

УДК 519.6

ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ПРИ РЕШЕНИИ ОБРАТНОЙ ЗАДАЧИ ВОЛНОВОЙ ДИАГНОСТИКИ

С. Л. Овчинников¹, С. Ю. Романов¹

Рассматриваются методы и алгоритмы решения нелинейной трехмерной обратной задачи волновой диагностики неоднородностей в однородных средах. Обсуждаются подходы к выбору алгоритмов и организации параллельных вычислений в рассматриваемой задаче. Проведено сравнение эффективности нескольких алгоритмов решения, описан способ построения матрицы задачи в компактной форме, реализованы процедуры табулирования элементов матрицы на целочисленной сетке и сохранения промежуточных результатов расчетов. Реализованные подходы привели к ускорению алгоритмов, а также позволили решать задачу с 45 000 неизвестных, довести количество вычислительных узлов до 500 и, следовательно, проводить расчеты для более сложного класса задач. В частности, появилась возможность моделировать нелинейные эффекты переотражения волн.

Ключевые слова: параллельные вычисления, компьютерное моделирование, нелинейные волновые модели, обратные задачи, диагностика, уравнение Гельмгольца.

1. Введение. В работе рассмотрена обратная задача волнового зондирования неоднородных включений в однородной среде. Методы ее решения находят приложение, например, при исследовании приповерхностных слоев Земли для глубин от нескольких метров до десятков метров, диагностике дорожных покрытий и т.п. Эта задача исследуется ведущими учеными как в нашей стране, так и за рубежом [1–6].

Рассматриваемая задача является трехмерной и нелинейной, т.е. вычислительно трудоемкой. Количество неизвестных задачи растет примерно как $O(n^4)$, где n — число точек в одном направлении трехмерной сетки. Единственный путь решения этой задачи состоит в использовании суперкомпьютеров [10–13].

В работах [10–13] были рассмотрены методы решения прямой и обратной задачи для томографической диагностики среды в линейном и нелинейном приближении для кластерных систем. В настоящей статье проведено сравнение эффективности нескольких алгоритмов решения и выбраны наиболее перспективные подходы. Выполнено существенное усовершенствование параллельной структуры программ на суперкомпьютере семейства СКИФ НИВЦ МГУ.

Количество элементов матрицы решаемой системы уравнений растет примерно как $O(n^8)$. Используя разреженность и эрмитовость матрицы, удалось построить компактную форму хранения матрицы с асимптотикой количества элементов $O(n^7)$. Структура используемого алгоритма позволяет разбить матрицу на части и распределить их вычисление и хранение по отдельным узлам вычислительной системы. Основная часть объема итерационных вычислений также выполняется независимо на отдельных вычислительных узлах.

С учетом специфики ядра уравнения была реализована процедура вычисления элементов матрицы на целочисленной сетке, что позволило на 20–30% повысить скорость расчета матрицы.

Несмотря на высокое быстродействие кластерного суперкомпьютера, объем задачи требует обеспечения возможности прерывания и продолжения расчетов с прерванного места. В связи с этим была реализована процедура сохранения промежуточных результатов.

Ускорение алгоритмов решения обратной задачи позволило повысить размерность сетки вычислений, довести количество вычислительных узлов до 500 и, следовательно, проводить расчеты для более сложных классов задач. В частности, появилась возможность моделировать нелинейные эффекты переотражения излучаемой волны. На модельных примерах было показано, что эти эффекты являются существенно нелинейными.

2. Математическая постановка обратной задачи. Приведем основные формулы и результаты работы [11]. Рассмотрим задачу, описываемую скалярным волновым уравнением

$$\Delta u(r, q, \omega) + \kappa^2(r, \omega)u(r, q, \omega) = f(r, q, \omega), \quad (1)$$

¹ Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, 119991, Москва; e-mail: romanov@srcc.msu.ru

где $r \in \mathbb{R}^3$ — положение точки в трехмерном пространстве и Δ — оператор Лапласа по переменной r . Уравнение (1) в скалярном приближении описывает акустическое или электромагнитное комплекснозначное поле $u(r, q, \omega)$, генерируемое источником, заданным функцией $f(r, q, \omega)$. Будем предполагать, что источник является точечным гармоническим осциллятором, расположенным в точке $q \in Q$. Тогда эта функция имеет вид $f(r, q, \omega) = -\delta(r - q)$, где ω — круговая частота источника излучения. Пусть неоднородность среды вызвана только изменениями фазовой скорости $c(r)$, тогда имеет место соотношение $\kappa(r, \omega) = \frac{\omega}{c(r)}$.

Вне области неоднородности будем предполагать для простоты, что $\kappa(r, \omega) = \kappa_0 = \frac{\omega}{c_0}$, где постоянная c_0 известна.

Рассмотрим функцию Грина

$$G(r, q, \omega) = \frac{1}{4\pi\|r - q\|} \exp\left(i \frac{\omega}{c_0} \|r - q\|\right), \tag{2}$$

которая удовлетворяет уравнению $\Delta G(r, q, \omega) + \frac{\omega^2}{c_0^2} G(r, q, \omega) = -\delta(r - q)$.

