Разработка алгоритмического и программного обеспечения для моделирования вулканических структур на гибридном кластере

Сапетина Анна Федоровна

Новосибирский национальный исследовательский государственный университет (НГУ), Россия

e-mail: afsapetina@gmail.com

Рассматривается численное моделирование распространения упругих волн в 3D неоднородных средах на основе решения полной системы уравнений теории упругости с соответствующими начальными и граничными условиями. Моделирование проводится на основе конечно-разностного метода. Предлагаются способы распараллеливания программы для численных расчетов на графических картах средствами технологии CUDA. Разработана параллельная программа, использующая одну графическую карту, проведены тестовые расчеты. Проведена оценка полученного ускорения работы программы. Разработанная программа включена в программный комплекс, решающий эту же задачу и ориентированный на использование всех узлов гибридного кластера ССКЦ ИВМиМГ СО РАН.

Ключевые слова: 3D численное моделирование, упругие волны, GPU, CUDA, неоднородная среда, грязевые вулканы.

Введение. В настоящее время грязевые вулканы широко распространены на земном шаре. Не смотря на то, что они известны достаточно давно, степень их изученности значительно уступает изученности магматических вулканов. Существует множество гипотез о происхождении грязевых вулканов, и процессов происходящих в них. Для их проверки необходимо создание математических моделей и проведение детальных геофизических исследований грязевых вулканов.

Такие исследования проводятся российскими и иностранными коллективами ученых. Одним из путей изучения строения грязевых вулканов является активный вибросейсмический мониторинг. Проведение натурных геофизических экспериментов на грязевых вулканах позволяет получить некоторые представления о скоростных параметрах упругой среды, а также о геометрии изучаемого объекта. Но в процессе обработки результатов полевого эксперимента могут возникнуть интересные эффекты, требующие дальнейшего исследования. Создание математических моделей грязевых вулканов и дальнейшее моделирование сейсмических полей помогают в изучении реальных объектов.

В связи с тем, что реальная область исследования имеет довольно сложный рельеф, не всегда удается поставить площадную систему наблюдения для решения обратной задачи геофизики. Поэтому приходится решать набор прямых задач с целью определения параметров изучаемой среды, соответствующих экспериментальным наблюдениям на поверхности изучаемых грязевых вулканов.

В связи со сложностью и масштабом моделируемой области, решение задачи численного моделирования распространения упругих волн от сосредоточенного источника может требовать значительных вычислительных ресурсов. Поэтому необходима разработка комплекса параллельных программ для уменьшения времени расчета и возможностью моделирования «больших» 3D моделей упругих сред.

Караваевым Д.А. были разработаны и обоснованы методы численного моделирования сейсмических полей для 3D сложно построенных сред, характерных для грязевых вулканов. Им же разработан инструментарий для решения прикладных задач численного моделирования сейсмических полей, включающий построитель 3D моделей неоднородных упругих сред и параллельную программу для численного моделирования распространения упругих волн, реализованную на кластерах с MPP-архитектурой, с использованием технологий MPI и OpenMP [1, 2].

Стоит отметить консервативность в развитии алгоритмов по сравнению с современными вычислительными системами, которые развиваются быстрее. Потому возникает проблема подстройки алгоритмов под архитектуру.

В последние несколько лет стали активно применять графические процессоры для вычислений общего назначения. Их современная архитектура позволяет эффективно выполнять вычисления с высокой степенью параллелизма. В начале 2012 года в ССКЦ при ИВМиМГ СО РАН запущен гибридный кластер с графическими процессорами NVidia Tesla M2090. Также подобный вычислительный комплекс функционирует на базе Информационно-вычислительного центра НГУ.

Разработанная параллельная программа предназначена для численного моделирования распространения упругих волн в трехмерно-неоднородных упругих средах с использованием конечно-разностного метода на графической карте.

