

Трехмерное моделирование течения эмульсии методом граничных элементов на гетерогенных системах¹

Солнышкина О.А.

Центр «Микро- и наномасштабная динамика дисперсных систем» БашГУ, Уфа
Институт механики им. Р.Р. Мавлютова УНЦ РАН, Уфа

В работе рассматривается динамика двух вязких несмешивающихся жидкостей в неограниченной области в трехмерной постановке при малых числах Рейнольдса. В основе используемой численной методики лежит метод граничных элементов, который очень эффективен для исследования трехмерных задач в бесконечных областях. Для ускорения расчетов и увеличения масштаба рассматриваемых задач применяется гетерогенный подход к распараллеливанию вычислений на центральном (CPU) и графических (GPU) процессорах. Для ускорения итерационного решателя (GMRES) и преодоления ограничений, связанных с размером памяти вычислительной системы, разработан модуль матрично-векторного произведения «на лету» и распараллелен на графических процессорах (GPU). Также реализована multi-GPU версия модуля. Представлены результаты по эффективности использования GPU.

1. Введение

Эмульсии возникают во многих областях промышленности: пищевая промышленность, переработка отходов, косметология, медицина. В нефтяной области их используют в потокоотклоняющих технологиях, для глушения скважин, для выравнивания профиля приемистости скважин. Таким образом, необходимо разработать эффективный численный аппарат, позволяющий детально исследовать поведение эмульсий, включая взаимодействие капель, их коалесценцию, и наблюдать многие эффекты на микроуровне в подобных системах «жидкость–жидкость».

Для более точного определения особенностей реологии и микроструктуры эмульсий требуется рассматривать большой объем дисперсной среды, что неизбежно приводит к необходимости разработки эффективного численного аппарата для исследования крупномасштабных задач. Для ускорения расчетов и увеличения масштаба рассматриваемых задач применяется гетерогенный подход к распараллеливанию вычислений на центральном (CPU) и графических (GPU) процессорах.

В работе исследуется трехмерное течение Стокса двух вязких несмешивающихся жидкостей

в неограниченной области [4]. Метод граничных элементов для Стоксовых течений изложен в [5] и успешно применялся для расчета взаимодействия капель и частиц в дисперсных течениях [3], в указанных статьях также можно найти обзор литературы.

Для решения поставленных задач используются новые эффективные подходы к численному моделированию трехмерных задач и современные информационные технологии. Программная реализация задач предусматривает выбор оптимальных алгоритмов в зависимости от количества узлов сетки. Для ускорения расчетов разработан модуль матрично-векторного произведения без хранения матрицы в памяти вычислительной системы, который используется в итерационном решателе GMRES, и распараллелен как на обычном многоядерном процессоре (CPU), так и на графических процессорах (GPU) с использованием технологии CUDA.

2. Математическая модель

Рассматривалась задача о движении деформируемых капель ньютоновской жидкости 2 плотности ρ_2 и вязкости μ_2 в потоке другой жидкости 1 плотности ρ_1 и вязкости μ_1 . Жидкости занимают объем V_2 и V_1 соответственно, S — граница раздела фаз. Гравитационная постоянная g и коэффици-

¹Работа выполнена при поддержке Министерства образования и науки РФ (грант 11.G34.31.0040).

ент поверхностного натяжения капли γ постоянны. Динамика дисперсной системы происходит под действием сдвигового потока $u_s(x)$ в направлении оси OX в поле силы тяжести с установившейся скоростью падения u_g в направлении оси OZ . При числах Рейнольдса Re_1, Re_2 значительно меньше 1, поведение внешнего потока жидкости 1 и поведение жидкости 2 внутри капли описывается уравнениями Стокса:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}_i &= -\nabla p + \mu_i \nabla^2 \mathbf{u}_i = 0, \\ \nabla \cdot \mathbf{u} &= 0, \quad i = 1, 2, \end{aligned} \quad (1)$$

где $\boldsymbol{\sigma}$ — тензор напряжений; p — давление.

Условия на границе двух жидкостей:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{u}_2 = \mathbf{u}, \quad \mathbf{f} = \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{f}_1 - \mathbf{f}_2 = f \mathbf{n}, \\ f &= \gamma(\nabla \cdot \mathbf{n}) + (\rho_1 - \rho_2)(\mathbf{g} \cdot \mathbf{x}), \quad \mathbf{x} \in S, \end{aligned} \quad (2)$$

где \mathbf{f} — поверхностное напряжение; \mathbf{n} — нормаль к поверхности, направленная в жидкость 1.

