# ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ И МАСШТАБИРУЕМОСТИ ПРОГРАММЫ РЕШЕНИЯ ЗАДАЧ УЛЬТРАЗВУКОВОЙ ТОМОГРАФИИ

**С.Ю. Романов**

## 1. Введение.

В настоящее время томографические комплексы являются неотъемлемой частью диагностических исследований в различных областях науки и техники и, конечно же, в медицине. В работе рассмотрены некоторые вопросы, относящиеся к разработке программного обеспечения для решения обратных задач ультразвуковой томографии в медицине. В связи с актуальностью проблемы в настоящее время существует несколько научных центров, интенсивно работающих в области решения обратных задач ультразвуковой томографии [1-6].

В предлагаемой работе обратная задача, в отличие от интегрального подхода [7-9], рассмотрена как коэффициентная обратная задача для дифференциального уравнения гиперболического типа. Одна из проблем решения рассматриваемой задачи состоит в необходимости выполнения огромного объема вычислений. Эффективным методом решения этой проблемы является использование параллельных вычислений на многопроцессорных системах. Разработанная программа позволяет решать обратные задачи волновой томографии в нелинейной постановке на сетках 1000x1000 элементов в томографическом слое. Полученные результаты являются новыми и находятся на мировом уровне [10]. В работе [11] рассмотрены вопросы оптимизации кода при параллельных расчетах.

В настоящей работе рассмотрены вопросы повышения эффективности, масштабируемости программного обеспечения в задаче ультразвуковой томографии. Расчеты проводились на кластерной вычислительной системе МГУ «Ломоносов». Проведено тестирование программы в конфигурации, обеспечивающей одновременную работу 20480 процессов. Относительная эффективность распараллеливания на 20480 процессов составила около 60%.

Работа выполнена при финансовой поддержке РФФИ проект № 12-07-00304-а.

## 2. Постановка и методы вычислений обратной задачи.

Рассмотрим волновое уравнение, которое в скалярном приближении описывает акустическое поле $u(r,t)$ в трехмерной области $\Omega \subset R^3$, ограниченной поверхностью *S* в течение времени *[0,T]* с точечным источником, располагающимся в точке $r_0$

$$\begin{aligned} c(r)u_{tt}(r,t) - \Delta u(r,t) &= \delta(r-r_0)\cdot f(t), \\ u(r,t=0) = u_t(r,t=0) &= 0, \quad \partial_n u\,|_{ST} = p(r,t). \end{aligned} \tag{1}$$

Здесь $c^{-0.5}(r)$ - является скоростью волны в среде, $r \in R^3$ - положение точки в пространстве, $\Delta$ - оператор Лапласа по переменной $r$. Генерируемый источником импульс описывается функцией $f(t)$, $\partial_n u\,|_{ST}$ - производная вдоль нормали к поверхности *S* в области $S \times T$, $p(r,t)$ - некоторая известная функция. Будем предполагать, что неоднородность среды вызвана только изменениями скорости, а вне области неоднородности скорость $c(r) \equiv c_0 = const$, где $c_0$ - известна.

Обратная задача состоит в нахождении функции $c(r)$, описывающей неоднородность, по экспериментальным данным измерения волны $U(s,t)$ на границе $S$ области за время $(0;T)$ при различных положениях $r_0$ источника. Следуя классическим традициям томографических исследований, обратная задача диагностики 3D объекта рассматривается как набор двумерных задач. Подробно математические аспекты проблемы рассмотрены в работах [4, 12].

Расчетная модель, на основе которой проведены вычисления, получена из дифференциального уравнения (1). Для решения обратной задачи в каждом из слоев, будем использовать метод конечных разностей. В такой постановке решение волновых дифференциальных уравнений сводится к решению разностных уравнений. На области изменения аргументов введем равномерную дискретную сетку

$$v_{ijmk} = \left\{ \begin{matrix} (x_i, y_j, z_m, t_k) : x_i = ih,\ 0 \le i \le N_x;\ y_j = jh,\ 0 \le j \le N_y; \\ z_m = mp,\ 0 \le m \le N_z;\ t_k = k\tau,\ 0 \le k \le N_t \end{matrix} \right\},$$

где *h* – шаг сетки по горизонтальным координатам, *p* – шаг сетки по вертикальной координате, $\tau$ - шаг сетки по времени. Параметры *h* и $\tau$ связаны условием устойчивости Куранта $c^{-0.5}\tau < h$. Параметры $N_x$, $N_y$ задают

количество точек сетки по горизонтальным координатам, $N_z$ –количество слоев регистрации данных по вертикали.

