

DOI: 10.15514/ISPRAS-2019-31(6)-16



Применение сеточно-характеристического метода для решения задач распространения динамических волновых возмущений на высокопроизводительных вычислительных системах

Н.И. Хохлов, ORCID: 0000-0002-2460-0137 <khokhlov.ni@mipt.ru>

И.Б. Петров, ORCID: 0000-0003-3978-9072 <petrov@mipt.ru>

Московский физико-технический институт

141701, Московская область, г. Долгопрудный, Институтский переулок, д.9

Аннотация. В работе рассматривается применение различных современных технологий высокопроизводительных вычислений для ускорения численного решения задач распространения динамических волновых возмущений с использованием сеточно-характеристического метода. Рассматриваются технологии как для центральных процессоров (CPU), так и для графических процессоров (GPU). Приведены сравнительные результаты применения технологий MPI, OpenMP, CUDA. В качестве примеров работы разработанного программного комплекса приводится ряд примеров расчета задач сейсмологии и геофизики. Отдельно рассмотрен вопрос распараллеливания задач с наличием контактов многих сеток и топографии дневной поверхности, используя криволинейные сетки.

Ключевые слова: сеточно-характеристический метод; сейсмология; геофизика; MPI; OpenMP; CUDA

Для цитирования: Хохлов Н.И., Петров И.Б. Применение сеточно-характеристического метода для решения задач распространения динамических волновых возмущений на высокопроизводительных вычислительных системах. Труды ИСП РАН, том 31, вып. 6, 2019 г., стр. 237-252. DOI: 10.15514/ISPRAS-2019-31(6)-16

Благодарности: Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00914.

Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems

N.I. Khokhlov, ORCID: 0000-0002-2460-0137 <khokhlov.ni@mipt.ru>

I.B. Petrov, ORCID: 0000-0003-3978-9072 <petrov@mipt.ru>

Moscow Institute of Physics and Technology,

9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation

Abstract. The paper considers the application of various modern technologies of high-performance computing to accelerate the numerical solution of the problems of propagation of dynamic wave disturbances using the grid-characteristic method. Technologies are considered both for central processing units (CPUs) and for graphic processors (GPUs). Comparative results of applying MPI, OpenMP, CUDA technologies are presented. As examples of the work of the developed software package, a number of

examples of calculating the problems of seismic and geophysics are given. Separately, the issue of parallelizing problems with the presence of contacts of many grids and the topography of the day surface using curvilinear grids is considered.

Keywords: grid-characteristic method; seismic; geophysics; MPI; OpenMP; CUDA

For citation: Khokhlov N.I., Petrov I.B. Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 6, 2019. pp. 237-252 (in Russian). DOI: 10.15514/ISPRAS-2019-31(6)-16

Acknowledgments: This work was supported by the Russian Foundation for Basic Research as part of research project No. 18-07-00914.

1. Введение

Численное моделирование распространения динамических волновых возмущений в твердых телах используется при решении широкого круга задач. К таким задачам, в том числе, относятся задачи сейсмоземлетрясения и вычислительной геофизики. Роль численного моделирования в каждой из данных областей очень важна. Поиск полезных ископаемых, таких как углеводороды, является дорогостоящей задачей, для решения которой применяются различные методы петрофизики, сейсмологии и геологии. В результате применения этих методов строится приближительная модель геологической среды. Далее эта модель используется для проверки различных гипотез о положении в пространстве геологических пластов и распределении полезных ископаемых. Один из методов такого анализа – точное решение прямой задачи распространения волн упругости с использованием численного моделирования. Кроме того, решение прямой задачи является составной частью методов инверсии и миграции, цель которых заключается в определении параметров среды и положений в пространстве разделов между средами, что в конечном итоге позволяет обнаружить полезные ископаемые в толще земли. Численное моделирование распространения сейсмических волн представляет существенную часть работ при проведении геологической разведки в нефтяной отрасли. Математическое моделирование проводится в различных геологических средах, в том числе, в слоистых средах и в средах с наличием различных неоднородностей (например, трещин или каверн). Задачи такого рода представляются очень ресурсоемкими с точки зрения вычислительных ресурсов.

Характерные размеры расчетных областей в таких задачах по каждому измерению около 1000-10000 узлов, а разумные времена расчета – около суток. Производительность современных настольных компьютеров позволяет решать прямую задачу явными численными методами в двумерном случае за разумное время. А производительности кластеров уже достаточно для решения трехмерных задач. Повышенный интерес промышленности к решению прямой задачи объясняется ростом производительности современных центральных процессоров, поскольку это позволило быстро получать результаты моделирования для крупных геологических моделей. Повышению интереса также способствует интенсивное развитие технологий параллельного программирования прикладных задач общего назначения на графических процессорах. Как правило, явные численные методы могут быть переписаны для эффективного исполнения на GPU, что в зависимости от типа задачи и численного метода может повысить производительность на несколько порядков по сравнению с реализацией для центрального процессора. Таким образом, возросший интерес к проведению численного моделирования в индустрии объясняет актуальность данной работы. В работе рассматривается сеточно-характеристический метод решения уравнения распространения волн в упругой среде. Данный метод обладает рядом особенностей, из-за которых его применение в

определенных постановках более целесообразно, чем использование классических методов наподобие конечно-разностного. В работе [1] было произведено сравнение разрывного метода Галеркина с сеточно-характеристическим методом на неструктурированной сетке и на регулярной сетке [9, 24]. Авторами была продемонстрирована более высокая скорость расчета с использованием СХ метода, а точность расчета показала применимость метода к практическим задачам.

В данной работе рассматривается программный комплекс, созданный для моделирования задач распространения динамических волновых возмущений в твердых телах. Программный комплекс использует двумерные и трехмерные структурные блочные сетки с наличием неоднородностей. Для численного интегрирования используются сеточно-характеристические [1,2] и конечно-объемные [3] методы 2-4 порядка точности. Алгоритм распараллелен используя различные технологии написания параллельных приложений. 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. Кроме того, код распараллелен с использованием технологии 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 8 устройств CUDA.

