

Н. И. Хохлов, И. Б. Петров

## Решение больших задач сейсмологии на высокопроизводительных вычислительных системах

**Аннотация.** В данной работе рассматривается созданный программный комплекс для моделирования задач распространения динамических волновых возмущений в твердых телах. Комплекс работает на двумерных и трехмерных структурных блочных сетках с наличием неоднородностей. Для численного интегрирования применяются сеточно-характеристические и конечно-объемные методы 2-4 порядка точности. Код распараллелен используя технологию MPI. Реализована низкоуровневая оптимизация кода с использованием потоковых SIMD инструкций центральных процессоров SSE и AVX. В системах с общей памятью алгоритм распараллелен используя технологию OpenMP. Также код распараллелен используя технологию CUDA. Приведены результаты тестирования ускорения и примеры расчетов.

**Ключевые слова и фразы:** MPI, CUDA, OpenMP, SIMD, SSE, AVX, сеточно-характеристический метод, системы гиперболических уравнений, механика деформируемого твердого тела.

### Введение

Численное моделирование распространения динамических волновых возмущений в твердых телах используется при решении широкого круга задач. К таким задачам относятся задачи сейсморазведки, сейсмологии и сейсмостойкости. Роль численного моделирования в каждой из данных областей очень важна. Численное моделирование распространения сейсмических волн представляет существенную часть работ при проведении геологоразведки в нефтяной отрасли. Математическое моделирование проводится в различных геологических средах, в том числе в слоистых средах и в средах с наличием различных неоднородностей (например, трещины или каверны). Задачи такого рода представляются очень ресурсоемкими с точки зрения вычислительных ресурсов. Область вычисления, как правило, представляет собой сейсмический куб с длиной ребра от 1 км до 10

км. В тоже время, неоднородности могут быть размером в несколько метров. При моделировании задач сейсмостойкости также приходится сталкиваться с множеством неоднородностей. Параметры зданий, такие как толщина стен, размеры проемов намного меньше размеров расчетных областей, включающих в себя порой большие массивы породы, размерами более 10 км вдоль одного направления. Таким образом, расчетная сетка должна быть достаточно подробной, чтобы иметь возможность правильно выделить все неоднородности. Для получения достаточной точности расчета и учета большого числа неоднородностей требуется использование больших вычислительных сеток, в реальных расчетах используются сетки размерами до нескольких десятков миллиардов узлов.

В данной работе рассматривается созданный программный комплекс для моделирования задач распространения динамических волновых возмущений в твердых телах. Комплекс работает на двумерных и трехмерных структурных блочных сетках с наличием неоднородностей. Для численного интегрирования применяются сеточно-характеристические и конечно-объемные методы 2-4 порядка точности.

Код распараллелен используя технологию MPI. Реализована низкоуровневая оптимизация кода с использованием потоковых SIMD инструкций центрального процессора SSE и AVX. В настоящее время достигнута эффективность распараллеливания до 70 % используя технологию MPI при масштабировании до 16 тысяч вычислительных ядер. В системах с общей памятью алгоритм распараллелен используя технологию OpenMP. Также код распараллелен используя технологию CUDA, что дает ускорение до 50 раз по сравнению с одним ядром CPU. Программа может использовать несколько карточек в рамках одного хоста.

В данной работе рассмотрены результаты одного и того же алгоритма используя различные технологии. Приведены тесты распараллеливания до 16 тысяч ядер CPU и 6 устройств CUDA. Приведены результаты тестовых расчетов.

## **1. Математическая модель**

### **1.1. Определяющие уравнения**

Сформулируем основные уравнения линейной динамической теории упругости, которым подчиняется состояние бесконечно малого

объема линейно-упругой среды. Рассмотрим нестационарные уравнения теории упругости для случая трех переменных, в некоторой ортонормированной системе координат  $(x_1, x_2, x_3)$ :

$$(1) \quad \begin{aligned} \rho \dot{v}_i &= \nabla_j \sigma_{ij}, \\ \dot{\sigma}_{ij} &= q_{ijkl} \dot{\epsilon}_{kl} + F_{ij}, \end{aligned}$$

здесь  $\rho$  – плотность среды,  $v_i$  – компоненты вектора скорости смещения,  $\sigma_{ij}$  и  $\epsilon_{ij}$  – компоненты тензоров напряжений Коши и деформации,  $\nabla_j$  – ковариантная производная по  $j$ -й координате,  $F_{ij}$  – добавочная правая часть. Вид компонент тензора 4-го порядка  $q_{ijkl}$  определяется реологией среды. Для линейно-упругого случая они имеют вид:

$$q_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}).$$

В этом соотношении, которое обобщает закон Гука,  $\lambda$  и  $\mu$  – параметры Ляме, а  $\delta_{ij}$  – символ Кронекера.

