

Технология моделирования распространения упругих волн в средах со сложной 3D геометрией поверхности на высокопроизводительных кластерах*

П.А. Титов¹

ИВМиМГ СО РАН¹

В работе рассмотрена технология моделирования 3D упругих волн на высокопроизводительных кластерах. Технология включает в себя математический метод и параллельный алгоритм. Отличительной особенностью метода является использование криволинейной трехмерной сетки, которая хорошо согласуется с геометрией строения свободной поверхности. Важным аспектом работы является со-дизайн – анализ эффективности параллельной реализации на всех этапах разработки алгоритма. Разработана программная реализация алгоритма для проведения расчетов на многоядерных системах. Рассмотрены особенности распараллеливания алгоритма. Представлены результаты тестовых расчетов для изучения ускорения и масштабируемости предлагаемого алгоритма.

Ключевые слова: Теория упругости, 3D, моделирование, криволинейная сетка, суперкомпьютер, со-дизайн

1. Введение

Моделирование 3D упругих волн в средах различного строения является важным аспектом создания геофизических 3D моделей, изучения особенностей волновых полей. Зачастую решить обратную задачу геофизики (восстановление строения и параметров среды по экспериментально полученным записям сигналов) сложно, и одним из методов является решение набора прямых задач (моделирование сейсмополей в среде с заданными параметрами и строением) с варьированием значений параметров и геометрии среды при сравнении реальных данных с результатами моделирования. Наиболее используемый метод для решения прямой задачи – метод конечных разностей (например, [1-4]). Отметим, что исследуемая область может иметь сложную 3D геометрию поверхности, поэтому важным отличительным моментом работы является построение криволинейной 3D сетки. Например, объектом исследования может быть магматический вулкан. Изучение строения среды и мониторинг подобного объекта является важной практической задачей. Теория построения и применения криволинейных сеток для решения реальных задач хорошо описана в работах [5-7]. Криволинейные сетки ранее применялись для задач моделирования 2D сейсмических полей в работе [8]. Работа автора является продолжением работы [9] и отличается от [8] способом построения сетки, а также решением задачи для трехмерного случая. Подобный подход к численному моделированию упругих волн подразумевает работу с большим количеством 3D данных. Учитывая масштабы области при решении реальных задач (десятки километров в каждом координатном направлении), задача численного моделирования становится невыполнима на персональной рабочей станции. Поэтому был разработан 3D параллельный алгоритм и его программная реализация для проведения расчетов на высокопроизводительных кластерах. Расчеты для данной работы проводились на кластере НКС-30T+GPU (Сибирский Суперкомпьютерный Центр, www2.sssc.ru). Также активно применяется и реализуется на практике идея со-дизайна. Его суть будет разобрана на каждом этапе работы. Раздел 2 включает в себя математическую постановку 3D задачи, а также ее преобразование

*Работы выполнены при поддержке грантов РФФИ 16-01-00455, 16-07-00434, а также программы фундаментальных исследований РАН 4.9 "Модельные и экспериментальные исследования вулканических структур методами активной и пассивной сейсмологии".

в связи с использованием криволинейной 3D сетки. Раздел 3 описывает численный метод для решения задачи. В разделе 4 разбираются особенности параллельного алгоритма и его программной реализации. В разделе 5 представлены результаты тестов ускорения работы и масштабируемости параллельного алгоритма в зависимости от количества вычислительных ядер. В разделе 6 перечислены основные результаты работы.

2. Постановка задачи

Задача является динамической. Решение прямой задачи геофизики связано с решением системы уравнений теории упругости в случае 3D среды. В данной работе рассматриваются упругие волны. Модель среды задаётся тремя параметрами: коэффициентами Ламэ λ , μ и плотностью ρ . Уже на данном этапе мы начинаем применять со-дизайн: выбирать вид постановки задачи, которая будет наиболее предпочтительна для численной реализации. Например, в работе [1] уравнения представлены в терминах скоростей смещений. В такой постановке необходимо работать с системой из 9 уравнений и, следовательно, с 9 параметрами на стадии численной реализации.

Поскольку данная работа существенным образом связана с разработкой метода для расчетов на криволинейных сетках, то использование подобного подхода было бы достаточно ресурсоемким. Поэтому было принято решение использовать систему, которая описывается с помощью трех компонент вектора смещений $(u, v, w)^T$. Согласно концепции со-дизайна, такой способ является более оптимальным с точки зрения экономии оперативной памяти и времени счета.