Постановка задачи. Численное моделирование распространения сейсмических волн в сложно построенных упругих неоднородных средах проводится на основе решения полной системы уравнений теории упругости с соответствующими начальными и граничными условиями, записанной в терминах вектора скоростей смещений $\vec{u} = (U,V,W)^T$ и тензора напряжений $\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$:

$$\rho \frac{\partial \vec{u}}{\partial t} = [A]\vec{\sigma} + \vec{F}(t,x,y,z), \qquad \frac{\partial \vec{\sigma}}{\partial t} = [B]\vec{u},$$

$$A = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}, \qquad B = \begin{bmatrix} (\lambda+2\mu)\frac{\partial}{\partial x} & \lambda\frac{\partial}{\partial y} & \lambda\frac{\partial}{\partial z} \\ \lambda\frac{\partial}{\partial x} & (\lambda+2\mu)\frac{\partial}{\partial y} & \lambda\frac{\partial}{\partial z} \\ \lambda\frac{\partial}{\partial x} & \lambda\frac{\partial}{\partial y} & (\lambda+2\mu)\frac{\partial}{\partial z} \\ \mu\frac{\partial}{\partial y} & \mu\frac{\partial}{\partial x} & 0 \\ \mu\frac{\partial}{\partial z} & 0 & \mu\frac{\partial}{\partial x} \\ 0 & \mu\frac{\partial}{\partial z} & \mu\frac{\partial}{\partial y} \end{bmatrix}$$

λ, μ – параметры Ламе, ρ – плотность. Предполагается, что параметры упругой среды зависят от трех пространственных переменных *X, Y* и *Z*.

Начальными условиями являются: $\sigma_{xz}|_{t=0} = 0$, $\sigma_{yz}|_{t=0} = 0$, $\sigma_{xy}|_{t=0} = 0$, $\sigma_{xx}|_{t=0} = 0$, $\sigma_{yy}|_{t=0} = 0$, $\sigma_{zz}|_{t=0} = 0$, $U(x,y,z)|_{t=0} = 0$, $V(x,y,z)|_{t=0} = 0$, $W(x,y,z)|_{t=0} = 0$, а граничными : $\sigma_{xz}|_{z=0} = 0$, $\sigma_{yz}|_{z=0} = 0$, $\sigma_{zz}|_{z=0} = 0$.

В данной постановке предполагается, что правая часть (массовая сила) имеет следующий вид: $\vec{F}(t,x,y,z) = F_x\vec{i} + F_y\vec{j} + F_z\vec{k}$.

В качестве области моделирования рассматривается параллелепипед, одна из граней которого является свободной поверхностью (плоскость z = 0).

Метод решения задачи. Метод решения поставленной задачи основан на использовании конечно-разностного метода. Этот метод является наиболее применяемым в случае трехмерных динамических задач теории упругости. Алгоритм построения конечно-разностной схемы наиболее подробно изложен в статье [3]. Расчет сеточных коэффициентов в разностной схеме проводится на основе интегральных законов сохранения (поскольку параметры λ, μ и ρ могут быть разрывными).

Используемая конечно-разностная схема имеет второй порядок аппроксимации по времени и пространству. Общий вид некоторых уравнений конечно-разностной схемы следующий:

$$\frac{\rho_{i,j,k} + \rho_{i-1,j,k}}{2} \frac{u^{n+1}_{i-\frac{1}{2},j,k} - u^{n}_{i-\frac{1}{2},j,k}}{\tau} = \frac{(\sigma^{n+\frac{1}{2}}_{xxi,j,k} - \sigma^{n+\frac{1}{2}}_{xxi-1,j,k})}{\Delta x} + \frac{(\sigma^{n+\frac{1}{2}}_{xyi-\frac{1}{2},j+\frac{1}{2},k} - \sigma^{n+\frac{1}{2}}_{xyi-\frac{1}{2},j-\frac{1}{2},k})}{\Delta y} + + \frac{(\sigma^{n+\frac{1}{2}}_{xzi-\frac{1}{2},j,k+\frac{1}{2}} - \sigma^{n+\frac{1}{2}}_{xzi-\frac{1}{2},j,k-\frac{1}{2}})}{\Delta z} + f^{n}_{xi,j,k},$$

