решить трехмерное интегральное уравнение Фредгольма 1-го рода для векторной функции, с использованием многопроцессорных систем [2].

В настоящей статье будут продемонстрированы эффективные подходы к решению таких задач, которые можно реализовать как на многопроцессорных кластерных системах, так и на современных персональных компьютерах с многоядерными процессорами. При разработке этих методов активно использовались идеи применения многопроцессорных систем для решения двумерных интегральных уравнений Фредгольма 1-го рода, заложенные в работах [3] и [4]. Эффективность данного подхода при решении рассматриваемого класса задач продемонстрирована на примере задачи восстановления параметров намагничности. Все результаты данной работы могут быть легко упрощены на случаи задач меньшей размерности либо задач, при решении которых необходимо восстанавливать скалярную функцию.

2. Постановка задачи и метод решения. Рассмотрим следующее трехмерное интегральное уравнение Фредгольма 1-го рода для векторной функции:

$$AM = \int_{L_x}^{R_x} \int_{L_y}^{R_y} \int_{L_z}^{R_z} K(s, t, r, x, y, z) M(x, y, z) dx dy dz = B(s, t, r). \quad (1)$$

Здесь $B(s, t, r)$ и $M(x, y, z)$ — векторные функции, а ядро $K(s, t, r, x, y, z)$ — матричная функция:

$$B(s, t, r) = \begin{pmatrix} B^1(s, t, r) \\ B^2(s, t, r) \\ B^3(s, t, r) \end{pmatrix}, \quad M(x, y, z) = \begin{pmatrix} M^1(x, y, z) \\ M^2(x, y, z) \\ M^3(x, y, z) \end{pmatrix}, \quad K = \begin{pmatrix} K^{11} & K^{12} & K^{13} \\ K^{21} & K^{22} & K^{23} \\ K^{31} & K^{32} & K^{33} \end{pmatrix}.$$

¹ Московский государственный университет им. М. В. Ломоносова, физический факультет, Ленинские горы, 119992, Москва; Д. В. Лукьяненко, инженер, e-mail: lukyanenko@physics.msu.ru; А. Г. Ягола, профессор, e-mail: yagola@physics.msu.ru

Обозначим

$$P = \{(x, y, z) : L_x \leq x \leq R_x, L_y \leq y \leq R_y, L_z \leq z \leq R_z\},$$

$$Q = \{(s, t, r) : L_s \leq s \leq R_s, L_t \leq t \leq R_t, L_r \leq r \leq R_r\},$$

где P — область определения векторной функции $\mathbf{M}(x, y, z)$, а Q — область определения векторной функции $\mathbf{B}(s, t, r)$. Будем предполагать, что $\mathbf{M} \in W_2^2(P)$ и $\mathbf{B} \in L_2(Q)$, а оператор A с ядром K непрерывен и однозначен. Нормы правой части уравнения (1) и решения вводятся следующим образом:

$$\|\mathbf{B}\|_{L_2} = \sqrt{\|B^1\|_{L_2}^2 + \|B^2\|_{L_2}^2 + \|B^3\|_{L_2}^2}, \quad \|\mathbf{M}\|_{W_2^2} = \sqrt{\|M^1\|_{W_2^2}^2 + \|M^2\|_{W_2^2}^2 + \|M^3\|_{W_2^2}^2}.$$

Пусть вместо точно известных $\overline{\mathbf{B}}$ и оператора A известны их приближенные значения \mathbf{B}_δ и A_h , такие, что $\|\mathbf{B}_\delta - \overline{\mathbf{B}}\|_{L_2} \leq \delta$, $\|A - A_h\|_{W_2^2 \rightarrow L_2} \leq h$. При выписанных условиях задача является некорректной, для ее решения необходимо построить регуляризирующий алгоритм. Воспользуемся алгоритмом, основанным на минимизации функционала Тихонова $F^\alpha[\mathbf{M}] = \|A_h \mathbf{M} - \mathbf{B}_\delta\|_{L_2}^2 + \alpha \|\mathbf{M}\|_{W_2^2}^2$, который в нашем случае примет вид

$$F^\alpha[\mathbf{M}] = \int_{L_s}^{R_s} \int_{L_t}^{R_t} \int_{L_r}^{R_r} ds dt dr \left\{ \int_{L_x}^{R_x} \int_{L_y}^{R_y} \int_{L_z}^{R_z} K(s, t, r, x, y, z) \mathbf{M}(x, y, z) dx dy dz - \mathbf{B}(s, t, r) \right\}^2 + \alpha \Omega[\mathbf{M}], \quad (2)$$

где $\Omega[\mathbf{M}] \equiv \|\mathbf{M}\|_{W_2^2}^2$ — сглаживающий функционал:

$$\Omega[\mathbf{M}] = \int_{L_x}^{R_x} \int_{L_y}^{R_y} \int_{L_z}^{R_z} \left\{ M^2 + \left(\frac{\partial^2 M}{\partial x^2} \right)^2 + \left(\frac{\partial^2 M}{\partial y^2} \right)^2 + \left(\frac{\partial^2 M}{\partial z^2} \right)^2 \right\} dx dy dz.$$

Здесь квадрат частной производной вектор-функции понимается в смысле суммы квадратов частных производных компонент вектор-функции.