В области, не содержащей источников, получаем явную разностную схему для дифференциального уравнения (1)

$$c_{ij}\frac{u_{ijk+1}-2u_{ijk}+u_{ijk-1}}{\tau^2}-\frac{u_{i+1jk}-2u_{ijk}+u_{i-1jk}}{h^2}-\frac{u_{ij+1k}-2u_{ijk}+u_{ij-1k}}{h^2}=0.$$

Здесь $u_{ijk}$ – значения $u(r,t)$ в точке ($x_i$, $y_j$) в момент времени $t_k$, $c_{ij}$ – значения $c(r)$ в точке ($x_i$, $y_j$).

Расчет распространения звуковой волны (расчет «в прямом времени») выполняется по временным слоям в явной форме

$$u_{ijk+1}=u_{ijk}(2-\frac{2\tau^2}{c_{ij}h^2}-\frac{2\tau^2}{c_{ij}h^2})+\frac{(u_{i+1jk}+u_{i-1jk})\tau^2}{c_{ij}h^2}+\frac{(u_{ij+1k}+u_{ij-1k})\tau^2}{c_{ij}h^2}-u_{ijk-1}, \qquad (2)$$

начальные условия задаются в виде $u_{ij0} = u_{ij1} = 0$, граничные условия задаются в виде:

$$\frac{(u_{i+1jk}-u_{i-1jk})}{2h}=\pm\frac{(u_{ijk+1}-u_{ijk-1})}{c_{ij}^{0.5}2\tau} \text{ для } i= 1 \text{ или } N_x\text{-}1,$$

$$\frac{(u_{ij+1k}-u_{ij-1k})}{2h}=\pm\frac{(u_{ijk+1}-u_{ijk-1})}{c_{ij}^{0.5}2\tau} \text{ для } j= 1 \text{ или } N_y\text{-}1.$$

Расчетная модель, описываемая формулой (2), аппроксимирует гиперболическое уравнение второго порядка (волновое уравнение) со вторым порядком точности. Эта модель описывает волновые эффекты дифракции, рефракции, переотражения и т.д. Для решения задачи в каждом томографическом слое необходимо выполнить $O(N_xN_yN_t)$ операций. Таким образом, объем вычислений растет не более, чем линейно от числа точек сетки по времени $N_t$ и числа точек сетки области вычислений $N_x * N_y$.

Градиент функционала вычислялся по разностной формуле

$$grad_{ij}=\sum_{k=0}^{m}\frac{u_{ijk+1}-u_{ijk}}{\tau}\frac{w_{ijk+1}-w_{ijk}}{\tau}\tau,$$

а невязка на текущей итерации

$$Nev=\frac{1}{2}\sum_{k=0}^{m}\sum_{(i,j)\in S}(u_{ijk}-U_{ijk})^2h\tau.$$

Здесь $S$ – граница, $U_{ijt}$ – значения $U(r,t)$ в точке ($x_i$, $y_j$) на границе $S$ в момент времени $t_k$, $grad_{ij}$ – градиент невязки в точке ($x_i$, $y_j$), $w_{ijk}$ - результаты расчета в обратном времени [12].

Зондирующий импульс задается формулой

$$u_{ij0}=sin\left(\frac{2\pi(R_{ij}\text{-}BegImp)}{WidthImp}-0.5\pi\right)+1, \text{ при}$$

$$BegImp < R_{ij} < BegImp+WidthImp$$

$$u_{ij0}=0, \text{ в остальных случаях}$$

где $R_{ij}=h\sqrt{(i-ISou_s)^2+(j-JSou_s)^2}$. Параметр *WidthImp* – начальная ширина импульса, *BegImp* — расстояние от источника до начального положения импульса. $ISou_s, JSou_s$ – координаты положения источников на границе $S$, $s=1, \ldots, N_s$. $N_s$ - количество источников.

