



Д.Г. Воротникова, Д.Л. Головашкин

ВЕКТОРНЫЙ АЛГОРИТМ РАЗНОСТНОГО РЕШЕНИЯ ВОЛНОВОГО УРАВНЕНИЯ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

(Институт систем обработки изображений РАН, Самарский национальный исследовательский университет имени академика С.П. Королёва)

Введение

Разностное решение уравнений Максвелла (FDTD-метод [1]) на протяжении последних 15 лет является незаменимым инструментом математического моделирования в быстроразвивающихся областях естествознания, изучающих генерацию, распространение и прием электромагнитного излучения: нанопластики [2], наноплазмоники [3], теории фотонных кристаллов [4] и др. Характеризуясь простотой программной реализации, общностью математической модели, естественностью подхода к ее решению, FDTD-метод вместе с тем известен высокой вычислительной сложностью, препятствующей его широкому распространению за пределами сообщества пользователей суперкомпьютерных вычислительных систем.

Решение указанной проблемы традиционно ищут в упрощении математической модели [1,5], либо в переходе к вычислениям на распространенных графических процессорах [1,6] (GPU). Настоящая работа посвящена комбинированному подходу, при котором разностное решение уравнение Даламбера реализуется на GPU серии GeForce компании NVidia.

1. Векторный алгоритм разностного решения волнового уравнения

Известны условия [7], при которых система уравнений Максвелла может быть сведена к одному скалярному уравнению Даламбера относительно выбранной проекции электрического либо магнитного полей. При их выполнении математическая модель для двумерного случая записывается в виде эллиптического уравнения

$$\frac{\partial^2 U}{\partial t^2} = c^2 \left(\frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} \right),$$
 где $U(t,x,y)$ – упомянутая проекция, а

c – скорость света (для простоты примем ее постоянной). Следуя [8] наложим на квадратную область вычислительного эксперимента D ($0 \leq t \leq T$, $0 \leq y \leq L$, $0 \leq z \leq L$) сеточную область D_h $\{(t_k, y_i, z_j): t_k = k\tau, k=0, 1, \dots, K=T/\tau, y_i = ih, i=0, \dots, N+1=L/h, z_j = jh, j=0, \dots, N+1\}$, определив на их границах условия Дирихле (электрическая стенка) и считая начальное условие известным. Тогда явная разностная схема для этого уравнения имеет вид [8]:

$$U_{i,j}^{k+1} = \alpha \left(U_{i,j-1}^k + U_{i-1,j}^k - 4U_{i,j}^k + U_{i+1,j}^k + U_{i,j+1}^k \right) + 2U_{i,j}^k - U_{i,j}^{k-1}, \alpha = \frac{c^2 \tau^2}{h^2}.$$



Составляя векторный алгоритм для реализации вычислений по ней на GPU, обратимся к приему из [9] (иллюстрированному там на примере уравнения теплопроводности) и основанному на построчном развертывании сеточной функции в вектор $U^k = (U_{0,0}^k, U_{0,1}^k, \dots, U_{0,N+1}^k, U_{1,0}^k, \dots, U_{N+1,N}^k, U_{N+1,N+1}^k)$ и повторном использовании попарных сумм дифференциального шаблона. Тогда искомым векторным алгоритм в нотации Джина Голуба [10] может быть представлен в следующем виде:

% Заполнение вектора T попарными суммами:

$$1. T(2: N(N-1)) = U^k(2: N(N-1)) + U^k(N+1: N \times N-1);$$

% Подсчёт значений для следующего временного слоя U^{k+1} :

$$2. U^{k+1}(N+2: N(N-1)-1) = \alpha(T(2: N(N-1)-1) + T(N+3: N(N-2)+N+1)) +$$

$$+ \alpha_1 U^k((N+2: N(N-1)-1) + U^{k-1}(N+2: N(N-1)-1)), \text{ где } \alpha_1 = 2-4\alpha;$$

3. Восстановление граничных значений.

Последнее действие необходимо в силу модификации значений сеточной функции на границе вычислительной области на втором шаге алгоритма.

Характеризуясь действиями с векторами, длиной пропорциональной N^2 , приведенный алгоритм может считаться масштабируемым (загрузка ядер видеокарты происходит пропорционально общему количеству узлов сеточной области).

Программная реализация алгоритма на языке CUDA C сопровождалась отказом от работы с потоками (низкоуровневый стиль программирования) и использованием векторных процедур CublasSum (шаги 1,2) и CublasSaxpy (шаг 2) библиотеки CUBLAS, что позволяет говорить о переносимости кода на разные системные платформы (в отличие от конкурирующего программного продукта [6]).

2. Вычислительные эксперименты

В распоряжении авторов находился компьютер, оснащенный видеокартой NVIDIA GeForce GTX 660Ti и процессором Intel(R) Core(TM) i7-3770, работающий под управлением операционной системы Debian 7 и CUDA Toolkit 5.5.

Дискретизация сеточной области по времени задавалась 250 слоями, по пространству варьировалась от 10 до 1000 узлов с шагом 10. Выбранные параметры обеспечивали хорошую воспроизводимость результатов экспериментов (самый короткий длился несколько секунд), оберегая их от заметного воздействия различных системных событий.