Будем предполагать, что поле $u(r, q, \omega)$ измеряется в области P ($r \in P$), т.е. приемники излучения расположены в области P . Точечные источники излучения размещены в области Q ($q \in Q$), а неоднородность находится в ограниченной области R . Раздельно записывая уравнения для областей R и P , получаем нелинейную систему уравнений [7, 8]

$$u(r, q, \omega) = u_0(r, q, \omega) + \omega^2 \int_R G(r', r, \omega) \xi(r') u(r', q, \omega) dr', \quad r \in R, \tag{3}$$

$$U(p, q, \omega) = \omega^2 \int_R G(r', p, \omega) \xi(r') u(r', q, \omega) dr', \quad p \in P, \tag{4}$$

где $U(p, q, \omega) = u(p, q, \omega) - u_0(p, q, \omega)$, $u_0(r, q, \omega) = - \int_Q G(r', r, \omega) f(r', q, \omega) dr'$ и $\xi(r) = c_0^{-2} - c^{-2}(r)$.

В обратной задаче неизвестными считаются свойства среды $\xi(r)$ и поле $u(r, q, \omega)$. Известными считаются уравнения связи (3) и (4), в которых функция $U(p, q, \omega)$ получается как результат измерений в области расположения приемников. Требуется по этим измерениям восстановить неоднородность среды $\xi(r)$.

3. Алгоритмы решения обратной задачи. В основе итерационного метода решения обратной задачи лежит регуляризованный метод Ньютона [7, 8]

$$X_{p+1} = X_p - \tau (F_p'^* F_p' + \alpha_p E)^{-1} (F_p'^* F(X_p) + \alpha_p (X_p - \zeta)), \tag{5}$$

где $X_p = (u_p^{(1)}, \dots, u_p^{(t)}, \zeta_p)$ — вектор неизвестных, $t = qd$, q — число используемых частот зондирования, d — число положений источников, E — единичная матрица, $0 < \tau < 1$, α_p — параметр регуляризации, ζ — вектор регуляризации, $F_p' = F'(X_p)$ — производная Фреше, $F_p'^*$ — матрица, сопряженная к матрице F_p' , и p — номер итерации. Формулы для $F(X_p)$, $F_p' = F'(X_p)$ и $F_p'^* F_p'$ выписаны в [11]. Матрица F_p' имеет размер $(n + m)t \times n(t + 1)$. Эрмитова положительно определенная матрица $F_p'^* F_p'$ размера $n(t + 1) \times n(t + 1)$ в клеточном представлении имеет вид

$$F_p'^* F_p' = \begin{pmatrix} (n \times n) & 0 & 0 & (n \times n) \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & (n \times n) & (n \times n) \\ (n \times n) & \dots & (n \times n) & (n \times n) \end{pmatrix} = \begin{pmatrix} C^{1*} C^1 & 0 & \dots & 0 & C^{1*} D^1 \\ 0 & C^{2*} C^2 & \dots & 0 & C^{2*} D^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & C^{t*} C^t & C^{t*} D^t \\ D^{1*} C^1 & D^{2*} C^2 & \dots & D^{t*} C^t & D^{1*} D^1 + D^{2*} D^2 + \dots + D^{t*} D^t \end{pmatrix}, \tag{6}$$

где n — количество узлов сетки в расчетной области R , в которой расположена неоднородность, m — набор точек измерения поля U , C и D — матрицы, расчетные формулы для которых приведены в [11]. Начальное приближение возьмем в виде $X_0 = (u_0^{(1)}, \dots, u_0^{(t)}, 0)$, где $u_0^{(t)}$ — поле прямой волны при некотором положении источника и частоте.

Наиболее вычислительно трудоемкой операцией в методе (5) является вычисление матрицы и решение линейной системы с матрицей $(F_p'^* F_p' + \alpha_p E)$ на каждом шаге итераций. Для сравнения алгоритмов нами были разработаны однопроцессорные варианты программ решения линейных систем методом Холецкого, методом простой итерации и методом сопряженных градиентов [9, 14]. Далее для алгоритма, показавшего наибольшую эффективность, был разработан многопроцессорный вариант программы с использованием технологии MPI.