Для любого $\alpha > 0$ существует единственная экстремаль функционала Тихонова \mathbf{M}_η^α , $\eta = \{\delta, h\}$, реализующая минимум $F^\alpha[\mathbf{M}]$. Для выбора параметра регуляризации можно использовать алгоритм конечномерного обобщенного принципа невязки [1]. При выборе параметра $\alpha = \alpha(\eta)$ по обобщенному принципу невязки: $\rho(\alpha) = \|A_h \mathbf{M}_\eta^\alpha - \mathbf{B}_\delta\|_{L_2}^2 - (\delta + h \|\mathbf{M}_\eta^\alpha\|_{W_2^2})^2 = 0$, \mathbf{M}_η^α стремится при $\eta \rightarrow 0$ к точному решению задачи в норме W_2^2 , а следовательно, и равномерно на P . В качестве метода минимизации функционала Тихонова применяется метод сопряженных градиентов.

3. Структура алгоритма. При решении задачи минимизации методом сопряженных градиентов необходимо вычислять значение функционала Тихонова $F^\alpha[\mathbf{M}]$ и его градиента $\text{grad } F^\alpha[\mathbf{M}]$. Для численного решения задачи переходим к конечномерным пространствам. Введем сетки по x, y, z, s, t, r с шагами $h_x, h_y, h_z, h_s, h_t, h_r$ и числом узлов $N_x, N_y, N_z, N_s, N_t, N_r$ соответственно:

$$x_{i_1} = L_x + (i_1 - 1)h_x, \quad i_1 = \overline{1, N_x}, \quad h_x = \frac{R_x - L_x}{N_x - 1},$$

$$y_{i_2} = L_y + (i_2 - 1)h_y, \quad i_2 = \overline{1, N_y}, \quad h_y = \frac{R_y - L_y}{N_y - 1},$$

$$z_{i_3} = L_z + (i_3 - 1)h_z, \quad i_3 = \overline{1, N_z}, \quad h_z = \frac{R_z - L_z}{N_z - 1},$$

$$s_{j_1} = L_s + (j_1 - 1)h_s, \quad j_1 = \overline{1, N_s}, \quad h_s = \frac{R_s - L_s}{N_s - 1},$$

$$t_{j_2} = L_t + (j_2 - 1)h_t, \quad j_2 = \overline{1, N_t}, \quad h_t = \frac{R_t - L_t}{N_t - 1},$$

$$r_{j_3} = L_r + (j_3 - 1)h_r, \quad j_3 = \overline{1, N_r}, \quad h_r = \frac{R_r - L_r}{N_r - 1}.$$

Примем, что для $n = \overline{1, 3}$ и $m = \overline{1, 3}$

$$M_{i_1 i_2 i_3}^m = M^m(x_{i_1}, y_{i_2}, z_{i_3}), \quad B_{j_1 j_2 j_3}^n = B^n(s_{j_1}, t_{j_2}, r_{j_3}), \quad K_{j_1 j_2 j_3 i_1 i_2 i_3}^{nm} = K^{nm}(s_{j_1}, t_{j_2}, r_{j_3}, x_{i_1}, y_{i_2}, z_{i_3}).$$

Все интегралы в формуле (2) аппроксимируются по формуле прямоугольников. Таким образом, получаем следующую конечно-разностную аппроксимацию функционала Тихонова (2):

$$F^\alpha[M] = \Phi[M] + \alpha\Omega[M], \quad (3)$$

где

$$\Phi[M] = \sum_{j_1=1}^{N_s} \sum_{j_2=1}^{N_t} \sum_{j_3=1}^{N_r} \sum_{n=1}^3 h_s h_t h_r \left[\sum_{i_1=1}^{N_x} \sum_{i_2=1}^{N_y} \sum_{i_3=1}^{N_z} \sum_{m=1}^3 h_x h_y h_z K_{j_1 j_2 j_3 i_1 i_2 i_3}^{nm} M_{i_1 i_2 i_3}^m - B_{j_1 j_2 j_3}^n \right]^2, \quad (4)$$