2. Математическая модель

В работе рассматривается модель сплошной среды, использующая определяющие соотношения линейной теории упругости. Здесь и далее будем использовать верхний индекс для обозначения номера компоненты вектора (или тензора). Рассмотрим физическую модель среды. В каждой точке пространства $x = (x^1, x^2, x^3)$ задан вектор смещения $u(x, t) = (u^1, u^2, u^3)$ частиц среды от начального положения в результате упругого взаимодействия. Тензор деформации $\varepsilon(x, t)$ вычисляется на основе вектора смещений следующим образом:

$$\varepsilon^{ij} = \frac{1}{2}(u_j^i + u_i^j), \quad i, j \in \{1, 2, 3\}. \quad (1)$$

Нижний индекс обозначает дифференцирование. Если там указано число, то имеется в виду дифференцирование по пространственной координате с данным номером. Например, дифференцирование некоторой переменной $a(x, t)$ будет выглядеть так:

$$a_t = \frac{\partial a}{\partial t}, \quad a_2 = \frac{\partial a}{\partial x_2}.$$

Для каждой точки среды можно записать второй закон Ньютона:

$$\rho u_{tt}^i - \sum_{j=1}^3 \sigma_j^{ij} - f^i = 0; \quad i \in \{1, 2, 3\}. \quad (2)$$

Здесь $\rho(x)$ – плотность в рассматриваемой точке пространства, а $f(x, t) = (f^1, f^2, f^3)$ – плотность силы, направленной из рассматриваемой точки пространства x , а σ^{ij} – тензор упругого напряжения.

Помимо уравнения движения приведенного выше, понадобится еще закон сохранения, который задаст взаимосвязь между тензором деформации и тензором напряжения. Обобщенный закон Гука как раз и устанавливает такую связь.

$$\sigma^{ij} = \sum_{k=1}^3 \sum_{l=1}^3 c^{ijkl} \varepsilon^{kl}, \quad i, j \in \{1, 2, 3\}. \quad (3)$$

Остается задать значения c_{ijkl} . Дальнейшие выкладки будут рассматриваться для модели идеальной линейно-упругой среды без поглощения волн. В программном комплексе также реализованы и другие модели, например: модель вязкоупругой среды, учитывающая затухание волн и модель упругопластичной среды, учитывающая гистерезисные свойства материала.

В случае изотропной линейно-упругой среды, равенство (3) значительно упрощается. В силу симметрии тензора напряжений, а также в силу изотропности среды, количество независимых переменных в c_{ijkl} снижается с 91 до 2. В качестве таких неизвестных удобно принять так называемые параметры Ламе λ и μ . Данные параметры выражаются через скорость продольных Р-волн $c_p = \sqrt{(\lambda + 2\mu)/\rho}$ и скорость поперечных S-волн $c_s = \sqrt{\mu/\rho}$. Тогда закон сохранения примет следующий вид:

$$\sigma^{ij} = \lambda \sum_{k=1}^3 \varepsilon^{kk} \delta^{ij} + 2\mu \varepsilon^{ij}, \quad i, j \in \{1, 2, 3\}. \quad (4)$$

где δ^{ij} – символ Кронекера. С учетом равенства 1, закон сохранения можно записать как связь между тензором напряжений и вектором смещений:

$$\sigma^{ij} = \lambda \sum_{k=1}^3 u_k^k \delta^{ij} + \mu(u_j^i + u_i^j), \quad i, j \in \{1, 2, 3\}. \quad (5)$$

Таким образом, мы получили систему уравнений (2), (5), которая описывает распространение волн в линейно-упругих средах.

Для того, чтобы перейти к двумерному случаю, положим производные компонент вектора смещений и тензора напряжений по оси x_3 равными 0. Это делается из соображения, что изменения параметров вдоль одной из осей трехмерного пространства в двумерном случае нет. Уравнения (2), (5) распадутся на две независимых системы уравнений. Первая описывает распространение упругих волн в двумерном случае и представляет для нас интерес:

$$\rho u_{tt}^i - \sum_{j=1}^2 \sigma_j^{ij} - f^i = 0, \quad i \in \{1, 2\}. \quad (6)$$

$$\sigma^{ij} = \lambda \sum_{k=1}^2 u_k^k \delta^{ij} + \mu(u_j^i + u_i^j), \quad i, j \in \{1, 2\}. \quad (7)$$

В двумерном случае вид системы уравнений остался тем же, как и в трехмерном случае, но количество уравнений уменьшилось с 9 до 5.

3. Сеточно-характеристический метод

Для того, чтобы продемонстрировать применение численного метода, надо записать систему (6), (7) в матричной форме [9]. Для ее записи будем использовать вектор скорости $v = u_t$ вместо вектора смещения u .

$$\begin{bmatrix} v_1^1 \\ v_1^2 \\ \sigma_1^{11} \\ \sigma_1^{22} \\ \sigma_1^{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ \lambda + 2\mu & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^1 \\ v_1^2 \\ \sigma_1^{11} \\ \sigma_1^{22} \\ \sigma_1^{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda + 2\mu & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_2^1 \\ v_2^2 \\ \sigma_2^{11} \\ \sigma_2^{22} \\ \sigma_2^{12} \end{bmatrix} + \begin{bmatrix} f^1 \\ f^2 \\ f^1 \\ f^2 \\ f^1 \end{bmatrix}. \quad (8)$$