Первая строка в системе уравнений (1) представляет три уравнения движения, вторая – шесть реологических соотношений. Вектор искомого функций, состоящий из 9-ти компонент имеет вид:

$$\mathbf{u} = \{v_1, v_2, v_3, \sigma_{11}, \sigma_{12}, \sigma_{13}, \sigma_{22}, \sigma_{23}, \sigma_{33}\}^T.$$

Тогда перечисленные модели твердого тела допускают запись системы уравнений (1) динамики деформируемого твердого тела в матричном виде [1]:

$$(2) \quad \frac{\partial \mathbf{u}}{\partial t} = \sum_{j=1}^3 \mathbf{A}_j \frac{\partial \mathbf{u}}{\partial x_j},$$

где  $\mathbf{A}_j$  – матрицы размера  $9 \times 9$ .

## 1.2. Численные методы

Для численного моделирования задач динамики деформируемого твердого тела широко применяется сеточно-характеристический метод [2, 3]. Вначале применяется метод расщепления по пространственным координатам, в результате чего имеем три одномерных системы:

$$(3) \quad \frac{\partial \mathbf{u}}{\partial t} = \mathbf{A}_j \frac{\partial \mathbf{u}}{\partial x_j}, \quad j = 1, 2, 3.$$

Каждая из этих систем является гиперболической и обладает полным набором собственных векторов с действительными собственными значениями, поэтому каждую из систем можно переписать в виде:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{\Omega}_j^{-1} \mathbf{\Lambda}_j \mathbf{\Omega}_j \frac{\partial \mathbf{u}}{\partial x_j},$$

где матрица  $\mathbf{\Omega}_j$  - матрица составленная из собственных векторов,  $\mathbf{\Lambda}_j$  - диагональная матрица, элементами которой являются собственные значения. Для всех координат матрица  $\mathbf{\Lambda}$  имеет вид:

$$\mathbf{\Lambda} = \text{diag}\{c_p, -c_p, c_s, -c_s, c_s, -c_s, 0, 0, 0\},$$

где  $c_p = \sqrt{(\lambda + 2\mu)/\rho}$  - продольная скорость звука в среде,  $c_s = \sqrt{\mu/\rho}$  - поперечная скорость звука.

После замены переменных  $\nu = \mathbf{\Omega} \mathbf{u}$  каждая из систем (3) распадается на девять независимых скалярных уравнений переноса (индекс  $j$  далее опускается, где это возможно):

$$\frac{\partial \nu}{\partial t} + \mathbf{\Lambda} \frac{\partial \nu}{\partial x} = 0.$$

Одномерные уравнения переноса решаются с помощью метода характеристик, либо обычными конечно-разностными схемами.

После того, как все компоненты  $\nu$  перенесены, восстанавливается само решение:

$$\mathbf{u}^{n+1} = \mathbf{\Omega}^{-1} \nu^{n+1}.$$

В данной работе использовались TVD-разностные схемы [4] 2-го порядка точности. В программе реализовано 15 различных лимитеров [5], в расчетах в основном использовался ограничитель superbee, предложенный в [6]:

$$\phi_{sb}(r) = \max[0, \min(2r, 1), \min(r, 2)].$$

Также использовались сеточно-характеристические монотонные разностные схемы, принцип построения которых описан в [7]. В программе реализованы схемы 2-4 порядка точности, большинство расчетов проводилось используя схему 4-го порядка точности. Приведем

