ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МОДЕЛИРОВАНИЯ ТРЕХМЕРНОГО ТЕЧЕНИЯ СТОКСА МЕТОДОМ ГРАНИЧНЫХ ЭЛЕМЕНТОВ

Э. Н. Середин, Д. Г. Прибыток
Объединённый институт проблем информатики Национальной Академии Наук Беларуси
Гданьский Политехнический Университет
Минск, Республика Беларусь;Гданьск, Польша
E-mail: {eduard.seredin, dpribytok}@tut.by

Рассматривается применение технологии параллельных вычислений для моделирования трёхмерного течения вязкой жидкости (течение Стокса) прямым методом граничных элементов. Решение задачи строится в три этапа: дискретизация и построение СЛАУ, её решение и нахождение вектора скорости жидкости в заданных точках. Для построения СЛАУ и нахождения вектора скорости были разработаны и реализованы параллельные алгоритмы с помощью технологии программирования видеокарт CUDA. Проведено сравнение временных затрат для алгоритмов на примере расчёта движения вязкой жидкости в трёхмерной каверне.

Введение

Для многих научных, инженерных и прикладных задач имеется достаточно большое количество алгоритмов, для которых довольно успешно разрабатываются их параллельные аналоги [1, 2]. Ниже рассматривается создание параллельного алгоритма на примере моделирования трехмерных течений Стокса и проводится сравнительный анализ быстродействия отдельных частей алгоритма с их последовательными версиями.

I. Краткое описание алгоритма

Течение Стокса [3, 4] – это модель движения жидкости, которая выводится из уравнений Навье-Стокса при допущении малости числа Рейнольдса, Re«1. Данная модель может использоваться для описания течения вязких полимеров, в теории смазки, течения лавы, для моделирования движения крови в организме человека и т. д. Математическая модель для трёхмерного случая может быть представлена в интегральной форме

$$\xi(x_0) \cdot u_j(x_0) = -\frac{1}{\mu} \oint_\Omega \phi_i(x) \cdot G_{ij}(x_0 - x) d\Omega \cdot$$

$$\cdot \oint_\Omega u_i \cdot T_{ijk}(x_0 - x) \cdot n_k(x) d\Omega, \qquad (1)$$

где i,j,k могут принимать значения 1, 2, 3, u – трёхмерный вектор скорости жидкости, $\mu = const$ – коэффициент динамической вязкости жидкости, Ω – граница области решения, x – точка, принадлежащая Ω, n(x) – вектор внешней нормали к Ω в точке x, x_0 – произвольная точка пространства, $\xi(x_0)$ – весовая функция, $G_{ij}(r)$ – тензорный потенциал Стокса [5], $T_{ijk}(r)$ – тензор, ассоциированный с G_{ij}, $\phi_i(x)$ – вектор напряжений на границе Ω в точке x.

Решение строится методом граничных элементов [4, 6], главной особенностью которого является необходимость в дискретизации лишь границы области решения, что уменьшает количество дискретных элементов (граничных элементов - ГЭ), а также позволяет решать задачи в областях со сложной геометрией, как например, в случае многосвязных областей [7]. Основным недостатком метода является большая потребность в оперативной памяти, вытекающая из высокой плотности матрицы СЛАУ, которая имеет размер 3N·3N, где N - количество ГЭ.

II. Последовательный алгоритм

Разобьем границу трехмерной области на N граничных элементов, представляющих собой пространственные прямоугольники либо треугольники. В центре каждого ГЭ зададим локальную систему координат. Будем предполагать, что функции напряжений $\phi_i(x)$ являются на нём постоянными. Для каждого ГЭ зададим в качестве граничного условия вектор скорости. В уравнении (1), поочередно помещая рассматриваемую точку x_0 в центры полученных элементов, получим следующий вид интегрального уравнения:

$$u(\xi_0^p) = -\frac{1}{4\pi\mu} \sum_{q=1}^{N} [\tilde{\phi}(\xi_0^q) \cdot A_{qp} \cdot \int\int_{S_q} G_{ij}(r_{pq}) dS_q] +$$

$$+\frac{1}{4} I^K(\xi_0^p), \qquad (2)$$

$$I^K(\xi_0^p) = \sum_{q=1}^{N} [\tilde{u}(\xi_0^q) \cdot$$

$$\cdot A_{qp} \cdot \int\int_{S_q} T_{ijk}(r_{pq}) \cdot (0,0,1) dS_q], \qquad (3)$$

где $r_{pq} = A_{gq} \cdot (\xi_0^q - \xi_0^p) - (\xi_x, \xi_y, 0)$, $i,j,k = 1,2,3$, $p,q = 1,2,3,...,N$, ξ_0^p, ξ_0^q - центры элементов с индексами p,q, A_{qp} - матрица перехода от q-й к p-й системе координат, A_{gq} - матрица перехода от глобальной к q-й системе координат, S - область

ГЭ, определенная в его декартовой системе координат $O\xi_x\xi_y$.