$$\frac{(\sigma^{n+\frac{1}{2}}_{xxi,j,k} - \sigma^{n-\frac{1}{2}}_{xxi,j,k})}{\tau} = (\lambda+2\mu)_{i,j,k} \frac{u^{n}_{i+\frac{1}{2},j,k} - u^{n}_{i-\frac{1}{2},j,k}}{\Delta x} + \lambda_{i,j,k} \frac{v^{n}_{i,j+\frac{1}{2},k} - v^{n}_{i,j-\frac{1}{2},k}}{\Delta y} + \lambda_{i,j,k} \frac{w^{n}_{i,j,k+\frac{1}{2}} - w^{n}_{i,j,k-\frac{1}{2}}}{\Delta z},$$

$$\frac{(\sigma^{n+\frac{1}{2}}_{xzi-\frac{1}{2},j,k-\frac{1}{2}} - \sigma^{n-\frac{1}{2}}_{xzi-\frac{1}{2},j,k-\frac{1}{2}})}{\tau} = \mu 1_{i-\frac{1}{2},j,k-\frac{1}{2}} \left(\frac{u^{n}_{i-\frac{1}{2},j,k} - u^{n}_{i-\frac{1}{2},j,k-1}}{\Delta z} + \frac{w^{n}_{i,j,k-\frac{1}{2}} - w^{n}_{i-1,j,k-\frac{1}{2}}}{\Delta x}\right).$$

Критерий устойчивости данной схемы приведен в работе [3] и имеет следующий вид:

$\tau \le \left(Vp_{\max} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \right)^{-1}$, где $Vp_{\max}$ – максимальная скорость распространения упругих волн.

Для схемы с одинаковыми шагами по пространственным переменным коэффициент $\mu 1$ для определения σ_{xz} имеет следующий вид: $\mu 1_{i-\frac{1}{2},j,k-\frac{1}{2}} = \left(\frac{1}{4} \left(\frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i-1,j,k}} + \frac{1}{\mu_{i,j,k-1}} + \frac{1}{\mu_{i-1,j,k-1}} \right) \right)^{-1}$. В данной работе рассматриваются только равномерные сетки.

Параллельная реализация программы численного моделирования. Для численного моделирования распространения упругих волн в трехмерно неоднородных упругих средах с использованием приведенного конечно-разностного метода, разработана программа, использующая одну графическую карту. Программная реализация основана на языке программирования C++ и технологии CUDA (Compute Unified Device Architecture), для организации процесса распараллеливания на GPU.

Для расчетов разработанной программе требуется набор входных файлов с информацией о моделируемой среде и параметрами расчета. Для моделирования упругих сред со сложной геометрией использовался построитель 3D моделей неоднородных упругих сред, разработанный Караваевым Д.А. С его помощью проводится подсчет параметров Ламе и плотности, соответствующих строящейся модели, в каждом узле расчетной сетки.

Результатом работы программы является набор бинарных файлов, содержащих снимки волнового поля в плоскостях, проходящих через точку расположения источника.

Разработанное программное обеспечение включено в программный комплекс, решающий эту же задачу и ориентированный на использование всех узлов гибридного кластера ССКЦ при ИВМиМГ СО РАН.

Гибридный кластер ССКЦ состоит из 40 вычислительных узлов HP SL390s G7, по три карты NVIDIA Tesla M2090 на архитектуре Fermi (Compute capability 2.0) на узле. У каждой карты 1 GPU с 512 ядрами и 6 ГБ оперативной памяти. Суммарно гибридный кластер содержит 80 процессоров (480 ядер) CPU и 120 процессоров (61440 ядер) GPU. Пиковая производительность – 85 Тфлопс.