2.1. Уравнения теории упругости в смещениях

Для решения прямой задачи необходимо построить 3D модель среды: определить размеры и форму области, а также задать параметры λ, μ, ρ для каждой из составляющих частей среды (среда может быть неоднородной). На рисунке 1 изображен пример исследуемой 3D области с криволинейной свободной поверхностью. Введем обозначения: $\partial\Gamma$ – граница области по бокам и снизу, ∂S – свободная криволинейная поверхность.

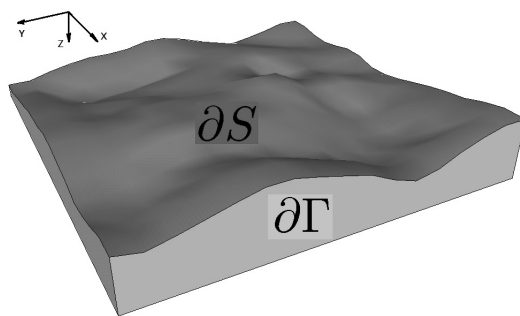


Рис. 1. Пример исследуемой области

В декартовой системе координат в переменных (x, y, z) уравнения имеют вид:

$$\begin{aligned} \rho \frac{\partial^2 u}{\partial t^2} &= \frac{\partial}{\partial x} \left((\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} + \lambda \frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial u}{\partial z} + \mu \frac{\partial w}{\partial x} \right) + F_x \\ \rho \frac{\partial^2 v}{\partial t^2} &= \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) + F_y \\ \rho \frac{\partial^2 w}{\partial t^2} &= \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial z} + \mu \frac{\partial w}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} + (\lambda + 2\mu) \frac{\partial w}{\partial z} \right) + F_z \end{aligned} \quad (1)$$

где F_x, F_y, F_z – компоненты массовой силы.

Граничные условия и начальные нулевые условия имеют вид :

$$u|_{\partial\Gamma} = v|_{\partial\Gamma} = w|_{\partial\Gamma} = 0, \quad \frac{\partial u}{\partial t} = \frac{\partial v}{\partial t} = \frac{\partial w}{\partial t} = 0 \quad (2)$$

Условия на свободной поверхности ∂S :

$$\begin{aligned} n_x \left((\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} + \lambda \frac{\partial w}{\partial z} \right) + n_y \left(\mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) + n_z \left(\mu \frac{\partial u}{\partial z} + \mu \frac{\partial w}{\partial x} \right) &= 0 \\ n_x \left(\mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) + n_y \left(\lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \frac{\partial w}{\partial z} \right) + n_z \left(\mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) &= 0 \\ n_x \left(\mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) + n_y \left(\mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) + n_z \left(\lambda \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} + (\lambda + 2\mu) \frac{\partial w}{\partial z} \right) &= 0 \end{aligned} \quad (3)$$

где (n_x, n_y, n_z) – единичный вектор нормали к свободной поверхности в каждой точке. Найдя решение системы из трех уравнений (1) с учетом начально-краевых условий (2),(3), мы тем самым сможем определить значения компонент вектора смещений $(u, v, w)^T$.

2.2. Построение криволинейной сетки и преобразование задачи

В работе автора используется построение криволинейной 3D сетки для расчетной области. Важнейшим моментом является ортогональность ребер ячеек возле свободной поверхности: все пересекающиеся ребра каждой криволинейной ячейки возле поверхности локально-ортогональны. Это означает, что в каждой точке поверхности вертикальные ребра ячеек перпендикулярны плоскости, касательной к поверхности в этой точке (рис. 2). В этих же точках ортогональны и ребра, соответствующие горизонтальным направлениям. Такой подход для решения подобных динамических задач геофизики впервые применен автором данной работы и позволяет увеличить точность аппроксимации граничных условий для численного решения задачи (для случая произвольной формы поверхности с первого порядка при использовании регулярной гексаэдральной сетки (работа [1]) до второго порядка при использовании криволинейных ячеек). С точки зрения со-дизайна, наиболее предпочтительным оказался метод трансфинитной интерполяции, разобранный в [5, стр. 62]. Он позволяет быстро строить 3D сетку при помощи параллельного 3D алгоритма, который был также разработан автором. Пример работы алгоритма можно видеть на рисунке 2. После построения сетки нужно преобразовать уравнения для новой системы координат (q^1, q^2, q^3) , где сетка является регулярной гексаэдральной. Используя замену $(x, y, z) \rightarrow (q^1, q^2, q^3)$,

