

О решении уравнения Навье-Стокса в переменных «функция тока – вихрь» на многопроцессорной вычислительной машине с использованием системы CUDA

Ясинский Ф.Н., д-р физ.-мат. наук, Евсеев А.В., асп.

Исследуется методика моделирования движения вязкой несжимаемой жидкости, описываемой уравнением Навье-Стокса в переменных «функция тока – вихрь», на графических ускорителях Nvidia в системе CUDA.

Ключевые слова: численное моделирование, вычислительная гидродинамика, уравнение Навье-Стокса, уравнение Пуассона, уравнение Лапласа, метод прогонки, метод переменных направлений, CUDA, GPU общего назначения, параллельный алгоритм, многопроцессорный, многопоточный.

On Solution of Navier–Stokes Equation in «Stream Function – Curl» Variable on Multiprocessor Computer with CUDA System

F.N. Yasinskiy, Doctor of Physics and Mathematics, A.V. Evseev, Post-Graduate Student

The article studies the technique of using Nvidia CUDA on GPUs for modeling the viscous incompressible fluid motion, described by the Navier-Stokes equations in the stream function from.

Keywords: numerical simulation, CFD, the Navier-Stokes equation, the Poisson equation, the Laplace equation, sweep method, alternating direction method, CUDA, GPUPU, parallel algorithm, multiprocessor, multithreaded.

Уравнение Навье-Стокса традиционно записывается в системе «скорость – давление» [1]:

$$\begin{cases} \frac{\partial \bar{u}}{\partial t} = -(\bar{u} \cdot \nabla) \bar{u} + \nu \Delta \bar{u} - \frac{1}{\rho} \nabla P + \bar{F}, \\ \operatorname{div} \bar{u} = 0, \end{cases} \quad (1)$$

где \bar{u} – вектор скорости; P – давление среды; \bar{F} – ускорение макрочастиц среды, вызванное обменом количеством движения с соседними макрочастицами; ν – кинематическая вязкость.

В некоторых задачах может оказаться предпочтительнее форма записи уравнения Навье-Стокса в переменных «функция тока – вихрь». Если ограничиться двумерным случаем, то можно ввести следующие определения:

- функции тока

$$\begin{cases} u_1 = \frac{\partial \Psi}{\partial x_2}, \\ u_2 = -\frac{\partial \Psi}{\partial x_1}; \end{cases} \quad (2)$$

- вихря

$$\omega = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}. \quad (3)$$

Соответственно, уравнение Навье-Стокса преобразуется в следующую форму [1]:

$$\frac{\partial \omega}{\partial t} + u_1 \frac{\partial \omega}{\partial x_1} + u_2 \frac{\partial \omega}{\partial x_2} = \nu \left(\frac{\partial^2 \omega}{\partial x_1^2} + \frac{\partial^2 \omega}{\partial x_2^2} \right), \quad (4)$$

$$\frac{\partial^2 \Psi}{\partial x_1^2} + \frac{\partial^2 \Psi}{\partial x_2^2} = -\omega. \quad (5)$$

На твердых и неподвижных поверхностях граничные условия задаются следующим образом:

$$u_1|_{\Gamma} = 0;$$

$$u_2|_{\Gamma} = 0;$$

$$\Psi|_{\Gamma} = \text{const};$$

$$\omega|_{\Gamma} = -\frac{\partial^2 \Psi}{\partial n^2} \Big|_{\Gamma},$$

где n – нормаль к границе Γ .

Константа для функции тока на стенках выбирается просто: на одной из граничных стенок функция тока равна нулю, на других – функция тока подбирается таким образом, что разница функций между двумя границами равна объемному расходу среды между ними:

$$Q = \Psi|_{\Gamma_2} - \Psi|_{\Gamma_1},$$

где $\Psi|_{\Gamma_1}$ – значение функции тока на границе Γ_1 ;

$\Psi|_{\Gamma_2}$ – на границе Γ_2 .

На входных и выходных отверстиях задается профиль скорости, а расчетные формулы для функции тока и вихря получаются либо интегрированием, либо дифференцированием, в соответствии с их определениями (2)–(3).

Таким образом, общий алгоритм решения можно описать следующей последовательностью действий:

1) устанавливаем $t = 0$ и задаем граничные условия для полей скорости и функции тока;

2) чтобы стартовые условия были оптимальными, находим начальное поле функции тока путем решения уравнения Лапласа:

$$\frac{\partial^2 \Psi}{\partial x_1^2} + \frac{\partial^2 \Psi}{\partial x_2^2} = 0;$$

- 3) обновляем граничные условия для вихря;
- 4) находим новое поле скоростей \vec{u}^{t+1} по значениям функции тока;
- 5) решаем уравнение Навье-Стокса для вихря ω^{t+1} (4);
- 6) используя полученные значения ω^{t+1} решаем уравнение Пуассона для функции тока Ψ^{t+1} (5);
- 7) если не достигнут конечный момент времени, то возвращение на шаг 3, иначе – окончание работы алгоритма.

Как можно видеть, в предложенном алгоритме присутствуют две сложные для вычисления части – это ОДУ (4) с конвективными членами и уравнение Пуассона (5).