ее для численного решения одномерного линейного уравнения упругости  $u_t + \lambda u_x = 0$ ,  $\sigma = \lambda \tau / h$ :

$$\begin{aligned} u_m^{n+1} &= u_m^n - \sigma(\Delta_1 - \sigma(\Delta_2 - \sigma(\Delta_3 - \sigma\Delta_4))), \\ \Delta_1 &= \frac{1}{24}(-2u_{m+2}^n + 16u_{m+1}^n - 16u_{m-1}^n + 2u_{m-2}^n), \\ \Delta_2 &= \frac{1}{24}(-u_{m+2}^n + 16u_{m+1}^n - 30u_m^n + 16u_{m-1}^n - u_{m-2}^n), \\ \Delta_3 &= \frac{1}{24}(2u_{m+2}^n - 4u_{m+1}^n + 4u_m^n - 2u_{m-2}^n), \\ \Delta_4 &= \frac{1}{24}(u_{m+2}^n - 4u_{m+1}^n + 6u_m^n - 4u_{m-1}^n + u_{m-2}^n). \end{aligned}$$

Кроме того, используется сеточно-характеристический критерий монотонности [7], что дает в областях с резкими градиентами решения 3-й порядок точности, а в областях с гладким изменением решения 4-й порядок точности.

## 2. Программный комплекс

Для моделирования задач распространения волновых возмущений в гетерогенных средах был создан программный комплекс, позволяющий численно решать поставленные задачи, используя сеточно-характеристические методы разных порядков аппроксимации.

При создании программного комплекса одним из основных требований было возможность проведения расчетов на достаточно больших расчетных сетках (более 10 млрд узлов), чтобы была возможность проводить моделирование в больших расчетных областях. В этом случае возникают значительные потребности, как в ресурсах процессорного времени, так и оперативной памяти ЭВМ. В работе рассмотрен ряд технологий, позволяющих существенно ускорить время работы вычислительного модуля.

### 2.1. Оптимизация работы с памятью

Рассмотрим базовую организацию работы расчетного алгоритма. В работе используются явные сеточные методы, переходы от одного шага по времени к следующему происходит путем обхода всей расчетной сетки и вычисления значения в новом узле исходя из значения в исходном узле и некоторой его окрестности, называемой шаблоном разностной схемы. Обход сетки организуется путем двух вложенных циклов. Исследование такой реализации показало, при достаточно большом размере расчетной сетки возникает определенное число промахов кеша процессора, что ведет к увеличению времени работы программы. Путем изменения алгоритма обхода расчетной сетки, а также за счет использования дополнительных массивов памяти и

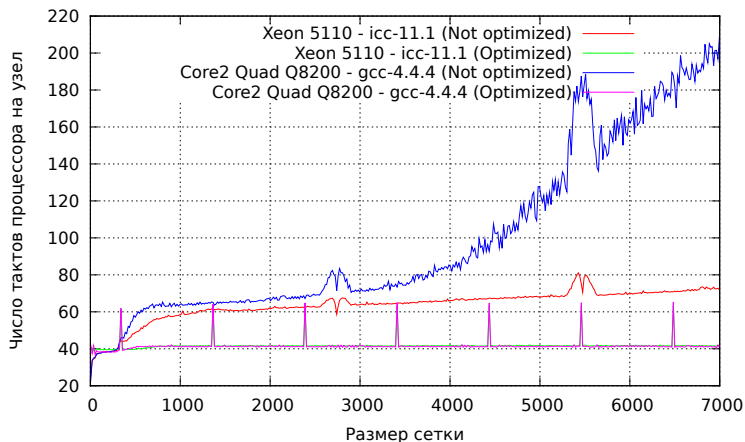


Рис. 1. Зависимость времени переноса одного узла разностной схемы от размера сетки

принудительного кэширования данных, удалось оптимизировать работу с памятью, что позволило в ряде случаев ускорить работы алгоритма на больших расчетных сетках. Величина ускорения зависит от модели центрального процессора.

На рис. 1 приведена зависимость числа тактов процессора, затраченных на перенос одного узла расчетной сетки в зависимости от размера сетки. Показаны результаты для базовой (Not optimized) и оптимизированной версии (Optimized). Размер сетки указан по одной из осей, тест проводился на двумерной задаче. В тесте участвовали следующие процессоры и компиляторы (везде указан размер кеша L2):

- Intel Xeon 5110, 4 МБ, icc-11.1;
- Intel Core2 Quad Q8200, 2 МБ, gcc-4.4.4.

При компиляции тестовых программ использовались следующие опции компиляторов: gcc -O3; icc -O3. Как видно из графиков, в зависимости от модели процессора ускорение может достигать 5 раз.

