

# ЧИСЛЕННЫЕ МЕТОДЫ И АНАЛИЗ ДАННЫХ

## БЛОЧНЫЕ АЛГОРИТМЫ СОВМЕСТНОГО РАЗНОСТНОГО РЕШЕНИЯ УРАВНЕНИЙ ДАЛАМБЕРА И МАКСВЕЛЛА

Л.В. Яблокова<sup>1</sup>, Д.Л. Головашкин<sup>1,2</sup>

<sup>1</sup> Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия,

<sup>2</sup> Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН, Самара, Россия

### Аннотация

Работа посвящена синтезу блочных алгоритмов FDTD-метода, в частности совместного разностного решения уравнений Даламбера и Максвелла. Учёт иерархической структуры памяти ЭВМ позволил до 6 раз сократить длительность вычислений по методу в сравнении с его известными программными реализациями.

**Ключевые слова:** FDTD-метод, блочные алгоритмы, ускорение вычислений.

**Цитирование:** Яблокова, Л.В. Блочные алгоритмы совместного разностного решения уравнений Даламбера и Максвелла / Л.В. Яблокова, Д.Л. Головашкин // Компьютерная оптика. – 2018. – Т. 42, № 2. – С. 320-327. – DOI: 10.18287/2412-6179-2018-42-2-320-327.

### Введение

Задача отображения численных методов на архитектуру вычислительных систем, впервые поставленная в 70-е годы директором Вычислительного центра СО АН СССР Гурием Ивановичем Марчуком [1], к настоящему моменту становится одной из центральных проблем вычислительной математики в связи с достижением «кремниевого тупика». Ещё 15 лет назад широкий круг специалистов связывал развитие численных методов в первую очередь с исследованием таких классических проблем, как снижение вычислительной сложности, повышение порядков аппроксимации, смягчение критериев устойчивости и т.п. Действительно, зачем математику интересоваться архитектурой вычислительного комплекса, если под алгоритмом можно понимать последовательность действий и этим ограничиться. Последнее обстоятельство обусловлено длительным периодом повышения производительности ЭВМ за счёт роста тактовой частоты центральных процессоров, пришедшимся на вторую половину прошлого века. К настоящему времени ситуация изменилась. Ведущие разработчики вычислительной техники наращивают производительность процессоров усложнением архитектурных решений (многоядерностью, многопоточностью, SIMD-расширениями), обеспечивающих параллелизацию, конвейеризацию и векторизацию вычислений [2]. В свою очередь, это объясняет интерес к синтезу соответствующих (параллельных, конвейерных, векторных) алгоритмов одного численного метода, учитывающих различные архитектурные особенности.

Изложенное хорошо иллюстрируется на примере FDTD-метода (*Finite-Difference Time-Domain*), отличие которого состоит в неизменном росте актуальности на протяжении трёх четвертей века [3], обусловленном в значительной мере развитием вычислительной техники. Так, известны его многочисленные параллельные [4], векторные [5, 6] и конвейерные [7] алгоритмы. Особый интерес представляет последняя работа, где используется устройство FPGA (*Field-Programmable Gate Array*), которое по сути является аппаратной реализацией алгоритма.

Вместе с упомянутыми новыми архитектурными решениями продолжают развиваться и давно известные, среди которых авторы настоящей работы желают выделить кэш процессора. Алгоритмический приём, связанный с учётом этого вида памяти ЭВМ и направленный на оптимизацию его использования, именуется блочностью [8, 9], если работа посвящена более математике, чем программированию, или «tiling'ом» [10] в обратном случае. При программной реализации «tiling» сводится к изменению порядка итераций циклической конструкции (с условием неизменности результатов вычислений) с целью минимизации коммуникаций между различными областями памяти ЭВМ. Долгое время он пользовался популярностью по большей части при организации матричных вычислений [8, 9], однако не так давно появились блочные алгоритмы решения явных сеточных уравнений [11] и, в частности, FDTD-метода [12]. Предлагаемое исследование является переосмыслением и развитием идеи совместного разностного решения уравнений Даламбера и Максвелла [13, 14] с использованием блочности.

### 1. Экспериментальное исследование совместного разностного решения уравнений Даламбера и Максвелла

Традиционно принято различать натурный и вычислительный эксперименты по обращению с предметом исследования. В первом случае манипуляции производятся непосредственно с самим объектом, во втором – с его математической моделью. Исследование вычислительного процесса при этом может быть классифицировано двояко. С одной стороны, он связан с решением уравнений математической модели, с другой – является физическим явлением, характеризующимся по крайней мере одной непосредственно измеряемой физической величиной (временем). Положим, что если важным результатом вычислительного процесса являются значения *рассчитываемых* величин, то следует говорить о вычислительном эксперименте; если значения *измеряемых* величин – об

эксперименте натурном. То есть один и тот же эксперимент может интерпретироваться как вычислительный и натурный одновременно. Далее речь пойдет о второй интерпретации. Натурному исследованию будут подлежать вычислительные процессы, определяющиеся (в смысле А.А. Маркова [15]) некоторыми алгоритмами, а следовательно (пусть и опосредованно), эти алгоритмы.

