Блочный алгоритм совместного разностного решения уравнений Даламбера и Максвелла

Л.В. Яблокова[1], Д.Л. Головашкин[1,2]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086
[2]Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН, Молодогвардейская 151, Самара, Россия, 443001

Аннотация. Характерной особенностью математического моделирования на современном этапе развития, по мнению авторов настоящего доклада, является учет архитектуры вычислительной системы не только на этапе составления программы для ЭВМ, но и в ходе разработки численного метода и синтеза математической модели. Указанный прием существенно расширит возможности исследователя по поиску оптимального (в смысле ускорения вычислений) отображения численного метода на упомянутую архитектуру. В предлагаемой работе данная идея иллюстрируется на примерах основной математической модели вычислительных электродинамики и оптики, уравнениях Максвелла, и FDTD-метода. Недавно были сформулированы блочные алгоритмы для разностного решения дифференциальных уравнений, в том числе уравнений Максвелла (FDTD-метод). Их успех вдохновил авторов доклада на следующий шаг — модификацию самой модели, ориентированную на сокращение упомянутых коммуникаций. Данная модификация основана на совместном решении уравнений Даламбера и Максвелла, которое позволяет с одной стороны в несколько раз сократить интенсивность обмена данных между оперативной и кэш-памятью за счет большего количества арифметических операций, приходящихся на одну сеточную функцию при решении уравнения Даламбера, с другой – свободно пользоваться детально разработанными за долгую историю FDTD-метода технологиями и готовыми программными реализациями задания падающей волны, наложения поглощающих слоев, учета дисперсии среды и др. за счет включения фрагментов сеточной области Yee в основную сеточную область.

1. Введение

Несмотря на давнее начало исследований [1] в области численного решения уравнений Максвелла, интерес к этой предметной области со временем только возрастает. Данное замечание в первую очередь относится к применению метода конечных разностей [2] — наиболее популярному к настоящему времени варианту численного решения указанных уравнений. В семидесятых годах прошлого века [3] он удостоился собственной аббревиатуры FDTD (Finite-Difference Time-Domain) под которой широко известен и по сей день.

Актуальность развития FDTD-метода обусловлена несколькими причинами. Во-первых, широкой востребованностью в новых предметных областях: нанофотонике [4], радиобиологии [5] и др. в силу общности математической модели. Действительно, с помощью уравнений

Максвелла описываются без ограничений любые процессы, связанные с волновой природой электромагнитного излучения. Во-вторых, необходимостью учета особенностей современных вычислительных архитектур: кластерной [6], векторной [7] и др. Если ранее производительность ЭВМ повышалась в основном за счет увеличения тактовой частоты центрального процессора, то сейчас она увеличивается использованием различных приемов параллельной обработки данных.

В предлагаемом авторами настоящей работы исследовании акцент при синтезе нового варианта FDTD-метода делается на учете иерархической структуры запоминающих устройств ЭВМ, в частности на возможности оптимизации коммуникаций меду оперативной и кэш-памятью процессора. Заданная тематика скупо освещается в научной печати в силу междисциплинарного характера задачи. Специалистов по вычислительной математике традиционно интересуют другие проблемы теории разностных схем: повышение порядка аппроксимации дифференциальной задачи [8], конструирование подвижных сеточных областей [9]. В свою очередь разработчики математического программного обеспечения, реализующего FDTD-метод, стремятся не отстать от развития «крупных» форм организации современных параллельных и распределенных вычислений: с использованием видеопроцессоров [10], на облаках [11].

Тем не менее, появляются свежие работы по блочным алгоритмам разностного решения основных уравнений электродинамики [12-14], свидетельствующие о внимании и к «малым» формам организации вычислений, учитывающим неприметные архитектурные особенности процессоров и демонстрирующие при этом значительное ускорение за счет такого учета. В вычислительной практике достаточно давно на примере матричных вычислений отмечена возможность управлять длительностью расчетов за счет изменения размеров блока [15,16] в блочных алгоритмах. К сожалению, особенности теории разностных схем (отсутствие необходимости в умножении плотных матриц, традиционный акцент на проблемах устойчивости) до последнего времени затрудняли проникновение блочности в эту предметную область.

Другим препятствием к развитию блочных методов является сложность управления загрузкой/выгрузкой данных в кэш память процессора в ходе вычислительного процесса. Операторы загрузки (но не выгрузки) присутствуют только на языке ассемблера и их использование не носит директивного характера: данные могут не загружаться в кэш после их исполнения, если блок уже там. В более распространенных для рассматриваемой предметной области языках программирования управление коммуникациями между оперативной и кэш-памятью достигается косвенным путем за счет приема разбиения циклических конструкций, упоминаемого в иностранной литературе как tiling [17,18].