Алгоритм, основанный на методе сопряженных градиентов, имеет следующий вид. Пусть решается уравнение

$$Cx = b, \quad \text{где } C = C^* > 0, \quad (7)$$

тогда

$$X^1 = X^0 - \tau_1 C X^0 + \tau_1 b, \quad X^{k+1} = \alpha_{k+1} X^k - \alpha_{k+1} \tau_{k+1} C X^k + (1 - \alpha_{k+1}) X^{k-1} + \alpha_{k+1} \tau_{k+1} b. \quad (8)$$

Здесь $\tau_{k+1} = \frac{(r_k, r_k)}{(C r_k, r_k)}$, $r_k = C X^k - b$, $k = 0, 1, \dots$, $\alpha_1 = 1$, $\alpha_{k+1} = \left(1 - \frac{\tau_{k+1}}{\tau_k} \frac{(r_k, r_k)}{(r_{k-1}, r_{k-1})} \frac{1}{\alpha_k}\right)^{-1}$, $k = 1, 2, \dots$. Для нашего случая

$$C = F_p'^* F_p' + \alpha_p E, \quad b = F_p'^* F(X_p) + \alpha_p (X_p - \zeta), \quad X_p = (u_p^{(1)}, \dots, u_p^{(t)}, \zeta_p). \quad (9)$$

Асимптотика алгоритмической сложности метода сопряженных градиентов имеет вид $\text{Nit} \times O((n^3)^2)$, где Nit — число итераций и n — число точек области решения по одной координате. Для метода простой итерации формула алгоритмической сложности выглядит так же, но Nit — другое число итераций. Метод Холецкого является прямым методом решения систем линейных уравнений. Его алгоритмическая сложность имеет вид $O((n^3)^3)$. Таким образом, итерационные методы предпочтительнее по количеству арифметических операций, если число итераций не слишком большое. Обычно итерации заканчивают, когда невязка становится меньше предполагаемой ошибки входных данных δ .

Были проведены численные эксперименты на однопроцессорных компьютерах по сравнению времени работы трех указанных алгоритмов. Рассматривалась задача с двумя источниками на двух частотах, исходные данные были известны на сетке 13×13 точек. Проводилось пять итераций метода Ньютона (т.е. пять раз решалась линейная система). Задача решалась на двух сетках $7 \times 7 \times 7$ точек (размер матрицы 1715×1715) и $9 \times 9 \times 9$ точек (размер матрицы 3645×3645). Итерации заканчивали при достижении невязки $\delta = 3 \times 10^{-4}$ и $\delta = 1 \times 10^{-4}$. Время расчета в секундах для каждого варианта приведено в таблице. К этому времени необходимо добавить время пятикратного вычисления матрицы, которое составило 255 сек для матрицы 1715×1715 и 1625 сек для матрицы 3645×3645 .

Сравнение времени работы различных алгоритмов

	Размер матрицы 1715×1715		Размер матрицы 3645×3645	
	$\delta = 3 \times 10^{-4}$	$\delta = 1 \times 10^{-4}$	$\delta = 3 \times 10^{-4}$	$\delta = 1 \times 10^{-4}$
Простая итерация	1113	3566	16102	27697
Сопряженных градиентов	29	64	137	304
Метод Холецкого	41	41	295	295

Из таблицы следует, что метод сопряженных градиентов сходится значительно быстрее метода простой итерации. Прямой метод Холецкого с увеличением размерности начинает уступать методу сопряженных градиентов. В отличие от метода Холецкого, кроме того, метод сопряженных градиентов обладает очень высоким потенциалом распараллеливания процедур вычислений. Поэтому для дальнейших расчетов на параллельном компьютере был выбран метод сопряженных градиентов.

3.1. Сложность вычислительной задачи. Как было показано выше, численное решение обратной задачи (3), (4) сводится к итерационному процессу (5), на каждом этапе которого необходимо решить систему линейных уравнений с матрицей $F_p'^* F_p' + \alpha_p E$. Основными вычислительными трудностями данной задачи являются большой объем требуемой оперативной памяти для хранения этой матрицы и большое количество собственно вычислительных операций. Например, для сетки $N_x = N_y = N_z = 21$ при количестве положений источников $d = 5$ и числе частот $q = 30$ необходимый объем памяти для хранения матрицы (6) составляет $8((dq + 1)(N_x N_y N_z))^2 \sim 14570$ Гб (здесь 8 — размер в байтах числа типа complex, Гб = 1024^3 байт).

Вычислительная сложность задачи также весьма велика. Как видно из (6), процедура вычисления матрицы $F_p'^* F_p'$ сводится в конечном итоге к перемножению $4dq$ раз матриц размера $N \times N$ ($N = N_x \times N_y \times N_z$) и состоит из порядка $4dq O(N^3)$ операций. Сложность решения системы линейных уравнений составляет порядка $\text{Nit} \times O(N^2)$ операций, где Nit — число итераций.

3.2. Реализация метода решения обратной задачи. Одним из возможных путей решения вышеуказанных трудностей может быть использование кластерных вычислительных систем. При таком подходе возможно как распределять необходимую оперативную память по различным вычислительным узлам, так и организовывать на этих узлах параллельные вычисления.

Ниже описываются способы оптимизации использования оперативной памяти и сокращения времени расчетов.

3.3. Общая структура реализации параллельных вычислений. Для решения обратной задачи нами была выбрана система организации параллельных вычислений типа “один ведущий, остальные ведомые”. Такая система позволяет сосредоточить всю логику работы программы, чтение/запись данных, выдачу диагностики и сообщений об этапах выполнения программы в одном месте — на одном вычислительном узле. Выбор номера этого ведущего узла непринципиален, нами был выбран номер 0.