Смысл совместного разностного решения уравнений Даламбера и Максвелла [13, 14] заключается в использовании выгодных особенностей обеих моделей в одном вычислительном процессе. Разностное решение уравнения Даламбера, в отличие от такового для Максвелла, сопровождается меньшими затратами памяти ЭВМ (вместо хранения значений четырех сеточных функций достаточно трёх) и снижением количества арифметических операций (9 вместо 11 матричных операций при переходе к следующему временному слою) [16, 17]. В свою очередь, решение на той же сеточной области уравнений Максвелла позволяет использовать многочисленные готовые технологии (TF/RF, наложения PML-слоёв и др.), не тратя время на разработку соответствующих фрагментов программного комплекса [18], что важно в инженерной практике. Недостатками такого подхода являются: ограничение области применения случаем ТМ-моды (в терминах фундаментальной работы [19] по FDTD-методу) и незначительность ожидаемого выигрыша от экономии системных ресурсов (памяти и времени вычислений). Однако недавние результаты по оптимизации работы с многоуровневой памятью ЭВМ [11, 12] побудили авторов настоящей статьи к проведению экспериментального исследования совместного разностного решения уравнений Даламбера и Максвелла.

С разностными схемами и особенностями построения сеточной области при совместном разностном решении читатель может познакомиться в работах [13, 14], здесь же внимание фокусируется на параметрах экспериментов. В качестве аппаратного обеспечения выбраны процессоры линейки Intel Core: i7-3770 3,4 ГГц (материнская плата Asus P8Z77 WS с частотой системной шины DMI 5000 МГц; объём ОЗУ 32 Гб, тип памяти DIMM DDR3-1333 МГц), i5-4200M 3,1 ГГц (ноутбук, материнская плата FUJITSU FJNB36 с частотой системной шины DMI 3000 МГц; объём ОЗУ 4 Гб, тип памяти DIMM DDR3-1600 МГц) и i3-4150 3,5 ГГц (материнская плата Gigabyte GA-Z97M-DS3H с частотой системной шины DMI 5000 МГц; объём ОЗУ 12 Гб, тип памяти DIMM DDR3-1333 МГц) – как мощные и доступные инструменты специалиста-вычислителя. Использование двух современных операционных систем: CentOS 7.2 и Ubuntu 16.04, являющихся бесплатными версиями основных дистрибутивов Linux (Red Hat и Debian), позволит выявить зависимость результатов от вида системного и программного обеспечения. С той же целью выбраны три различных компилятора: свободный распространяемый GCC 5.3, коммерческие PG

Fortran 2016 и Intel Fortran 2017. В силу широкой популярности программных кодов FDTD-метода на языке MatLab [18, 19] программирование велось с использованием стандарта Fortran 90, синтаксические правила которого наиболее близки к MatLab'у.

Для сравнения привлекалось несколько программных комплексов (табл. 1). Первый из них – это пакет MEEP 1.3 (скомпилированный с помощью GCC) разработки Массачусетского технологического института, реализующий FDTD-метод и фактически являющийся эталонным для широкого круга исследователей в оптике и электродинамике [20]. В табл. 1 представлен в двух вариантах: MEEP – без поглощающих слоёв в вычислительной области и MEEP+PML – с ними. Вторым программным комплексом для сравнения были выбраны переведённые с языка MatLab на Fortran практически без изменений программы для реализации FDTD-метода Susan C. Hagness из диска-приложения к [19]. В табл. 1: FDTD – без поглощающих слоёв, FDTD+PML – с ними. Также рассматривалась переведённая с MatLab'a на Fortran с небольшими правками, учитывающими специфику предметной области, программа для разностного решения уравнения Даламбера из [21], в табл. 1 – колонка DLM. В ней не предполагается учёт поглощающих слоёв, в электродинамике они ассоциируются с решением уравнений Максвелла [19]. Четвёртым комплексом является авторская программа совместного разностного решения уравнений Даламбера и Максвелла [22], в которой уравнения Максвелла решаются в области задания падающей волны и наложения PML-слоёв по коду из [19], в области оптического элемента решается уравнение Даламбера по коду из [21]. Основной особенностью программы [22] является согласование обоих решений. В табл. 1 ей соответствует колонка JS (*Joint Solution*).

Дискретизация сеточной области по пространству составляла  $10000 \times 10000$  узлов – максимальное значение, удовлетворяющее ограничению на объём оперативной памяти (4 Гб) с учётом размещения в ней операционной системы. Приемлемую длительность расчётов обеспечивала дискретизация сеточной области в 200 временных слоёв, очевидно, недостаточная для постановки *вычислительного* эксперимента – за это модельное время волновой фронт в силу условия устойчивости Куранта [19] не успеет достичь всех границ вычислительной области независимо от расположения источника излучения, однако вполне подходящая для эксперимента *натурного*, как обеспечивающая стабильность результатов (независимость их от случайных системных событий) при сравнении длительности расчётов по различным алгоритмам. Полагалось, что за указанное модельное время в 200 слоёв плоский однородный волновой фронт (заданный по технологии TF/RF) преодолет в вакууме расстояние, связанное со 100 узлами области по пространству – соотношение, принятое в пакете MEEP по умолчанию. В случае наложения на границы вычислительной области PML-слоёв, последние объединялись по методике из [23], и их совокупная толщина по одному пространственному направ-

лению составляла 40 узлов. В табл. 1 представлены результаты экспериментов с программными реализациями различных алгоритмов для заданной сеточной области.

Отметим, что смена компилятора и операционной системы практически не влияет на результаты экспери-

ментов, представленных в табл. 1, что позволяет уверенно сравнивать программные реализации, фактически являющиеся алгоритмами, по вычислительным процессам, ими определённым. Также в силу этого далее обсуждаются результаты только для Ubuntu и GCC.