Для использования такой архитектуры при решении поставленной задачи предложенным конечно разностным методом вычислительные мощности распределяются следующим образом. Для распараллеливания данной задачи используется декомпозиция области на слои вдоль направления одной из координатных осей (например, вдоль оси *Oz*). Каждый слой рассчитывался на отдельном узле (рис. 1), где в свою очередь он разбивался ещё на три подслоя вдоль той же координатной оси, по числу графических карт на узле. При реализации этого метода каждая графическая карта рассчитывает свою сеточную область на каждом временном шаге (используя полученную в ходе этой работы программу) независимо от других, за исключением точек, находящихся на границе между двумя соседними областями. Эти точки являются общими для каждой из областей и для продолжения счета необходимо производить обмен информацией об искомых величинах между «соседями». Обмены производятся при помощи технологии MPI (Message Passing Interface). При этом на каждой графической карте параллельная часть кода выполняется как большое количество нитей. Такой гибридный метод обеспечивает высокую степень параллелизма.

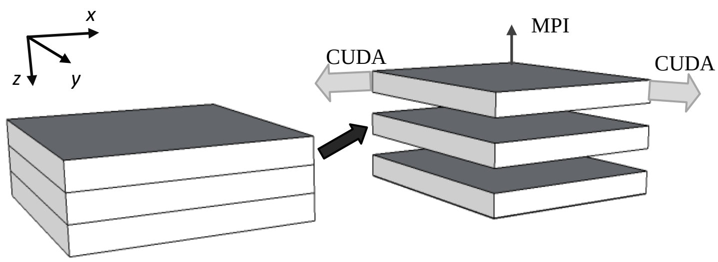

Рис. 1. Декомпозиция расчетной области в рамках узла.

Адаптация алгоритма для реализации с использованием GPU. Для решения задач CUDA использует очень большое количество параллельно выполняемых нитей, для эффективной загрузки GPU необходимо использовать много тысяч отдельных нитей. При этом все запущенные на выполнение нити организованы в иерархию, состоящую из сети блоков и сети нитей в блоке и задаваемую при запуске функции исполняемой на устройстве.

В ходе работы выбраны и реализованы два наиболее целесообразных варианта разбиения на сеть блоков и сеть нитей в блоке. Остальные варианты откинуты по причине малой степени параллелизма заложенной в них, либо если они существенно ограничивают размер разностной сетки.

Первый вариант (далее будем называть его вариантом с 3D сеткой) предусматривает трехмерную сетку блоков и трехмерную сетку нитей в блоке, и каждая нить осуществляет расчет компонент $\vec{\sigma}$ и $\vec{u}$ на текущем шаге по времени в одной точке разностной сетки. Индексы *i, j, k* сопоставляются глобальным индексам нити по трем компонентам (.x, .y, .z соответственно), что позволяет обратиться к каждому узлу разностной сетки.

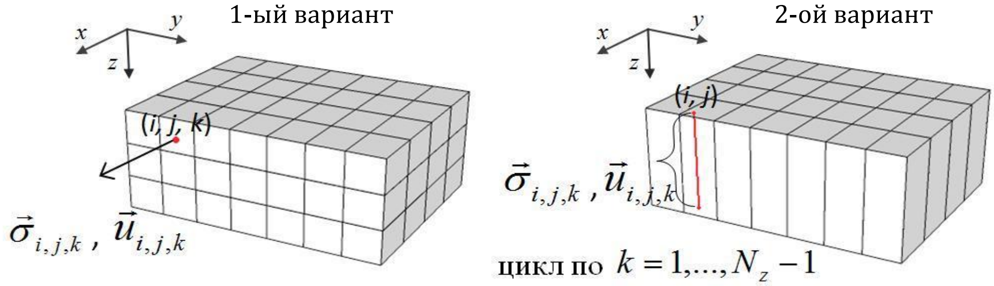

Рис. 2. Выбранные варианты размера и размерностей сетки блоков и блоков нитей

Второй вариант (далее будем называть его вариантом с 2D сеткой) предусматривает двухмерную сетку блоков и двухмерную сетку нитей в блоке, и каждая нить осуществляет расчет компонент $\vec{\sigma}$ и $\vec{u}$ на текущем шаге по времени, сопоставляя индексам *i* и *j* глобальные индексы нити (.x, .y соответственно) и проходя циклом по индексу *k*.

