Ускорение вычислений по блочным алгоритмам разностного решения уравнения теплопроводности

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

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

Аннотация. Учет особенностей архитектуры конкретной вычислительной системы при разработке нового алгоритма известного численного метода давно считается необходимым при синтезе параллельных и векторных алгоритмов. В настоящей работе предлагается принимать во внимание архитектурные особенности процессора еще на этапе конструирования самого численного метода, как это когда-то предлагалось академиком Гурием Ивановичем Марчуком, однако так и не закрепилось в широкой вычислительной практике. Данная идея иллюстрируется на примере синтеза новой разностной схемы для уравнения теплопроводности, традиционно являющимся объектом для испытания новшеств в теории разностных схем. В качестве упомянутой архитектурной особенности выбрана иерархическая структура памяти ЭВМ, обуславливающая появление коммуникационных издержек даже при использовании одного аппаратного вычислительного потока для организации расчетов. Учет данной особенности в вычислительной линейной алгебре связывают с блочными алгоритмами, в теории разностных схем – с приемом программирования «tiling». Однако для двухслойных разностных схем блочных алгоритмов решения сеточных уравнений до предлагаемой работы известно не было в силу невозможности организации блочных вычислений по существующим схемам. Для восполнения этого пробела авторы предлагают новый прием конструирования двухслойных разностных схем и смешанную схему со сдвигом как пример применения этого приема. В ходе экспериментов демонстрируется пятикратное ускорение расчетов по такой схеме относительно традиционной явной при той же вычислительной сложности.

1. Введение

Математическое моделирование, как самостоятельная предметная область, получила признание в середине прошлого века в связи с необходимостью реализации крупных научно-технических проектов, давших импульс развитию математической физике, вычислительной математике и вычислительной технике. Классическое понимание математического моделирования [1] связывает указанные научные отрасли с соответствующими компонентами триады академика Александра Андреевича Самарского: моделью, численным методом и программным комплексом. В течение длительного времени вторая и тем более третья компоненты триады считались второстепенными, к их разработке приступали после завершения работы над математической моделью. В свою очередь, создание численного метода как правило предшествовало выбору вычислительной системы, на которой он впоследствии

реализовывался. Положение изменилось в 70-е годы прошлого века, когда директор Вычислительного центра СО АН СССР Гурий Иванович Марчук поставил задачу отображения численного метода на архитектуру вычислительной системы [2], решение которой подразумевало учет архитектурных особенностей при синтезе метода.

Предлагаемая работа является приложением этой задачи к теории разностных схем. Среди упомянутых особенностей выбрана иерархическая структура памяти ЭВМ, как наименее изученная с точки зрения теории разностных схем [3] с одной стороны, и наиболее важная для этой теории, по мнению авторов ряда публикаций [4-7], с другой. В смежной предметной области, вычислительной линейной алгебре, прием составления блочных алгоритмов давно считается классическим [8,9] и зарекомендовал себя как надежный инструмент сокращения длительности вычислений за счет минимизации коммуникаций между оперативной памятью и кэш-памятью центрального процессора или видеопамятью графического процессора. С точки зрения разработчиков программного обеспечения этот прием именуется «tiling» [10] и используется при оптимизации кода независимо от его предназначения [11,12].

Наиболее общее представление о блочности [9] согласуется с требованием интенсификации использования локального фрагмента общих данных, загруженного на верхний иерархический уровень памяти ЭВМ, до его выгрузки. Чем больше арифметических операций будет произведено над этим фрагментом, тем реже придется загружать его в дальнейшем. Следовательно, снизится длительность коммуникаций между различными уровнями памяти, зачастую определяющая общую длительность вычислений по алгоритму. Программно такая стратегия реализуется удвоением циклических конструкций («tiling»), позволяющих перераспределять общее количество итераций между различными уровнями вложенности при неизменном объеме арифметических операций в алгоритме.

Применительно к теории разностных схем обсуждаемая методика сокращения длительности вычислений начала применяться недавно [6] и еще не успела получить широкого распространения. В отличие от тривиального подхода, при котором вычисления на следующем временном слое сеточной области начинаются после их завершения на предыдущем, блочные алгоритмы характеризуются локализацией вычислительного процесса внутри заданной подобласти сеточной области (блока), пересекающей несколько (от единиц до сотен) слоев по времени. В блоке расчеты производятся послойно, а блоки перебираются в установленном порядке. Популярной формой блока признается параллелепипед («Diamond Torre» [4,5]).