## 2.2. Векторизация при помощи SIMD инструкций

Данный алгоритм также был оптимизировать задействовав потоковые SIMD-расширения центрального процессора (SSE). SSE вклю-

чает в архитектуру несколько расширенных регистров (8 на современных процессорах) размером 128 бит, либо 256 бит в случае технологии AVX. Преимущество в производительности достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными. В таком случае блоком SSE осуществляется распараллеливание вычислительного процесса между данными.

Для возможности оперировать данными в SSE регистрах потребовалось изменение формата хранения расчетной сетки. Чтобы произвести какие-либо операции используя SSE требуется поместить в регистр необходимые расчетные данные, в случае когда эта операция занимает много времени все преимущество потоковых инструкций теряется. В связи с этим возникает необходимость быстрой возможности загрузки и выгрузки данных в SSE регистры. Изначально данные хранились в виде массива структур, или AOS (Array of structures), при таком подходе необходимо по одному числу типа float или double копировать данные в SSE, затем производить вычисление и копировать данные обратно. При этом переменные, предназначенные для использования в одном SSE регистре будут расположены в памяти не последовательно.

Для ускорения данного процесса необходимо расположить все компоненты структуры в памяти последовательно, и затем подгружать их по необходимости в SSE регистры. Был произведен переход к хранению структуры массивов, или SOA (Structure of arrays), что позволило быстрее осуществлять копирование в SSE регистры.

На рис. 2 показана время работы различных версий алгоритма на одних и тех же данных.

Результаты приведены для базовой реализации, версии оптимизированной по памяти и использующей систему хранения SOA (Cache-friendly), а также двух вариантов потоковых инструкций SSE и AVX. Размер расчетной сетки, использованной в тесте –  $4096 \times 4096$ . Тестирование проводилось для различных вариантов процессоров и компиляторов, все процессоры поддерживали инструкции AVX:

- Intel Xeon E5-2690, gcc-4.4.6;
- Intel Xeon E5-2690, icc-12.1.0;
- AMD Opteron 6276, gcc-4.7.2.

Также менялся тип используемых данных: float и double.

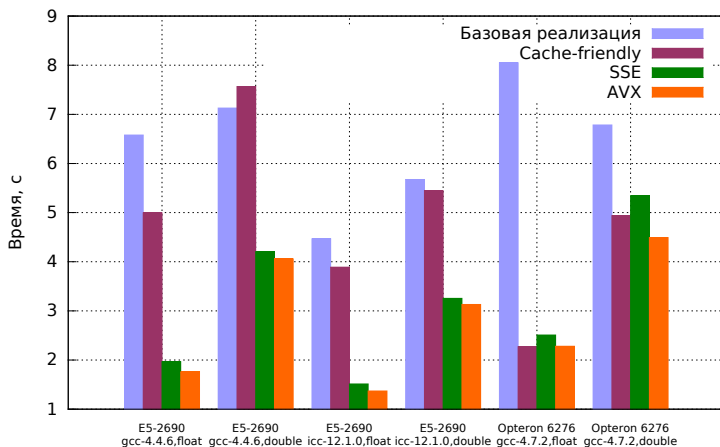


Рис. 2. Время работы различных вариантов оптимизации

По результатам тестирования видно ускорение до семи раз при использовании чисел одинарной точности и до двух раз при числах с двойной точностью. В зависимости от версии компилятора и процессора результат несколько отличается, в частности в ряде случаев имеется автоматическая векторизация. Скорость работы алгоритма, использующего AVX инструкции, практически не отличается от скорости работы алгоритма, построенного на использовании инструкций SSE. Эта особенность по-видимому связана с тем, что скорость работы памяти при загрузке данных в регистры AVX такая же как и у SSE.

### 2.3. Распараллеливание MPI

Для работы в системах с распределенной памятью программный комплекс распараллелен используя технологию MPI. При распараллеливании использовались стандартные алгоритмы декомпозиции расчетной области и обмена приграничными ячейками.

Одно из основных требований к алгоритму – работа на большом числе вычислительных ядер (тысячи) для обеспечения приемлемого времени расчета. На рис. 3 приведены результаты тестирования масштабирования алгоритма при увеличении числа расчетных ядер от 128 до 16384.