Выделение одного вычислительного узла в качестве “ведущего” позволяет значительно упростить отладку и сопровождение программы. Все логические решения принимаются только на “ведущем” узле, остальные узлы работают в режиме “управления потоком данных”. Они ожидают прихода исходных данных для расчета матрицы, после приема этих данных производят расчет своей части матрицы и правой части уравнения и далее вызывают процедуру решения системы линейных уравнений.

“Ведущий” узел вычисляет начальные данные для итерационного процесса, производит чтение промежуточных данных из файла сохранения, создает этот файл в случае необходимости, осуществляет вычисления после решения системы линейных уравнений и принимает все логические решения: об окончании итераций, об окончании выполнения программы и т.п.

3.4. Табулирование численных значений элементов матриц. Как видно из (6), при вычислении матрицы $F_p'^* F_p'$ используется функция Грина $G(r, q, \omega)$ (см. (2)). Табулирование ее значений на целочисленной сетке для некоторого диапазона позволяет значительно повысить скорость расчета матрицы $F_p'^* F_p'$ (выигрыш порядка 20–30%). Табулирование основано на следующем свойстве функции $G(r, q, \omega)$. Как видно из (2), функция $G(r, q, \omega)$ зависит от $\|r - q\|$. Наложим достаточно слабое ограничения на шаг расчетной сетки: шаги сетки по всем координатам должны быть равны. Пусть шаг равен Δ . Имеем

$$\|r - q\| = \sqrt{(r_1 - q_1)^2 + (r_2 - q_2)^2 + (r_3 - q_3)^2} = \Delta \sqrt{(i_1 - j_1)^2 + (i_2 - j_2)^2 + (i_3 - j_3)^2} = \Delta \sqrt{I},$$

где r_k, q_k — координаты точек сетки и I — целое число. Таким образом, для вычисления функции $G(r, q, \omega)$ достаточно знать ее значения для довольно малого диапазона возможных значений I на дискретной сетке: от 0 до $(3(\max(N_x^2, N_y^2, N_z^2) - 1) + 1)$, где $N_x \times N_y \times N_z$ — размер сетки.

В связи с этим в программе заранее вычисляются значения функции для всех возможных значений аргумента из диапазона, а затем сохраняются в некотором массиве.

3.5. Использование специального вида матрицы. Как видно из (6), матрица $F_p'^* F_p'$ является весьма разреженной. Пусть $N = N_x \times N_y \times N_z$, тогда очевидно, что для всех строк матрицы меньше dqN ненулевыми являются только $2N$ элементов. Кроме того, матрица $F_p'^* F_p'$ является эрмитовой. Все это позволяет хранить только $2(dq + 1)$ ее клеток размером $N \times N$. Компактная форма хранения этой матрицы имеет вид

$$F_p'^* F_p' \rightarrow \begin{pmatrix} C^{1*} C^1 & C^{1*} D^1 \\ C^{2*} C^2 & C^{2*} D^2 \\ \vdots & \vdots \\ C^{t*} C^t & C^{t*} D^t \\ D^{1*} C^1 & D^{1*} D^1 + D^{2*} D^2 + \dots + D^{t*} D^t \end{pmatrix}. \tag{10}$$

Таким образом, для хранения матрицы при $N = 21^3$, $d = 5$ и $q = 30$ необходимый объем памяти составляет 193 Гб вместо 14570 Гб при использовании плотной матрицы.

Компактная форма хранения матрицы усложняет алгоритм решения системы линейных алгебраических уравнений, однако это практически не сказывается на времени решения системы.

3.6. Сохранение промежуточных результатов. Несмотря на высокое быстродействие кластерного суперкомпьютера, объем задачи требует обеспечения возможности прерывания и продолжения расчета задачи с прерванного места. Мы можем сохранять значения вектора $X = (u^{(1)}, \dots, u^{(t)}, \zeta)$, вычисленные на каждом шаге итерации метода Ньютона. Для того чтобы не испортить вычисленные на предыдущем

шаге значения при прерывании задачи в процессе записи, массив сохранения создается двойного размера: сначала сохраняемые значения записываются в первую половину, затем во вторую, затем снова в первую и т.д.

При каждом старте программа проверяет, существует ли файл сохранения. Если существует, то промежуточные данные загружаются из файла, а счет продолжается с прерванного места; если файл сохранения не существует, то он создается, и в него записываются исходные значения параметров счета.

Для внутренних итераций решения системы линейных уравнений также используются файлы сохранения. Для этих итераций промежуточными результатами являются значения одного или нескольких (в зависимости от метода решения) комплексных векторов. Относительно малый объем промежуточных данных позволяет сохранять эти данные на каждом шаге итерации. Алгоритм сохранения/восстановления промежуточных данных аналогичен алгоритму для внешних итераций за одним исключением: файл сохранения уничтожается после окончания всех итераций, отсутствие файла сохранения служит признаком того, что систему уравнений следует решать с самого начала.