Табл. 1. Зависимость длительности вычислений (в секундах) по алгоритмам от типа процессора, системного и программного обеспечения

Процессор	Операционная система и компилятор	Алгоритмы и пакеты					
		МЕЕР	FDTD	DLM	МЕЕР+PML	FDTD+PML	JS
i7-3770	Ubuntu и GCC	121,5	105,7	41,31	122,68	109,16	42,43
	Cent OS и PGI Fortran		106,95	41,52		108,53	43,18
i5-4200M	Ubuntu и GCC	204,54	172,47	63,13	206,40	173,60	66,09
	Ubuntu и Intel Fortran		172,42	62,81		173,56	65,96
i3-4150	Ubuntu и GCC	183,96	154,98	56,98	186,2	157,99	59,79

Первые три серии экспериментов (столбцы 3–5 табл. 1) характеризуются отсутствием поглощающих слоёв и совместного решения. Это допускает сравнение наиболее простых реализаций, когда на результат влияет наименьшее количество факторов. Длительность вычислений по пакету МЕЕР на 13–16 % превышает таковую для реализации FDTD-метода из [19], что, по-видимому, объясняется простотой кода последней. Очевидно, более широкие возможности, включённые в пакет, даже не будучи задействованными, несколько замедляют вычисления. Например, учёт дисперсии среды проводится добавлением слагаемого к разностному выражению. Возможно, в МЕЕР такое суммирование производится независимо от того, задана дисперсия или нет. В последнем случае слагаемое равно нулю, что, тем не менее, увеличивает длительность расчётов.

Определённое недоумение вызывают результаты для разностного решения уравнения Даламбера, длительность вычислений по которому в 2,56–2,73 раз меньше, чем по реализации FDTD-метода из [19]. Такая разница не может объясняться умеренным (20 %) отличием в вычислительной сложности между методами. Следует ли из этого, что критерий сравнения разностных схем по количеству арифметических операций недостаточен и необходимы поиски более точного критерия? Во второй части предлагаемой работы авторы вернутся к этому вопросу.

Наложение на границы сеточной области PML-слоёв, в большинстве случаев обязательное при проведении вычислительных экспериментов в оптике и электродинамике, приводит к незначительному увеличению длительности расчётов (столбцы 6–8 табл. 1) в силу малой площади этих слоёв – менее процента от всей вычислительной области. Переход к совместному разностному решению позволяет ускорить вычисления при выбранных параметрах в 2,89–3,12 и 2,57–2,64 раза по сравнению с пакетом МЕЕР и реализацией FDTD-метода из [19] соответственно при той же точности.

Изменение дискретизации сеточной области, необходимое для постановки вычислительного эксперимента, связано с увеличением количества временных слоёв (при том же шаге по времени) в 200 раз. За это модельное время плоский волновой фронт дойдёт до противоположной границы вычислительной об-

ласти и вернётся обратно, что позволит изучить отражённую от оптического элемента волну. Длительности вычислительных процессов увеличатся во столько же при неизменных ускорениях. Тогда переход от использования пакета МЕЕР к совместному решению, например, на процессоре i3-4150, позволит ожидать результатов расчётов 3,32 часа вместо 10,34.

Сравнивая зависимость представленных в табл. 1 результатов от марки процессора, отметим совпадение ускорений вычислений при переходе от пакета МЕЕР к совместному решению (3,12 и 3,11) и от реализации FDTD из [19] к совместному (2,63 и 2,64) для последних двух процессоров и отличие аналогичных ускорений (2,89 при первом переходе и 2,57 при втором) для первого процессора. Забегая вперед, свяжем указанные совпадения и отличие с одинаковым объёмом кэш-памяти L3 в 3 Мб у i3-4150, i5-4200M и разным по сравнению с i7-3770, у которого объём L3 равен 8 Мб. Также остановим внимание на падении ускорения с увеличением ёмкости L3.

## 2. Учёт многоуровневой структуры памяти ЭВМ при совместном разностном решении уравнений Даламбера и Максвелла

Несоответствие теоретических ожиданий и экспериментальных результатов, выявленное в прошлом параграфе, побудило к дальнейшему анализу FDTD-метода и совместного разностного решения. В классической модели теории параллельного программирования «канал/задача» [24], задача определяется как последовательная программа и данные, с которыми она работает. Возможно, Ян Фостер имел в виду даже холизм программы и данных. Не стремясь к коренной модификации теории разностных схем [25], авторы настоящей работы отмечают, что было бы уместным учитывать при разработке схем особенности хранения значений сеточных функций. А именно, многоуровневую структуру оперативной памяти ЭВМ, как это давно принято при синтезе алгоритмов вычислительной линейной алгебры [26].

Действительно, наличие такой многоуровневости обуславливает необходимость в производстве коммуникаций между различными областями памяти; операций, которые могут оказать определяющее влияние на длительность вычислений. В рассматриваемом

примере FDTD-метода для двумерной сеточной области классический алгоритм Yee [19] подразумевает последовательное решение трёх явных сеточных уравнений, с участием двух и четырёх сеточных функций в каждом соответственно. То есть при переходе на следующий временной слой возникает необходимость в организации трёх серий коммуникаций между оперативной памятью и кэш-памятью процессора. Причём количество посылок/приёмов данных в последней серии вдвое выше, чем в двух первых. Если принять за  $m$  количество пересылок значений двух сеточных функций при решении первого уравнения, то для кэша того же объёма получим всего  $m + m + 2m = 4m$  пересылок.

Разностное решение уравнения Даламбера сопровождается при переходе на следующий временной слой одной серией коммуникаций с участием трёх сеточных функций – всего  $3/2m$  пересылок. Таким образом, общее количество пересылок при решении уравнений Максвелла в 2,67 раза больше, чем для Даламбера. Данное значение хорошо коррелирует с разницей в 2,56–2,73, полученной в предыдущем параграфе экспериментально при сравнении обоих алгоритмов. Наложение PML-слоёв на границы вычислительной области приводит к увеличению доли арифметических операций в общей длительности вычислений, что обуславливает небольшое падение ускорения (уже смешанного решения по сравнению с классическим FDTD) до 2,57–2,64 раз. При этом следует помнить, что значительная часть арифметических и коммуникационных операций производится одновременно и точно оценить их долю в общем времени расчётов без обращения к низкоуровневым программным средствам невозможно. В силу чего при обсуждении данного аспекта вычислений проводится лишь качественный анализ.