$$\begin{aligned} \Omega[M] = & h_x h_y h_z \sum_{i_1=1}^{N_x} \sum_{i_2=1}^{N_y} \sum_{i_3=1}^{N_z} \sum_{m=1}^3 (M_{i_1 i_2 i_3}^m)^2 + \\ & + \frac{h_y h_z}{h_x^3} \sum_{i_1=2}^{N_x-1} \sum_{i_2=1}^{N_y} \sum_{i_3=1}^{N_z} \sum_{m=1}^3 (M_{i_1+1 i_2 i_3}^m - 2M_{i_1 i_2 i_3}^m + M_{i_1-1 i_2 i_3}^m)^2 + \\ & + \frac{h_x h_z}{h_y^3} \sum_{i_1=1}^{N_x} \sum_{i_2=2}^{N_y-1} \sum_{i_3=1}^{N_z} \sum_{m=1}^3 (M_{i_1 i_2+1 i_3}^m - 2M_{i_1 i_2 i_3}^m + M_{i_1 i_2-1 i_3}^m)^2 + \\ & + \frac{h_x h_y}{h_z^3} \sum_{i_1=1}^{N_x} \sum_{i_2=1}^{N_y} \sum_{i_3=2}^{N_z-1} \sum_{m=1}^3 (M_{i_1 i_2 i_3+1}^m - 2M_{i_1 i_2 i_3}^m + M_{i_1 i_2 i_3-1}^m)^2. \end{aligned} \quad (5)$$

Таким образом, исходная задача сводится к задаче минимизации в N -мерном пространстве, где $N = 3 \times N_x \times N_y \times N_z$, с последующим выбором параметра регуляризации по обобщенному принципу невязки.

Итерационный процесс минимизации функционала строится следующим образом: задается некоторое начальное приближение — вектор $\mathbf{M}^{(0)}$. Для заданной точки $\mathbf{M}^{(i)}$ минимизирующей последовательности вычисляется градиент $\mathbf{g}^{(i)} = \text{grad } F^\alpha[\mathbf{M}^{(i)}]$, который легко получить из формул (4) и (5):

$$\begin{aligned} \left(\text{grad } F^\alpha[\mathbf{M}] \right)_{i_1 i_2 i_3}^m &= \frac{\partial F^\alpha[M_{i_1 i_2 i_3}^m]}{\partial M_{i_1 i_2 i_3}^m} = 2h_x h_y h_z \sum_{j_1=1}^{N_s} \sum_{j_2=1}^{N_t} \sum_{j_3=1}^{N_r} \sum_{n=1}^3 h_s h_t h_r K_{j_1 j_2 j_3 i_1 i_2 i_3}^{nm} \times \\ &\times \left[\sum_{i_1=1}^{N_x} \sum_{i_2=1}^{N_y} \sum_{i_3=1}^{N_z} \sum_{m=1}^3 h_x h_y h_z K_{j_1 j_2 j_3 i_1 i_2 i_3}^{nm} M_{i_1 i_2 i_3}^m - B_{j_1 j_2 j_3}^n \right] + \alpha \frac{\partial \Omega[M_{i_1 i_2 i_3}^m]}{\partial M_{i_1 i_2 i_3}^m}, \quad (6) \\ \frac{\partial \Omega[M_{i_1 i_2 i_3}^m]}{\partial M_{i_1 i_2 i_3}^m} &= 2h_x h_y h_z M_{i_1 i_2 i_3}^m + 2 \frac{h_y h_z}{h_x^3} \Omega'_1 + 2 \frac{h_x h_z}{h_y^3} \Omega'_2 + 2 \frac{h_x h_y}{h_z^3} \Omega'_3, \\ \Omega'_1 &= \begin{cases} -2M_{i_1+1 i_2 i_3}^m + M_{i_1 i_2 i_3}^m + M_{i_1+2 i_2 i_3}^m, & i_1 = 1, \\ -4M_{i_1+1 i_2 i_3}^m + 5M_{i_1 i_2 i_3}^m - 2M_{i_1-1 i_2 i_3}^m + M_{i_1+2 i_2 i_3}^m, & i_1 = 2, \\ -4M_{i_1+1 i_2 i_3}^m + 6M_{i_1 i_2 i_3}^m - 4M_{i_1-1 i_2 i_3}^m + M_{i_1-2 i_2 i_3}^m + M_{i_1+2 i_2 i_3}^m, & i_1 = \overline{3, N_x-2}, \\ -2M_{i_1+1 i_2 i_3}^m + 5M_{i_1 i_2 i_3}^m - 4M_{i_1-1 i_2 i_3}^m + M_{i_1-2 i_2 i_3}^m, & i_1 = N_x - 1, \\ -2M_{i_1-1 i_2 i_3}^m + M_{i_1 i_2 i_3}^m + M_{i_1-2 i_2 i_3}^m, & i_1 = N_x. \end{cases} \end{aligned}$$

Выражения для Ω'_2 и Ω'_3 имеют вид, аналогичный Ω'_1 , с учетом соответствующих замен выражений в коэффициентах i_2 и i_3 на выражения для i_1 .