Применению этого приема для совместного разностного решения уравнений Даламбера и Максвелла и посвящена предлагаемая публикация.

2. Особенности совместного разностного решения уравнений Даламбера и Максвелла

В работах [19,20] детально описывается теория совместного разностного решения уравнений Даламбера и Максвелла, здесь же перечисляются результаты его экспериментального исследования, значимые для выбранной тематики. При постановке экспериментов использовался следующий программно-аппаратный инструментарий: процессор Intel Core i7-3770, операционная система Ubuntu 16.04.1 (ядро 4.4), компилятор gcc 5.3 и пакет Meep 1.3 Массачусетского технологического института (скомпилированный для корректности сравнения этим же компилятором), реализующий FDTD-метод и фактически являющийся эталонным для широкого круга исследователей [21]. Сеточная область выбиралась размерами 10000×10000 узлов по пространству и 200 по времени, что обеспечивало достаточную загрузку оперативной памяти (как для реальных вычислительных экспериментов в нанофотонике) при приемлемой длительности расчетов.

В случае моделирования распространения TM-моды (в терминах фундаментальной работы [22]) разностные решения уравнений Даламбера и Максвелла отличаются друг от друга на величину машинной точности и сходятся к аналитическому. При использовании пакета MEEP

длительность вычислений составила 124,75 сек., расчеты по авторской программной реализации разностного решения уравнений Максвелла длились 112,71 сек., для разностного решения уравнения Даламбера — 41,34 сек. Ускорение последнего в 3,02 и 2,73 раза соответственно нельзя объяснить снижением вычислительной сложности на 10% при разностном решении уравнений Даламбера по сравнению с таковым для Максвелла. Авторы связывают наблюдаемый эффект с увеличением отношения количества арифметических операций, приходящихся на вычисления по дифференциальному шаблону, к объему памяти, задействованному при работе с тем же шаблоном. Для двумерной схемы Yee указанная величина равна 3/2 (производится три операции над двумя разными проекциями поля) для первого и второго разностных уравнений и 7/4 (7 операций над 3 проекциями поля и значением диэлектрической проницаемости) для третьего; для второй схемы — 9/3 (9 операций над двумя разными временными слоями одной проекции и значением диэлектрической проницаемости), что, по-видимому, приводит к значительному снижению интенсивности коммуникаций между оперативной и кэш-памятью.

Совместное разностное решение сочетает в себе достоинства обоих подходов: ускорение для Даламбера и разработанный инструментарий (наложения поглощающих слоев, задания падающей волны и т. д.) для Максвелла. В ходе следующего эксперимента уравнения Максвелла решались в PML-слоях [22] толщиной 100 узлов по краям сеточной области, а в центре — уравнение Даламбера. Длительность расчетов в этом случае составила 47,47 сек. и незначительно превосходит предыдущий результат (при получении которого вычислений в PML слоях не проводилось), характеризуясь ускорениями в 2,63 раз и 2,37 раз по сравнению с пакетом Meep и авторской реализацией схемы Yee соответственно.

Далее авторы используют обнаруженный эффект сокращения длительности вычислений при снижении интенсивности коммуникаций между оперативной и кэш-памятью для дальнейшего ускорения вычислений в случае совместного решения.

3. Блочный алгоритм совместного разностного решения

К настоящему времени известны блочные алгоритмы как для разностного решения уравнений Максвелла [12] (2009 год), так и для Даламбера [14] (2015 год). Предложенный в последней работе прием перехода к блочным вычислениям с алмазной тороидальной формой блока (Diamond Torre Algorithm) послужил основой для синтеза авторского блочного алгоритма совместного разностного решения, который состоит из следующих двух этапов.

На первом производятся вычисления на h временных слоях сеточной области для обновления значений сеточных функций в подобласти расположения PML. На рисунке 1 это «ручка ковша». В силу информационной зависимости на пространстве итераций, совпадающем с сеточной областью, вычислению на данном этапе также подлежат значения в узлах, прилегающих на расстояние h и меньшее к PML, относящиеся уже к волновому уравнению. Так в узле, отстоящем на k других узлов от поглощающих слоев ($k \leq h$) будет сформировано значение сеточной функции на временном слое k. Все такие узлы составят «стенки ковша» на рисунке 1.

Второй этап характеризуется организацией вычислений во внутреннем объеме «ковша» по алгоритму, который авторы назвали волновым по аналогии с методикой перехода к блочности из [17]. Как показано на рисунке 2, в ходе вычислительного процесса внутри «ковша» значения сеточных функций рассчитываются по направлению слева направо таким образом, что передний фронт такого процесса имеет вид наклонной плоскости. В узлах сеточной области перед ним значения функций еще не определены, после него — найдены на слое h. На самой поверхности фронта вычисляются значения на разных слоях в порядке возрастания с увеличением высоты и уменьшением абсциссы узла. После второго этапа алгоритма опять наступает первый, и их чередование продолжается, пока значения сеточных функций на всех слоях области не окажутся найденными.