3.7. Распределение матрицы по узлам вычислительной системы. Основным “потребителем” оперативной памяти является матрица $F_p'^* F_p'$. Все остальные массивы значительно меньше и могут иметь копии на каждом узле вычислительной системы. При тех же параметрах $N = 21^3$, $d = 5$ и $q = 30$, например, размер массива правой части системы линейных уравнений и массива решения составляют около 11 Мб (для сравнения, размер матрицы 193 Гб). В рассматриваемых итерационных методах решения системы линейных уравнений основная операция — это умножение строки матрицы на столбец промежуточного результата, поэтому естественным методом распределения матрицы по узлам вычислительной системы является выделение для каждого узла своего диапазона строк матрицы. В данной реализации принято решение о выделении каждому узлу равного числа строк матрицы. Этот метод не является оптимальным с точки зрения равномерного распределения нагрузки на вычислительные узлы, но значительно упрощает алгоритм синхронизации работы узлов.

В результате данного подхода минимальное количество узлов вычислительной системы, требуемых для решения конкретной задачи, приблизительно равно размеру матрицы в байтах, деленному на 1 Гб (при оперативной памяти узла, равной 1 Гб). Это число может быть немного больше или меньше в зависимости от распределения задачи по конкретным вычислительным узлам.

3.8. Параллельное вычисление матрицы. Как было сказано в предыдущем разделе, каждому вычислительному узлу выделяется свой диапазон строк матрицы $F_p'^* F_p'$. Оперативная память для этих строк выделяется в динамически распределяемом диапазоне памяти каждого вычислительного узла. Указатель Matrix указывает на первую строку диапазона. Для того чтобы при вычислении матрицы на каждом узле можно было использовать одну и ту же процедуру, внутри этой процедуры указатель корректируется таким образом, чтобы он указывал на нулевую строку матрицы. Внутри цикла вычисления матрицы индексы переменных проверяются на условие попадания внутрь конкретного диапазона строк для каждого вычислительного узла. Если это условие не выполнено, вычисления не производятся.

Матрица вычисляется сразу в компактной форме. С учетом того, что она является эрмитовой, вычисление нижней “полосы” матрицы (последние N строк) не производится.

Кроме того, в процедуре расчета матрицы предусмотрена возможность добавления регуляризирующей диагональной матрицы αE .

3.9. Параллельное решение системы линейных уравнений. Для решения системы линейных уравнений был выбран метод сопряженных градиентов (8). Как видим, основной операцией в данном методе является умножение матрицы на вектор. Для плотной матрицы реализация данного алгоритма весьма проста, для уплотненной матрицы (10) — более сложная. Рассмотрим подробнее параллельную работу алгоритма.

Алгоритм работает с матрицей, представленной в компактной форме. Сначала осуществляется чтение промежуточных значений из файла сохранения. Эту операцию выполняет узел с номером 0, а затем прочитанные значения рассылаются на все узлы. Рассылка осуществляется процедурой MPI_Bcast; эта процедура обеспечивает также синхронизацию, т.е. узел ожидает получения данных и не продолжает работу, пока обмен данными не будет завершен.

Далее реализуется собственно итерационный процесс (8) решения линейной системы (7), (9). Поскольку каждый узел имеет значения всех элементов массивов X^k и X^{k-1} , то в каждом узле могут быть независимо вычислены значения CX^k для того диапазона строк матрицы, который этот узел содержит, за исключением последних N строк. В силу компактной формы хранения матрицы вычисления для $C^{n*} C^n$, $C^{m*} D^n$ и $D^{n*} D^n$ приходится выполнять отдельно. Поскольку уплотненная матрица не содержит всех элементов для последних N строк, значения CX^k вычисляются отдельно с использованием того факта, что

матрица $F_p'^* F_p'$ является эрмитовой. Частичные суммы для последних N строк матрицы вычисляются в отдельном массиве в каждом узле. Далее эти частичные суммы передаются в последний узел, где суммируются (с помощью процедуры `MPI_Reduce`). Вычисленные окончательные суммы рассылаются всем узлам с помощью процедуры `MPI_Bcast`.

После вычисления всех элементов вектора CX^k каждый узел имеет возможность вычислить X^{k+1} для тех строк матрицы, которые содержатся в данном узле. После этого узлы обмениваются вычисленными значениями X^{k+1} с помощью процедуры `MPI_Bcast`. Очередная итерация завершена.

Важным элементом итерационного алгоритма является критерий прекращения итераций. Нами был выбран критерий, основанный на величине невязки $\|CX^{k+1} - b\|$. Вычисление невязки производится аналогично вычислению CX^k в цикле итерации: в каждом узле параллельно вычисляются частичные суммы отдельно для $C^n * C^n$, $C^n * D^n$ и частичные суммы для последних N строк матрицы. Далее эти частичные суммы передаются для окончательного суммирования в узел 0, где и принимается решение о продолжении или окончании итераций. Затем это решение, зафиксированное во флаговой переменной, передается на все остальные узлы при помощи процедуры `MPI_Bcast`.