Далее по направлению градиента $\mathbf{g}^{(i-1)}$ и направлению спуска $\mathbf{h}^{(i-1)}$ на предыдущем шаге вычисляется новое направление спуска по формуле $\mathbf{h}^{(i)} = \mathbf{g}^{(i)} + \gamma_i \mathbf{h}^{(i-1)}$, $\gamma_i = \frac{(\mathbf{g}^{(i)} - \mathbf{g}^{(i-1)}, \mathbf{g}^{(i)})}{\mathbf{g}^{(i-1)}, \mathbf{g}^{(i-1)}}$ (вариант

Полака–Рибьера метода сопряженных градиентов). После этого строится однопараметрическое множество, состоящее из элементов $\mathbf{M}_\lambda = \mathbf{M}^{(i)} + \lambda \mathbf{h}^{(i)}$. Затем решается задача одномерной минимизации функционала $F^\alpha[\mathbf{M}_\lambda]$. Точка минимума функционала $F^\alpha[\mathbf{M}_\lambda]$ на рассматриваемом множестве принимается за следующий элемент $\mathbf{M}^{(i+1)}$ минимизирующей последовательности. При решении одномерной задачи минимизации используется квадратичная аппроксимация функции [4] $\sigma(\lambda) \equiv F^\alpha[\mathbf{M}^{(i)} + \lambda \mathbf{h}^{(i)}] - F^\alpha[\mathbf{M}_\lambda^{(i)}]$

по трем точкам: $0, \lambda_{\text{step}}, 2\lambda_{\text{step}}$. После этого приближенное решение одномерной задачи минимизации берется в виде $\lambda_{\text{min}} = \lambda_{\text{step}} \frac{1}{2} \frac{\sigma_2 - 4\sigma_1}{\sigma_2 - 2\sigma_1}$, $\sigma_1 \equiv \sigma(\lambda_{\text{step}})$, $\sigma_2 \equiv \sigma(2\lambda_{\text{step}})$. При необходимости производится дополнительное уточнение этой точки (этот вопрос подробно рассмотрен в работе [4]). За N шагов методом сопряженных градиентов находится минимум функционала Тихонова (2) для фиксированного параметра регуляризации α . Параметр регуляризации α выбирается по обобщенному принципу невязки (соответствующие численные методы выбора параметра регуляризации по обобщенному принципу невязки подробно описаны в работе [1]).

4. Распараллеливание задачи минимизации. При решении задачи минимизации методом сопряженных градиентов основные вычисления приходятся на вычисление функционала Тихонова $F^\alpha[M]$ и его градиента $\text{grad } F^\alpha[M]$. Из формул (3)–(6) для функционала и его градиента видно, что они состоят из не связанных между собой групп слагаемых (по j_1, j_2, j_3 и i_1, i_2, i_3 соответственно). Это дает возможность применения многопроцессорной системы. Задачу можно распараллелить, т.е. переписать программу таким образом, чтобы независимые части программы выполнялись на разных процессорах. Изложенные в этом разделе подходы основаны на результатах, которые были представлены в работах [3] и [4] для двумерных задач.

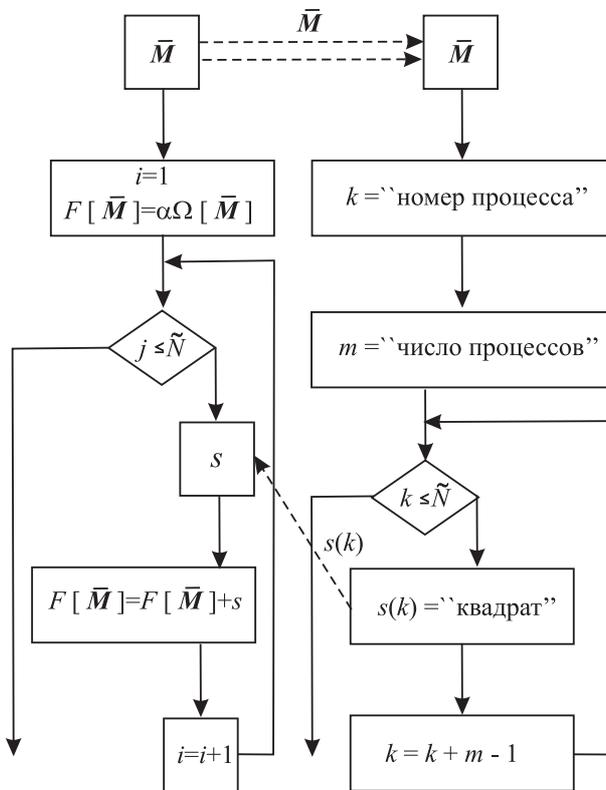


Рис. 1. Схема вычисления функционала для нулевого процесса (слева) и для ненулевого процесса (справа)