уравнения (1)-(3) примут новый вид. Покажем для примера только преобразование (1):

$$\begin{aligned} \rho \frac{\partial^2 u}{\partial t^2} &= \sum_{j=1}^3 \frac{\partial q^j}{\partial x} \frac{\partial}{\partial q^j} \left((\lambda + 2\mu) \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial u}{\partial q^i} + \lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial v}{\partial q^i} + \lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial w}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial y} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial u}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial v}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial z} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial u}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial w}{\partial q^i} \right) + F_x \\ \\ \rho \frac{\partial^2 v}{\partial t^2} &= \sum_{j=1}^3 \frac{\partial q^j}{\partial x} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial u}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial v}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial y} \frac{\partial}{\partial q^j} \left(\lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial u}{\partial q^i} + (\lambda + 2\mu) \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial v}{\partial q^i} + \lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial w}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial z} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial v}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial w}{\partial q^i} \right) + F_y \\ \\ \rho \frac{\partial^2 w}{\partial t^2} &= \sum_{j=1}^3 \frac{\partial q^j}{\partial x} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial u}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial w}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial y} \frac{\partial}{\partial q^j} \left(\mu \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial v}{\partial q^i} + \mu \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial w}{\partial q^i} \right) + \\ &\quad \sum_{j=1}^3 \frac{\partial q^j}{\partial y} \frac{\partial}{\partial q^j} \left(\lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial x} \frac{\partial u}{\partial q^i} + \lambda \sum_{i=1}^3 \frac{\partial q^i}{\partial y} \frac{\partial v}{\partial q^i} + (\lambda + 2\mu) \sum_{i=1}^3 \frac{\partial q^i}{\partial z} \frac{\partial w}{\partial q^i} \right) + F_z \end{aligned} \quad (4)$$

Уравнения (2) не изменятся, (3) преобразуются аналогично (1). Таким образом, мы получаем новый вид уравнений, на основе которых далее строится алгоритм решения задачи.

3. Численный метод

Численное решение задачи производится на основе метода конечных разностей. Опять же, с точки зрения со-дизайна, этот метод зарекомендовал себя хорошо для создания на его основе эффективного 3D параллельного алгоритма (например, [10,11]).

Для уравнений (4) строится разностный аналог. За основу берутся формулы из работы [8], адаптированные для 3D-случая. Схема имеет второй порядок аппроксимации по пространству и по времени.

4. Параллельный алгоритм и его программная реализация

Численный алгоритм построен на основе широко известного метода конечных разностей, как это было отмечено выше. Использование представленного метода решения задачи связано с вычислениями для большого количества 3D данных, поэтому важным моментом является применение 3D декомпозиции области моделирования. В этом случае область разбивается на 3D «блоки». Каждый из блоков располагается непосредственно в оперативной памяти вычислительного устройства. Под вычислительным устройством в общем случае следует понимать CPU, графический процессор GPU или сопроцессор Xeon Phi. В данной работе в качестве вычислительного устройства используется CPU. Для проведения параллельных расчетов для каждого блока необходимо разместить и хранить в оперативной памяти девять 3D массивов со значениями для компоненты смещений u, v, w , параметры