Если процесс итерации должен быть продолжен, узел 0 производит сохранение значений X^k и X^{k+1} в файле сохранения. В случае же прекращения итераций файл сохранения удаляется и происходит выход из процедуры решения системы линейных уравнений.

4. Математическое моделирование. Численные эксперименты были проведены на суперкомпьютере семейства СКИФ НИВЦ МГУ. В численном эксперименте рассматривалась следующая модель. На глубине 45 м под поверхностью Земли располагались два отражающих объекта. Фазовая скорость в среде равна $c_0 = 1800$ м/с. Первый объект отражает в два раза больше, чем второй. На поверхности Земли над вторым объектом в квадрате размером 80×80 м располагались 40×40 приемников в узлах решетки с равномерным шагом. Источник располагался на поверхности посередине между объектами. Зондирование проводилось на 40 частотах в диапазоне от 30 до 250 Гц, что соответствует длинам волн от 7 до 60 м. При решении обратной задачи область расчетов имела размеры $180 \times 180 \times 180$ м, в ней была введена сетка $9 \times 9 \times 9$ точек. Число неизвестных составило около 30000 комплексных чисел. Всего было проведено 11 итераций метода Ньютона. Расчет линейной системы заканчивался при достижении невязки $\delta = 3 \times 10^{-4}$. Задача считалась на 512 процессорах. Время расчета одной внутренней итерации составило около 8 сек.

В этом численном эксперименте предполагалось промоделировать следующий эффект. При излучении зондирующего импульса источником волна, отражаясь от двух объектов, регистрируется приемниками. Однако кроме этих импульсов будет зарегистрирована также переотраженная волна, т.е. волна, отразившаяся от первого объекта, далее попавшая на второй объект и далее отраженная вторым объектом на приемники. Аналогично и для первого объекта. Эффект переотражения является нелинейным эффектом и потому не может быть учтен при решении задачи линейными методами. Поэтому при решении линейными методами должны получаться ложные объекты, располагающиеся под реальными объектами глубже на половину расстояния между первым и вторым объектом. Разработанные же нами нелинейные методы должны учесть этот эффект и реконструировать только реальные объекты.

В качестве результатов решения линейным методом мы брали первый шаг процесса Ньютона, поскольку для выбранных начальных приближений он совпадает с линейным методом.

На рис. 1 приведено полутоновое изображение амплитуды восстановленной линейным методом функции в вертикальных сечениях, проходящих через оба объекта вдоль оси x (а) и через второй объект вдоль оси z (б). На рис. 1б внизу видно наличие размытого переотраженного ложного объекта, что согласуется с приведенной выше теорией.

На рис. 2 приведено полутоновое изображение амплитуды восстановленной нелинейным методом функции в двух перпендикулярных вертикальных сечениях, проходящих через оба объекта (а) и через второй объект (б). На рис. 2а видны два истинных объекта с разным коэффициентом отражения. На рис. 2б ложный объект исчез.

В другом численном эксперименте рассматривалась следующая модель. На глубине 45 и 135 м под поверхностью Земли располагались по два одинаково отражающих объекта на расстоянии 90 м друг от друга. Фазовая скорость в среде: $c_0 = 1800$ м/с. На поверхности Земли над объектами симметрично в квадрате размером 380×380 м располагались 40×40 приемников в узлах решетки с равномерным шагом. Источник располагался на поверхности посередине между объектами. Зондирование проводилось на частотах 40 и 60 Гц, что соответствует длинам волн 45 и 30 м. При решении обратной задачи область расчетов имела размеры $180 \times 180 \times 180$ м, в ней была введена сетка $25 \times 25 \times 25$ точек. Расчет линейной системы заканчивался при достижении невязки $\delta = 1 \times 10^{-3}$. Время расчета одной внешней итерации на 512 процессорах составило около 3 час. Время расчета одной внутренней итерации составило 9 сек.