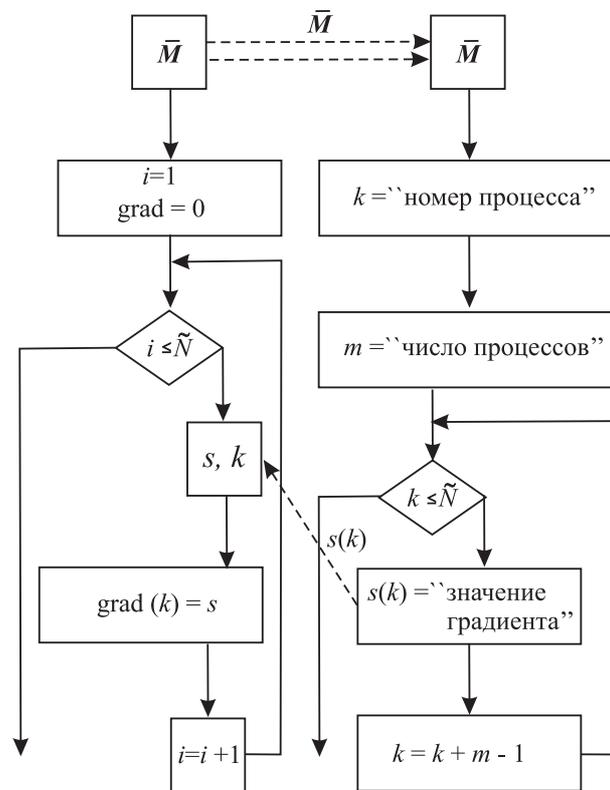


Рис. 2. Схема вычисления градиента функционала: для нулевого процесса (слева); и для ненулевого процесса (справа)

При решении модельных задач использовался суперкомпьютер СКИФ МГУ “Чебышев” Научно-исследовательского вычислительного центра МГУ (подробную информацию о кластере можно найти в [5]). При написании программ для решения модельных задач на многопроцессорных системах использовалась библиотека MPI (Message Passing Interface). В качестве языка программирования был выбран Fortran 90.

Исходная задача решалась с использованием N параллельных процессов $0, 1, \dots, N - 1$, выполняющихся на отдельных процессорах и при необходимости взаимодействующих друг с другом. Нулевой процесс выполнял нераспараллеливаемые действия: чтение исходных данных из файла, запись результатов, сбор результатов вычислений остальных процессов и перераспределение между ними новых задач. Распараллеливание происходило только при минимизации функционала: до и после минимизации ненулевые процессы не задействованы. При распараллеливании нулевой процесс рассматривается только как

процесс, передающий и получающий информацию от других процессов.

Рассмотрим подробнее задачу вычисления значения функционала $F^\alpha[\mathbf{M}]$ в некоторой точке \mathbf{M} (рис. 1) [4]. Сначала нулевой процесс передает всем остальным процессам вектор \mathbf{M} . После этого каждый ненулевой процесс вычисляет свой квадрат выражения для (4) для заданных переменных j_1, j_2, j_3 . Все многомерные массивы можно записать как одномерные, поэтому в схеме переменные j_1, j_2, j_3 заменяются переменной k , которая изменяется от 1 до $\tilde{N} = 3 \times N_s \times N_t \times N_r$. Каждый процесс вычисляет значения для своих значений k . После этого это значение ($s(k)$ в схеме) передается нулевому процессу, где происходит суммирование со значениями, переданными от других процессов. Основной особенностью вычисления функционала является тот факт, что при повторных вычислениях могут получаться различные значения функционала. Это связано с тем, что переменные s могут поступать в разном порядке от разных процессов, в связи с чем округление суммирования каждый раз будет давать различный результат.

Алгоритм распараллеливания задачи вычисления градиента функционала $\text{grad } F^\alpha[\mathbf{M}]$ (рис. 2) [4] аналогичен схеме вычисления функционала $F^\alpha[\mathbf{M}]$. Но есть и отличие: каждый ненулевой процесс вычисляет k -й элемент вектора градиента $\text{grad } F^\alpha[\mathbf{M}]$ (массив grad в схеме) и передает нулевому процессу вычисленный элемент s вместе с номером k этого элемента в массиве. Таким образом, при формировании массива grad не важно, в какой последовательности нулевой процесс получает элементы s , так как числа k точно определяют их позиции в массиве grad . Кроме того, важным отличием вычисления градиента от вычисления функционала является то, что вычисления градиента сглаживающего функционала $\Omega[\mathbf{M}]$ распараллелены.

Очень важным вопросом является эффективность алгоритмов распараллеливания. Ограничения на эффективность этих алгоритмов накладывает закон Амдала. Согласно закону Амдала, небольшая часть алгоритма, которая не может быть распараллелена, ограничивает ускорение вычислений, которое может быть достигнуто за счет распараллеливания. Любая практическая задача содержит распараллеливаемые вычисления и нераспараллеливаемые (последовательные). Закон Амдала устанавливает следующие ограничения на ускорение работы алгоритма, которые дает распараллеливание: $S = \frac{1}{(1-P) + P/(N-1)}$, где S — ускорение работы программы (относительно времени работы последовательной версии программы), N — число используемых при вычислениях процессов, P — часть вычислений, которые можно распараллелить. Таким образом, если, например, часть нераспараллеливаемых действий равна 10% от общего количества вычислений, то невозможно получить ускорения работы программы более чем в 10 раз вне зависимости от числа процессов N .