Эта же гипотеза об определяющем влиянии многоуровневой структуры памяти на длительность вычислений хорошо объясняет совпадение ускорений для разных процессоров с одинаковым объёмом кэш-памяти и несовпадение для процессоров с разным. Ведь только упомянутый объём и определяет количество коммуникаций при равной дискретизации сеточной области. С помощью той же гипотезы можно объяснить падение ускорения с увеличением объёма кэша. Отмеченное увеличение влечёт за собой снижение интенсивности коммуникаций, следовательно, и уменьшение вклада длительности коммуникаций в общее время вычислений. В итоге, переход от решения уравнений Максвелла к уравнению Даламбера приводит к более скромному выигрышу при росте объёма кэш-памяти процессора в силу незначительной разницы в вычислительной сложности между обоими алгоритмами. Оптимизируется величина, влияние которой на итоговый результат уменьшилось.

Естественно желание исследователя научиться управлять обнаруженным эффектом. Нельзя ли ввести в алгоритм решения сеточного уравнения параметр, изменение которого позволит варьировать количество коммуникаций между различными областями памяти ЭВМ, не меняя при этом математическую

модель? Ведь переход от разностного решения уравнений Максвелла к совместному решению [13, 14] является именно такой сменой.

В матричных вычислениях для этого синтезируют блочные алгоритмы. Говоря о сути перехода к блочности, Джин Голуб [8] акцентирует внимание на такой трансформации алгоритма (будь то LU-разложение или что-нибудь другое), при которой основной операцией над матрицами (блоками) станет их умножение. Если значения сеточных функций на двумерной по пространству области действительно удобно хранить в виде матриц, то представить вычисления по явной разностной схеме через операции умножения плотных матриц решительно невозможно. Каноническая форма записи разностных схем [25] содержит лишь ленточные матрицы с достаточно узкой лентой. С другой стороны, Джеймс Деммель [9] говорит о блочности как о средстве управления коммуникациями между различными уровнями памяти ЭВМ, то есть в необходимом здесь ключе. К сожалению, он не предлагает универсальных рецептов конструирования блочных алгоритмов, которые можно применить в другой предметной области.

Лишь недавно [11, 12] такие алгоритмы появились в практике решения сеточных уравнений FDTD-метода и уравнения Даламбера. Их авторы обращаются к приему «tiling» удвоения циклических конструкций [10], который по сути и является программистской основой блочности в любой предметной области. Смысл такой блочности состоит в многократном (как можно более интенсивном) использовании локального участка общих данных, загруженного в верхний уровень иерархической памяти ЭВМ. В [11, 12] обсуждается выбор оптимальной формы фрагмента сеточной области, составляющего блок, и указывается на наклонную башню с ромбовидным основанием, называемую «DiamondTogge» [11] или «diamond tiling» [12], в качестве таковой.

Формулируя блочный алгоритм совместного разностного решения уравнений Даламбера и Максвелла, разделим пространственную часть сеточной области на две подобласти. К первой отнесём зону расположения PML-слоёв у границ вычислительной области (там решаются уравнения Максвелла) и полосу шириной  $h$  (блочный параметр), прилегающую к этой зоне (в ней решается уравнение Даламбера). Данную подобласть не будем разделять на блоки в силу её небольшой площади и высокой сложности организации блочных вычислений в PML-слоях. В ходе первого, неблочного этапа вычислений производится пересчёт сеточных функций в оговоренной зоне на  $h$  временных слоях и  $h-i$  слоях для сеточных функций, определённых в упомянутой полосе и отстоящих на  $i$  узлов от зоны. Фигура, образованная заданными таким образом узлами, напоминает ковш (рис. 1), дно которого относится ко второй пространственной подобласти. Значения сеточных функций, там определённых, на первом этапе не вычислялись. Наклонные стенки ковша соответствуют временным слоям  $h-i$  (полоса,

прилегающая к PML), а ручка – временному слою  $h$  (зона расположения PML).

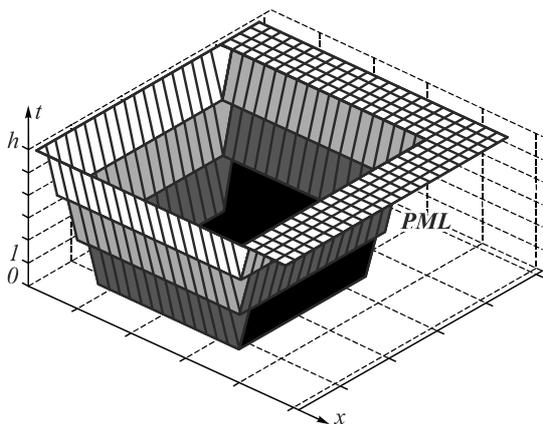


Рис. 1. Распределение сеточных функций по временным слоям перед началом блочного этапа вычислений. От белого (слои на максимальной высоте  $h$  и рядом) к черному (слои на минимальной высоте и около неё). Под ручкой ковша расположены PML-слои

На втором, блочном этапе алгоритма вычисляются значения сеточных функций внутри ковша. Наклонную башню с ромбовидным основанием из [11] трансформируем в наклоненную усеченную пирамиду с прямоугольным основанием, большая сторона которого (далее ширина) равна ширине дна ковша из рис. 1, а меньшая (далее длина) –  $h$ . С ростом высоты пирамиды (тоже  $h$ ) длина её горизонтального сечения не изменятся, а сама пирамида наклоняется в сторону, обратную направлению перебора блоков. Ширина сечения увеличивается в соответствии с расширением стенок ковша. На рис. 2 изображена проекция блока на плоскость, разрезающую ковш вдоль направления перебора. Большой стрелкой указано направление обхода блоков в ходе вычислительного процесса, малыми – направления обхода узлов сеточной области внутри блока: снизу вверх и слева направо.