Кинематическое условие, описывающее динамику капли

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}), \quad \mathbf{x} \in S. \quad (3)$$

3. Численное моделирование

Задача решалась методом граничных элементов [1]. Суть метода состоит в переходе от дифференциального уравнения в частных производных, описывающего поведение неизвестной функции внутри и на границе области, к интегральному уравнению, определяющему только граничные значения, и поиске численного решения этого уравнения. При необходимости определения значения потенциала во внутренних точках области, их можно вычислить, используя известные решения на границе.

Таким образом, размерность задачи уменьшается на единицу. Все это делает метод граничных элементов достаточно эффективным при моделировании трехмерных задач в бесконечных областях или областях со сложной геометрией.

Для дискретизации поверхности каплей использовалась триангуляция Делоне. Наиболее трудоемкой процедурой является вычисление средней кривизны поверхности в каждом узле сетки. Средняя кривизна вычислялась двумя методами: контурных интегралов и установления параболоида. Последний дает более точное значение средней кривизны с погрешностью не более 3%, но для его использования валентность всех узлов сетки должна быть не менее пяти.

В процессе эволюции во времени узлы сетки движутся согласно уравнению (3), которое на первых итерациях решается методом Рунге–Кутты 4-го

порядка, а затем методом Адамса–Бэшфорта. Для того чтобы сохранить устойчивости сетки и избежать перекрытия поверхностей капель, был выбран достаточно маленький шаг по времени.

Для данной поверхности S скорость \mathbf{u} в произвольной точке \mathbf{y} может быть вычислена через граничные интегралы [3, 4]

$$\left. \begin{aligned} \mathbf{y} \in V_1, \quad \mathbf{u}(\mathbf{y}) - 2\mathbf{u}_\infty(\mathbf{y}) \\ \mathbf{y} \in V_2, \quad \lambda \mathbf{u}(\mathbf{y}) \\ \mathbf{y} \in S, \quad \frac{1+\lambda}{2} \mathbf{u}(\mathbf{y}) - \mathbf{u}_\infty(\mathbf{y}) \end{aligned} \right\} = \int_S \left\{ -\frac{1}{\mu} \mathbf{G}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{f}(\mathbf{x}) - (1-\lambda)[\mathbf{T}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{n}(\mathbf{x})] \cdot \mathbf{u}(\mathbf{x}) \right\} dS(\mathbf{x}), \quad (4)$$

где $\mu = \mu_1$ — вязкость обтекающей жидкости; $\lambda = \mu_2/\mu_1$ — отношение вязкостей внутренней и внешней жидкостей; \mathbf{G} — фундаментальное решение уравнения Стокса и \mathbf{T} — тензор напряжений, определяются как

$$\mathbf{G}(\mathbf{x}, \mathbf{y}) = \frac{1}{8\pi} \left(\frac{\mathbf{I}}{r} + \frac{\mathbf{r}\mathbf{r}}{r^3} \right), \quad (5)$$

$$\mathbf{T}(\mathbf{x}, \mathbf{y}) = -\frac{3}{4\pi} \frac{\mathbf{r}\mathbf{r}\mathbf{r}}{r^5}, \quad \mathbf{r} = \mathbf{x} - \mathbf{y}, \quad r = |\mathbf{r}|.$$

Поверхность S разбивается на N узлов \mathbf{x}_i , по которым мы строим квадратурные формулы граничных интегралов.

Используя метод вершинных коллокаций, последнее уравнение в граничных интегралах (4) в точках $\mathbf{y} = \mathbf{x}_j$ можно записать

$$\sum_{i=1}^N \left[\frac{1+\lambda}{2} \mathbf{I}_{ji} + (1-\lambda) \mathbf{I}_{ji}^{(T)} \right] \cdot \mathbf{u}_i = \mathbf{u}_{\infty, j} - \frac{1}{\mu} \sum_{i=1}^N \mathbf{I}_{ji}^{(G)} f_i, \quad j = 1, \dots, N, \quad (6)$$

$$\mathbf{I}_i^{(G)}(\mathbf{y}) = \int_{S_i} \mathbf{G}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x}), \quad (7)$$

$$\mathbf{I}_i^{(T)}(\mathbf{y}) = \int_{S_i} \mathbf{T}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x}), \quad (8)$$

где \mathbf{I}_{ji} — единичная матрица, остальные пространственные функции записаны в рассматриваемой точке коллокации.