При решении нашей задачи распараллеливание осуществлялось только при вычислении функционала Тихонова $F^\alpha[\mathbf{M}]$ и его градиента $\text{grad } F^\alpha[\mathbf{M}]$. Ясно, что все остальные вычисления производились последовательно. Однако так как время работы данного последовательного кода пренебрежимо мало при больших размерностях сетки, то время, затрачиваемое на все эти вычисления, можно считать равным нулю. Остается открытым вопрос о том, как влияют на эффективность распараллеливания последовательные вычисления сглаживающего функционала $\Omega[\mathbf{M}]$ при вычислении функционала Тихонова.

Из формулы (4) для конечно-разностной аппроксимации функционала Тихонова видно, что невязка состоит из $3 \times N_s \times N_t \times N_r$ независимо вычисляемых групп слагаемых, а сглаживающий функционал состоит из $3 \times (\leq 4)$ эквивалентных (в смысле затрачиваемого на вычисления время) групп слагаемых. Соответственно для вычисления функционала Тихонова часть распараллеливаемых действий оценивается как

$$P \geq \frac{3N_s N_t N_r}{3N_s N_t N_r + 12} = \frac{N_s N_t N_r}{N_s N_t N_r + 4},$$

откуда видно, что даже при достаточно небольшом количестве входных данных, например для $N_s = N_t = N_r = 10$ (напомним, что эти сетки соответствуют области определения известной векторной функции \mathbf{B}), распараллеливаемая часть вычислений составляет более 97%. Современные прикладные задачи требуют обработки гораздо большего числа входных данных, в связи с чем доля распараллеливаемых вычислений стремится к 100%, что доказывает очень высокую эффективность рассматриваемых алгоритмов распараллеливания.

Следует также заметить, что рассмотренный параллельный алгоритм имеет следующую особенность: каждый процесс вычисляет достаточно объемные (в смысле затрачиваемого на вычисления времени) группы слагаемых, в связи с чем время обмена сообщениями между процессами пренебрежимо мало по сравнению с временем работы каждого процесса и его можно не учитывать при оценке эффективности алгоритма. По этой самой причине вычисления сглаживающего функционала $\Omega[\mathbf{M}]$ не распараллеливаются: как видно из формулы (5), при распараллеливании каждый процесс вынужден был бы производить объем действий, время вычисления которых было бы сопоставимо (и даже было бы меньше с увеличением

числа процессов) с временем передачи сообщений между процессами, что привело бы к значительному замедлению работы программы.

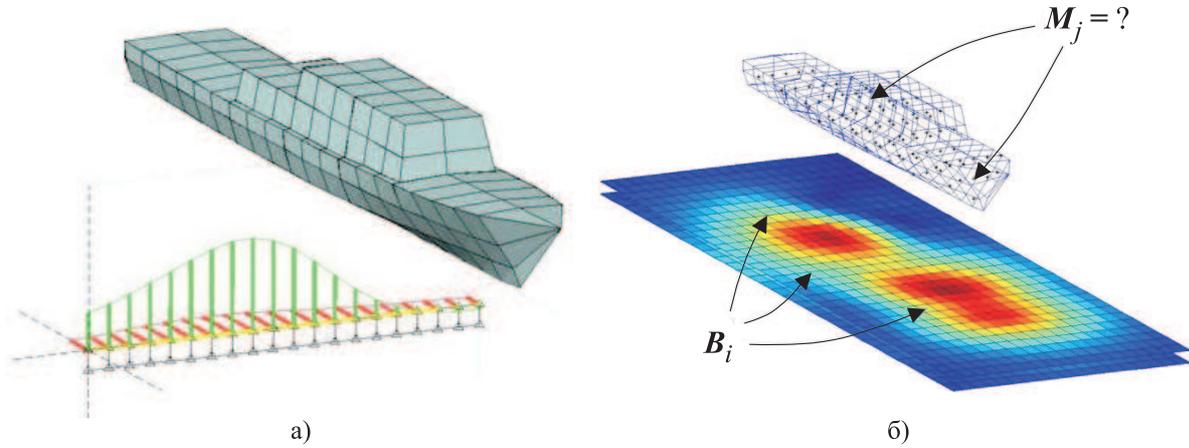


Рис. 3. Корабль проходит над массивом триаксиальных сенсоров, которые измеряют значения магнитного поля, индуцируемого магнитными диполями, распределенными по объему корабля (а). Эквивалентная постановка задачи: статичный корабль над системой плоскостей, на которых измерены значения магнитного поля (б)