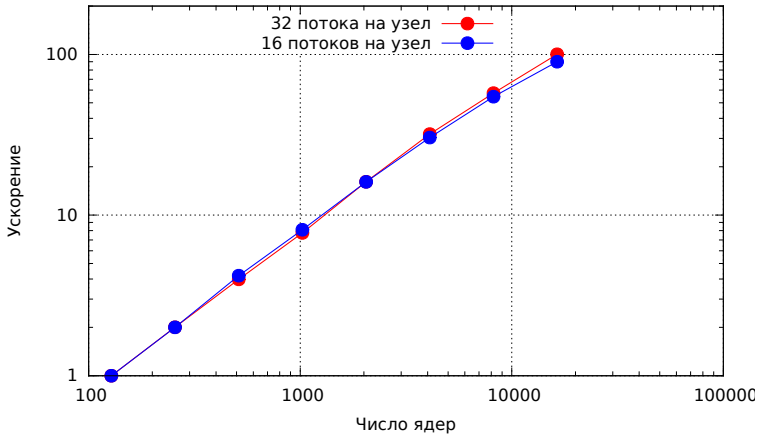


Рис. 3. Масштабируемость параллельного алгоритма

В тесте использовалась расчетная сетка размером  $1000 \times 1000 \times 1000$  узлов. Тестирование проводилось на кластере НЕСТoR. Данный суперкомпьютер состоит из 2816 вычислительных узлов. Каждый из узлов оснащен двумя процессорами 16-core AMD Opteron 2.3GHz Interlagos. Оперативная память составляет 32 ГБ на узел. Ускорение составляет до 100 раз при увеличении числа вычислительных ядер в 128 раз. На рис. 4 приведены графики эффективности работы параллельного алгоритма.

Тестирование показало, что эффективность составляет около 70-80 %, что является хорошим результатом для такого количества вычислительных ядер.

Результаты несколько отличаются при запуске 16 и 32 потоков на один узел суперкомпьютера. Это связано с особенностями работы железа вычислительных узлов. В случае, когда на каждом из ядер висит свой процесс и интенсивно использует шину памяти, происходит провал в скорости работы данной шины. В результате чего, если на одном узле запускать по 32 потока, происходит провал с скорости работы с памятью. Данная особенность относится только к текущему вычислительному комплексу и не является особенностью работы алгоритма.

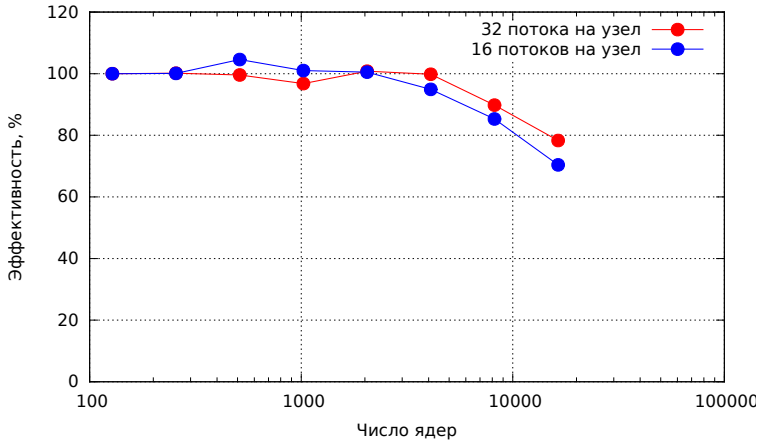


Рис. 4. Эффективность параллельного алгоритма

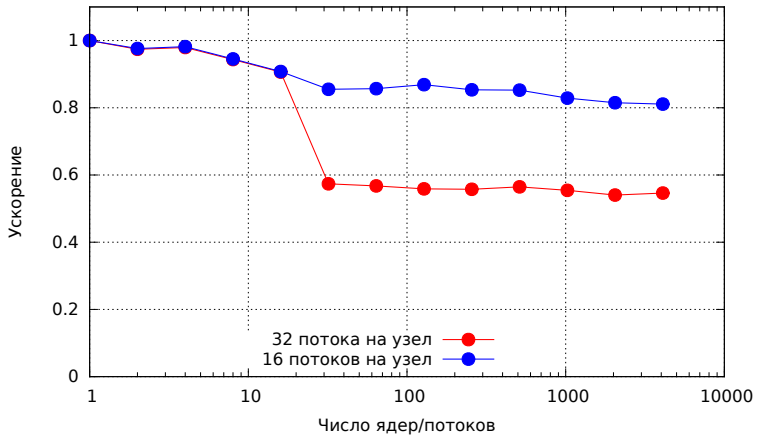


Рис. 5. Ускорение параллельного алгоритма при увеличении размера задачи

Еще один тест – масштабируемость размеров задачи при одновременном увеличении числа ядер. В данном тесте на каждое вычислительное ядро приходится одинаковое число узлов расчетной сетки. На рис. 5 приведены результаты такого тестирования.