Уравнение Пуассона – это эллиптическое дифференциальное уравнение в частных производных [2, 4], которое записывается в следующем виде:

$$\Delta u = f.$$

Его решение не является тривиальной задачей. В то же время это уравнение является основополагающим в некоторых областях науки. Так, оно используется в электростатике, аэрогидродинамике, в расчетах плазменных систем, при определении стационарного распределения тепла в теле и в задачах магнитостатики.

К сожалению, уравнение Пуассона в вычислительном смысле является «тяжелым». Большая часть расчетов при решении уравнения Навье-Стокса связана с интегрированием уравнения Пуассона.

Существует множество методов решения уравнения Пуассона, отличающихся скоростью, точностью, применимостью, расширяемостью. Более подробно такие методы рассмотрены в [1, 2].

Одним из способов ускорения вычислений является их распараллеливание, т.е. получение их эффективных параллельных расчетных схем. Широко распространены кластерные системы на основе универсальных ЦПУ, в которых программирование параллельных программ происходит с использованием MPI, OpenMP или других систем.

В последнее время в сфере научных вычислений все большую популярность набирают вычисления с использованием графических ускорителей. Современные видеоакселераторы представляют собой массивно-параллельные процессоры с общей памятью. В отличие от центрального процессора с несколькими ядрами, один графический процессор содержит несколько сотен ядер, которые могут проводить вычисления параллельно. Учитывая разницу в стоимости одного графического ускорителя и кластерной системы, использование таких вычислителей становится экономически выгодным.

Наиболее развитой на данный момент технологией является система CUDA, предложенная компанией Nvidia в 2007 г. [6, 7]. В данной технологии вычисления производятся множеством блоков, состоящих из некоторого числа обрабатывающих потоков. В отличие от MPI, Nvidia CUDA представляет собой систему с общей памятью. Однако каждое обращение к общей памяти приводит к существенным задержкам, во время которых вычисления потоком не

проводятся. Чтобы избежать подобных задержек, каждое ядро имеет некоторый объем разделяемой памяти, которая является быстрой общей памятью для всех потоков одного блока (рис. 1).

Главной задачей при построении параллельных алгоритмов для Nvidia CUDA становится эффективное разделение расчетных данных задачи между блоками и потоками.

Отличительной особенностью CUDA является отсутствие возможности осуществить глобальную синхронизацию вычислений между блоками иным способом, кроме как закончив вычисления во всех блоках. В связи с этим каждый этап вычислительного алгоритма может использовать собственное, оптимальное для него разделение задачи между блоками.

Как было отмечено выше, в предложенном алгоритме решения уравнения Навье-Стокса присутствуют две наиболее существенные части. Так, для нахождения решения уравнения (4) для вихря использовалась схема переменных направлений [1]. Как известно, она сводится к вычислениям набора трехдиагональных систем, сначала, на первом этапе, для каждого столбца, а затем, на втором этапе, для каждой строки.

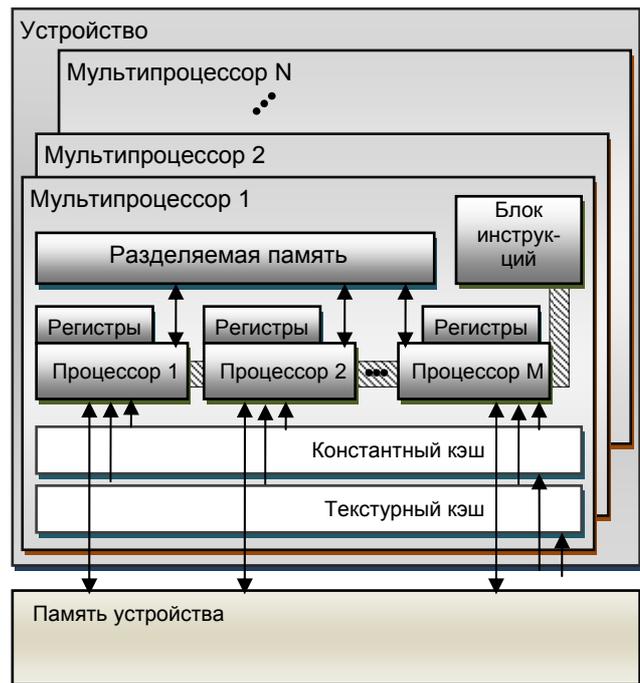


Рис. 1. Архитектура CUDA

Трехдиагональные системы, в последовательном варианте обычно решаемые с использованием алгоритма прогонки, вычисляются с помощью параллельной циклической редукции, так как данный алгоритм эффективно может быть реализован для массивно-параллельных систем с общей памятью [5]. Указанные этапы требуют подготовки данных для обработки и их преобразования после вычислений. Эти части могут быть выполнены без специального разделения на области.

В отличие от (4), для уравнения Пуассона (5) важно оптимальное разбиение задачи между блоками, т.е. необходимо использовать геометрический

параллелизм, когда каждому блоку назначается свой сектор исходных данных, а каждый поток выполняет вычисления только с одним элементом данных. При этом граничные условия становятся существенной проблемой, так как они необходимы для вычислений, но не должны вычисляться по общему алгоритму. Следовательно, требуется, чтобы каждый расчетный блок использовал разделяемую память с учетом граничных условий и обновлял их на каждой итерации.