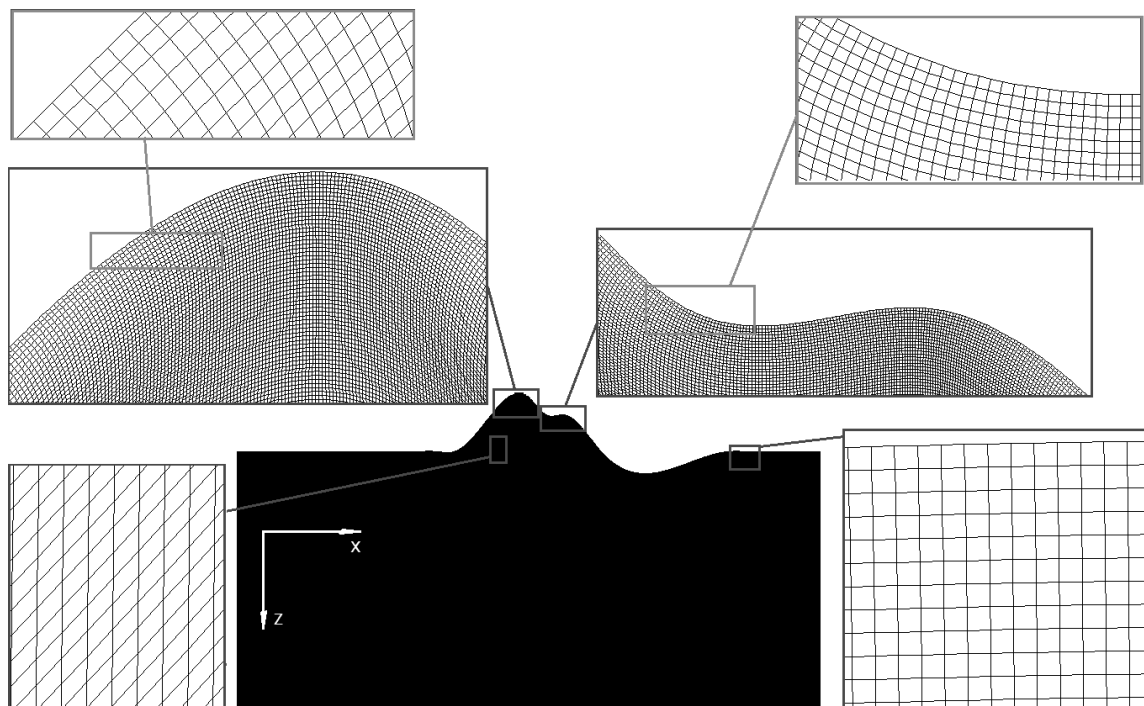


Рис. 2. Пример криволинейной сетки, 2D-срез

среды λ, μ, ρ , а также координаты узлов криволинейной сетки X, Y, Z . Для данной работы были разработаны параллельная программа построения сетки и параллельная программа, реализующая расчеты по разностной схеме для задачи (4). При разработке автор объединил их в одну программу. Таким образом, можно не записывать координаты узлов сетки на жесткий диск на стадии работы "строителя"сетки, а потом их считывать на стадии работы расчетной программы. Предложенный подход ускоряет работу программы значительно и зависит от размеров сетки и скорости записи/чтения информации на жесткие диски. Дать какую-либо процентную оценку сложно, поскольку время работы программы зависит не только от размеров области, но и требуемого времени моделирования сейсмополей.

Для массивов с u, v, w нужно организовать обмен данными между соседними блоками. Коммуникации осуществляются через созданную 3D-топологию. Каждому вычислительному процессу присваивается тройка номеров – декартовы координаты внутри этой топологии (рисунок 3).

Каждый блок имеет "слои перекрытия" в которые он принимает данные от своих соседей. Соответственно свои данные он шлет в аналогичные "слои перекрытия" своих соседей. Важно отметить, что пересылки осуществляются не только между процессами, соседними по одному из координатных направлений (q^1, q^2, q^3). Обмен данными ведется также между процессами, соседствующими по всем диагональным направлениям. Соответственно, на рисунке 4 можно видеть представление слоев перекрытия для одного блока. Разными цветами обозначены принимаемые данные с разных направлений. Общее количество пересылок – 26. Сам блок изображен белым цветом.