Тестирование проводилось при изменении числа вычислительных ядер от 1 до 4096. На графике виден характерный провал при расчете в 32 потока, связанный с особенностями работы железа, описанными выше. Максимальный размер сетки в данном тесте – около 33 млрд узлов. Как можно видеть, алгоритм показывает хорошее ускорение и может вести расчет на больших задачах.

## 2.4. Распараллеливание OpenMP

Алгоритм также распараллелен в системах с общей памятью используя технологию OpenMP. В данном направлении работа только начата и при увеличении числа ядер от 1 до 32 алгоритм показывает эффективность около 60 %. Столь низкие показатели связаны с недостаточно эффективной работой с памятью, вычислительным ядрам приходится много обращаться к областям памяти у соседних ядер, что вызывает замедление расчета.

Ведутся работы по повышению эффективности работы с технологией OpenMP.

## 2.5. Распараллеливание CUDA

Алгоритм также был распараллелен на графических GPGPU процессорах NVidia используя технологию CUDA. Потребовалось полное переписывание части расчетного модуля под данную архитектуру. На текущий момент достигнуто ускорение около 80 раз по сравнению с одним ядром CPU используя числа с одинарной точностью и 30 раз используя числа с двойной точностью. В тестах использовались карточки nVidia Tesla s2050 и процессоры Intel Xeon X5670 2.7 ГГц.

Текущая реализация алгоритма позволяет использовать несколько GPU карточек в рамках одного расчетного узла. Это сделано средствами технологии CUDA. На рис. 6 приведен тест ускорения при увеличении числа карточек от 1 до 6, в зависимости от размера расчетной сетки.

При небольших размерах сетки алгоритм не дает прироста скорости работы при увеличении числа устройств. Это связано с большими накладными расходами на взаимодействие между устройствами. Однако на больших расчетных сетках реализация дает ускорение до 4 раз при увеличении числа устройств в 6 раз.

Ведутся работы по переносу алгоритма на гибридные архитектуры MPI+CUDA.

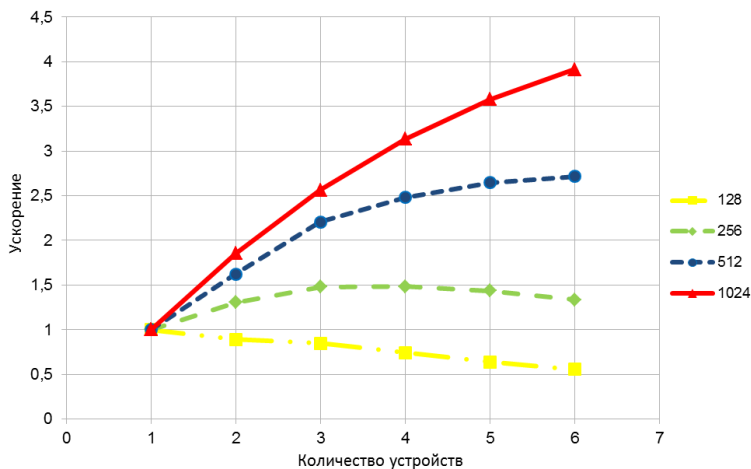


Рис. 6. Распараллеливание на нескольких карточках

### 3. Результаты расчетов

Приведем некоторые результаты тестовых расчетов, выполненных на созданном программном комплексе.

#### 3.1. Моделирование распространения возмущений от гипоцентра землетрясения

В данной работе исследуется процесс распространения упругих волн, возникающих в процессе землетрясения в гетерогенных средах.

Для моделирования очага землетрясения была выбрана сдвиговая модель возмущения в гипоцентре. В этой модели задавалась прямоугольная область шириной 40 м и длиной 500 м с некоторой ненулевой скоростью. Одна часть области двигается в одну сторону, другая в - противоположную. Такая модель физически соответствует ситуации, когда в земной коре существует разлом, по которому происходит подвижка при землетрясении. В данной работе проводилось сравнение распространения возмущений от двух типов землетрясений с горизонтальным и вертикальным сдвигом.