3. **Исследование эффективности и масштабируемости программы.** Возможность эффективного использования параллельных вычислений при решении обратной задачи определяется структурой алгоритма, возможностью выделения большого числа максимально вычислительно независимых блоков. Предложенная архитектура программы, позволила выделить большое количество таких блоков, на которых параллельно выполняются процессы общего потока задачи.

Во-первых, трёхмерная область делится на Nslice независимо обрабатываемых сечений (Nslice ~= 40) в соответствии с рисунком 1.

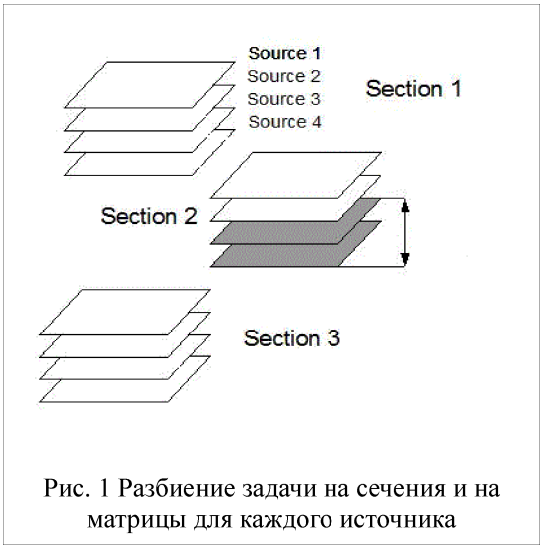



Рис. 1 Разбиение задачи на сечения и на матрицы для каждого источника

Далее, в каждом сечении расчёт для каждого ультразвукового источника (их количество *Nsou* ~ 10 ÷ 20) может быть выполнен в значительной степени независимо (результаты расчетов должны быть лишь просуммированы для вычисления градиента на каждой итерации градиентного спуска). Таким образом, в целом имеем набор *Nslice * Nsou* матриц, вычисление которых может быть выполнено независимо.

Для дальнейшего распараллеливания вычислений воспользуемся известным методом распараллеливания по пространству, который состоит в том, что общее поле вычислений размером *NyAll * NxAll* точек матрицы разбивается на *NyPar * NxPar* частей-блоков размером *Nx*Ny*, вычисления в которых производятся различными вычислительными ядрами параллельно (*NyPar * NxPar* - ядер).

Один из основных принципов разбиения на блоки состоит в выравнивании загрузки процессоров. Для минимизации времени ожидания процессором данных с соседних процессоров расчёты всех блоков должны быть завершены синхронно, поэтому следует разбивать матрицу на блоки с равным объёмом вычислений. В нашем случае явной разностной схемы, одинаковой во всех точках, используем простейший вариант разбиения на блоки одинакового размера в соответствии с рисунком 2.

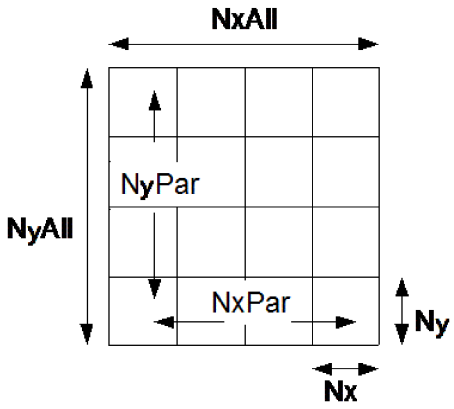



Рис. 2 Разбиение матрицы на блоки

Для минимизации межпроцессорных обменов следует минимизировать периметр блоков (в данном случае периметр равен 2*(*Ny* + *Nx*) по отношению к их площади (в данном случае площадь равна *Ny * Nx*). Как известно, минимум периметра у прямоугольника заданной площади достигается при *Ny = Nx*. Это требование было проверено экспериментально на модельных задачах. Для этого были проведены эксперименты на сетках размером 502x502 и 1002x1002, результаты которых приведены в таблицах 1 и 2. В верхней строке указаны значения *NX_PAR*NY_PAR* для различных разбиений сетки по процессам, в нижней - время расчета 10 итераций. При перестановке значений *NX_PAR* и *NY_PAR* результат не меняется. Из данных таблиц видно, что чем ближе значения *NX_PAR* и *NY_PAR* друг к другу, тем меньше время счета.