Количество чередований обоих этапов алгоритма определяется отношением общего количества временных слоёв сеточной области к блочному параметру.

Следуя традиции [11, 12], назовем сформулированный алгоритм пирамидальным. Его упрощением будет второй алгоритм, в котором блок станет представлять собой правую грань пирамиды. Ход вычислительного процесса в этом случае будет напоминать распространение плоских волн в ковше от левого края к правому. По этой аналогии и следуя за авторами [27], назовём такой алгоритм волновым.

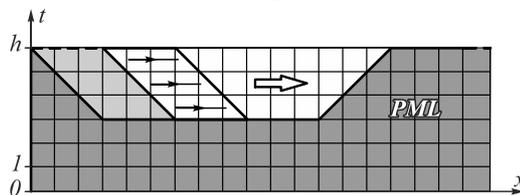


Рис. 2. Распределение сеточных функций по временным слоям на блочном этапе вычислений. Тёмно-серый цвет – значения функций, определённые до второго этапа алгоритма, светло-серый – уже определены в ходе второго этапа, внутри параллелограмма – вычисляются на настоящий момент, белый – ожидают определения

Коренным отличием обоих алгоритмов совместного разностного решения уравнений Даламбера и Максвелла от сформулированного в [14] является изменение направления обхода узлов сеточной области при расчёте значений сеточных функций. Ранее такой обход производился внутри одного временного слоя, а слои чередовались в порядке возрастания номера. Теперь сеточная область обходится по блокам, причём один блок пересекает  $h$  временных слоёв. Сам же численный метод и его характеристики (аппроксимация, устойчивость) остаётся без изменений в силу неизменности дифференциального шаблона.

В табл. 2 представлены результаты экспериментального исследования пирамидального и волнового алгоритмов для прежнего аппаратного, системного, программного обеспечения и той же сеточной области. Выбранные значения блочного параметра  $h$  – чётные в силу особенностей программной реализации разностного решения уравнения Даламбера [22], и общее количество слоёв по времени сеточной области (200) делится на них нацело.

Табл. 2. Зависимость длительности вычислений (в секундах) по блочным алгоритмам от аппаратного, системного и программного обеспечения. Верхнее значение в ячейке для пирамидального алгоритма, нижнее – для волнового

Процессор	Операционная система и компилятор	$h$							
		2	4	8	10	20	40	50	100
i7-3770	Ubuntu и GCC	37,91 37,75	31,72 32,15	28,69 28,41	28,41 27,85	28,12 26,76	32,26 31,90	36,35 36,06	43,63 43,95
	Cent OS и PGI Fortran	38,04 37,90	31,55 31,23	28,55 27,98	28,22 27,43	27,84 26,18	32,41 32,19	36,44 36,60	44,40 44,35
i5-4200M	Ubuntu и GCC	56,19 55,81	44,42 43,57	43,95 37,96	42,18 38,19	65,83 68,30	67,18 68,43	67,07 68,32	67,23 68,08
	Ubuntu и Intel Fortran	56,88 56,16	45,29 44,36	44,90 38,73	44,56 39,57	65,18 64,64	67,58 68,76	67,71 68,80	67,57 68,81
i3-4150	Ubuntu и GCC	49,80 49,51	38,66 38,20	38,27 32,61	36,16 31,65	56,75 53,80	60,96 60,88	60,98 60,73	60,97 60,61

Как и ранее (табл. 1), смена компилятора и операционной системы почти не влияют на результаты экспериментов, что позволяет сравнивать алгоритмы с помощью их программных реализаций и в блочном

случае. Далее будем обсуждать значения, полученные на Ubuntu посредством GCC.

Как видно из табл. 2, увеличение блочного параметра сначала приводит к ускорению вычислений, за-

тем, по достижении некоторой минимальной величины, длительность расчётов по блочным алгоритмам начинает расти. Ключом к объяснению данного эффекта является сопоставление экспериментальных данных и зависимости объёма блока от  $h$ . Так, для  $h = 2, 4, 8, 10, 20, 40, 50$  и  $100$  соответствующие блоки занимают 0,45; 0,9; 1,79; 2,24; 4,49; 8,97; 11,22 и 22,43 Мб. Наилучший по времени результат для первого процессора соответствует значению блочного параметра  $h = 20$  узлов – максимальному, при котором блок ещё полностью умещается в кэш-память (8 Мб), для второго и третьего (3 Мб) – 10 (или близкое к 10 значение 8) узлов. Для блоков с меньшей высотой кэш используется не полностью, для блоков с большей – неоднократно при работе с одним блоком. Оба обстоятельства приводят к увеличению коммуникаций между оперативной и кэш-памятью и, как следствие, замедлению вычислений.

Сравнивая алгоритмы, следует отдать предпочтение волновому перед пирамидальным, особенно при оптимальных значениях блочного параметра. Для процессоров i5-4200M и i3-4150 разница между алгоритмами по длительности вычислений составила 16 % и 14 % соответственно, для i7-3770 – 5 %. Возможно, это объясняется простотой волнового алгоритма по сравнению с пирамидальным, за счёт которой компилятору удаётся сформировать лучший в смысле быстройдействия код.