Выбор величины модуля скорости был произведён путём сопоставления данных моделирования и реальных экспериментальных

$H$ , м	$V_p$ , км/с	$V_s$ , км/с
300	4,19	2,79
400	4,65	3,1
500	5,85	3,5
2800	6,13	3,9

ТАБЛИЦА 1. Параметры слоистой среды

данных. Данные были взяты из землетрясения произошедшего вблизи Guadalupe Victoria 6 июля 2010 года. По оценкам автоматической системы, очаг землетрясения располагался на глубине 1,5 км. Из карты максимальных скоростей известно, что максимальные скорости грунта на поверхности земли составили порядка 1 см/с. Путём задания различных по величине скоростей в очаге землетрясения и численного моделирования было установлено, что для получения амплитуд скоростей на поверхности того же порядка, необходимо задавать скорость начального возмущения равной 10 см/с.

Было проведено моделирование распространения волн при землетрясении, происходящем в слоистой среде. Плотность всех слоёв считалась постоянной и равной 2500 кг/м<sup>3</sup>. Толщины слоёв, а также скорости продольной и поперечной волн приведены в таблице 1.

На рис. 7 представлены распределения модуля скорости при распространении волн от гипоцентра до дневной поверхности для землетрясения с горизонтальным и вертикальным сдвигами.

Результаты представлены для одних и тех же моментов времени. Хорошо заметен сильный второй фронт поперечной волны, он имеет несколько меньшую скорость и доходит до земной поверхности позднее. Также заметны отражения от различных геологических слоёв. Картины от различных видов возмущений отличаются и имеют хорошее качественное сходство с экспериментальными данными.

### 3.2. Задачи сейсмостойкости

Рассмотрим процесс прохождения плоских продольной и поперечной волн через наземный объект, в нашем случае небольшое сооружение из бетона. Фронт волны параллелен свободной поверхности земли, распространение волны от центра земли к поверхности. В данном расчете такой импульс рассматривался как один пик от землетрясения.

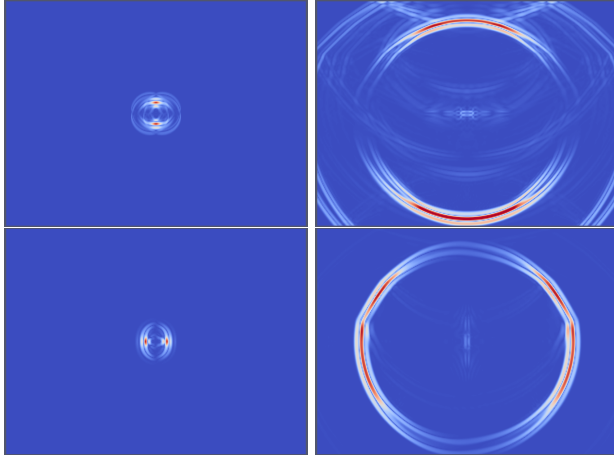


Рис. 7. Результаты моделирования для землетрясения в горизонтальном и вертикальном сдвиге (сверху вниз)

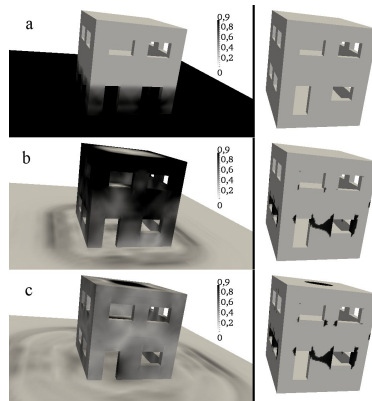


Рис. 8. Результаты расчета для продольной волны: а)  $t = 0,0013$  с, б)  $t = 0,0039$  с, в)  $t = 0,0079$  с

На рис. 8 показано прохождение продольной волны в различные моменты времени. Слева отображен модуль скорости среды, а справа цветом показаны участки где хотя бы раз за расчет выполнялось условие Мизеса, т. е. возможны разрушения.