Для сравнения скорости работы описанных выше способов проведены сравнительные тесты. Результаты тестов получены с использованием вычислительных серверов HP SL390s G7 и пакета CUDA Toolkit 4.1 на базе Информационно-вычислительного центра НГУ. Для сравнения брались разные по размеру разностные сетки и разные размеры блока нитей. Количество шагов по времени (Nt) для всех тестов одинаково и равно 1000. Размеры блока обозначены как: A = DimBlock.x, B = DimBlock.y, C = DimBlock.z. Тесты проводились на кубических моделях, т. е. при Nx = Ny = Nz = N. Время указано в секундах.

Таблица 1. Время тестовых расчетов для варианта с 3D сеткой

A×B×C	*N* = 360	*N* = 280	*N* = 200
$2^3 = 8$	349,34	152,70	51,90
$4^3 = 64$	196,39	92,07	33,03
$8^3 = 512$	119,76	57,48	20,98
$10^3 = 1000$	165,92	77,10	28,00

Таблица 2. Время тестовых расчетов для варианта с 2D сеткой

A×B	*N* = 360	*N* = 280	*N* = 200
$4^2 = 16$	542,42	228,97	63,49
$8^2 = 64$	460,84	201,14	69,98
16×4 = 64	308,9	122,89	43,09
4×16 = 64	920,3	421,53	150,77
$16^2 = 256$	299,75	131,06	47,40
32×16 = 512	234,5	88,25	32,20
16×32 = 512	289,5	130,74	47,74
$32^2 = 1024$	123,98	53,41	20,00

Анализируя полученные данные можно сказать следующее. Чем больше общее количество нитей в блоке, тем меньше время работы программы. (Это можно объяснить тем, что планировщик задач, встроенный в GPU, тем эффективнее загружает потоковый мультипроцессор, чем больше нитей задано для исполнения на нем.) Вариант с 3D сеткой работает быстрее, чем вариант с 2D сеткой. Для максимальной рассматриваемой модели *N* = 360 при 512 нитях в блоке вариант с 3D сеткой практически в 2 раза быстрее, чем вариант с 2D сеткой.

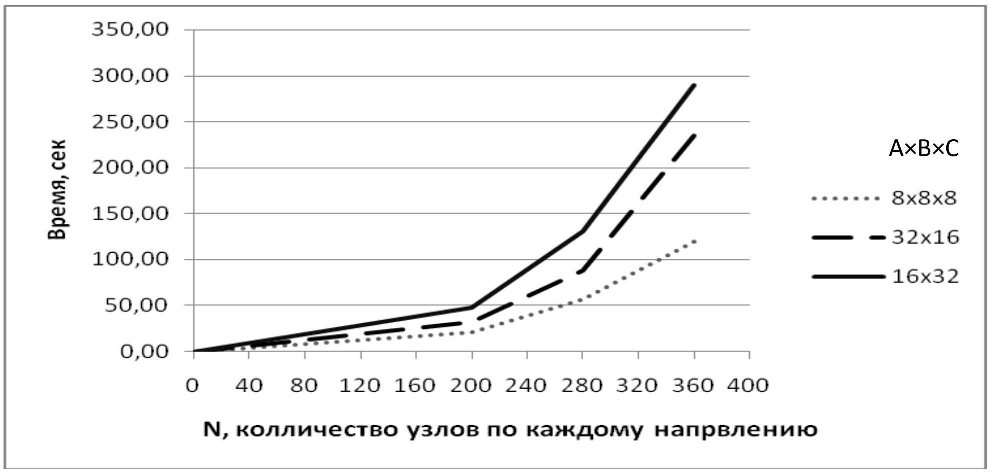

Рис. 3. Сравнение быстродействия программы на разных сетках блоков и нитей с общим количеством нитей в блоке равным 512

Рис. 4. Сравнение быстродействия программы на разных сетках блоков и нитей с общим количеством нитей в блоке равным 64

При исполнении на GPU нити в каждом блоке разбиваются на группы по 32, называемые warp'ами. Только нити в пределах одного warp'а выполняются физически одновременно. Все нити одного warp'а всегда принадлежат одному блоку. Нити из разных warp'ов могут находиться на разных стадиях выполнения программы. При этом управление warp'ами прозрачно осуществляет сам GPU.