Сравнению подлежали результаты, полученные с помощью пакета WCALM Калифорнийского технологического университета [6], авторского алгоритма на основе канонического представления разностной схемы Йи для уравнений Максвелла [9] и алгоритма из настоящей работы. При этом вычислительная сложность (количество арифметических операций со скалярными величинами) во всех трех алгоритмах примерно (отличия не превышают 10%) одинакова, что позволяет говорить о проведенном (рис. 1) сравнении, как о корректном.

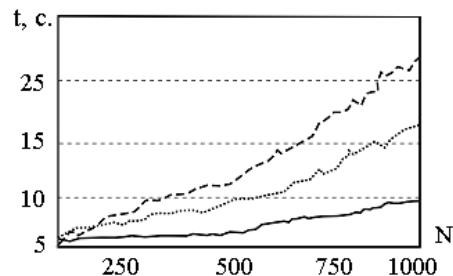


Рис. 1 Длительность вычислений по пакету B-CALM (пунктир), алгоритму на основе канонической формы для уравнений Максвелла (точки) и предложенному алгоритму (непрерывная кривая).

Отмечая более чем трехкратное превосходство предложенного алгоритма по сравнению с B-CALM и более чем двукратное по сравнению с предыдущей авторской разработкой, объясним эти отличия во-первых переходом к векторному представлению операций. Разработчики B-CALM работают с потоками, что, видимо, менее соответствует конвейерной архитектуре GPU. Во-вторых, повторным использованием сдвоенных сумм и переходом к другому типу векторных операций (от покомпонентного произведения векторов в [9] к *saхru*). Как известно [10], одна векторная операция типа *saхru* содержит больше скалярных операций, чем покомпонентное произведение.

Благодарности

Публикация подготовлена в рамках работ по гранту РФФИ № 14-07-00291 А.

Заключение

Программная реализация предложенного векторного алгоритма решения сеточных уравнений явной разностной схемы для дифференциального уравнения Даламбера позволила ускорить вычисления на видеопроцессоре по сравнению с известными авторам реализациями за счет перехода к использованию векторных нотации алгоритма и процедур библиотеки CUBLAS.

Литература

1. Taflove, A. Computational Electrodynamics: The Finite-Difference Time-Domain Method / A. Taflove, S. Hagness. ed. 3-d. – Boston: Artech House Publishers, 2005. – 852 p.
2. Дифракционная нанофотоника / под редакцией В.А. Сойфера. - М.: Физматлит, 2011. — 680 с.
3. Климов, В.В. Наноплазмоника/ В.В. Климов. - М.: Физматлит, 2010. — 480 с.
4. Lourtioz, J.-M. Photonic Crystals. Towards Nanoscale Photonic Devices/ J.-M. Lourtioz, H. Benisty, V. Berger, J.-M. Gerard, D. Maestre, A. Tchebnokov. - Springer, 2008 514 p.



5. Козлова, Е.С. Моделирование распространения короткого двумерного импульса света / Е.С. Козлова, В.В. Котляр // Компьютерная оптика. – 2012. – Т. 36, No 2. – С. 158–164
6. Wahl, P. B-CALM: An open-source GPU-based 3D-FDTD with multi-pole dispersion for plasmonics / P. Wahl, D.-S. LyGagnon, Ch. Debaes, D.A.B. Miller, H. Thienpont // Optical and Quantum Electronics. – 2012. – Vol. 44. – P. 285-290.
7. Неганов, В.А. Линейная макроскопическая электродинамика. Том 1 / В.А.Неганов, С.Б.Раевский, Г.П.Яровой. - М: Радио и Связь, 2000. – 512 с.
8. Самарский, А.А. Теория разностных схем/ А.А. Самарский. - М.: Наука, 1977. — 656 с.
9. Воротникова, Д.Г. Алгоритмы с «длинными» векторами решения сеточных уравнений явных разностных схем / Д.Г. Воротникова, Д.Л. Головашкин // Компьютерная оптика. – 2015. – Т. 39, № 1. – С. 87-93.
10. Golub, G.H. Matrix Computations / G.H. Golub, Ch.F. Van Loan. – Baltimore: Johns Hopkins University Press, 1989. – 747 p.

С.В. Востокин

ПРИМЕНЕНИЕ ПРЕДМЕТНЫХ ЯЗЫКОВ ДЛЯ АВТОМАТИЗАЦИИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

Предметные языки (domain-specific language, DSL) широко применяются для автоматизации научных расчетов. Наиболее известные из них: MATLAB (векторные вычисления); Wolfram Language (комбинация символьной и численной математики); Maple (символьные вычисления); Mathcad (интерактивные вычисления, визуализация). Преимущество данных языков в предметной области численного моделирования состоит в том, что их синтаксис и семантика адаптированы к традиционной математической нотации, что существенно снижает трудоёмкость программирования.

Однако, если требуется оптимизация кода, что характерно для области высокопроизводительных научных вычислений, предметные языки уступают языкам общего назначения высокого уровня C/C++. Поэтому критичные к производительности части кода перечисленных выше предметных языков реализуются именно на языках C/C++.

Для ускоренной разработки программного обеспечения были предложены методики, позволяющие сочетать предметные языки и языки общего назначения в одном проекте [1,2]. В данной работе рассматривается применение этого подхода для автоматизации высокопроизводительных научных вычислений на базе облачного сервиса Templet Web [3].