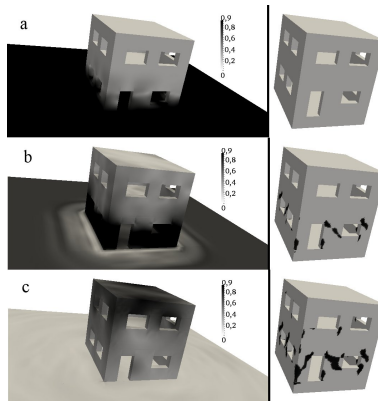


Рис. 9. Результаты расчета для поперечной волны: а)  $t = 0,0020$  с, б)  $t = 0,0039$  с, в)  $t = 0,0186$  с

Как видно из рисунков, при таком типе волны основные разрушения происходят на первом этаже и откол на потолке второго этажа здания. При землетрясениях основной урон происходит от поперечных волн, поскольку их амплитуда намного больше. Результаты аналогичного расчета для поперечной волны представлены на рис. 9.

Для продольной волны картина похожая, но основные разрушения находятся в углах, дверных и оконных проемах. Нет поврежденных потолков, как это хорошо было видно в случае продольной волны.

#### 4. Заключение

В работе представлено описание программного комплекса для решения широкого круга динамических волновых задач. Основные направления – задачи сейсмики, сейсмостойкости, задачи геологоразведки. Комплекс распараллелен и оптимизирован для работы на различных архитектурах и позволяет производить моделирование больших задач за приемлемое время.

#### Список литературы

- [1] Петров И. Б., Холодов А. С. Численное исследование некоторых динамических задач механики деформируемого твердого тела сеточно-характеристическим методом: // Журнал вычислительной математики и математической физики, 1984. Т. 5, № 24, с. 722–739. (russian) ↑1.1

- [2] Магомедов К. М., Холодов А. С. *О построении разностных схем для уравнений гиперболического типа на основе характеристических соотношений* : . // Журнал вычислительной математики и математической физики, 1969. Т. 9, № 2, с. 373–386. (russian) ↑1.2
- [3] Магомедов К. М., Холодов А. С. *Сеточно-характеристические численные методы*. Москва : Наука, 1988. ↑1.2
- [4] Harten A. *High Resolution Schemes for Hyperbolic Conservation Laws* // Journal of Computational Physics, 1997. Vol. 135, no. 2, p. 260 - 278. ↑1.2
- [5] Петров И. Б., Хохлов Н. И. *Сравнение TVD лимитеров для численного решения уравнений динамики деформируемого твердого тела сеточно-характеристическим методом* // Математические модели и задачи управления. Сборник научных трудов, 2011, с. 104–111. ↑1.2
- [6] Roe P. *Characteristic-based schemes for the Euler equations* // Annual review of fluid mechanics, 1986. Vol. 18, no. 1, p. 337–365. ↑1.2
- [7] Холодов А. С., Холодов Я. А. *О критериях монотонности разностных схем для уравнений гиперболического типа* // Журнал выч. математики и мат. физики, 2006. Т. 46, № 9, с. 1560–1588. ↑1.2

*Об авторах:*

### **Николай Игоревич Хохлов**



Выпускник МФТИ 2008, заведующий лабораторией на кафедре информатики МФТИ, кандидат физико-математических наук. Автор более 10 научных работ. Область научных интересов: математическое моделирование волновых процессов в твердых телах, в задачах сейсмоки и сейсморазведки полезных ископаемых, параллельные вычисления, технологии программирования.

*e-mail:*

[k\\_h@inbox.ru](mailto:k_h@inbox.ru)

### **Игорь Борисович Петров**



Выпускник МФТИ 1976 года, член-корреспондент РАН, профессор, заведующий кафедрой информатики МФТИ. Автор более 200 научных работ, нескольких учебников и учебных пособий. Область научных интересов: математическое моделирование волновых процессов в твердых телах, задач сейсмоки и сейсморазведки полезных ископаемых, вопросов безопасности сооружений, математическое моделирование в медицине и биологии.

*e-mail:*

[petrov@mipt.ru](mailto:petrov@mipt.ru)



*Образец ссылки на эту публикацию:*

Н. И. Хохлов, И. Б. Петров. *Решение больших задач сейсмологии на высокопроизводительных вычислительных системах* // Программные системы: теория и приложения : электрон. научн. журн. 2013?. Т. 4?, № 4(17)?, с.??-??.

*URL:*

<http://psta.psiras.ru/read/>

ABSTRACT. *(in Russian)*.

*Key Words and Phrases:*