Таблица 1. Сравнение вариантов разбиения сетки, размер сетки – 502x502, число процессов 36.

| Варианты разбиения сетки | 1*36 | 2*18 | 3*12 | 6*6 |
|---|---|---|---|---|
| Время, сек | 83 | 59 | 55 | 49 |

Таблица 2. Сравнение вариантов разбиения сетки, размер сетки – 1002x1002, число процессов 64.

| Варианты разбиения сетки | 1*64 | 2*32 | 4*16 | 8*8 |
|---|---|---|---|---|
| Время, сек | 692 | 386 | 254 | 220 |

Для явной, 7-точечной, 4-связной в плоскости ($x,y$) разностной схемы, при разбиении на прямоугольные параллельно обрабатываемые блоки, возникает 4-связная сетка межпроцессорных обменов. На каждом процессоре (вычислительном ядре) для одного шага разностной схемы по времени выполняется $\sim Nx^2$ арифметических операций и производится передача и приём $\sim Nx$ данных в 4 направлениях. Для обеспечения масштабируемости коммуникационная сеть кластера должна содержать, как минимум, такое же число связей, тогда все передачи данных могут выполняться параллельно.

Отметим, что поскольку в явной разностной схеме значения во всех точках сетки на следующем шаге по времени не зависят друг от друга и могут быть вычислены параллельно как SIMD-алгоритм, то задача может эффективно выполняться на широком диапазоне архитектур процессоров: общего назначения, векторных, массивно-параллельных, графических и т.п.

Рассмотрим вопросы эффективности вычислительной системы в рассматриваемой задаче ультразвуковой томографии в зависимости от количества вычислительных ядер на одну матрицу (или размера блока $Nx$ ). Качественно опишем математической параметрической моделью процесс вычислений для нашей задачи.

Исходными параметрами, определяющими конфигурацию системы, являются:

1. $V$ – объём данных одного блока, $V = Nx * Ny = NyAll * NxAll / N_{proc}$;
2. $Vc$ – объём кэша вычислительного узла;
3. $Vm$ – объём данных, не поместившихся в кэш $Vm = V - min(V,Vc)$;
4. $Bc$ – производительность вычислений в кэше (cache bandwidth);
5. $Bm$ – производительность вычислений в памяти (memory bandwidth) ;
6. $Lcomm$ – задержка распространения сообщений, comm latency;
7. $Bcomm$ – производительность коммуникационной сети, comm bandwidth.

Времена выполнения одного шага явной разностной схемы для одного процесса с точностью до коэффициентов равны:

- $Tcache = min(V,Vc) / Bc$ – для операций в кэше,
- $Tmem = Vm / Bm$ – для операций в памяти,
- $Tcomm = 4 * Nx / Bcomm$ – для коммуникаций.

$$T = (Lcomm + Tcomm + Tcache + Tmem) * T\_all, \quad (3)$$

Общее время выполнения вычислений имеет вид
где $T\_all$ – число шагов по времени разностной схемы.

Рассмотрим эффективность $E$ функционирования каждого процесса в выбранной схеме распараллеливания. С точностью до коэффициента эффективность можно понимать как отношение времени $T_0$ расчетов задачи на $N_{proc}$ процессах с пиковой производительностью к времени $T$ расчета на $N_{proc}$ процессах в выбранной схеме распараллеливания. Типично пиковой производительностью системы является

$$E = \frac{T_0}{T} = \frac{K}{N_{proc} \cdot T}, \quad (4)$$

производительность вычислений в кэш-памяти.

где $K$ – некоторый постоянный коэффициент, $T$ - из формулы (3) (зависит от $N_{proc}$).