По полученным данным видно что программа быстрее работает на сетке блоков, размер которых кратен длине warp'а. При совпадении длины warp'а с размерностью блока по компоненте x, время работы программы уменьшается, за счет сокращения времени доступа к памяти. В общем, чем ближе размерность блока по компоненте x к длине warp'а, тем меньше время работы программы (при условии, что длина warp'а кратна выбранному *DimBlock.x*). Это объясняется принципом формирования warp'а и выбранным способом расположения используемых трехмерных массивов в памяти.

Сравнение времени работы последовательной и параллельной программ. Проведено сравнение времени разработанной программы, использующей одно GPU, с аналогичной программой, работающей только на CPU. Рассматривалась сетка Nx = Ny = Nz = 120, Nt = 195. Параллельная программа запускалась на сервере HP SL390s G7 (использовала один CPU и один GPU). Последовательная версия запускалась на процессоре Intel Core2 Duo.

Таблица 3. Сравнение времени работы последовательной и параллельной программ

	Время, сек
Последовательная программа	15
Параллельная программа	1,32
Ускорение	11,38

Тестовый расчет. В качестве примера работоспособности разработанной программы численного моделирования распространения сейсмических волн представлены результаты в виде наглядных снимков волнового поля, которые получены в ходе работы программы.

В качестве тестовой задачи рассматривалась однородная изотропная среда со следующими параметрами:

Lx = Ly = Lz = 5.0 км – размеры моделируемой области,

ρ = 1.0 г/см^3 – плотность,

Vp = 2.0 км/с, Vs = 1.0 км/с – скорости распространения p-волн и s-волн соответственно,

несущая частота источника равна 5.0 Гц,

количество узлов Nx = Ny = Nz = 360,

шаг по времени τ = 0.004 с,

источник расположен в центре области, в точке с координатами (180, 180, 180).

На рис. 5 изображено изменение во времени искомой компоненты U вектора скорости смещения в плоскости z = 180 (проходящей через точку в которой расположен источник).