Введем вектор $\varphi(x, t) = (v^1, v^2, \sigma^{11}, \sigma^{22}, \sigma^{12})$. Обозначим матрицы в уравнении (8) как A^1 и A^2 . Вектор с функциями, определяющими источник упругих колебаний, обозначим как $s = (f^1/\rho, f^2/\rho, 0, 0, 0)$. Тогда уравнение (10) можно будет записать в более короткой форме:

$$\varphi_t = A^1 \varphi_1 + A^2 \varphi_2 + s. \quad (9)$$

Далее, используя метод расщепления (или метод дробных шагов) по пространству, описанный например в [5], перейдем к решению одномерных уравнений переноса:

$$\varphi_t = A^1 \varphi_1, \quad \varphi_t = A^2 \varphi_2. \quad (10)$$

В дальнейшем пересчет с одного временного слоя на следующий согласно формулам (10) будем называть соответственно шагом по X и шагом по Y. Обозначим их в виде операторов: $\varphi^{t+\Delta t} = S_x(\Delta t, \varphi^t)$, $\varphi^{t+\Delta t} = S_y(\Delta t, \varphi^t)$. Последовательность вычислений этих шагов накладывает ограничение на порядок схемы по пространству. В работе [6] описано так называемое расщепление Странга (Strang splitting), определяющее последовательность вычисления одномерных уравнений переноса, которое позволяет получить второй порядок точности при условии, что каждый шаг по отдельности имеет порядок не меньше второго. Расщепление Странга заключается в применении следующего порядка вычислений отдельных шагов для пересчета с момента времени t в $t + \Delta t$:

$$\varphi^{t+\frac{\Delta t}{4}} = S_x\left(\frac{\Delta t}{4}, \varphi^t\right), \quad (11)$$

$$\varphi^{t+\frac{3\Delta t}{4}} = S_y\left(\frac{\Delta t}{2}, \varphi^{t+\frac{\Delta t}{4}}\right), \quad (12)$$

$$\varphi^{t+\Delta t} = S_x\left(\frac{\Delta t}{4}, \varphi^{t+\frac{3\Delta t}{4}}\right). \quad (13)$$

Как показала практика, применение более простого способа расчета шагов по X и Y хоть и имеет первый порядок точности, но позволяет получить результат, близкий к расщеплению Странга:

$$\varphi^{t+\frac{\Delta t}{2}} = S_x\left(\frac{\Delta t}{2}, \varphi^t\right), \quad (14)$$

$$\varphi^{t+\Delta t} = S_y\left(\frac{\Delta t}{2}, \varphi^{t+\frac{\Delta t}{2}}\right). \quad (15)$$

Поскольку выбор способа расщепления не вносит существенных изменений в пригодность к повышению производительности и распараллеливанию метода, было принято решение в тестовой программной реализации воспользоваться вторым, более простым вариантом.

Уравнения (10) решаются при помощи приведения матриц к диагональному виду [4]: $A^1 = L^1 \Lambda^1 R^1$ и $A^2 = L^2 \Lambda^2 R^2$, где $L^1 R^1 = E$, $L^2 R^2 = E$, а матрицы Λ^1 и Λ^2 – диагональные. Такое приведение возможно, поскольку матрицы A^1 и A^2 имеют действительные собственные числа, которые и стоят на диагоналях матриц Λ^1, Λ^2 :

$$\Lambda^1 = \Lambda^2 = \text{diag}(c_p, -c_p, c_s, -c_s, 0). \quad (16)$$

Поскольку матрицы рассматриваемой системы уравнений могут быть приведены к диагональному виду, система уравнений является гиперболической. Матрицы L^1 и L^2 содержат в столбцах собственные векторы A^1 и A^2 соответственно:

$$L^1 = \begin{bmatrix} c_p & -c_p & 0 & 0 & 0 \\ 0 & 0 & c_s & -c_s & 0 \\ \lambda + 2\mu & \lambda + 2\mu & 0 & 0 & 0 \\ \lambda & \lambda & 0 & 0 & 1 \\ 0 & 0 & \mu & \mu & 0 \end{bmatrix}, \quad L^2 = \begin{bmatrix} 0 & 0 & c_s & -c_s & 0 \\ c_p & -c_p & 0 & 0 & 0 \\ \lambda & \lambda & 0 & 0 & 1 \\ \lambda + 2\mu & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & \mu & \mu & 0 \end{bmatrix}. \quad (17)$$

Матрицы R^1 и R^2 получаются обращением матриц L^1 и L^2 :

$$R^1 = \begin{bmatrix} +\frac{1}{2c_p} & 0 & \frac{1}{2(\lambda + 2\mu)} & 0 & 0 \\ -\frac{1}{2c_p} & 0 & \frac{1}{2(\lambda + 2\mu)} & 0 & 0 \\ 0 & +\frac{1}{2c_s} & 0 & \frac{1}{2\mu} & 0 \\ 0 & -\frac{1}{2c_s} & 0 & \frac{1}{2\mu} & 0 \\ 0 & 0 & -\frac{\lambda}{\lambda + 2\mu} & 1 & 0 \end{bmatrix}, \quad R^2 = \begin{bmatrix} 0 & +\frac{1}{2c_p} & 0 & \frac{1}{2(\lambda + 2\mu)} & 0 \\ 0 & -\frac{1}{2c_p} & 0 & \frac{1}{2(\lambda + 2\mu)} & 0 \\ +\frac{1}{2c_s} & 0 & 0 & 0 & \frac{1}{2\mu} \\ -\frac{1}{2c_s} & 0 & 0 & 0 & \frac{1}{2\mu} \\ 0 & 0 & 1 & -\frac{\lambda}{\lambda + 2\mu} & 0 \end{bmatrix}. \quad (18)$$

Умножим слева уравнения (10) на R^1 и R^2 соответственно. Тогда получим:

$$R^1 \varphi_t = R^1 (L^1 \Lambda^1 R^1) \varphi_1, \quad R^2 \varphi_t = R^2 (L^2 \Lambda^2 R^2) \varphi_2. \quad (19)$$

Перейдем от исходных неизвестных φ к новым: $\omega^1 = R^1 \varphi$ и $\omega^2 = R^2 \varphi$, а также вспомним, что $R^1 L^1 = E$, $R^2 L^2 = E$. Тогда придем к следующим системам уравнений:

$$\omega_t^i = \Lambda^i \omega_i^i, \quad i \in 1, 2. \quad (20)$$

В каждом окажется по 5 скалярных независимых уравнений переноса, так как Λ^i – диагональные матрицы.