Вычисление сингулярных элементов производилось методом пробных функций на основе известных интегральных тождеств [5].

Решая систему линейных алгебраических уравнений (6), получаем компоненты скорости на границе.

4. Результаты использования графических процессоров для ускорения расчетов

При трехмерном численном моделировании физических процессов для областей со сложной геометрией, например, течение эмульсии в микроканалах переменного сечения, необходимо построение сеток с большим количеством узлов. Решение подобных многомасштабных задач требует разработки и применения эффективных численных методов. Для сеток маленького размера при решении СЛАУ применялись прямые методы, но при увеличении масштаба задачи их использование затрудняется. Это связано с тем, что размер необходимой памяти пропорционален квадрату числа узлов сетки, также при их увеличении возрастает время вычислений. При использовании прямых методов, начиная с некоторого количества узлов, возникает нехватка памяти вычислительной системы.

Эту проблему можно решить используя итерационные методы решения, которые существенно снижают затраты памяти и времени. Наиболее эффективными и устойчивыми среди итерационных методов решения таких систем уравнений являются так называемые проекционные методы, и особенно тот их класс, который связан с проектированием на подпространства Крылова (например, GMRES). Эти методы обладают целым рядом достоинств: они устойчивы, допускают эффективное распараллеливание, работу с различными строчными (столбцовыми) форматами и предобуславливателями разных типов [6].

Для эффективной программной реализации итерационного метода необходимо разработать подпрограмму, быстро умножающую матрицу на вектор, и ускорить сходимость метода с помощью предобуславливателя.

В рамках данной работы для решения проблем, связанных с использованием памяти, в среде Matlab был разработан программный модуль умножения матрицы на вектор без хранения матрицы в памяти системы (MV product on the fly), который используется в GMRES при решении СЛАУ. Каждый элемент матриц G и K вычислялся по следующим формулам:

$$\begin{aligned} G_{mn}^{ij} &= S_n \mathbf{G}(\mathbf{x}_m - \mathbf{x}_n) = \\ &= \frac{1}{8\pi i} S_n \left(\frac{\delta_{ij}}{|\mathbf{x}_m - \mathbf{x}_n|} - \right. \\ &\quad \left. - \frac{(x_m^i - x_n^i)(x_m^j - x_n^j)}{|\mathbf{x}_m - \mathbf{x}_n|^3} \right), \end{aligned} \quad (9)$$

$$\begin{aligned} K_{mn}^{ij} &= S_n \mathbf{K}(\mathbf{x}_m - \mathbf{x}_n) = \\ &= -\frac{3}{4\pi i} S_n \frac{(x_m^i - x_n^i)(x_m^j - x_n^j)}{|\mathbf{x}_m - \mathbf{x}_n|^5} \times \\ &\quad \times \sum_{k=1}^N (x_m^k - x_n^k) n_n^k, \end{aligned} \quad (10)$$

$$m, n = 1, \dots, N, \quad i, j, k = 1, 2, 3,$$

где N — количество узлов дискретизации области. Таким образом матрицы G и K имеют размер $3N \times 3N$ и $3N \times 3N$.

Применение модуля матрично-векторного произведения позволяет решить проблему ограничения по памяти вычислительной системы, но матричные вычисления являются вычислительно-трудоемкими, поэтому представляют собой классическую область применения параллельных вычислений. Для ускорения расчетов модуль был распараллелен на центральном многоядерном процессоре (CPU) средствами Matlab Parallel Computing Toolbox и с помощью программно-аппаратной технологии CUDA на графических процессорах (GPU) [7].

Использование графических процессоров для данной задачи обусловлено тем, что выполнение расчетов на GPU показывает отличные результаты в алгоритмах, использующих параллельную обработку данных (применение одной и той же последовательности математических операций ко множеству данных). При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико.

Для многих методов матрично-векторных вычислений характерно наличие параллелизма по данным и в большинстве случаев распараллеливание таких операций сводится к разделению обрабатываемых матриц между процессорами используемой вычислительной системы. Наиболее общие способы разделения матриц состоят в разбиении данных на полосы (горизонтальные или вертикальные) или на прямоугольные фрагменты (блоки).

В рамках данной работы распараллеливание модуля «MV product on the fly» на GPU основывалось на разбиении матриц горизонтальными полосами на m частей так, что $N = m \times L$, где N — размерность матрицы; L — число строк матрицы в блоке; m — количество потоков на GPU.