5. Пример решения прикладной задачи восстановления параметров намагниченности некоторого объекта по измеренным значениям магнитного поля вне этого объекта. Рассмотрим важную прикладную задачу восстановления пространственного распределения магнитных диполей M_j по измеренным значениям магнитного поля B_i вне исследуемого объекта. Такой тип задач возникает, например, в случае, когда необходимо определить области намагниченности корабля по значениям магнитного поля, измеренным триаксиальными сенсорами, расположенными на дне залива (рис. 3). Уравнение, описывающее магнитное поле B_i , индуцируемое магнитными диполями M_j , имеет вид

$$B_i(x_s, y_s, z_s) = \sum_{j=1}^N \frac{\mu_0}{4\pi} \left[\frac{3(M_j \cdot r_{ij})r_{ij}}{r_{ij}^5} - \frac{M_j}{r_{ij}^3} \right], \quad (7)$$

где x_s, y_s, z_s — координаты точек (соответствуют расположению триаксиальных сенсоров), в которых определена векторная функция B , r_{ij} — расстояние между точкой (x_s, y_s, z_s) и точкой (x, y, z) , в которой расположен магнитный диполь с номером j , μ_0 — величина магнитной проницаемости в вакууме, N — число магнитных диполей.

Уравнение (7) может быть описано эквивалентным ему интегральным уравнением Фредгольма 1-го рода для векторной функции:

$$B_i(x_s, y_s, z_s) = \int_V K(x - x_s, y - y_s, z - z_s) M(x, y, z) dv. \quad (8)$$

В этом случае функция B является векторной функцией, определенной на системе измерительных сенсорных плоскостей и неизвестная функция M также векторная функция, определенная по объему корабля V . Здесь x_s, y_s, z_s — координаты точек, расположенных внутри объема корабля V . Ядро K интегрального уравнения (8) соответствует уравнению (7) и может быть записано в виде

$$K(x - x_s, y - y_s, z - z_s) = \frac{\mu_0}{4\pi r^5} \begin{bmatrix} 3(x - x_s)^2 - r^2 & 3(x - x_s)(y - y_s) & 3(x - x_s)(z - z_s) \\ 3(y - y_s)(x - x_s) & 3(y - y_s)^2 - r^2 & 3(y - y_s)(z - z_s) \\ 3(z - z_s)(x - x_s) & 3(z - z_s)(y - y_s) & 3(z - z_s)^2 - r^2 \end{bmatrix},$$

где $r = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2}$. Если мы предположим, что

$$V \subset P = \{(x, y, z) : L_x \leq x \leq R_x, L_y \leq y \leq R_y, L_z \leq z \leq R_z\}$$

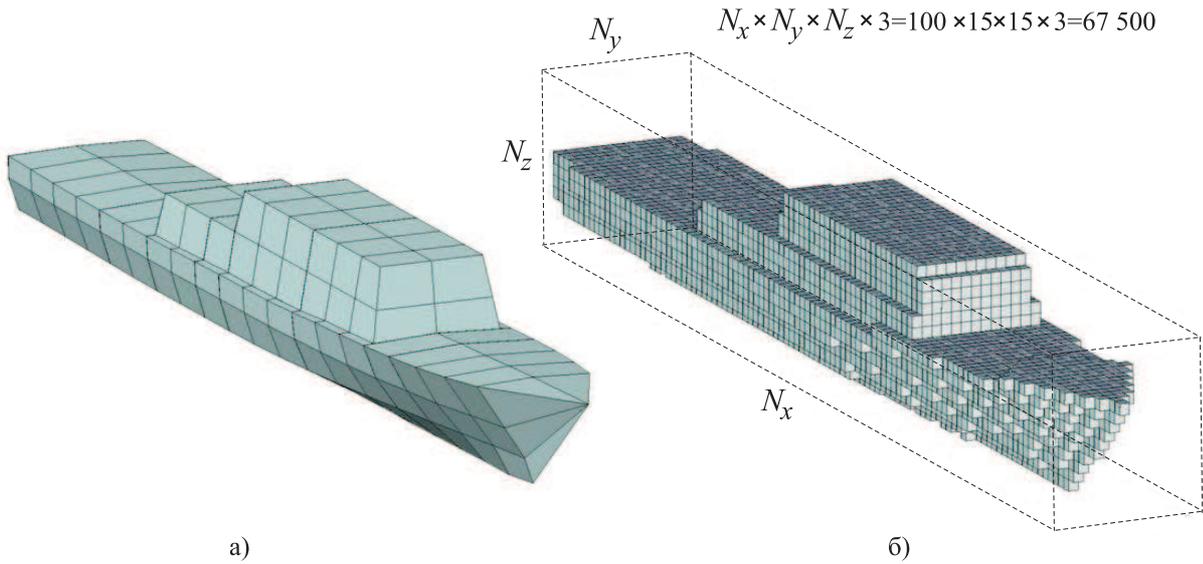


Рис. 4. Модель корабля (а), аппроксимация корпуса корабля прямоугольными параллелепипедами (б)