В отличие от алмазного тороидального алгоритма из [14] такой подход проще программно реализуется и приводит к незначительному, но стабильному сокращению длительности вычислений по сравнению с другой методикой перехода к блочности.

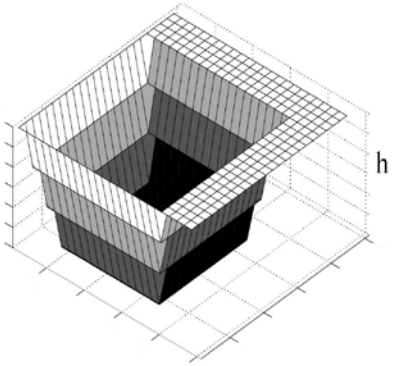

Рисунок 1. Распределение сеточных функций по временным слоям перед началом блочного этапа вычислений. От белого (слои на максимальной высоте h и рядом) к черному (слои на минимальной высоте и около нее).

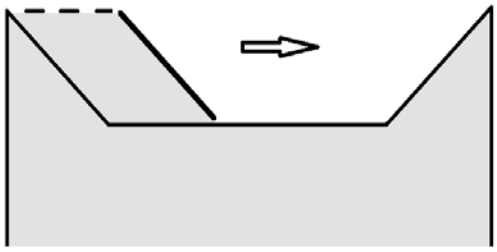

Рисунок 2. Распределение сеточных функций по временным слоям на блочном этапе вычислений.

Таблица 1. Зависимость длительности вычислений (сек.) от высоты блока h. T_1 – длительность для волнового алгоритма, T_2 – для алмазного тороидального. Величина V – объем блока в мегабайтах.

h	2	4	8	10	20	40	50	100
T_1	42,63	36,88	33,88	33,49	32,54	36,83	40,80	48,42
T_2	42,75	37,15	34,13	33,57	33,64	37,16	40,90	48,44
V	0,45	0,9	1,79	2,24	4,49	8,97	11,22	22,43

Примечательно, что лучшие результаты для обоих алгоритмов достигнуты для h=20 — максимальной высоте волны (или тора) при которой блок еще умещался в кэш-память целиком (объем кэш L3 для Intel Core i7-3770 составляет 8 Мб). Действительно, при h>20 блок уже не помещается в кэш целиком и приходится загружать его по частям, а для h<20 быстрая кэш-память используется не целиком. Оба указанных обстоятельства приводят к росту коммуникаций и, как следствие, общей длительности вычислений.

4. Заключение

Переход к блочному алгоритму позволил почти в полтора раза сократить длительность вычислений в случае совместного разностного решения уравнений Максвелла и Даламбера и довести общее ускорение по сравнению с пакетом Meep Массачусетского технологического института до 3,84 раз.

5. Благодарности

Работа выполнена при поддержке Федерального агентства научных организаций (соглашение № 007-ГЗ/Ч3363/26) и гранта РФФИ 16-47-630560-р_а.

6. Литература

[1] Cron, G. Equivalent circuit of the field equations of Maxwell / I.G. Cron // Proc. IRE. – 1944. – Vol. 32. – P. 289-299.