На каждой итерации каждый из m потоков вычисляет свою часть вектора решения. То есть при умножении матрицы G размера $3N \times 3N$ на вектор b размера $3N \times 1$ на одной итерации поток вычисляет очередной свой блок матрицы размером 3×3 по формулам (9)–(10) и умножает полученный блок на соответствующую часть вектора b размером 3×1

и прибавляет результат к полученному на предыдущей итерации вектору (рис. 1). Каждый поток хранит часть результирующего вектора, которая по окончании вычислений копируется в global memory для получения полного вектора решения.

На рис. 2 показано сравнение времени выполнения модуля матрично-векторного произведения на одном ядре CPU (MVP on the fly), на 8 ядрах CPU (parallel MVP on the fly), на GPU (parallel GPU MVP on the fly) и встроенной функции умножения матрицы на вектор Matlab (Matlab MVP).

5. Результаты численного моделирования

Все расчеты для single-GPU версии модуля проводились на вычислительной системе с CPU Intel Xeon 5660, 2.8GHz, GPU NVIDIA Tesla C2050. При проведении вычислительных экспериментов было установлено, что для различных размеров задачи наилучшие результаты по времени достигаются при размере блока равным 256 потоков.

На рис. 3 приведено ускорение расчета на GPU по сравнению с CPU для операций с двойной и одинарной точностью. При расчетах на графической карте NVIDIA Tesla C2050 для количества узлов сетки N от 1 000 до 100 000 для уравнений Стокса получено ускорение до 520 раз для операций с двойной точностью и до 700 раз для операций с одинарной точностью.

При проведении тестовых расчетов достигнута следующая производительность: до 220 Gflops для операций с двойной точностью и 300 Gflops для операций с одинарной точностью (рис. 4). Учитывая, что пиковая производительность графической карты Tesla C2050 для чисел с плавающей точкой двойной точности составляет 515 Gflops, одинарной точности — 1,03 Tflops, получены хорошие результаты.

Также была реализована multi-GPU версия модуля и проведено несколько тестовых расчетов на рабочей станции с 3 графическими картами NVIDIA Tesla C2075 и с двумя CPU Intel Xeon 5660, 2.8GHz. На рис. 5 представлены некоторые результаты по ускорению расчетов на GPU в зависимости от размера задачи.

Результаты тестов на графической карте NVIDIA Tesla C2050 показали возможность решения граничных задач для уравнений Стокса размером до 100 000 элементов на одной рабочей станции.

Численные расчеты проводились для обратной водонефтяной эмульсии с плотностью дисперсной фазы $\rho_2 = 10^3$, вязкостью $\mu_2 = 10^{-3}$ и коэффициентом поверхностного натяжения $\gamma = 0.05$, с плотностью несущей фазы $\rho_1 = 0.8 \cdot 10^3$ и вязкостью

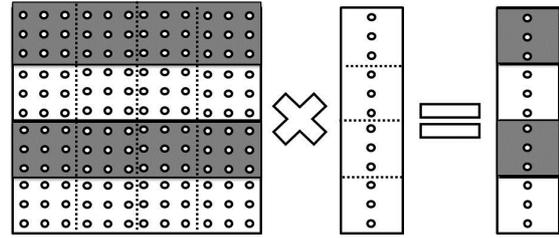


Рис. 1. Разделение данных и организация вычислений при выполнении параллельного модуля матрично-векторного произведения на GPU