и система сенсорных плоскостей ограничена прямоугольным параллелепипедом

$$Q = \{(x_s, y_s, z_s) \equiv (s, t, r) : L_s \leq s \leq R_s, L_t \leq t \leq R_t, L_r \leq r \leq R_r\},$$

то мы получаем трехмерное интегральное уравнение Фредгольма 1-го рода для векторной функции:

$$\mathbf{B}(s, t, r) = \int_{L_x}^{R_x} \int_{L_y}^{R_y} \int_{L_z}^{R_z} K(x - s, y - t, z - r) \mathbf{M}(x, y, z) dx dy dz,$$

которое в точности соответствует уравнению (1), методы решения которого уже нами разработаны.

Типичные размерности, соответствующие реальным приложениям: $N_x = 100$, $N_y = 15$, $N_z = 15$ (рис. 4). Входные данные симулировали данные реального эксперимента, когда измерения производились шестью триаксиальными сенсорами (расположены друг над другом в два ряда по три сенсора), и каждым из них было сделано 4000 измерений, что соответствует $N_s = 4000$, $N_t = 3$, $N_r = 2$. Эти размерности сеток соответствуют задаче восстановления 67500 неизвестных по 72000 входных данным. В результате применения описанных в данной работе методов было восстановлено распределение областей намагниченности по объему корабля. На рис. 5 представлено несколько срезов модуля восстановленной векторной функции \mathbf{M} . Входные данные были заданы с ошибкой, равной 1.5%.

Из сказанного выше следует, что рассмотренные методы позволили достаточно точно восстановить очертания корпуса корабля (как часть корабля, которая наиболее сильно подвержена намагничиванию) и даже некоторую внутреннюю структуру (отчетливо можно различить двигательную установку).

Время вычислений примерно составило 29 часов при использовании 200 процессоров (Intel Xeon E5472 3.0 GHz). Столь длительное время вычислений связано с применением регуляризирующих алгоритмов, которые требуют повторных находжений минимума минимизируемого функционала для каждого значения параметра регуляризации α .

6. Выводы. Применение многопроцессорных систем при решении трехмерных интегральных уравнений Фредгольма 1-го рода для векторных функций дает возможность существенно сократить время

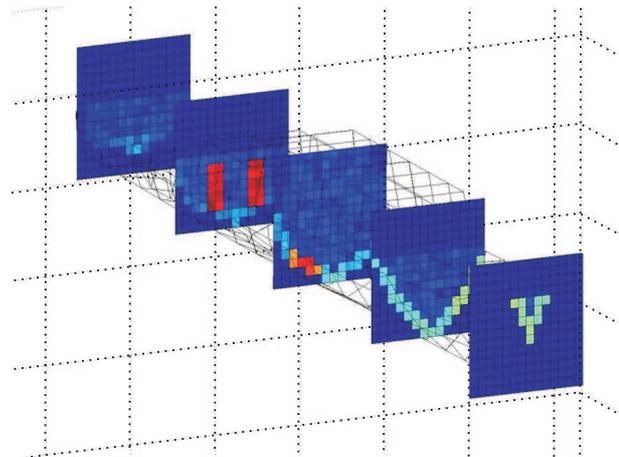


Рис. 5. Результаты восстановления распределения намагниченности по объему корабля (представлено 5 срезов модуля восстановленной векторной функции \mathbf{M})

работы программы, что позволяет эффективно решать современные многомерные задачи. Алгоритм метода сопряженных градиентов для минимизации функционала Тихонова является пригодным для распараллеливания, что обеспечивает эффективность рассматриваемого подхода. Данные методы могут быть эффективно применены и для решения задач меньшей размерности либо задач, при решении которых необходимо восстановить скалярную функцию.

СПИСОК ЛИТЕРАТУРЫ

1. *Тихонов А.Н., Гончарский А.В., Степанов В.В., Ягола А.Г.* Численные методы решения некорректных задач. М.: Наука, 1990.
2. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
3. *Васильев М.П., Ягола А.Г.* Применение многопроцессорных систем для решения двумерных интегральных уравнений Фредгольма 1-го рода // Вычислительные методы и программирование. 2003. **4**, № 2. 156–159.
4. *Ягола А.Г., Титаренко В.Н., Васильев М.П., Шимановская Е.В.* Особенности решения задач картирования распределения химических элементов по поверхностям звезд как некорректных задач с использованием многопроцессорных систем // Вычислительные методы и программирование. 2002. **3**, № 1. 5–17.
5. Вычислительный кластер НИВЦ МГУ (<http://parallel.ru/cluster>).

Поступила в редакцию
22.09.2010