Параллельная программа для данного алгоритма создана при помощи языка Fortran и технологии MPI. На этой стадии, пользуясь соображениями со-дизайна, необходимо максимально оптимизировать программу для эффективного проведения расчетов на многоядерной архитектуре. Важно помнить, что при написании программы на языке Fortran массивы в памяти располагаются в ином порядке, нежели при написании на языке C++. Это стоит учитывать при определении порядка обхода массивов во вложенных циклах. Использование метода конечных разностей позволяет осуществлять обмен данными между процессами

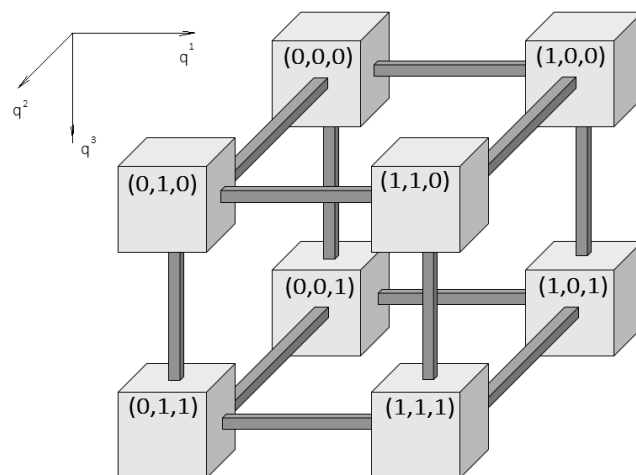


Рис. 3. 3D-топология

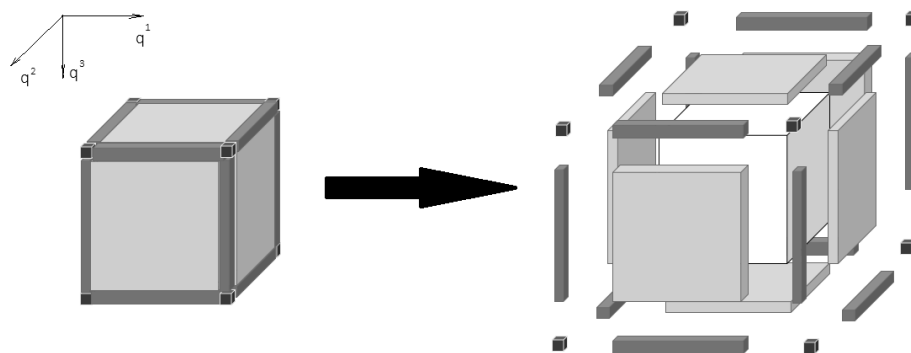


Рис. 4. Слои перекрытия

при помощи неблокирующих пересылок MPI: `Isend`, `Irecv`. При этом используется буферный массив, куда копируются значения u, v, w , что позволяет для двух соседних блоков все данные переслать в одном сообщении вместо трех, если бы для каждой компоненты была отдельная пересылка. Далее, вычисление компонент u, v, w для каждого блока разбито на 2 независимые части: внутри блока и на гранях блока (рисунок 5).

Схематично работа программы для каждого блока может быть представлена в следующем виде:

0) Определение всех постоянных данных: координат узлов криволинейной сетки X, Y, Z и параметров среды λ, μ, ρ .

1) Вычисление новых значений u, v, w на гранях блока.

2) Запуск неблокирующих пересылок `Isend`, `Irecv` новых значений u, v, w на гранях блока его соседям.

3) Вычислений новых значений u, v, w для внутренних точек блока.

4) Проверка завершения всех пересылок (MPI `Waitall`).

5) Переход на новый шаг итерации по времени, переход к пункту 1).

Данный подход к организации вычислений позволяет получить хорошие результаты в тестах масштабируемости и ускорения работы программы для различного числа процессов.

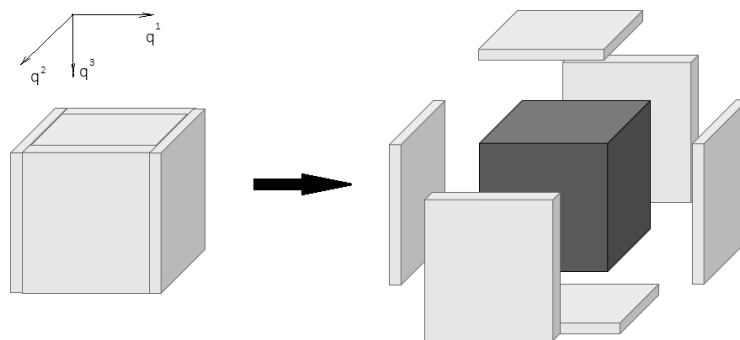


Рис. 5. Разбиение блока на внутреннюю часть и на грани

5. Тестирование программы на многопроцессорной архитектуре

В данном разделе изложены результаты двух тестов: на масштабируемость и на ускорение. В первом случае мы полагаем размер области неизменным и варьируем количество вычислительных ядер. Во втором случае мы изменяем размер общей области моделирования пропорционально количеству вычислительных ресурсов, так что объем вычислений для каждого процесса остается неизменным. Все расчеты были проведены на Кластере НКС-30Т+GPU, сервер g6 (64 двойных блейд-сервера HP BL2x220G6: 128 вычислительных модулей, RAM модуля—16 Гбайт, 256 (1024ядра) процессоров Intel Xeon E5540. Производительность – 6,1 Тфлопс).