На рисунке 3 приведено качественное поведение графика теоретической зависимости эффективности $E$ от $N_{proc}$, построенное с использованием формул (3)-(4). (сплошная линия), размер сетки матрицы $NyAll = NxAll$

= 1002. При построении графика в модели использованы типичные значения параметров для процессоров общего назначения.

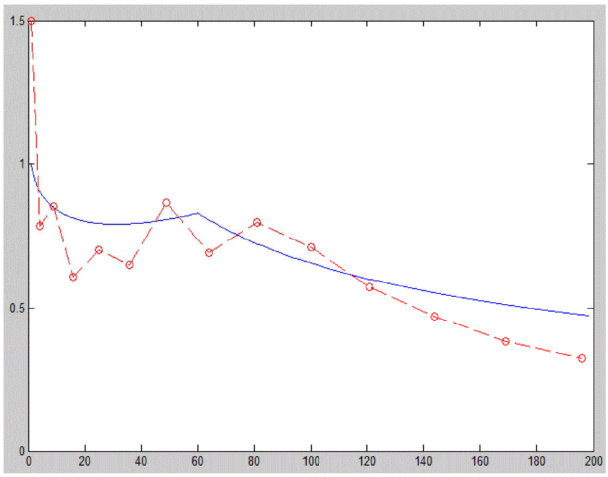



Рис. 3 График зависимости эффективности E от Nproc на одной матрице: теоретическая зависимость (сплошная линия) и экспериментальная зависимость (пунктирная линия)

Ниже в таблице 3 приведены результаты тестирования разработанной программы в рассматриваемой задаче ультразвуковой томографии, проведённые на суперкомпьютере «Ломоносов».

Таблица 3. Результаты тестирования на суперкомпьютере «Ломоносов»

| Число процессов ($N_{proc}$) | 1 | 2*2 | 3*3 | 4*4 | 5*5 | 6*6 | 7*7 | 8*8 | 9*9 | 10*10 | 11*11 | 12*12 | 13*13 | 14*14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Время выполнения 15 итераций (T) сек | 1893 | 903 | 370 | 291 | 162 | 121 | 67 | 64 | 44 | 40 | 41 | 42 | 44 | 45 |

На основании данных приведенных в таблице 3, на рисунке 3 пунктиром приведен экспериментально полученный график эффективности в зависимости от числа процессов. Видно, что графики теоретической и экспериментальной кривых имеют похожие закономерности.

Первоначально при увеличении количества ядер на матрицу эффективность несколько падает, поскольку при малом количестве ядер объем обменов по границам блоков невелик (для 1 ядра на матрицу отсутствуют вообще). Далее эффективность начинает расти, что связано с увеличением доли вычислений в кэше. При дальнейшем увеличении числа процессов эффективность падает из-за увеличения ожидания процессом данных с соседних процессов.

Зависимость эффективности вычислений от количества процессов на матрицу зависит от параметров конкретной вычислительной системы. Тем не менее, существует некоторое пороговое значение количества процессов на матрицу (в нашей программе примерно 100 процессов), выше которого эффективность падает. Этот пик эффективности связан с расчетами в кэше. На современных вычислительных системах производительность арифметического устройства и кэш-памяти во много (10-100) раз выше производительности памяти, и соответствие размера данных размеру кэша оказывает сильное влияние на эффективность [14].

Знание этого порогового значения количества процессов в рассматриваемой задаче особенно важно в том случае, если количество доступных процессоров ограничено. Поскольку правильное распределение процессоров по матрицам влияет на производительность вычислительной системы. Например, если доступно 200 вычислительных узлов, то использование их всех для вычисления на одной матрице или использование менее 50 узлов на матрицу было бы не эффективно.

Для вычислительных систем, в которых возможно параллельное выполнение обмена данными и вычислений, использование этой возможности должно приводить к существенному повышению эффективности расчетов. Однако проведенные эксперименты с неблокирующими коммуникационными операциями не привели к сокращению времени расчетов, что требует дополнительных исследований.