Известные блочные алгоритмы [4-7] предназначены для организации расчетов по явным трехслойным разностным схемам. Вычисления по неявным схемам сопровождаются решением систем линейных уравнений (многократное усложнение задачи), а явные двухслойные не годятся для блочности. Традиционно при их программной реализации значения сеточной функции на новом слое пишутся поверх значений на старом с целью экономии памяти. Введение дополнительного массива для хранения прежних значений, без которых блочность невозможна, лишь увеличит коммуникации внутри иерархической структуры памяти ЭВМ, что противоречит самой идее блочности. Поэтому авторы настоящей работы обратились к созданию нового приема построения разностных схем, лишенных указанного недостатка.

2. Смешанная разностная схема со сдвигом

Выбирая пример, иллюстрирующий предлагаемый прием, авторы остановились на одномерном однородном линейном нестационарном уравнении теплопроводности – традиционном объекте для демонстрации методик составления сеточных уравнений в теории разностных схем [3]. Кроме упомянутой методической ценности, безусловной научной будет характеризоваться обобщение приема на случай больших размерностей [13]. Однако, прежде необходимо засвидетельствовать его эффективность для выбранного примера, как простейшего.

Так, для уравнения

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \tag{1}$$

известны «школьные» явная

$$\frac{U_i^n - U_i^{n-1}}{\tau} = \frac{U_{i-1}^{n-1} - 2U_i^{n-1} + U_{i+1}^{n-1}}{h^2} \quad (2)$$

и неявная

$$\frac{U_i^n - U_i^{n-1}}{\tau} = \frac{U_{i-1}^{n} - 2U_i^{n} + U_{i+1}^{n}}{h^2} \quad (3)$$

разностные схемы, где функция U определена на вычислительной области $\omega=\{(t,x)$: $0\le t\le T$; $0\le x\le L\}$, а ее сеточный аналог U_i^n на сеточной области $\omega_h=\{(t_n,x_i)$: $t_n=\tau\cdot n$; $n=0,..,N$; $\tau=T/N$; $x_i=h\cdot i$; $i=0,..,I+1$; $\tau=L/(I+1)\}$. Краевые условия первого рода, начальное условие и их дискретизация дополняют дифференциальную задачу и разностные схемы для нее.

Как отмечалось ранее, известные методики составления блочных алгоритмов [4-7] не подходят для работы с (2) и (3) в силу упомянутых причин. Однако комбинация (2) и (3) в одной разностной схеме меняет ситуацию.

Пусть для определенности I – нечетное. Тогда примем, что на временном слое n=1 в четных его узлах (i=2,4,6,...,I-1) значения сеточной функции вычисляются по явному дифференциальному шаблону и формуле (2). Очевидно, этому ничего не препятствует, ведь значения на предыдущем слое при n=0 известны (дискретизация начального условия)

$$U_i^n = U_i^{n-1} + \frac{\tau}{h^2}\left(U_{i-1}^{n-1} - 2U_i^{n-1} + U_{i+1}^{n-1}\right) \quad (4)$$

Найденные таким образом U_i^1, помещаем в некоторый одномерный массив поверх U_i^0 при четных i.

Затем производим вычисления на том же слое для определения значений сеточной функции в нечетных узлах i=1,3,5,...,I по неявному дифференциальному шаблону и формуле (3). И этому также нет препятствий: соседние для U_i^1 значения U_{i-1}^1 и U_{i+1}^1 только что найдены по явному шаблону или определены из краевых условий. Таким образом, вычисления по неявной части новой разностной схемы производятся в явном виде:

$$U_i^n = \left(U_i^{n-1} + \frac{\tau}{h^2}\left(U_{i-1}^{n} + U_{i+1}^{n}\right)\right)\left(1 + 2\frac{\tau}{h^2}\right)^{-1} \quad (5)$$

Найденные U_i^1 помещаются в тот же массив поверх U_i^0 при нечетных i, в итоге данный массив хранит значения сеточных функций только на слое 1.