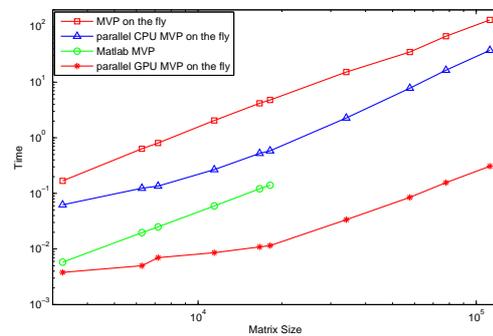


Рис. 2. Время вычисления матрично-векторного произведения

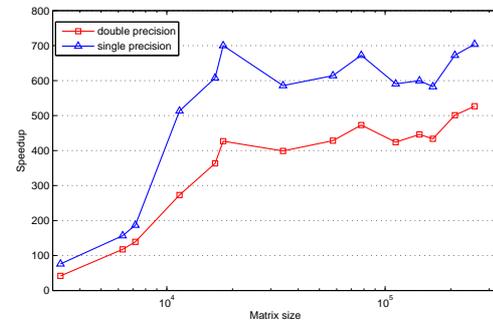


Рис. 3. Ускорение расчетов в зависимости от размера матрицы

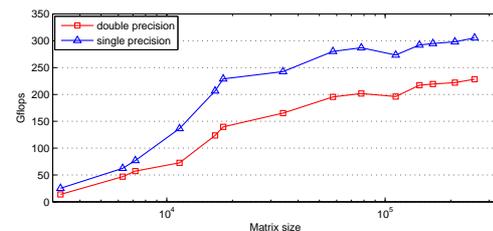


Рис. 4. Производительность в зависимости от размера матрицы

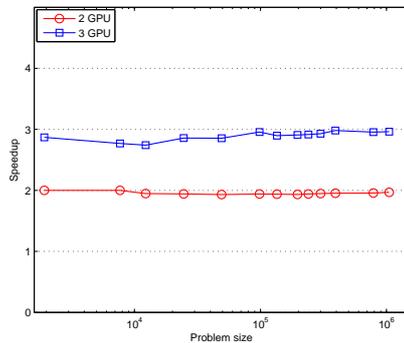


Рис. 5. Ускорение на нескольких GPU в зависимости от размера матрицы

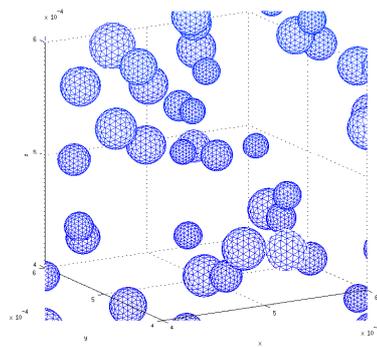


Рис. 6. Капли эмульсии в сдвиговом потоке в неограниченной области в момент времени $t = 0$ (фрагмент вычислительной области)

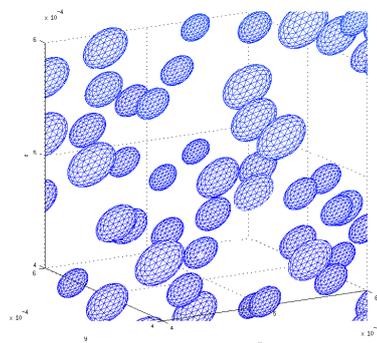


Рис. 7. Капли эмульсии в сдвиговом потоке в неограниченной области в момент времени $t = 100$ (фрагмент вычислительной области)

$\mu_2 = 2 \cdot 10^{-3}$. Течение жидкой двухфазной среды рассматривалось в сдвиговом потоке $u_s(x) = 5 \cdot 10^5 \cdot x_z + 10$ в неограниченной области с капиллярным числом $Ca = 0.2$, числом Рейнольдса $Re = 4$, числом Бонда $Bo = 4 \cdot 10^{-6}$. Все переменные заданы в системе СИ.

На рис. 6, 7 изображен фрагмент расчетной области $(4 \cdot 10^{-4}; 6 \cdot 10^{-4})^3 m^3$ в различные моменты времени. Изначально в расчетной области задавалось произвольное распределение сферических капель разного радиуса. В процессе моделирования капли вытягиваются по потоку, действие силы тяжести при заданных параметрах за короткий промежуток времени практически не наблюдается, поскольку скорость осаждения капель значительно меньше скорости сдвигового потока.

Список литературы

- [1] Бреббия К. Методы граничных элементов. Пер. с англ. М.: Мир, 1987. С. 524.
- [2] Хаппель Дж., Бреннер Г. Гидродинамика при малых числах Рейнольдса. Пер. с англ. М.: Мир, 1976. С. 623.
- [3] Zinchenko A.Z. and Davis R.H. An efficient algorithm for hydrodynamical interaction of many deformable drops // J. Comp. Phys. 2000. Vol. 157. Pp. 539–587.
- [4] Rallison J.M. A numerical study of the deformation and burst of a viscous drop in an extensional flow // J. Fluid Mech. 1978. Vol. 89, part 1. P. 191–200.
- [5] Pozrikidis C. Boundary Integral and Singularity Methods for Linearized Viscous Flow. 1992 (Cambridge University Press, Cambridge, MA).
- [6] Saad Y. Iterative Methods for Sparse Linear System. 2000, SIAM.
- [7] NVIDIA Corporation. NVIDIA CUDA Compute Unified Device Architecture Programming Guide. Version 3.2.2010.