На рисунке 4 пунктиром приведен график функции *1/T*, где $T(N_{proc})$ время расчетов $N_{proc}$ процессоров на одной матрице. Этот график описывает также производительность или ускорение вычислительной системы. Как

видно, производительность растет с линейной скоростью и выше вплоть до 100 процессоров на матрицу. Далее производительность ухудшается.

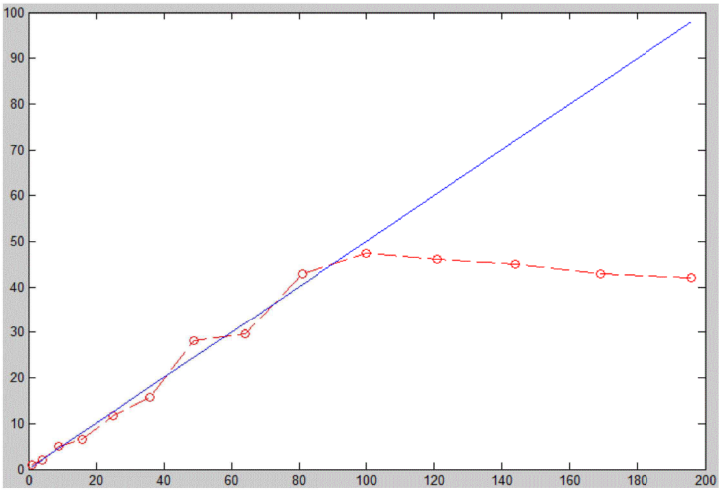


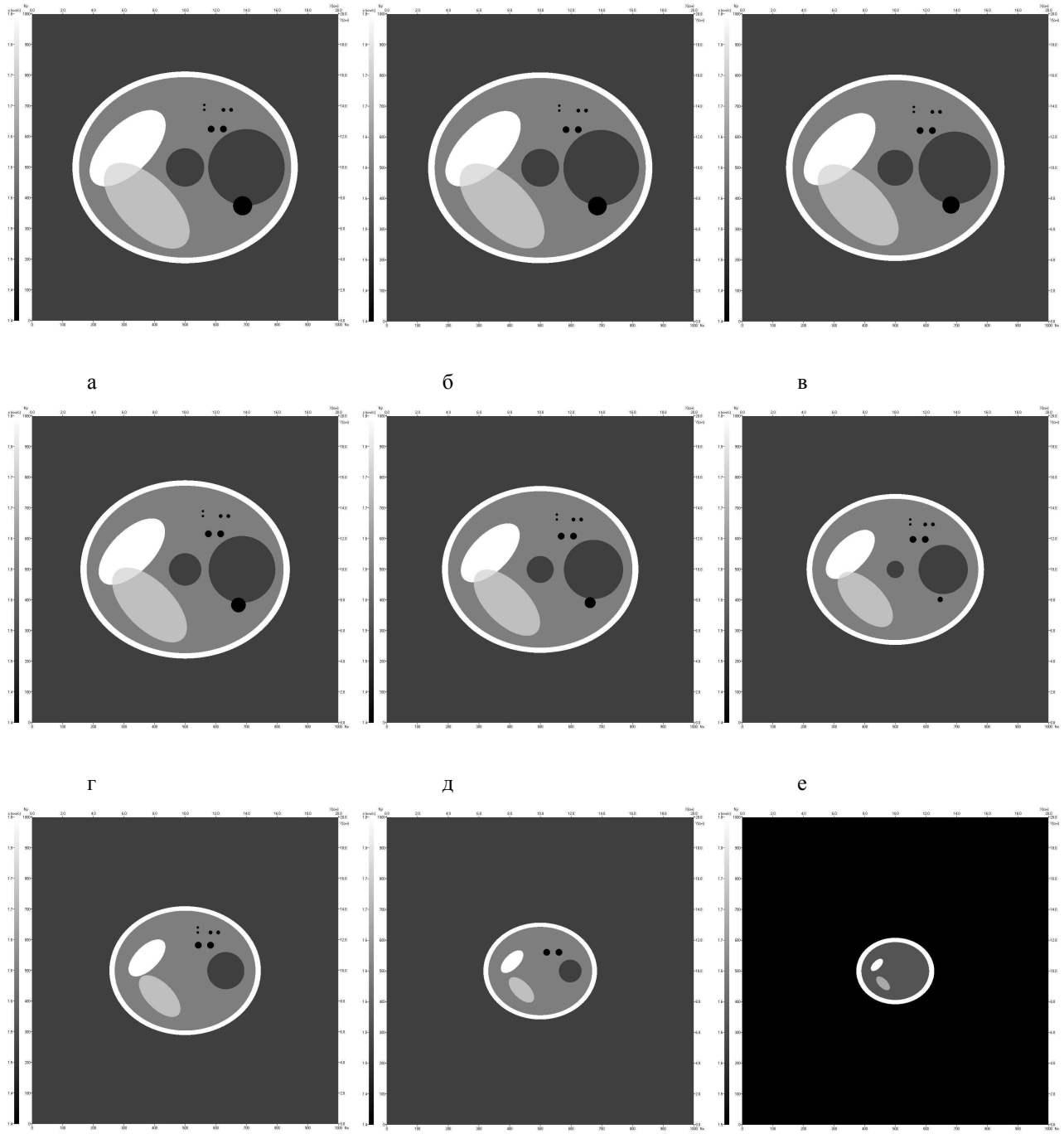

Рис. 4 Зависимость производительности системы от числа процессоров на одну матрицу (пунктир).

4. **Проведение тестирования программы в конфигурации, обеспечивающей одновременную работу 20480 процессов.** Рассматриваемая программа решения прямых и обратных задач ультразвуковой томографической диагностики тестировалась в конфигурации, обеспечивающей одновременную работу 20480 процессов на суперкомпьютере. Экспериментальные исследования проводились на компьютерно-синтезированном 3D объекте с модельными неоднородностями. На рисунках 5а-к приведены результаты реконструкции каждого 5-ого сечения функции скорости распространения ультразвуковой волны $c(x, y, z)$ в исследуемом объекте как функции от $x, y$ при фиксированном $z = z_i$ ($i$ = 1, … 40). Расстояние между сечениями по оси $z$ составляло 5 мм. Потемнение в каждой точке рисунка пропорционально $c(x, y, z)$. Минимальный размер неоднородности 3мм. Вариация скорости $c(x, y, z)$ не превышала 20%. В ходе экспериментальных исследований программы решалась прямая задача распространения ультразвуковой волны в каждом слое. По полученным данным решалась обратная задача восстановления функции $c(x, y, z)$ в каждом слое. Расчет прямой и обратной задачи производился одновременно на 40 слоях. Параметры расчетной модели:

- длина волны излучения 5.0 мм;
- шаг регистрации сигналов по пространству 2.4 мм;
- количество отсчетов данных по времени на период волны 20;
- уровень относительной погрешности входных данных 0;
- размер области ультразвукового зондирования по горизонтали 200x200 мм, по вертикали 200 мм.

Для решения обратной задачи использовался итерационный процесс с начального приближения $c(x, y, z) = const$. Количество итераций 700, время расчета около 4 часов. Расчеты проводились на сетке 1002x1002 точек в 40 слоях для 8 источников излучения. Распараллеливания по пространству по координатам X и Y состояло в том, что общее поле вычислений размером 1002x1002 точек разбивалось на *NyPar* * *NxPar* одинаковых частей (*NyPar = 8, NxPar = 8*), вычисления в которых производятся различными вычислительными ядрами. В результате получили распараллеливание задачи на 40 * 8 * 8 * 8 = 20480 процессов. В соответствие с предыдущим разделом такое разбиение по процессам позволяет достичь высокой эффективности.

В ходе экспериментальных исследований проводилось 2 запуска программы на 20480 процессорах. Первый запуск осуществлялся в виде одновременного запуска 3 программ, выполняющих расчет восстановления неоднородностей в 15, 15 и 10 слоях (7680 + 7680 + 5120 процессов). Второй запуск программы осуществлялся в виде запуска одной программы с использованием 20480 процессоров. Результаты расчетов не отличались друг от друга. Относительная эффективность распараллеливания на 20480 процессов составила около 60% (если сравнивать ускорение расчетов по отношению к расчетам на 1 процессоре).