Итоговые результаты в табл. 3 приведены для волнового алгоритма с оптимальным для каждого процессора блочным параметром при работе с операционной системой Ubuntu и компилятором GCC. Ускорение  $S_1$  определено как отношение длительности расчётов с использованием пакета МЕЕР и блочного алгоритма,  $S_2$  – программы Susan C. Hagness и блочного алгоритма,  $S_3$  – неблочного и блочного алгоритмов совместного разностного решения уравнений Даламбера и Максвелла.

Табл. 3. Сравнение волнового блочного алгоритма с пакетом МЕЕР, программой Susan C. Hagness и неблочной реализацией совместного разностного решения уравнений Даламбера и Максвелла

Процессор	Ускорение		
	$S_1$	$S_2$	$S_3$
i7-3770	4,59	4,08	1,59
i5-4200M	5,44	4,57	1,74
i3-4150	5,88	4,99	1,89

Продолжая пример для процессора i3-4150 из предыдущего параграфа с прогнозированием длительности вычислительного эксперимента, отметим, что если для МЕЕР он длился бы 10,34 часа, то для волнового блочного алгоритма из настоящего параграфа – всего 1,76 часа.

### Заключение

Конструирование и программная реализация блочных алгоритмов совместного разностного решения уравнений Даламбера и Максвелла позволили ускорить вычисления по FDTD-методу до 6 раз по

сравнению с пакетом МЕЕР Массачусетского технологического института за счёт оптимизации коммуникаций между оперативной памятью и кэш-памятью процессора. Развитие предложенного подхода авторы видят в синтезе параллельных вариантов блочных алгоритмов совместного разностного решения.

### Благодарности

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

### Литература

1. **Воеводин, В.В.** Математические модели и методы в параллельных процессах / В.В. Воеводин. – М.: Наука, 1986. – 296 с.
2. **Shen, J.P.** Modern processor design: Fundamentals of super-scalar processors / J.P. Shen, M.H. Lipasti. – 2<sup>nd</sup> ed. – Long Grove, Illinois: Waveland Press, Inc, 2013. – 642 p. – ISBN: 978-1-4786-0783-0.
3. **Cron, G.** Equivalent circuit of the field equations of Maxwell / G. Cron // Proceedings of the IRE. – 1944. – Vol. 32, Issue 5. – P. 289-299. – DOI: 10.1109/JRPROC.1944.231021.
4. **Yu, W.** Parallel finite-difference time-domain method / W. Yu, R. Mittra, T. Su, Y. Liu, X. Yang. – Boston: Artech House, 2006. – 272 p. – ISBN: 978-1-59693-085-8.
5. **Jordan, H.F.** Experience with FDTD techniques on the Cray MTA supercomputer / H.F. Jordan, S. Bokhari, S. Staker, J.R. Sauer, M.A. ElHelbawy, M.J. Picket-May // Proceedings of the SPIE. – 2001. – Vol. 4528. – P. 68-76. – DOI: 10.1117/12.434878.
6. **Inman, M.J.** Optimization and parameter exploration using GPU based FDTD solvers / M.J. Inman, A.Z. Elsherbeni // IEEE MTT-S International Microwave Symposium Digest. – 2008. – P. 149-152. – DOI: 10.1109/MWSYM.2008.4633125.
7. **Waidyasooriya, H.M.** FPGA-based deep-pipelined architecture for FDTD acceleration using OpenCL / H.M. Waidyasooriya, M. Hariyama // IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). – 2016. – P. 109-114. – DOI: 10.1109/ICIS.2016.7550742.
8. **Golub, G.H.** Matrix computations / G.H. Golub, Ch.F. Van Loan. – 3<sup>rd</sup> ed. – Baltimore, London: Johns Hopkins University Press, 1996. – 694 p. – ISBN: 0-8018-5414-8.
9. **Деммель, Дж.** Вычислительная линейная алгебра. Теория и приложения / Дж. Деммель. – М.: Мир, 2001. – 435 с. – ISBN: 5-03-003402-1.
10. **Wolfe, M.** More iteration space tiling / M. Wolfe // Proceedings of the ACM/IEEE Conference on Supercomputing (Supercomputing '89). – 1989. – P. 655-664. – DOI: 10.1145/76263.76337.
11. **Perepelkina, A.Yu.** DiamondTorre algorithm for high-performance wave modeling / A.Yu. Perepelkina, V.D. Levenchenko // Keldysh Institute Preprints. – 2015. – No. 18. – 20 p.
12. **Orozco, D.** Mapping the FDTD application to many-core chip architectures / D. Orozco, G. Guang // International Conference on Parallel Processing (ICPP '09). – 2009. – P. 309-316. – DOI: 10.1109/ICPP.2009.44.
13. **Головашкин, Д.Л.** Совместное разностное решение уравнений Даламбера и Максвелла. Одномерный случай / Д.Л. Головашкин, Л.В. Яблокова // Компьютерная оптика. – 2012. – Т. 36, № 4. – С. 527-533.
14. **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.
15. **Марков, А.А.** Теория алгоритмов / А.А. Марков. – М., Л.: Издательство Академии Наук СССР, 1954. – 377 с.
  16. **Golovashkin, D.L.** Use of the finite-difference method for solving the problem of H-wave diffraction by two-dimensional dielectric gratings / D.L. Golovashkin, N.L. Kazansky, V.N. Safina // Optical Memory and Neural Networks. – 2004. – Vol. 13, No. 1. – P. 55-62.
  17. **Козлова, Е.С.** Моделирование предвестников Зоммерфельда и Бриллюэна в среде с частотной дисперсией на основе разностного решения волнового уравнения / Е.С. Козлова, В.В. Котляр // Компьютерная оптика. – 2013. – Т. 37, № 2. – С. 146-154.
  18. **Elsherbeni, A.Z.** The finite-difference time-domain method for electromagnetics with MATLAB simulations / A.Z. Elsherbeni, V. Demir. – Raleigh, NC: SciTech Publishing, Inc., 2009. – 426 p. – ISBN: 978-1-891121-71-5.
  19. **Taflove, A.** Computational electrodynamics: The finite-difference time-domain method / A. Taflove, S. Hagness. – 3<sup>rd</sup> ed. – Boston: Artech House Publishers, 2005. – 1006 p. – ISBN: 978-1-58053-832-9.
  20. **Oskooi, A.F.** MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method / A.F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J.D. Joannopoulos, S.G. Johnson // Computer Physics Communications. – 2010. – Vol. 181, Issue 3. – P. 687-702. – DOI: 10.1016/j.cpc.2009.11.008.
  21. **Плохотников, К.Э.** Вычислительные методы. Теория и практика в среде MATLAB: курс лекций / К.Э. Плохотников. – 2-е изд. – М.: МГУ, 2015. – 496 с. – ISBN 978-5-9912-0354-8.
  22. Свидетельство о государственной регистрации программы для ЭВМ № 2017613903 «Совместное разностное решение уравнений Даламбера и Максвелла».
  23. **Головашкин, Д.Л.** Постановка излучающего условия при моделировании работы цилиндрических дифракционных оптических элементов методом разностного решения уравнений Максвелла // Математическое моделирование. – 2007. – Т. 19, № 3. – С. 3-14.
  24. **Foster, I.** Designing and building parallel programs: Concepts and tools for parallel software engineering / I. Foster. – Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1995. – 430 p. – ISBN: 978-0-2015-7594-1.
  25. **Самарский, А.А.** Теория разностных схем. – М.: Наука, 1977. – 656 с.
  26. **Gallivan, K.** Impact of hierarchical memory system on linear algebra algorithm design / K. Gallivan, W. Jalby, U. Meier, A.H. Sameh // The International Journal of Supercomputer Applications. – 1988. – Vol. 2, Issue 1. – P. 12-48. – DOI: 10.1177/109434208800200103.
  27. **Wolfe, M.** Loops skewing: The wavefront method revisited / M. Wolfe // International Journal of Parallel Programming. – 1986. – Vol. 15, Issue 4. – P. 279-293. – DOI: 10.1007/BF01407876.