Известно большое количество методов решения скалярных одномерных уравнений переноса $u_t + C u_x = 0$. Самым простым примером является разностная схема первого порядка CIR, предложенная в работе [7]. Она записывается следующим образом:

$$\frac{u^{m,n+1} - u^{m,n}}{\Delta t} + C \frac{u^{m+1,n} - u^{m,n}}{\Delta x} = 0. \quad (21)$$

Данную схему можно интерпретировать как результат линейной интерполяции значения в точке $u^{m*,n}$ на основе значений в точках $u^{m,n}$ и $u^{m+1,n}$ для $C < 0$ (или между $u^{m-1,n}$ и $u^{m,n}$ для $C > 0$). Поскольку решение вдоль характеристики постоянно, $u^{m,n+1} = u^{m*,n}$.

Для практических расчетов используются более сложные методы решения уравнений переноса, учитывающие дополнительно законы сохранения и позволяющие получать точные решения даже при наличии разрывов в решении и параметрах среды, которые могут возникать на границах пластов земли с разными скоростями.

Для расчетов представленных в работе применялась схема Рузанова третьего порядка [8]. После того, как решения ω^i будут пересчитаны с временного слоя t на слой $t + \Delta t$, значения исходных неизвестных находится по формуле $\varphi = L^i \omega^i, i \in 1, 2$. Можно переходить к следующей итерации расчета, для вычисления значений φ в момент времени $t + 2\Delta t$, а потом $t + 3\Delta t$ и т.д., пока не будет достигнуто желаемое количество шагов по времени.

4. Распараллеливание MPI

Для работы в системах с распределенной памятью программный комплекс распараллелен с использованием технологии MPI [16]. При распараллеливании использовались стандартные алгоритмы декомпозиции расчетной области и обмена приграничными ячейками, широко применяемые для явных сеточных методов [17-20]. Поскольку в данной реализации сеточно-характеристического решателя используются регулярные сетки, узлы могут храниться в 2D/3D массивах (в памяти они хранятся в виде

непрерывных одномерных массивов). На вход программе поступают номера процессов для каждой сетки, которые будут пересчитывать узлы внутри этих сеток. Алгоритм определяет наилучшее разбиение массивов на равные по размеру блоки для распределения работы между процессами путем вызова `MPI_Dims_create`. Кроме этого возможно вручную задать желаемое распределение процессов. Количество блоков, на которые разбита сетка, такое же, как и число процессов, ассоциированное с сеткой. Это значит, что каждый процесс пересчитывает узлы на следующий временной шаг не более чем для одного блока сетки. В то время как процессы могут быть ответственны не более чем за один блок сетки, они могут одновременно обрабатывать несколько блоков с разных сеток.

Для упрощения синхронизации узлов на границах блоков внутри одной сетки, создается специальный коммуникатор (`MPI_Comm`) и специальную группу (`MPI_Group`) для этой сетки. Число процессов, которые даны на входе, определяют список процессов из коммуникатора `MPI_COMM_WORLD`, на основе которых создается новый коммуникатор. Синхронизация внутри сетки производится между шагами по времени, и каждый процесс может получить информацию о процессах, обрабатывающих соседние блоки. Зная точные номера процессов, которые готовы обмениваться пересчитанными узлами на границах блоков, процессы могут выполнять асинхронные операции приема и передачи (`MPI_Isend`, `MPI_Irecv`).

В противоположность синхронизации на границе блоков, при синхронизации на границах контактных поверхностей между двумя сетками, процессы не знают номеров процессов, с которыми обмениваться данными, так как геометрия контактов может быть произвольной. В первоначальной реализации использовалась функция `MPI_Alltoall` для того, чтобы все процессы коммуникатора `MPI_COMM_WORLD` содержали одинаковую информацию о декомпозиции. Основной причиной неэффективности такой стратегии являлось то, что процессы, которые не нуждаются в синхронизации для продолжения расчета, все равно были обязаны участвовать в этом взаимодействии. Другими словами была создана точка глобальной синхронизации на каждом временном шаге, которую можно было устранить.

Тогда были введены доступные всем процессам списки со свойствами всех сеток. Это свойства содержат информацию о размерах сеток, номерах процессов, связанных с сетками, точные положения блоков и их размеры для каждого из этих процессов. Списки свойств создаются в начале инициализации расчетного алгоритма синхронизируются между всеми процессами и далее остаются неизменными (это возможно благодаря статичной структуре сеток и связей между ними). Для каждой сетки возможно получить номера процессов из `MPI_COMM_WORLD`, которые обрабатывают каждый блок. Следовательно, когда некоторый процесс должен обновить узлы на контактной границе для синхронизации с другим процессом, он знает какой именно процесс содержит данные из другой сетки с этим контактом, поэтому он может установить обмен данными только с этим процессом. Более того, данный подход позволяет учитывать случай, в котором несколько процессов на границе одной сетки обмениваются данными с независимо распределенными процессами на границе другой сетки. Это значит, что декомпозиции сеток могут быть независимы друг от друга.

Информация о контактах на границах сеток задается обобщенным способом. Положение области в одной сетке, которая должна быть отправлена в заданную область другой сетки, поступает на вход решателю. Список таких указаний содержит полную информацию о контактах в модели с несколькими сетками. Данная информация используется на каждом шаге по времени, когда происходит синхронизация. Если положения узлов сетки не совпадают в точности, то производится переинтерполяция.

Рассмотрим пример декомпозиции блочных сеток. Разбиение выполнено для модели слоистой среды с криволинейными контактными поверхностями между геологическими слоями. Модель имеет 8 слоев, каждый из которых обозначен отдельным цветом на рис. 1.