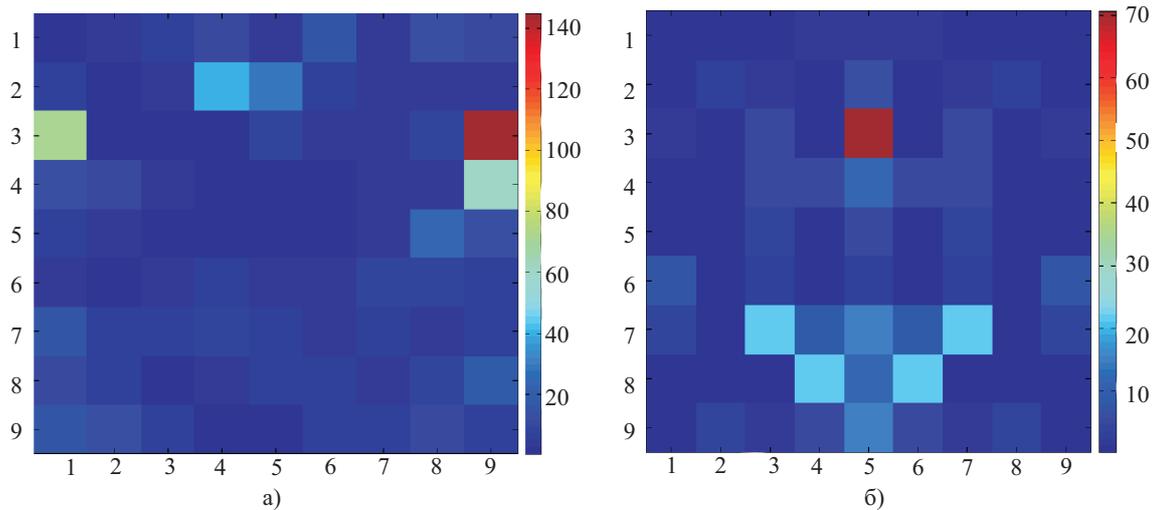


Рис. 1. Полутоновое изображение двух перпендикулярных вертикальных сечений амплитуды восстановленной линейным методом функции

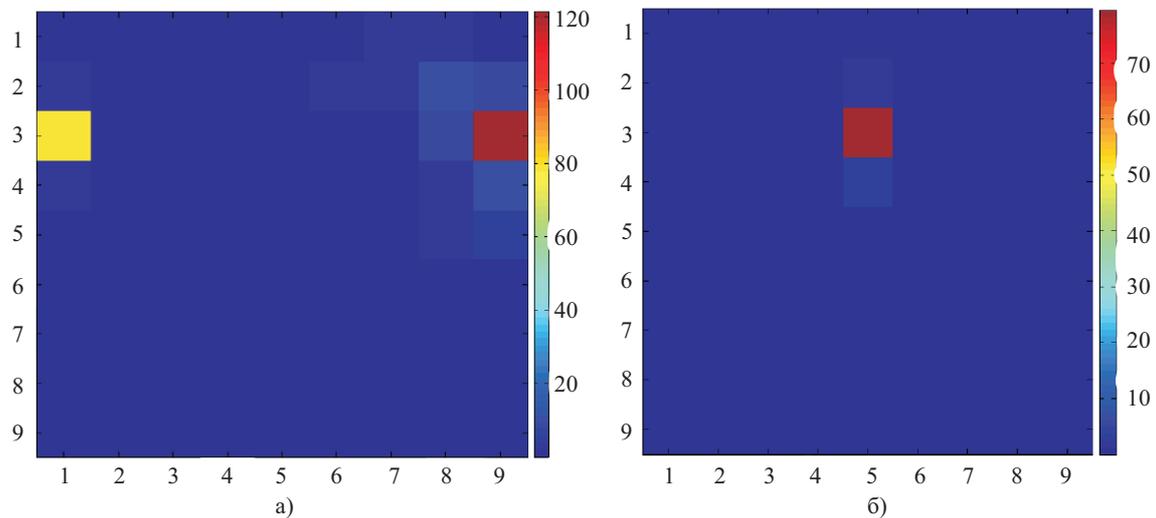


Рис. 2. Полутоновое изображение двух перпендикулярных вертикальных сечений амплитуды восстановленной нелинейным методом функции

В этом численном эксперименте предполагалось промоделировать возможность решения задачи на большой сетке в области решения. Число неизвестных составило около 45000 комплексных чисел.

На рис. 3а приведено полутоновое изображение амплитуды восстановленной функции в вертикальном сечении, проходящем через все четыре объекта. На рис. 3б — восстановленное изображение в горизонтальном сечении, проходящем через два объекта. Видно, что разрешение в горизонтальном направлении лучше, чем в глубину, а также видно наличие некоторого шума. Это связано с малым количеством использованных частот зондирования и положений источников.

5. Заключение. По результатам работы можно сделать следующие выводы.

Использование суперкомпьютеров семейства СКИФ позволяет в десятки раз повысить размерность рассматриваемой задачи. С увеличением размерности удастся рассматривать все более содержательные задачи. В частности, в работе приведены модельные примеры, для которых рассматриваемая задача диагностики является существенно нелинейной.

Использование итерационных процедур решения линейных уравнений позволяет с высокой эффективностью использовать параллельную структуру кластерных систем.