[2] Yee, K.S. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media / K.S. Yee // IEEE Trans. Antennas Propag. – 1966. – Vol.14. – P. 302-307.
[3] Taflove, A. Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations / A. Taflove, M. Brodwin // IEEE Transactions of microwave theory and techniques. – 1975. – Vol. 2(8). – P. 623-630.
[4] Gavrilov, A.V. Diffracrive Nanophotonics / A.V. Gavrilov, D.L. Golovashkin, L.L. Doskolovich, P.N. Dyachenko, S.N. Khonina, V.V. Kotlyar, A.A. Kovalev, A.G. Nalimov, D.V. Nesterenko, V.S. Pavelyev, Y.O. Shuyupova, R.V. Skidanov, V.A. Soifer // CRC Press, Taylor & Francis Group, CISP, Boca Raton, 2014. – 679 p.
[5] Перов, С.Ю. Теоретическая и экспериментальная дозиметрия в оценке биологического действия электромагнитных полей носимых радиостанций. Сообщение 1. Плоские фантомы / С.Ю. Перов, Е.В. Богачёва // Радиационная биология: Радиоэкология. – 2014. – Т. 54, № 1. – С. 57-61.
[6] Guiffaut, C. A parallel FDTD algorithm using the MPI library / C. Guiffaut, K. Mahdjoubi // IEEE Antennas and Propagation Magazine. – 2001. – Vol. 43(2). – P. 94-103.
[7] Vorotnikova, D.G. CUBLAS-aided Long Vector Algorithms / D.G. Vorotnikova, D.L. Golovashkin // Journal of Mathematical Modelling and Algorithms in Operations Research. – 2014. – Vol. 13(4). – P. 425-431.
[8] Xiao, F. High-order accurate split-step FDTD method for solution of Maxwell's equations / F. Xiao // Electronics Letters. – 2007. – Vol. 43(2). – P. 72.
[9] Fidel, B. Hybrid ray-FDTD moving window approach to pulsepropagation / B. Fidel, E. Heyman, R. Kastner, R.W. Zioklowski // Journal of Computational Physics. – 1997. – Vol. 138(2). – P. 480-500.
[10] FDTD solver [Электронный ресурс]. – URL: http://www.acceleware.com/fdtd-solvers.
[11] EMA3D Version 3.3 parallel modification. [Online]. Available: http://www.ema3d.com
[12] Orozco, D. Mapping the FDTD Application to Many-Core Chip Architectures / D. Orozco, G. Guang // Parallel Processing. ICPP '09. International Conference, 2009. – P. 309-316.
[13] Grosser, T. Split Tiling for GPUs: Automatic Parallelization Using Trapezoidal Tiles / T. Grosser, A. Cohen // Proc. of the GPGPU-6. – New York, USA: ACM, 2013. – P. 24-31.
[14] Perepelkina, A.Yu. Diamond Torre Algorithm for High-Performance Wave Modeling / A.Yu. Perepelkina, V.D. Levchenko // Keldysh Institute Preprints. – 2015. – Vol. 18. – P. 20.
[15] Golub, G.H. Matrix Computations / G.H. Golub, Ch.F. Van Loan. – Baltimor: Johns Hopkins University Press, 1996. – 726 p.
[16] Деммель, Дж. Вычислительная линейная алгебра. Теория и приложения / Дж. Деммель. – М.: Мир, 2001. – 435с.
[17] Wolfe, M. Loops skewing: The wavefront method revisited / M. Wolfe // International Journal of Parallel Programming. – 1986. – Vol. 15(4). – P. 279-293.
[18] Wolfe, M. More Iteration Space Tiling / M. Wolfe // Proceedings of the 1989 ACM/IEEE Conference on Supercomputing. Supercomputing '89. – New York, USA: ACM, 1989. – P. 655-664.
[19] Головашкин, Д.Л. Совместное разностное решение уравнений Даламбера и Максвелла. Одномерный случай / Д.Л. Головашкин, Л.В. Яблокова // Компьтерная оптика. – 2012. – Т. 36, №4. – С. 526-532.
[20] Булдыгин, Е.Ю. Совместное разностное решение уравнений Даламбера и Максвелла. Двумерный случай / Е.Ю. Булдыгин, Д.Л. Головашкин, Л.В. Яблокова // Компьютерная оптика. – 2014. – Т. 38, №1. – С. 20-27.
[21] Oskooi, A.F. Meep: A flexible free-software package for electromagnetic simulations by the FDTD method / A.F. Oskooi, D. Roundyb, M. Ibanescua // Computer Physics Communications. – 2010. – Vol. 181. – P. 687-702.
[22] Taflove, A. Computational Electrodynamics: The Finite-Difference Time-Domain Method / A. Taflove, S. Hagness. – Boston: Arthech House Publishers (Thrid Edition), 2005. – 1006 p.

Block algorithm for the joint difference solution of the d'Alembert and Maxwell equations

L.V. Yablokova[1], D.L. Golovashkin[1,2]

[1]Samara National Research University, Moscow Shosse 34A, Samara, Russia, 443086
[2]Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

Abstract. A characteristic feature of mathematical modelling at the present stage of development of this scientific branch, according to the authors of this report, is the consideration of the architecture of the computer system not only at the stage of compiling a computer program, but also in the development of a numerical method and the synthesis of a mathematical model. This method significantly broadens the researcher's ability to search for the optimal (in the sense of accelerating computations) mapping of the numerical method to the mentioned architecture. In this paper, this idea is illustrated by examples of the basic mathematical model of computational electrodynamics and optics, Maxwell's equations, and the FDTD. This modification is based on the joint solution of the d'Alembert and Maxwell equations, which allows one to several times reduce the data exchange rate between the operational and cache memory due to the greater number of arithmetic operations per one grid function in solving the d'Alembert equation, on the other hand, freely use the technologies developed in the long history of the FDTD method and ready-made software implementations for setting the incident wave, imposing the absorbing layers, taking into account the dispersion of the medium and others.

Keywords: finite-difference time-domain method, block algorithms, acceleration of calculations.