Явное выражение искомой сеточной функции при расчетах по неявной схеме не ново, известны схемы бегущего счета, в том числе и для уравнения (1) [14]. Однако, в отличие от них, здесь на одном временном слое применяются разные дифференциальные шаблоны и при переходе к следующему узлу по пространству шаблон не сдвигается на одну позицию, а «перепрыгивает» через узел. В силу этого обстоятельства авторы предлагают называть предложенную схему смешанной.

Переход на следующий временной слой n=2 сопровождается сдвигом шаблонов. Теперь в нечетных узлах производятся расчеты по (4), затем в четных по (5). Использование разных шаблонов на разных слоях также принято в уже упомянутой схеме бегущего счета из [14], новизна предлагаемой здесь обусловлена сдвигом шаблонов, используемых на предыдущем слое вместо введения нового. С учетом этого обстоятельства полное название предлагаемой семы – смешанная со сдвигом.

В итоге, на нечетных временных слоях сначала производятся вычисления по явному шаблону для четных по пространству узлов (шаг 1), потом по неявному для нечетных (шаг 2). На четных наоборот: по явному для нечетных узлов (шаг3) и по неявному для четных (шаг 4). В любом случае при переходе на следующий временной слой в первую очередь применяется явный дифференциальный шаблон. Случай четных I отличается от рассмотренного лишь видом шаблона, используемого для расчетов в узле I.

3. Блочный алгоритм

Поясним идею блочного алгоритма примером на рисунке 1 и в таблице 1.

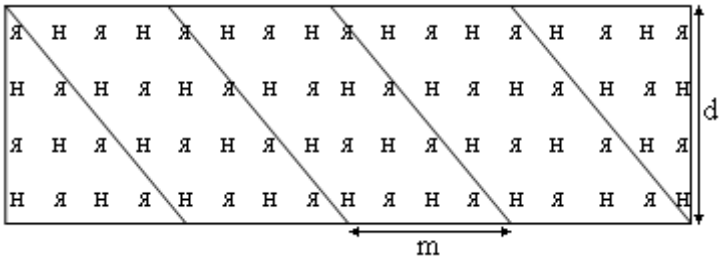

Рисунок 1. Пример разбиения на блоки участка сеточной области. Узлы, вычисления в которых производятся по явному шаблону, маркированы буквой «я», по неявному – «н».

Таблица 1. Изменение содержимого массива A в зависимости от этапа алгоритма.

этап алгоритма	содержание массива A
до начала вычислений	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
после вычислений в первом блоке	4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
после вычислений во втором блоке	4 4 4 4 4 3 2 1 0 0 0 0 0 0 0 0 0
после вычислений в третьем блоке	4 4 4 4 4 4 4 4 4 3 2 1 0 0 0 0 0
после вычислений в четвертом блоке	4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1 0
после вычислений в пятом блоке	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Положим в ω_h I=17 и N=4 и разобьем сеточную область на 5 блоков: левый (первый) в виде треугольника, правый (пятый) – треугольник, дополняющий первый до квадрата и три (второй, третий и четвертый) центральных параллелограмма. Границы между блоками назовем внутренними. Тогда на левой внутренней границе каждого блока окажутся узлы, связанные с расчетами исключительно по неявному шаблону, а на правой – по явному. Это обусловлено упомянутым ранее сдвигом дифференциального шаблона при переходе на следующий временной слой и обеспечивает, как будет показано далее, возможность довольствоваться одним одномерным массивом длины I+2 при хранении значений сеточной функции (краевых в том числе) в ходе вычислений по блочному алгоритму.

В отличие от тривиального подхода, когда в ходе вычислительно процесса узлы сеточной области обходятся послойно, блочность предполагает перебор узлов внутри блока, пересекающего несколько временных слоев, с последующим переходом к другому блоку. Говоря о рассматриваемом примере, вычислительный процесс для него начнется послойным перебором узлов сеточной области внутри левого треугольника. Сопроводим описание алгоритма демонстрацией содержимого одномерного массива A (таблица 1) длины I (не путать с основным массивом той же длины, хранящим значения сеточной функции), i-ая ячейка которого будет содержать значение индекса n, соответствующее номеру слоя у U_i^n из основного массива. Так, после завершения вычисления значений сеточных функций в узлах, расположенных внутри первого блока (рисунок 1, таблица 1) первые четыре элемента массива A окажутся различными: при i=1 сеточная функция определена на 4-ом слое, при i=2 на 3-ем, i=3 на 2-ом, i=4 на первом, а для i>4 вычислений еще не производилось.