Отметим, что время работы программы зависит только от размеров вычислительной области и не зависит от значений параметров и сложности геометрического строения среды. В тесте на масштабируемость использовалась область размером 240^3 точек (по 240 точек в каждом координатном направлении) на 1 процесс. Технология Intel hyper-threading была отключена, то есть одному вычислительному ядру соответствовал 1 процесс. Было выполнено 1100 итераций по времени. Результаты представлены на рисунке 6. Стоит отметить, что при переходе от 1 вычислительного ядра к 8 наблюдается скачок времени работы программы. Вероятно, это связано с тем, что при использовании 1 ядра не задействовались пересылки `Isend`, `Irecv`. Когда же используется более одного ядра, время расчетов находится в диапазоне от 3428,4 секунд (8 ядер) до 3561,2 секунд (120 ядер). Такой разброс времени работы программы может быть связан с увеличением числа пересылок. Тем не менее, эффективность масштабируемости алгоритма в сравнении для 8 и 120 ядер составила более 96%.

В тестах используется либо 1 процесс, либо число, кратное 8, чтобы всегда ресурсы каждого узла были задействованы полностью. В тесте на ускорение мы рассматриваем область размера 480^3 точек (по 480 точек в каждом координатном направлении) и различное число вычислительных процессов. Технология Intel hyper-threading также не использовалась (1 процесс – 1 MPI-поток). Было выполнено 1100 итераций по времени. На рисунке 7 представлены результаты. Поскольку размер области взят больше, чем при тесте на масштабируемость, тест начинается с 32 процессов. Заметим, что в сравнении времени работы программы для 32 процессов (1002,6 секунд) и 64 процессов (484,7 секунд) было получено ускорение в 2,07 раза. Возможно, это объясняется уменьшением размера пересылаемых сообщений при увеличении числа процессов, а также особенностями строения коммуникационной сети кластера. Аналогичная ситуация наблюдается при сравнении времени работы программы для 48 ядер (671,3 секунды) и 72 ядра (435,2 секунды): ускорение составило в 1,54 раза. Однако, при сравнении результатов для 40 ядер (809,2 секунды) и 80 ядер (415,9

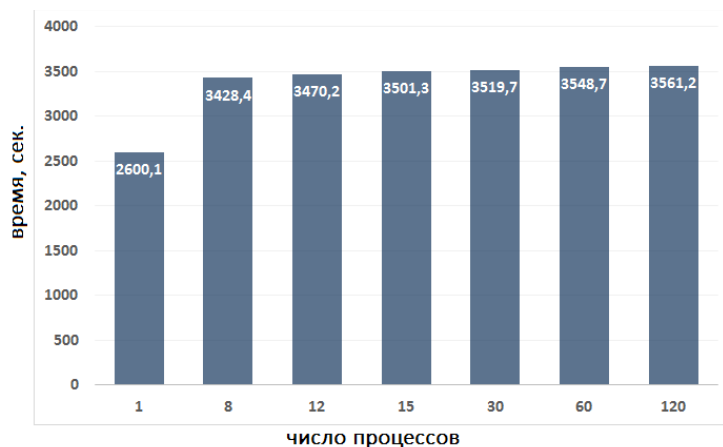


Рис. 6. Тест на масштабируемость

секунд) сверхлинейного ускорения не наблюдалось и работа программы ускорилась в 1,94 раза.