#### Сведения об авторах

**Яблокова Людмила Вениаминовна**, старший преподаватель кафедры прикладной математики Самарского национального исследовательского университета имени академика С. П. Королева. Область научных интересов: разностные схемы. E-mail: [lyablokova@yandex.ru](mailto:lyablokova@yandex.ru).

**Головашкин Дмитрий Львович**, доктор физико-математических наук, ведущий научный сотрудник лаборатории дифракционной оптики Института систем обработки изображений РАН – филиала Федерального государственного учреждения Федеральный научно-исследовательский центр «Кристаллография и фотоника» Российской академии наук, профессор кафедры прикладной математики Самарского национального исследовательского университета имени академика С. П. Королева. Область научных интересов: FDTD-метод, векторные и матричные вычисления, математическое моделирование оптических элементов и устройств нанофотоники. E-mail: [dimitriy@smr.ru](mailto:dimitriy@smr.ru).

ГРПТИ: 27.41.19.

Поступила в редакцию 26 декабря 2017 г. Окончательный вариант – 14 марта 2018 г.

#### BLOCK ALGORITHMS OF A SIMULTANEOUS DIFFERENCE SOLUTION OF D'ALEMBERT'S AND MAXWELL'S EQUATIONS

L.V. Yablokova<sup>1</sup>, D.L. Golovashkin<sup>1,2</sup>

<sup>1</sup> Samara National Research University, Samara, Russia,

<sup>2</sup> Image Processing Systems Institute of RAS – Branch of the FSRC “Crystallography and Photonics” RAS, Samara, Russia

#### Abstract

The work is devoted to the synthesis of block algorithms of the FDTD method. In particular, the simultaneous difference solution of d'Alembert's and Maxwell's equations is considered. Accounting for the computer memory hierarchical structure allows the calculation time to be reduced up to six times when compared with the known software implementations of the method.

**Keywords:** FDTD-method, block algorithms, computational speed-up.

**Citation:** Yablokova LV, Golovashkin DL. Block algorithms of a simultaneous difference solution of d'Alembert's and Maxwell's equations. Computer Optics 2018; 42(2): 320-327. DOI: 10.18287/2412-6179-2018-42-2-320-327.

**Acknowledgements:** The work was funded by the Russian Foundation for Basic Research under grant 16-47-630560 p\_a and by the Federal Agency of Scientific Organizations (agreement No. 007-GZ/C3363/26).