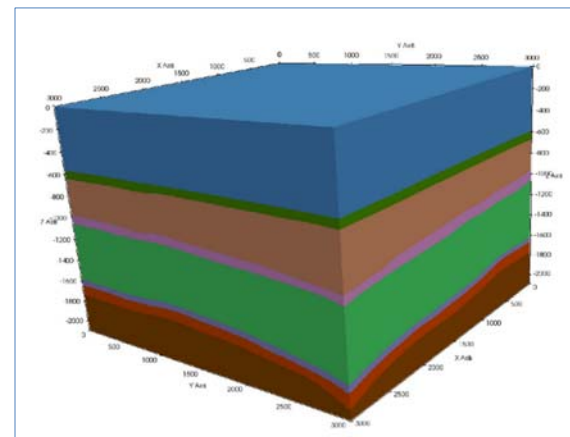


Рис. 1. Структура геологических слоев
Fig. 1. The structure of geological layers

Количество узлов во всей расчетной области $601 \times 601 \times 421$. На рис. 2 разными цветами изображены блоки с узлами, пересчитываемые разными процессами. Это упрощенный вариант с 16 процессами, каждый цвет обозначает один процесс. Для наглядности слои также были разделены по оси Z, хотя в реальном расчете для сеток с небольшим количеством узлов по оси Z такое разбиение не производилось, поскольку значительно ухудшало производительность.

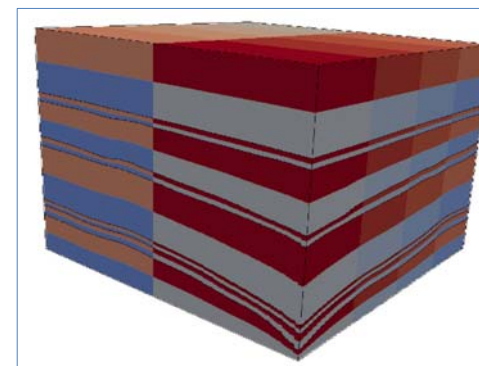


Рис. 2. Распределение 16 процессов по сеткам. Каждая сетка разделена на $4 \times 2 \times 2$ блоков
Fig. 2. Distribution of 16 processes on grids. Each grid is divided into $4 \times 2 \times 2$ blocks

Была измерена производительность нашей параллельной реализации сеточно-характеристического метода. Результаты изображены на рис. 3. В тесте использовалась расчетная сетка размером $1000 \times 1000 \times 1000$ узлов. Тестирование проводилось на кластере

НЕСТОР. 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. Отсчет идет от 128 процессов MPI, для того, чтобы получить корректные результаты, так и при малом числе процессов времена расчетов могут быть неприемлемо долгими. Можно заметить, что представленная реализация масштабируется до 4096 процессов практически без потери эффективности. При увеличении числа процессов наблюдается ожидаемое снижение эффективности, однако алгоритм все еще демонстрирует хорошие показатели даже на 16384 процессах.

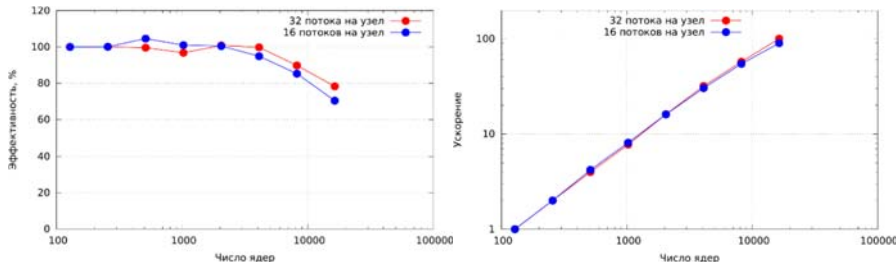


Рис. 3. Эффективность (слева) и масштабируемость (справа) параллельного алгоритма в среде MPI

Fig. 3. Efficiency (left) and scalability (right) of the parallel algorithm in the MPI environment

5. Распараллеливание в средах с общей памятью

Было проведено распараллеливание для систем с общей памятью с использованием технологий OpenMP [10, 21] и POSIX Threads. Полученное ускорение приведено на рис. 4. В силу несущественных различий в ускорении и простоты внедрения параллельного кода с применением OpenMP из них было решено использовать в расчетах только технологию OpenMP. Код распараллеливается согласно принципу геометрического параллелизма. Производится разбиение расчетной сетки на прямоугольные области, каждая из которых пересчитывается одним потоком OpenMP. Таким же образом разбиваются наложенные сетки, используемые для задания трещин. Глобальная синхронизация потоков происходит между шагами по времени. При этом происходит обмен значениями в узлах, которые находятся на границе разбиений и должны быть доступны одновременно нескольким потокам.

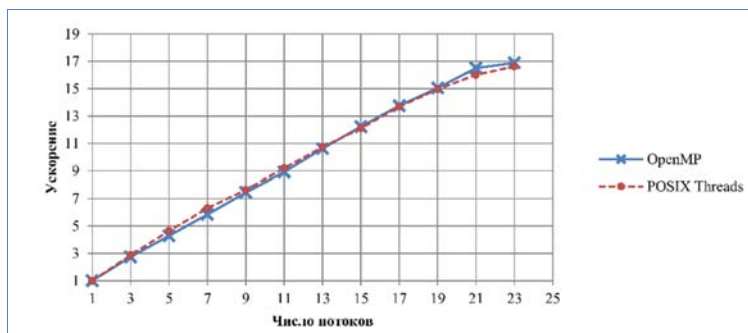


Рис. 4. Зависимость ускорения от числа потоков для систем с общей памятью
Fig. 4. Dependence of acceleration on the number of threads for systems with shared memory

Достигнута 74% эффективность ускорения полученного алгоритма на машине с процессорами Intel Xeon E5-2697 (на 24 ядрах).