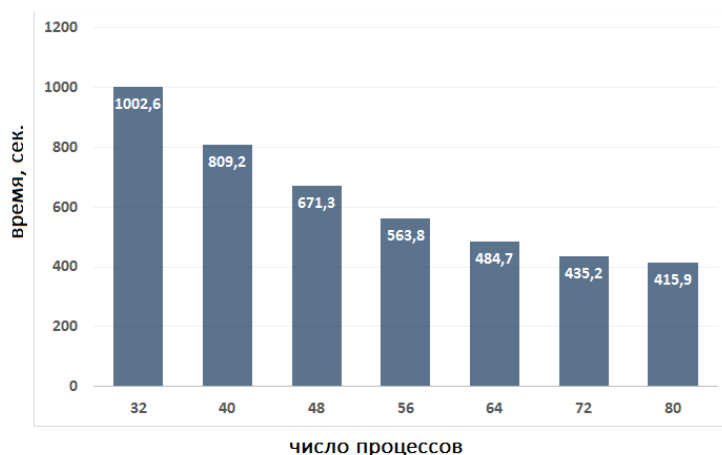


Рис. 7. Тест на ускорение

Во всех тестах использование большего числа ядер затруднено в связи с загруженностью кластера, поскольку это центр общего пользования и его ресурсами пользуются многие институты.

6. Заключение

В работе была представлена технология моделирования 3D сейсмических полей в упругих средах для областей со сложной геометрией свободной поверхности. Технология включает в себя выбор постановки задачи, разработку и адаптацию метода построения криволинейной 3D сетки, разработку метода численного решения задачи, использование 3D декомпозиции области для реализации на многопроцессорной архитектуре. Важным результатом работы является создание параллельного алгоритма и его программной реализации для построения криволинейной 3D сетки, локально-ортогональной возле свободной поверхности, а также создание параллельного 3D алгоритма и его программной реализации для численного моделирования упругих волн в изотропных средах со сложной геометрией свободной поверхности. Были проведены расчеты на кластере НКС-30T+GPU. На каждом этапе работы выбор того или иного подхода происходил при использовании со-дизайна –

анализа эффективности параллельной реализации на всех этапах разработки алгоритма. Проведенные тесты показали хорошие результаты масштабируемости и ускорения работы программы. Эти данные могут быть в дальнейшем применены для исследования поведения алгоритма методами имитационного моделирования [12] для кластеров экзафлопсного уровня. Это поможет в разработке и оптимизации масштабируемых алгоритмов для подобных вычислительных систем. В дальнейшем подобные алгоритмы могут быть применены для решения реальных задач, связанных, например, с мониторингом и изучением строения сред, характерных для магматических вулканов.

Литература

1. Глинский Б.М., Караваев Д.А., Ковалевский В.В., Мартынов В.Н. Численное моделирование и экспериментальное исследование грязевого вулкана "Гора Карabetова" выбросейсмическими методами // Вычислительные методы и программирование. Том 11, стр. 95-104, 2010
2. R.W. Graves Simulating seismic wave propagation in 3D elastic media using staggered grid finite differences. // Bull. Seism. soc. Am., vol.86, pp. 1091-1106, 1996
3. Virieux J. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. // Geophysics, Volume 51, Number 4, pp. 889-901 (1986)
4. Levander A.R. Fourth-order finite-difference P-SV seismograms. // Geophysics, vol. 53, issue 11, pp. 1425-1436 (1988)
5. Лисейкин В.Д. Разностные сетки. Теория и приложения // Издательство СО РАН, 2014, 254 стр.
6. Хакимзянов Г.С., Шокин Ю.И. Разностные схемы на адаптивных сетках // Редакционно-издательский центр НГУ, 2005, 130 стр.
7. Шокин Ю.И., Данаев Н.Т., Хакимзянов Г.С., Шокина Н.Ю. Лекции по разностным схемам на подвижных сетках // Редакционно-издательский центр КазНУ им. аль-Фараби, 183 стр.
8. Daniel Appelo, N. Anders Petersson A Stable Finite Difference method for the Elastic wave equation on complex geometries with free surfaces // Commun. Comput. Phys. Vol.5, No. 1, pp. 84-107, 2009
9. Титов П.А. Алгоритм и программа моделирования 2D-волновых полей в областях с криволинейной свободной поверхностью // Материалы конференции «Научный сервис в сети Интернет – 2014» Новороссийск, Абрау-Дюрсо, 21-26 сентября 2014, стр.446-455
10. Dimitri Komatitsch, Gordon Erlebacher, Dominik Goddeke, David Michea High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster // Journal of Computational Physics, Volume 229, Issue 20, pp. 7692-7714 (2010)
11. Karavaev D.A., Glinsky B.M., Kovalevsky V.V. A Technology of 3D Elastic Wave Propagation Simulation Using Hybrid Supercomputers // Russian Supercomputing Days 2015, pp. 26-33
12. Chernykh I., Glinskiy B., Kulikov I., Marchenko M., Rodionov A., Podkorytov D., Karavaev D. Using Simulation System AGNES for Modeling Execution of Parallel Algorithms on Supercomputers // Computers, Automatic Control, Signal Processing and Systems Science. The 2014 Int. Conf. on Applied Mathematics and Computational Methods in Engineering, pp. 66-70 (2014)