### References

- [1] Voevodin VV. Mathematical models and methods in parallel processes [In Russian]. Moscow: "Nauka" Publisher; 1986.
- [2] Shen JP, Lipasti MH. Modern processor design: Fundamentals of superscalar processors. 2nd ed. Long Grove, Illinois: Waveland Press, Inc; 2013. ISBN: 978-1-4786-0783-0.
- [3] Cron G. Equivalent circuit of the field equations of Maxwell. Proc IRE 1944; 32(5): 289-299. DOI: 10.1109/JRPROC.1944.231021.
- [4] Yu W, Mittra R, Su T, Liu Y, Yang X. Parallel finite-difference time-domain method. Boston: Artech House; 2006. ISBN: 978-1-59693-085-8.
- [5] Jordan HF, Bokhari S, Staker S, Sauer JR, ElHelbawy MA, Picket-May MJ. Experience with FDTD techniques on the Cray MTA supercomputer. Proc SPIE 2001; 4528: 68-76. DOI: 10.1117/12.434878.
- [6] Inman MJ, Elsherbeni AZ. Optimization and parameter exploration using GPU based FDTD solvers. IEEE MTT-S International Microwave Symposium Digest 2008: 149-152. DOI: 10.1109/MWSYM.2008.4633125.
- [7] Waidyasooriya HM, Hariyama M. FPGA-based deep-pipelined architecture for FDTD acceleration using OpenCL. ICIS 2016: 109-114. DOI: 10.1109/ICIS.2016.7550742.
- [8] Golub GH, Van Loan ChF. Matrix Computations. 3<sup>rd</sup> ed. Baltimore, London: Johns Hopkins University Press; 1996. ISBN: 0-8018-5414-8.
- [9] Demmel J. Applied numerical linear algebra. Philadelphia: SIAM; 1997. ISBN: 0-89871-389-7.
- [10] Wolfe M. More iteration space tiling. Proc Supercomputing '89 1989: 655-664. DOI: 10.1145/76263.76337.
- [11] Perepelkina AYu, Levchenko VD. DiamondTorre algorithm for high-performance wave modeling. Keldysh Institute Preprints 2015; 18: 1-20.
- [12] Orozco D, Guang G. Mapping the FDTD application to many-core chip architectures. ICPP '09 2009: 309-316. DOI: 10.1109/ICPP.2009.44.
- [13] Golovashkin DL, Yablokova LV. Joint finite-difference solution of the Dalamber and Maxwell's equations. One-dimensional case. Computer Optics 2012; 36(4): 527-533.
- [14] Buldygin EYu, Golovashkin DL, Yablokova LV. Joint finite-difference solution of the D'Alembert and Maxwell's equations. Two-dimensional case [In Russian]. Computer Optics 2014; 38(1): 20-27.
- [15] Markov AA, Nagorny NM. The theory of algorithms. Dordrecht, Netherlands: Springer, 1988. ISBN: 978-90-277-2773-2.
- [16] Golovashkin DL, Kazansky NL, Safina VN. Use of the finite-difference method for solving the problem of H-wave diffraction by two-dimensional dielectric gratings. Optical Memory and Neural Networks 2004; 13(1): 55-62.
- [17] Kozlova ES, Kotlyar VV. Simulations of Sommerfeld and Brillouin precursors in the medium with frequency dispersion using numerical method of solving wave equations [In Russian]. Computer Optics 2013; 37(2): 146-154.
- [18] Elsherbeni AZ, Demir V. The finite-difference time-domain method for electromagnetics with MATLAB simulations. Ralrigh, NC: SciTech Publishing Inc 2009. ISBN: 978-1-891121-71-5.
- [19] Taflove A, Hagness S. Computational electrodynamics: The finite-difference time-domain method. 3<sup>th</sup> ed. Boston: Artech House Publishers, 2005. ISBN: 978-1-58053-832-9.
- [20] Oskooi AF, Roundyb D, Ibanescua M, Bermel P, Joannopoulos JD, Johnson SG. MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method. Comput Phys Commun 2010; 181(3): 687-702. DOI: 10.1016/j.cpc.2009.11.008.
- [21] Plokhotnikov KE. Computational methods: Theory and practice in the MATLAB environment. The course of lectures [In Russian]. Moscow: "MGU" Publisher; 2007.
- [22] Certificate of state registration of the computer program No. 2017613903 "Joint difference solution of the d'Alembert and Maxwell equations" [In Russian].
- [23] Golovashkin DL. Formulation of the radiation condition for modeling the cylindrical doe operation using a finite difference solution of Maxwell's equations. Matem Mod 2007; 19(3): 3-14.
- [24] Foster I. Designing and building parallel programs: Concepts and tools for parallel software engineering. Boston, MA: Addison-Wesley Longman Publishing; 1995. ISBN: 978-0-2015-7594-1.
- [25] Samarskii AA. The theory of difference schemes. New York, NY: Marcel Dekker Inc; 2001. ISBN: 978-0-8247-0468-1.
- [26] Gallivan K, Jalby W, Meier U, Sameh AH. Impact of hierarchical memory system on linear algebra algorithm design. The International Journal of Supercomputer Applications 1988; 2(1): 12-48. DOI: 10.1177/109434208800200103.
- [27] Wolfe M. Loops skewing: The wavefront method revisited. International Journal of Parallel Programming 1986; 15(4): 279-293. DOI: 10.1007/BF01407876.

### Authors' information

**Lyudmila Veniaminovna Yablokova** works as senior teacher of the Applied Mathematics sub-department at Samara University. Research interests: finite-difference approximations. E-mail: [lyablokova@yandex.ru](mailto:lyablokova@yandex.ru).

**Dimitriy Lvovich Golovashkin** Doctor of Physics and Mathematics, leader researcher Diffractive Optics laboratory at the Image Processing Systems Institute of RAS – Branch of the FSRC "Crystallography and Photonics" RAS. Works as professor of the Applied Mathematics department at Samara University. Research interests are FDTD-method, vectors and matrix computation, mathematical modeling of optical components and nanophotonic devices. E-mail: [dimitriy@smr.ru](mailto:dimitriy@smr.ru).

*Received December 26, 2017. The final version – March 14, 2018.*