В результате получается система линейных уравнений относительно неизвестных значений ϕ

$$C \cdot \phi = B, \tag{4}$$

$$C_{pq} = \frac{1}{4\pi\mu} A_{qp} \cdot \int\int_{S_q} G_{ij}(r_{pq} dS_q), \tag{5}$$

$$B_p = -\tilde{u_p} + \frac{1}{4}\sum_{q=1}^{N}[\tilde{u}(\xi_0^q)\cdot$$

$$\cdot A_{qp} \cdot \int\int_{S_q} T_{ijk}(r_{pq}) \cdot (0,0,1) dS_q]. \tag{6}$$

После решения СЛАУ (4) скорость в любой точке x внутри области решения можно найти из выражения:

$$u(x) = \sum_{q=1}^{N} A_{qg} \cdot (\tilde{u}(\xi_0^q) \cdot U_q(x) - \tilde{\phi}(\xi_0^q) \cdot F_q(x)), \tag{7}$$

$$U_q(x) = \frac{1}{8}\int\int G_{ij}(r_{xq} dS_q), \tag{8}$$

$$F_q(x) = \frac{1}{8\pi\mu}\int\int_{S_q} T_{ijk}(r_{xq} \cdot (0,0,1) dS_q), \tag{8}$$

$r_{xq} = A_{gq} \cdot (x - \xi_0^q) - (\xi_x, \xi_y, 0)$, A_{qg} - матрица перехода от q-й к глобальной системе координат.

III. Параллельный алгоритм построения СЛАУ

Преимуществом параллельного алгоритма построения СЛАУ (4) с помощью технологии программирования видеокарт CUDA является возможность вычислять коэффициенты матрицы полностью независимо друг от друга, параллельно. Для свободного члена (6) необходимо применять специальную методику вычисления суммы, называемую редукцией [8]. Ускорение расчетов, получаемое от разработанного параллельного алгоритма заполнения матрицы, будет зависеть от используемой видеокарты с поддержкой технологии CUDA.

IV. Параллельный алгоритм нахождения скорости в заданных точках

После нахождения неизвестных значений напряжений необходимо использовать выражение (7) для поиска скоростей жидкости в наборе заданных точек. Создается прямоугольная сетка блоков размером N · M, где M – общее число точек, для которых необходимо найти вектор скорости потока. В память видеокарты помещаются массивы: граничных элементов, граничных условий, найденных значений напряжений ϕ и координат точек для которых необходимо найти скорости жидкости. Для каждой нити в блоке с координатами (q, i) ставится в соответствие и вычисляется значение слагаемого суммы $A_{qg} \cdot [\tilde{u} \cdot U_q(x_i) + \tilde{\phi_q} \cdot F_q(x_i)]$. По аналогии с вычислением свободного члена СЛАУ используется методика редукции для вычисления суммы (7).

V. Течение вязкой жидкости в трехмерной каверне

Для алгоритма построения СЛАУ ускорение работы программы при использовании видеокарты достигает до 200 раз, для алгоритма решения СЛАУ – до 943 раз, для алгоритма поиска вектора скорости в заданных точках – до 242 раз.

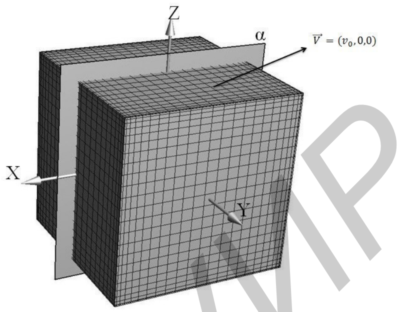

Рис. 1 – Граничноэлементная сетка каверны

VI. Заключение

Полученные результаты показывают [11], что за счет применения параллельных алгоритмов можно существенно увеличить скорость вычислений, связанных с моделированием трёхмерного течения Стокса, что может быть особенно полезно в динамических задачах.

Список литературы

1. Сандерс, Дж. Технолония CUDA в примерах: введение в программирование графических процессоров / Дж. Сандерс, Э. Кэндрот // Москва: ДМК Пресс. – 2011. – 232 с.
2. Середин, Э.Н. Фильтрация и корреляционная обработка изображений с помощью технологии CUDA / Э.Н. Середин, Б.А. Залесский // Весцi НАН Беларусi. Сер. физ.-мат. навук. – 2015. – №1. – С. 106-116.
3. Brebbia, C. A. Boundary Element Methods in Engineering / C.A. Brebbia // Springer-Verlag. – 1982.
4. Pozrikidis, C. Boundary Integral and Singularity Methods for Linearized Viscous Flow / C. Pozrikidis // Cambridge University Press. – 1992.
5. Lisicki, M. Four approaches to hydrodynamic Green's functions - the Oseen tensors / M. Lisicki // Institute of Theoretical Physics, Faculty of Physics, University of Warsaw. – 2013.
6. Cheng, A. Heritage and early history of the boundary element method / A. Cheng // Elsevier Science. – 2003.
7. Katsikadelis, J. Boundary Elements: Theory and Applications / J. Katsikadelis // Elsevier Science. – 2002.
8. Боресков, А.В. Технолония CUDA в примерах: введение в программирование графических процессоров / А.В. Боресков, А.А. Харламов // Москва: ДМК Пресс. – 2010. – 232 с.
9. CUDA Toolkit Documentation [Элетронный ресурс] / Режим доступа: http://docs.nvidia.com/cuda/index.html. – Дата доступа: 20.01.2016.
10. CUDA Toolkit Documentation [Элетронный ресурс] / Режим доступа: http://docs.nvidia.com/cuda/cusolver/index.html. – Дата доступа: 20.01.2016.
11. Прибыток, Д.Г. Параллельный алгоритм моделирования трехмерного течения стокса методом граничных элементов / Д.Г. Прибыток, Э.Н. Середин // Информатика. – 2016. – №2. – С. 52-58.