—

Technology of simulation of elastic wave propagation in media with complex 3D geometry surface on high-performance clusters

P.A. Titov¹

ICMMG SB RAS¹

The paper presents technology of 3D elastic waves simulation on high-performance clusters. The technology includes a mathematical method and parallel algorithm. A distinctive feature of the method is the use of a curved three-dimensional grid, which is consistent with the geometry of free surface. An important aspect of the work is co-design – analysis of parallel implementation effectiveness at all stages of the algorithm development. A software implementation of the algorithm for the calculation on multi-core systems is presented. The features of algorithm parallelization were reviewed. The results of test calculations for the study of acceleration and scalability of the proposed algorithm are presented.

Keywords: Elasticity, 3D, modeling, curved grid, supercomputer, co-design

References

1. Glinskiy B.M., Karavaev D.A., Kovalevskiy V.V., Martynov V.N. Chislennoe modelirovaniye i eksperimental'noye issledovanie gryazevogo vulkana "Gora Karabetova" vibroseismicheskimi metodami // Vychislitel'nye metody i programmirovaniye. Tom 11, str. 95-104, 2010
2. R.W. Graves Simulating seismic wave propagation in 3D elastic media using staggered grid finite differences. // Bull. Seism. soc. Am., vol.86, pp. 1091-1106, 1996
3. Virieux J. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. // Geophysics, Volume 51, Number 4, pp. 889–901 (1986)
4. Levander A.R. Fourth-order finite-difference P-SV seismograms. // Geophysics, vol. 53, issue 11, pp. 1425–1436 (1988)
5. Liseykin V.D. raznostnye setki, teoriya i prilozheniya // Izdatel'stvo SB RAS, 2014, 254 str.
6. Khakimzyanov G.S., Shokin Yu.I. Raznostnye skhemy na adaptivnykh setkakh // Redaktsionno-izdatel'skiy tsentr NGU, 2005, 130 str.
7. Shokin Yu.I., Danaev N.T., Khakimzyanov G.S., Shokina N.Yu. Lektsii po raznostnym skhemam na podvizhnykh setkakh // Redaktsionno-izdatel'skiy tsentr KazNU im. al'-Farabi, 183 str.
8. Daniel Appelo, N. Anders Petersson A Stable Finite Difference method for the Elastic wave equation on complex geometries with free surfaces // Commun. Comput. Phys. Vol.5, No. 1, pp. 84-107, 2009
9. Titov P.A. Algoritm i programma modelirovaniya 2D-volnovykh poley v oblastiakh s krivolineynoy svobodnoy poverkhnost'yu // Materialy konferentsii «Nauchnyy servis v seti Internet – 2014» Novorossiysk, Abrau-Dyurso, 21-26 sentyabrya 2014, str.446-455
10. Dimitri Komatitsch, Gordon Erlebacher, Dominik Goddeke, David Michea High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster // Journal of Computational Physics, Volume 229, Issue 20, pp. 7692–7714 (2010)

11. Karavaev D.A., Glinsky B.M., Kovalevsky V.V. A Technology of 3D Elastic Wave Propagation Simulation Using Hybrid Supercomputers // Russian Supercomputing Days 2015, pp. 26-33
12. Chernykh I., Glinskiy B., Kulikov I., Marchenko M., Rodionov A., Podkorytov D., Karavaev D. Using Simulation System AGNES for Modeling Execution of Parallel Algorithms on Supercomputers // Computers, Automatic Control, Signal Processing and Systems Science. The 2014 Int. Conf. on Applied Mathematics and Computational Methods in Engineering, pp. 66–70 (2014)