6. Распараллеливание на графических процессорах

Алгоритм также был распараллелен на графических GPGPU процессорах NVidia используя технологию CUDA. Данная технология широко применяется для распараллеливания, в том числе явных, вычислительных алгоритмов [11-15]. Потребовалось полное переписывание части расчетного модуля под данную архитектуру [21]. За основу для реализации алгоритма на графических процессорах была взята оптимизированная для выполнения на центральном процессоре версия данной программы. Оптимизациям была подвергнута наиболее затратная в вычислительном плане часть алгоритма. Так как использовалось расщепление по пространственной координате, для пересчета всей сетки требовалось два шага: по оси X и по оси Y. В первоначальной версии (cuda1 на рис. 5) на графическом устройстве выделяется в 2 раза больше памяти, чем требуется для хранения расчетной сетки. В памяти хранятся две ее копии.

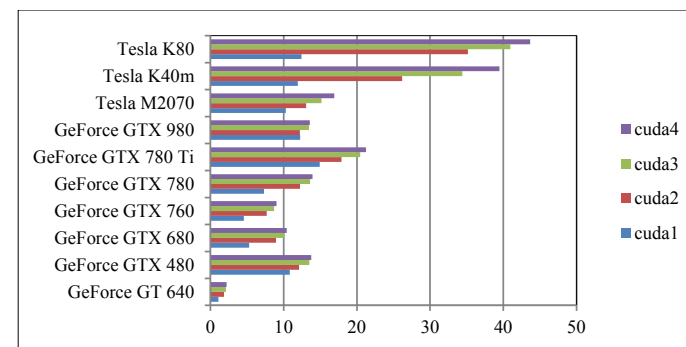


Рис. 5. Ускорение по сравнению с CPU
Fig. 5. Acceleration over CPU

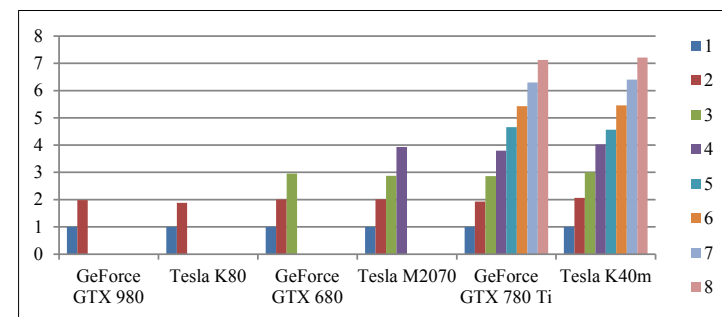


Рис. 6. Ускорение в зависимости от количества графических устройств
Fig. 6. Acceleration depending on the number of graphics devices

На каждом шаге происходит пересчет значений узлов сетки, хранящихся в одной из копий. При этом результат вычислений записывается в другую копию расчетной сетки. В результате происходит синхронизация только между вызовами kernel'ов CUDA. Это было

сделано по причине того, что глобальная синхронизация всего устройства создает большие временные задержки. Таким образом, удалось уменьшить количество глобальных синхронизаций до двух раз на один шаг по времени. Следующая оптимизация (cuda2) состояла в использовании общей памяти (shared memory) устройства. Чтение из глобальной памяти происходит только один раз на каждом шаге, причем обращения становятся последовательными (coalesced), что также ведет к повышению производительности. Далее используются оптимизации, дающие незначительный прирост производительности. В версии cuda3 дополнительные данные вычисляются на CPU и передаются через параметры вызова kernel'a. Таким образом, ожидается, что у каждого потока будет создана локальная копия этих значений. В версии cuda4 подбираются размеры блоков таким образом, чтобы минимизировать количество узлов, для которых требуются обмены памятью между блоками.

Для выполнения на нескольких GPU использовался наиболее оптимизированный вариант. Также была применена технология GPUDirect, позволяющая передавать данные через шину PCI Express в обход центрального процессора. Максимальное полученное ускорение по сравнению с CPU на одном графическом устройстве – в 55 раз на GeForce GTX 780 Ti в вычислениях с одинарной точностью и в 44 раза на Tesla K80 в вычислениях с двойной точностью (рис. 5). Максимальное достигнутое ускорение от количества графических устройств - в 7.1 раз на 8 графических устройствах (рис. 6) для двойной точности. Технология GPUDirect повысила ускорение на 10% от того, что было достигнуто без нее при вычислениях с одинарной точностью. При вычислениях с двойной точностью ускорение составило 2.4%.

7. Примеры расчетов

Используется модель слоистой среды с криволинейными контактами между слоями [22]. Модель имеет 8 слоев, каждый из которых обозначен отдельным цветом на рис. 1. Источник волн задан в виде взрыва с помощью применения вертикальной силы в объеме верхнего слоя. Изменение этой силы во времени задано с помощью импульса Рикера с частотой 30 Гц. Поверхность верхнего слоя считается плоской. На дневной поверхности задано условие свободной границы, а на всех остальных границах – поглощающее граничное условие. На рис. 7 приведена сейсмограмма с удаленным импульсом с источника, приведен вертикальный компонент скорости.

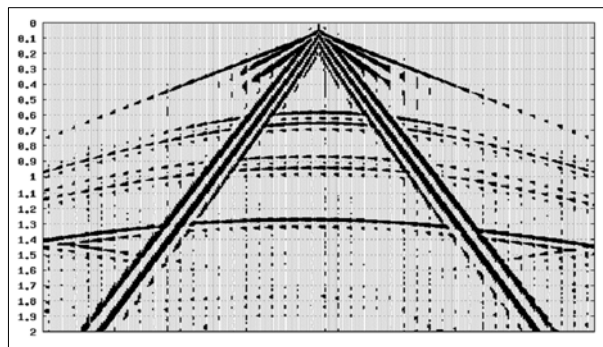


Рис. 7. Расчетная сейсмограмма, вертикальный компонент скорости
Fig. 7. Design seismogram, vertical velocity component

Еще один пример работы алгоритма проводился для стандартной модели SEG C3 NA [23]. На рис. 8 приведено распределение скорости продольных волн в пространстве.