Во втором, третьем и четвертом блоках последовательно производятся вычисления по смешанной схеме со сдвигом, послойно сверху вниз (рисунок 1) и со смещением на один пространственный узел влево при переходе на следующий временной слой. Во всех ранее вычисленных блоках значения сеточной функции определены на последнем временном слое, в только что вычисленном – на всех временных слоях кроме начального, в блоках с большими номерами – на начальном (таблица 1).

Действия по алгоритму завершаются расчетами в пятом блоке треугольной формы, после чего искомые значения сеточной функции на последнем слое считаются известными. Здесь, в отличие от первого блока, с увеличением номера слоя количество арифметических операций возрастает.

Важной характеристикой блочного алгоритма является очевидность векторизации вычислений по нему. Действительно, действия на каждом временном слое любого блока можно разбить на две векторных операции типа saxpy [8], связанных с расчетами по явному и неявному дифференциальным шаблонам.
Не вызовет затруднений и обобщение алгоритма на случай произвольных N и I. Введем при этом два параметра: ширину (m) и высоту (d) блока, определяя количество блоков в одном блочном временном слое как [I/m]+1, а число таких слоев как]N/d[, где [..] и]..[- операции округления до наименьшего и наибольшего целого соответственно.

4. Ускорение вычислений по блочному алгоритму
Развернутое исследование предложенных разностной схемы и блочного алгоритма для нее представляются авторам настоящей работы содержанием отдельной публикации, здесь же они ограничатся демонстрацией превосходства, изложенного математического и алгоритмического аппарата над классической явной схемой и тривиальным алгоритмом.
Эксперименты проводились на следующей аппаратной и системной базе: процессор Intel Core i5-4460, операционная система Linux Ubuntu 16.04, компилятор gfortran 5.4.
В первой серии экспериментов сравнивались длительности вычислений по классической явной и авторской смешанной со сдвигом разностным схемам при тривиальном подходе к организации вычислительного процесса. Ожидалось равное время расчетов или небольшое ускорение классического метода по сравнению с авторским в силу большего объема кода при реализации последнего. Однако, в широком диапазоне изменения параметров дискретизации сеточной области авторский численный метод неизменно демонстрировал двукратное ускорение (2,3 раза) по сравнению с классическим. Наблюдение за объемом выделяемой памяти в ходе вычислений привело к выводу о влиянии этого параметра на полученный результат. Несмотря на использование в обеих программных реализациях только одного массива для хранения значений сеточной функции, при расчетах по классической явной схеме такой массив в действительности удваивается (хотя в программе этого не предусмотрено!) для предупреждения преждевременного затирания значений сеточной функции на предыдущем слое по времени. При расчетах по авторской схеме в упомянутом удваивании необходимости не возникает. Ценным выводом из этого является заключение об определяющем влиянии длительности коммуникаций между различными уровнями памяти ЭВМ по сравнению с длительностью производства арифметических операций (которых в обоих методах одинаковое количество) на общее время вычислений. А, следовательно, и целесообразность применения блочных алгоритмов, предназначенных для сокращение коммуникационных издержек.
Во второй серии экспериментов исследовалось ускорение блочного алгоритма при фиксированной дискретизации сеточной области и варьируемых параметрах блочности (высоте и ширине блока). Получено два U-образных графика зависимости длительности вычислений от каждого из перечисленных параметров. Действительно, при объеме блока существенно меньше или больше оптимального, кэш-память процессора используется не рационально, что приводит к увеличению длительности коммуникаций. Наибольшее ускорение вычислений по сравнению с тривиальным алгоритмом для авторской схемы составило 2,2 раза; по сравнению с классической явной схемой – 5 раз.