а б в

г д е

ж и к

Рис. 5. Слои восстановленного изображения фантома

5. **Выводы.** Продемонстрирована возможность эффективного использования разработанного программного обеспечения для большого числа процессоров. Программа обеспечивает высокий уровень масштабирования для нескольких десятков тысяч процессов.

Максимальная производительность каждого ядра (и производительность всей системы при ограниченном количестве ядер) достигается при определённом значении *Nx* = *Nxonm*, которое может быть определено для конкретной системы. При отклонении *Nx* от оптимального значения, как в большую, так и в меньшую сторону, эффективность будет падать. Исходя из этого, следует сначала выбирать *Nx*, а затем, исходя из имеющихся ресурсов системы при данном *Nx*, выбирать число параллельно рассчитываемых на системе матриц.

Разработанное программное обеспечение позволяет решать трехмерные задачи восстановления неоднородностей в томографических слоях за приемлемое для медицинских целей время.

ЛИТЕРАТУРА:


1. R. Jiřík, I. Peterlík, N. Ruiter, J. Fousek, R. Dapp, M. Zapf, J. Jan "Sound-Speed Image Reconstruction in Sparse-Aperture 3-D Ultrasound Transmission Tomography" // IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control. 2012. 59, N 2, c.254-264.
2. C. Li, N. Duric, P. Littrup, L. Huang "In vivo breast soundspeed imaging with ultrasound tomography" // Ultrasound Med. Biol. 2009. 35, N 10, c.1615–1628.
3. F. Natterer, F. Wubbeling "A propagation-backpropagation method for uzltrasound tomography" // Inverse Problems. 1995. 11, N 6, c.1225–1232.
4. L. Beilina, M.V. Klibanov "Approximate global convergence and adaptivity for coefficient inverse problems". New York: Springer, 2012.
5. A. Backushinsky, A. Goncharsky, S. Romanov, S. Seatzu "On the identification of velocity in seismics and in acoustic sounding" // Pubblicazioni dell'istituto di analisa globale e applicazioni, Serie "Problemi non ben posti ed inversi». Issue 71. Firenze, 1994.
6. А.Б. Бакушинский, А.И. Козлов, М.Ю. Кокурин "Об одной обратной задаче для трехмерного волнового уравнения" // Журн. вычисл. матем. и матем. физики. 2003. 43, № 8, с.1201–1209.
7. С.Г. Головина, С.Ю. Романов, В.В. Степанов "Об одной обратной задаче сейсмики" // Вестн. МГУ. Сер. 15. Выч. мат. и киб. 1994. № 4, с.16–21.
8. А.В. Гончарский, С.Л. Овчинников, С.Ю. Романов "Об одной задаче волновой диагностики" // Вест. моск. ун-та. Сер.15. Вычисл. матем. и киберн. 2010. №1, с.7-13.
9. С.Л. Овчинников, С.Ю. Романов "Организация параллельных вычислений при решении обратной задачи волновой диагностики" // Вычислительные методы и программирование. 2008. Т.9. №1, с.338-345.
10. А.В. Гончарский, С.Ю. Романов "Суперкомпьютерные технологии в разработке методов решения обратных задач в УЗИ-томографии" // Вычислительные методы и программирование. 2012. Т.13. №1, с.235-238.
11. Вад. В.Воеводин, С.Л. Овчинников, С.Ю. Романов "Разработка высокоэффективных масштабируемых программ в задаче ультразвуковой томографии" // Вычислительные методы и программирование: новые вычислительные технологии. 2012. Т.13. №1, с.307-315.
12. А.В. Гончарский, С.Ю. Романов "О двух подходах к решению коэффициентных обратных задач для волновых уравнений" // Журн. вычисл. матем. и матем. физики. 2012. 52, № 2, с.1–7.
13. I. Foster "Designing and building parallel programs: concepts and tools for parallel software engineering". Reading: Addison Wesley, 1995.