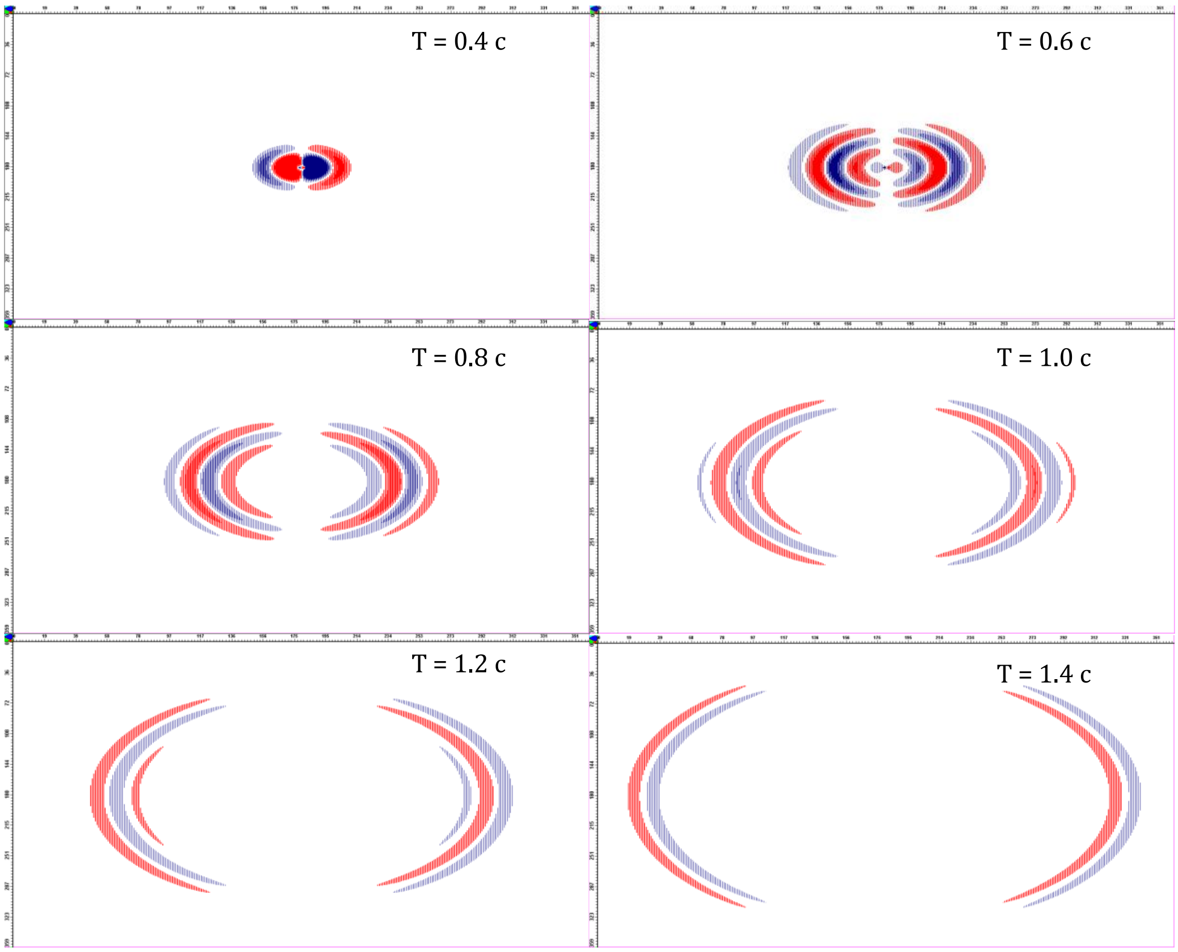

Рис 5. Тестовый расчет

Заключение. Разработана программа, решающая задачу численного моделирования распространения упругих волн в 3D неоднородных средах с использованием GPU.

Предложены два разных способа реализации данной программы, проведены сравнительные тесты для подбора оптимальных параметров сетки блоков и блоков нитей для запуска вычислений на GPU. Проведен анализ полученных данных и выделены основные зависимости времени работы программы от параметров сетки блоков и блоков нитей.

Получено ускорение во времени работы разработанной программы использующей GPU, по сравнению с аналогичной программой, использующей один CPU более чем в 10 раз. Это позволит ускорить процесс решения набора прямых задач распространения упругих волн с целью определения параметров изучаемой среды, соответствующих экспериментальным наблюдениям на поверхности изучаемых грязевых вулканов.

В будущем планируется исследовать время работы программного комплекса, использующего разработанную программу и ориентированного на использование всех вычислительных узлов гибридного кластера ССКЦ при ИВМиМГ СО РАН. Также планируется по возможности задействовать разделяемую память GPU для уменьшения времени работы программы. Предполагается применить вспомогательные алгоритмы для решения поставленной задачи для поглощения отражений упругих волн от границ расчетной области, например метод PML (Perfectly Matched Layers).

Автор выражает благодарность Караваеву Д.А и научному руководителю Глинскому Б.М.

Список литературы.

1. Караваев Д. А. Параллельная реализация метода численного моделирования волновых полей в трехмерных моделях неоднородных сред // Вестник Нижегородского университета им. Н. И. Лобачевского. 2009. № 6 (1). С. 203–209.
2. Глинский Б. М., Караваев Д. А., Ковалевский В. В., Мартынов В. Н. Численное моделирование и экспериментальные исследования грязевого вулкана «Гора Карабетова» вибросейсмическими методами // Вычислительные методы и программирование. Москва, 2010. Т. 11, С. 95–104.
3. Bihn M., Weiland T. A Stable Discretization Scheme for the Simulation of Elastic Waves // Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (IMACS 1997). T. 2, C. 75-80.
4. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графически процессоров: Пер. с англ. Слинкина А. А.// М.:ДМК Пресс, 2011. 232 с.
5. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA // М.:ДМК Пресс, 2010. 232 с.