С увеличением размерности задачи значительное время тратится на вычисление матрицы задачи. Поэтому в дальнейшем перспективной представляется разработка быстрых алгоритмов расчета матрицы, а также иных оптимизирующих подходов, в частности сохранение промежуточных результатов расчетов

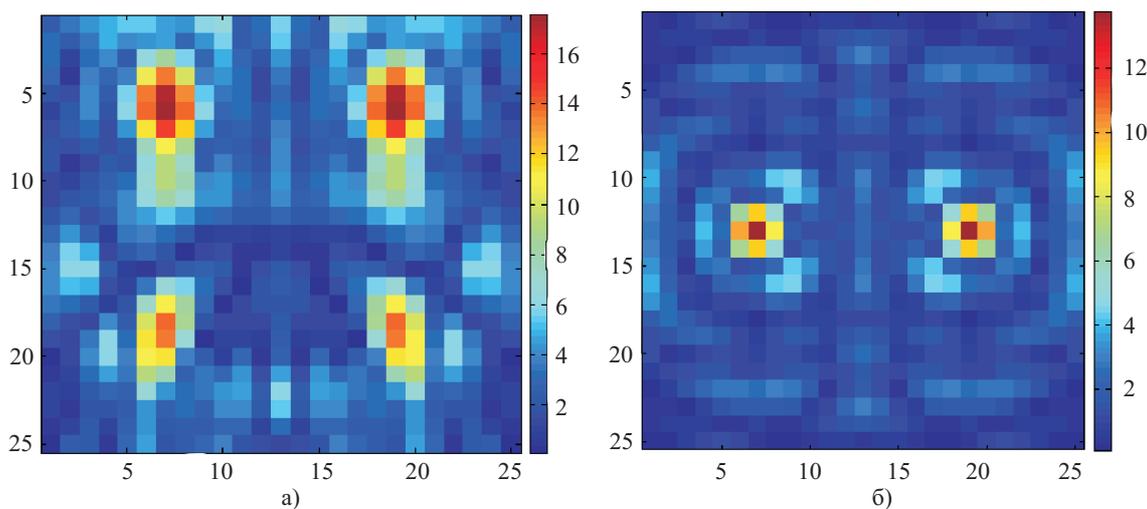


Рис. 3. Полутонное изображение четырех объектов: а) сечение в вертикальной плоскости, б) сечение в горизонтальной плоскости

матрицы объемом до 1 Тб.

Как показала практика расчетов, на выполнение первой итерации метода Ньютона расходуется значительно больше времени, чем на последующие. Это связано с тем, что на первом шаге итерационный процесс начинается с нулевого (грубого) начального приближения, а на последующих шагах — начинается с уточненного начального приближения. Для ускорения процесса перспективно на первом шаге решать (более простую) линейную задачу.

СПИСОК ЛИТЕРАТУРЫ

1. Claerbout J. Imaging the Earth's interior. Boston: Blackwell Scientific Publishers, 1986.
2. Robinson E., Durrani T.S., Peardon L.G. Geophysical signal processing. Englewood Cliffs: Prentice Hall, 1986.
3. Gazdag J., Squazzero P. Migration of seismic data // IEEE. 1984. **72**. 1302–1315.
4. Yilmaz O. Seismic data processing. Tulsa: Society of Exploration Geophysicists, 1987.
5. Baysal E., Kosloff D.D., Sherwood J.W.C. Reverse time migration // Geophysics. 1983. **48**. 1514–1524.
6. Ramm A.G. Multidimensional inverse scattering problems. London: Longman Group, 1992.
7. Bakushinsky A.B., Goncharsky A.V. Ill-posed Problems. Theory and applications. Dordrecht: Kluwer Academic Publ., 1994.
8. Bakushinsky A.B., Goncharsky A.V., Romanov S.Yu., Seatzu S. On the identification of velocity in seismic and in acoustic sounding. Firenze, 1994.
9. Рябенский В.С. Введение в вычислительную математику. М.: Физматлит, 1994.
10. Гончарский А.В., Романов С.Ю., Харченко С.А. Обратные задачи диагностики с синтезированной апертурой // Труды Международной научной конференции “Параллельные вычислительные технологии” (ПаВТ’2007). Челябинск, 29.01.2007 – 2.02.2007. Челябинск: Изд-во ЮУрГУ, 2007. Том 1. 109–114.
11. Гончарский А.В., Романов С.Ю., Харченко С.А. Обратная задача акустической диагностики трехмерных сред // Вычислительные методы и программирование. 2006. **7**, № 1. 117–125.
12. Виноградов Н.С., Гончарский А.В., Овчинников С.Л., Романов С.Ю. Об одной задаче волновой томографии // Труды Всероссийской научной конференции “Научный сервис в сети ИНТЕРНЕТ: многоядерный компьютерный мир”. Новороссийск, 24.09.07 – 29.09.07. М.: Изд-во Моск. ун-та, 2007. 129–133.
13. Агаян Г.М., Виноградов Н.С., Гончарский А.В., Овчинников С.Л., Романов С.Ю. Диагностика трехмерных сред методами синтезированной апертуры // Вычислительные методы и программирование. 2007. **8**, № 2. 5–10.
14. Тыртышников Е.Е. Методы численного анализа. М.: Академия, 2007.

Поступила в редакцию
20.09.2008