5. Заключение
В публикации демонстрируется перспективность разработки двухслойных явных разностных схем с учетом иерархической структуры памяти ЭВМ на примере смешанной схемы со сдвигом для однородного одномерного нестационарного линейного уравнения теплопроводности.
В качестве перспективных направлений развития данной тематики авторы видят: аналитическое исследование свойств новой схемы (аппроксимации и устойчивости), переход к случаям больших размерностей (двумерному и трехмерному), применение предложенного подхода к синтезу разностных схем для других параболических уравнений математической физики, распространение авторской технологии решения сеточных уравнений на случай неявных разностных схем.

6. Литература

[1] Самарский, А.А. Математическое моделирование: Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. – М.: ФИЗМАТЛИТ, 2002. – 320 с.
[2] Воеводин, В.В. Математические модели и методы в параллельных процессах / В.В. Воеводин. – М.: Наука, 1986. – 296 с.
[3] Самарский, А.А. Теория разностных схем. – М.: Наука, 1977. – 656 с.
[4] Перепелкина, А.Ю. Алгоритм DiamondTorre и высокопроизводительная реализация FDTD метода для суперкомпьютеров с графическими ускорителями / А.Ю. Перепелкина, А.В. Закиров, В.Д. Левченко. – М.: Ин-т прикладной математики им. М.В. Келдыша РАН, 2015. – 22 с.
[5] Perepelkina, A.Yu. Diamond Torre Algorithm for High-Performance Wave Mod-eling / A.Yu Perepelkina, V.D. Levchenko // Keldysh Institute Preprints. – 2015. – Vol. 18. – P. 20.
[6] Orozco, D.A. Mapping the FDTD Application to Many-Core Chip Architectures / D.A. Orozco, G.R. Gao // Parallel Processing, 2009. – P. 309-316.
[7] Takeshi, M. Automatic Parameter Tuning of Three-Dimensional Tiled FDTD Kernel // High Performance Computing for Computational Science – VECPAR. – 2014. – Vol. 8969. – P. 284-297.
[8] Голуб, Дж. Матричные вычисления / Дж. Голуб, Ч. Ван Лоун. – М.: Мир, 1999. – 548 с.
[9] Деммель, Дж. Вычислительная линейная алгебра. Теория и приложения. – М.: Мир, 2001. – 430 с.
[10] Gallivan, K. Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design / K. Gallivan, W. Jalby, U. Meier, A.H. Sameh // International Journal of High Performance Computing Applications. – 1988. – Vol. 2(1). – P. 12-48.
[11] Wolfe, M. More Iteration Space Tiling / M. Wolfe // IEEE Conference on Supercomputing, 1989. – P. 655-664.
[12] Wolfe, M. Loops skewing: The wave front method revisited // International Journal of Parallel Programming. – 1986. – Vol. 15(4). – P. 279-293.
[13] Самарский, А.А. Вычислительная теплопередача / А.А. Самарский, П.Н. Вабищевич. – М.: Едиториал УРСС, 2003. – 784 с.
[14] Калиткин, Н.Н. Численные методы. – М.: Наука, 1987. – 512 с.

Благодарности
Работа выполнена при поддержке гранта РФФИ 19-07-00423 А.

Acceleration of calculations using block algorithms for the difference solution of the heat equation

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

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

Abstract. In this paper, proposing to take into account the architectural features of the processor at the stage of constructing the numerical method itself. This idea is illustrating by the example of the synthesis of a new difference scheme for the heat conduction equation, which has traditionally been the object of testing innovations in the theory of difference schemes. Architectural feature hierarchical structure of computer memory chosen causes the appearance of communication costs even when using a single hardware computational flow for organizing calculations. Accounting for this feature in computational linear algebra is associated with block algorithms. Accounting for this feature in the theory of difference schemes is associated with the technique of programming "tiling". However, for two-layer difference schemes of block algorithms for solving grid equations, prior to the proposed work, it was not known due to the impossibility of organizing block calculations using existing schemes. The authors propose a new method of constructing two-layer difference schemes and a mixed scheme with a shift as an example of the application of this method. In the course of the experiments, a fivefold acceleration of calculations according to this scheme is demonstrated relative to the traditional explicit, with the same computational complexity.