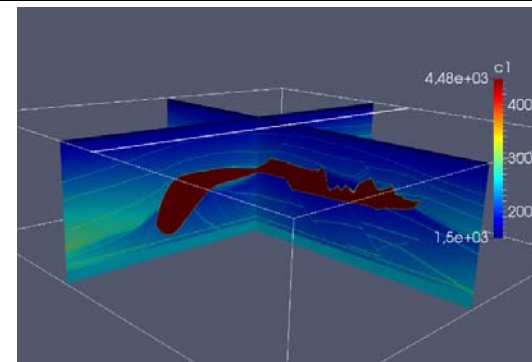


Рис. 8. Модель SEG C3 NA, цветом отображено распределение скорости продольных волн в среде
Fig. 8. Model SEG C3 NA, color shows the distribution of the velocity of longitudinal waves in the medium

Расчет проводился на сетке размером 676x676x201 узел. Источник – импульс Рикера с частотой 30.0 Гц. На рис. 9 приведена сейсмограмма, полученная от источника. Отобразен вертикальный компонент скорости.

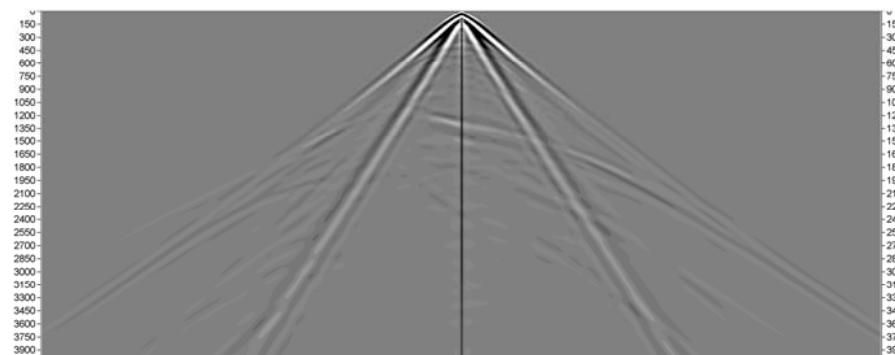


Рис. 9. Сейсмограмма вертикальной составляющей скорости для модели SEG C3 NA
Fig. 9. Seismogram of the vertical velocity component for the SEG C3 NA model

На рис. 10 приведены волновые картины в среде в последовательные моменты времени.

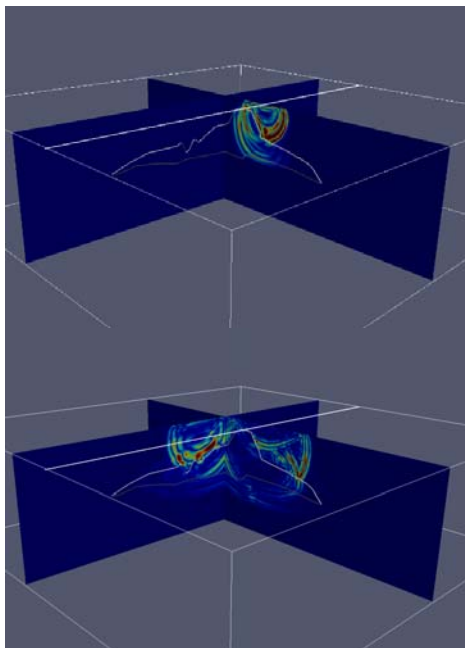


Рис. 10. Волновые картины для модели SEG C3 NA в последовательные моменты времени. Линией обозначен контур соляного купола

Fig. 10. Wave patterns for the SEG C3 NA model at successive times. The line indicates the outline of the salt dome

8. Выводы и результаты

В работе рассмотрен программный комплекс, предназначенный для моделирования распространения динамических волновых возмущений в твердых телах. В основе расчетного алгоритма лежит сеточно-характеристический метод решения систем гиперболических уравнений в частных производных. Данный метод позволяет учитывать волновую структуру уравнения и производить корректную постановку граничных и контактных условий. Реализован алгоритм, позволяющий производить расчет используя блочно-структурные сетки. Алгоритм распараллелен используя технологии MPI, OpenMP, CUDA. В настоящее время достигнута эффективность распараллеливания до 70% с использованием технологии MPI при масштабировании до 16 тысяч вычислительных ядер. В системах с общей памятью алгоритм распараллелен с использованием технологии OpenMP. Кроме того, код распараллелен с использованием технологии 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 8 устройств CUDA. Показана работоспособность алгоритма на тестовых примерах.