Для итерационных методов обычно на каждой итерации требуется проверка сходимости и синхронизация, поэтому удобно вычислять на ускорителе по одной итерации. Это позволяет решить для каждого блока проблему обновления граничных условий, а также использовать алгоритм параллельной редукции для проверки сходимости на всем массиве расчетных данных.

В итоге, чтобы вычисления проводились наиболее оптимально, каждый блок содержит 256 потоков в виде сетки 16x16, что, по используемым ресурсам ускорителя, позволяет выполнять по 3 блока на каждом ядре одновременно. Однако это накладывает ограничения на входные данные. Количество внутренних точек в общей расчетной области должно быть не менее 16 и быть степенью двойки.

Более подробные результаты испытаний различных вариантов решения уравнения Пуассона с использованием CUDA приведены в [3].

При решении уравнения Навье-Стокса не является принципиальным, какой метод решения уравнения Пуассона используется. Главное, чтобы требовались к размерам расчетной области согласовывались между отдельными этапами общего алгоритма решения.

На рис. 2 представлено изображение расчетной программы, где показано движение вязкой несжимаемой жидкости в виде векторов скорости движения в некоторый момент времени.

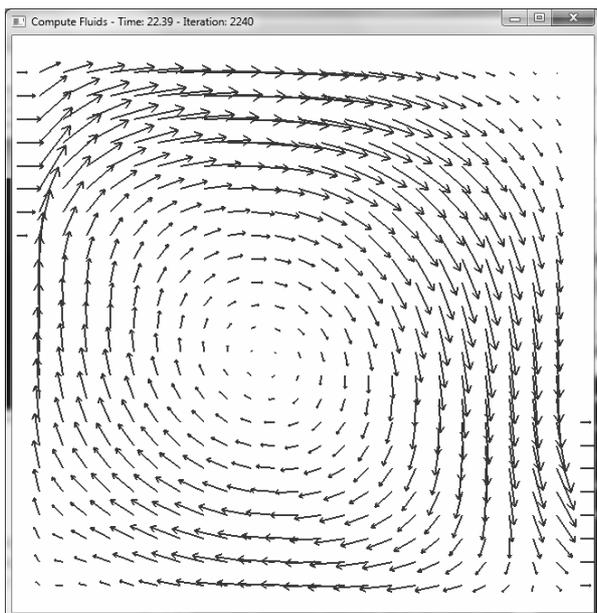


Рис. 2. Движение вязкой несжимаемой жидкости в некоторой области с входным и выходным отверстиями

Использование многопроцессорной вычислительной машины с несколькими графическими ускорителями позволяет достаточно эффективно распределять расчетные задания. В таком варианте для оптимизации расчетов становится возможным проводить некоторые легкие подготовительные вычисления на центральных процессорах с использованием MPI или OpenMP с дальнейшей передачей данных на ускорители для осуществления более сложных и продолжительных расчетов.

Заключение

Предложенный вариант решения уравнения Навье-Стокса в переменных «функция тока – вихрь» с использованием системы CUDA позволяет увеличить скорость счета и сократить затраты на решение. Показана возможность использования графических ускорителей при вычислении задач гидродинамики. Общее ускорение решения уравнения Навье-Стокса зависит от оптимальности подпрограмм вычисления трехдиагональных систем и уравнения Пуассона [3, 5].

Список литературы

1. **Алгоритмы** и программы для многопроцессорных суперкомпьютеров / В.В. Пекунов, С.Г. Сидоров, Л.П. Чернышева и др.; Иван. гос. энерг. ун-т. – Иваново, 2007.
2. **Евсеев А.В.** Методы решения уравнения Пуассона. – Иваново: ИГТА, 2009.
3. **Евсеев А.В.** Вопросы распараллеливания уравнения Пуассона и сравнение эффективности различных вариантов // Высокие технологии, исследования, промышленность. Т. 3: Сб. тр. IX Междунар. науч.-практич. конф. «Исследование, разработка и применение высоких технологий в промышленности». – СПб.: Изд-во Политехн. ун-та, 2010.
4. **Самарский А.А.** Введение в численные методы. – М.: Наука, 1982.
5. **Zhang Y., Cohen J., Owens J.D.** Fast tridiagonal solvers on the GPU // Principles and Practice of Parallel Programming, 2010. – P.127–136.
6. <http://cuda.cs.msu.su/> – специализированный курс «Архитектура и программирование массивно-параллельных вычислительных систем» на основе технологии CUDA в МГУ им. Ломоносова.
7. <http://developer.nvidia.com/object/gpucomputing.html> – NVIDIA GPU Computing Developer.

Ясинский Федор Николаевич,
ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
доктор физико-математических наук, профессор кафедры высокопроизводительных вычислительных систем,
телефон (4932) 26-98-29.

Евсеев Александр Владимирович,
ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
аспирант кафедры высокопроизводительных вычислительных систем,
телефон (4932) 26-98-29.