Список литературы / References

- [1]. V.A. Biryukov, V.A. Miryakha, I.B. Petrov, and N.I. Khokhlov. Simulation of elastic wave propagation in geological media: Intercomparison of three numerical methods. *Computational Mathematics and Mathematical Physics*, vol. 56, no. 6, 2016, pp. 1086–1095.
- [2]. Голубев В.И., Петров И.Б., Хохлов Н.И. Численное моделирование сейсмической активности сеточно-характеристическим методом. *Журнал вычислительной математики и математической физики*, том 53, № 10, 2013 г., стр. 1709 – 1720 / Golubev V.I., Petrov I.B., Khokhlov N.I. Numerical simulation of seismic activity by the grid-characteristic method. *Computational Mathematics and Mathematical Physics*, vol. 53, № 10, 2013, pp. 1523–1533.
- [3]. P.L. Roe. Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics*, № 18, 1986, pp. 337-365.
- [4]. LeVeque R.J. Finite volume methods for hyperbolic problems. *Cambridge Texts in Applied Mathematics*, vol. 31. Cambridge university press, 2002, 552 p.
- [5]. LeVeque R.J. Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. *Other Titles in Applied Mathematics*, vol. 98. Siam, 2007, 328 p.
- [6]. Strang G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, vol. 5, no. 3, 1968, pp. 506–517.
- [7]. Courant R., Isaacson E., Rees M. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, vol. 5, no. 3, 1952, pp. 243–255.
- [8]. Русанов В.В. Разностные схемы третьего порядка точности для сквозного счета разрывных решений. *Доклады АН СССР*, том 180, no. 6, 1968 г., стр. 1303–1305 / Rusanov V.V. Difference schemes of the third order of accuracy for the forward calculation of discontinuous solutions. *Doklady Akademii Nauk SSSR*, vol. 180, no. 6, pp. 1303–1305 (in Russian).
- [9]. Favorskaya A.V., Petrov I.B. Grid-characteristic method. *Innovations in Wave Processes Modelling and Decision Making*. In *Innovations in Wave Processes Modelling and Decision Making*, Springer, 2018, pp. 117–160.
- [10]. Dagum L., Menon R. OpenMP: an industry standard API for shared-memory programming. *IEEE computational science and engineering*, vol. 5, no. 1, 1998, pp. 46–55.
- [11]. Nakata N., Tsuji T., Matsuoka T. Acceleration of computation speed for elastic wave simulation using a Graphic Processing Unit. *Exploration Geophysics*, vol. 42, no. 1, 2011, pp. 98–104.
- [12]. Weiss R. M., Shragge J. Solving 3D anisotropic elastic wave equations on parallel GPU devices. *Geophysics*, vol. 78, no. 2, 2013, 22 p.
- [13]. F. Rubio et al. Finite-difference staggered grids in GPUs for anisotropic elastic wave propagation simulation. *Computers & geosciences*, vol. 70, issue C, 2014, pp. 181–189.
- [14]. Komatitsch D., Michéa D., Erlebacher G. Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA. *Journal of Parallel and Distributed Computing*, vol. 69, no. 5, 2009, pp. 451–460.
- [15]. D. Komatitsch et al. Modeling the propagation of elastic waves using spectral elements on a cluster of 192 GPUs. *Computer Science – Research and Development*, vol. 25, no. 1/2, 2010, pp. 75–82.
- [16]. Message Passing Interface Forum. MPI: a Message-Passing Interface Standard. Technical Report. University of Tennessee, 1994.
- [17]. Якововский М.В. Введение в параллельные методы решения задач: Учебное пособие. М., Издательство Московского университета, 2013, 328 стр. / Yakobovsky M.V. An Introduction to Parallel Problem Solving Methods: A Training Manual. M., Moscow University Publishers, 2013, 328 p.
- [18]. Ivanov A.M., Khokhlov N.I. Efficient Inter-process Communication in Parallel Implementation of Grid-Characteristic Method. In *Smart Innovation, Systems and Technologies*, vol. 133, 2019, pp. 91–102.
- [19]. Иванов А.М., Хохлов Н.И. Параллельная реализация сеточно-характеристического метода в случае явного выделения контактных границ. *Компьютерные исследования и моделирование*, том 10, № 5, 2018 г., стр. 667-678 / Ivanov A.M., Khokhlov N.I. Parallel implementation of the grid-characteristic method in the case of explicit contact boundaries. *Computer Research and Modeling*, vol. 10, № 5, 2018, pp. 667–678 (in Russian).
- [20]. Khokhlov N. et al. Solution of Large-scale Seismic Modeling Problems. *Procedia Computer Science*,

- vol. 66, 2015, pp. 191–199.
- [21]. Khokhlov N. et al. Applying OpenCL Technology for Modelling Seismic Processes Using Grid-Characteristic Methods. *Communications in Computer and Information Science*, vol. 678, 2016, pp. 577–588.
- [22]. Golubev V.I. et al. Simulation of dynamic processes in three-dimensional layered fractured media with the use of the grid-characteristic numerical method. *Journal of Applied Mechanics and Technical Physics*, vol. 58. № 3, 2017, 539–545.
- [23]. Aminzadeh F., Brac J., and Kunz T. 3D Salt and Overthrust models. In *Distribution CD of Salt and Overthrust models, SEG/EAGE Modeling Series*, No. 1, 1997.
- [24]. Фаворская А.В., Петров И.Б. Численное моделирование волновых процессов в скальных массивах сеточно-характеристическим методом. *Математическое моделирование*, том 30, № 3, 2018 г., стр. 37–51 / Favorskaya A. V., Petrov I.B. Numerical Modeling of Wave Processes in Rocks by the Grid-Characteristic Method. *Mathematical Models and Computer Simulations*, vol. 10. issue 5, 2018, pp. 639–647.

Информация об авторах / Information about authors

Николай Игоревич ХОХЛОВ, кандидат физико-математических наук, работает в МФТИ в должности старшего научного сотрудника, заместителя заведующего лабораторией прикладной вычислительной геофизики. Научные интересы включают численное моделирование, параллельные алгоритмы, высокопроизводительные вычислительные системы и технологии написания параллельных приложений.

Nikolai Igorevich KHOKHLOV, candidate of physical and mathematical sciences, works at MIPT as a senior researcher, deputy head of the laboratory of applied computational geophysics. His research interests include numerical modeling, parallel algorithms, high-performance computing systems, and parallel application writing technologies.

Игорь Борисович ПЕТРОВ является доктором физико-математических наук, профессором, членом-корреспондентом РАН, заведующим кафедры информатики МФТИ. В число научных интересов входят сеточно-характеристический метод, математическое моделирование, высокопроизводительные вычислительные системы

Igor Borisovich PETROV is a doctor of physical and mathematical sciences, professor, corresponding member of the Russian Academy of Sciences, head of the department of computer science at the Moscow Institute of Physics and Technology. His research interests include the grid-characteristic method, mathematical modeling, and high-performance computing systems.