

Синтез блочных алгоритмов разностного решения уравнения переноса

З.Р. Сунагатов¹

¹Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. В современных ЭВМ, зачастую, время, затраченное на вычисление арифметических операций гораздо меньше времени затраченного на коммуникации между различными областями памяти. Причем время этих коммуникаций определяет общее время вычислений. Данную особенность учитывает идея блочности. Применение блочности для решения дифференциальных уравнений началось недавно. К сожалению, в исследуемой автором литературе не нашлось упоминания о блочных алгоритмах для разностного решения уравнения переноса. В связи с этим, в данной работе, впервые была использована идея блочности для решения данного уравнения. В ходе выполнения данной работы был синтезирован и исследован блочный алгоритм разностного решения одномерного уравнения переноса на основе смешанной разностной схемы со смещением. Построенная схема пригодна для использования в ходе вычислений по блочному алгоритму. Особенностью данного алгоритма является отличие в линейных размерах блоков по горизонтали и вертикали от известных алмазных алгоритмов. Указанное отличие позволяет гибко управлять вычислительным процессом. В ходе экспериментального исследования программной реализации блочного алгоритма продемонстрирована его эффективность. Ускорение по сравнению с тривиальным вариантом на основе смешанной разностной схемы со смещением составило 1.5 раза, а по сравнению с тривиальным вариантом на основе явной разностной схемы 2 раза.

1. Введение

В современных ЭВМ, зачастую, время, затраченное на вычисление арифметических операций гораздо меньше времени затраченного на коммуникации между различными областями памяти [1]. Причем время этих коммуникаций определяет общее время вычислений. Данную особенность учитывает идея блочности [2]. Идея блочности существует и очень давно. Блочные алгоритмы часто применяются для ускорения операций умножения матриц. Они позволяют существенно сократить количество коммуникаций в памяти ЭВМ, что существенно ускоряет процесс умножения матриц [3].

Кроме того, в последние годы идея блочности активно применяется при решении сеточных уравнений. Так, в своей работе [4] японский ученый, Такеши Минами сформулировал блочный алгоритм решения уравнения Максвелла и экспериментально подтвердил его эффективность. Данная идея была развита в работе [5] сотрудника института прикладной математики имени М.В. Келдыша, А.В. Перепелкиной для волнового уравнения. В работе [6] Головашкина Д.Л. был предложен блочный алгоритм совместного решения обоих упомянутых уравнений.

Применение блочности – это действительно эффективный способ ускорения вычислений при использовании современной техники и применять его следует везде, где это возможно. К

сожалению, в исследуемой автором литературе не нашлось упоминания о блочных алгоритмах для разностного решения уравнения переноса. В связи с этим, в данной работе, впервые была использована идея блочности для решения данного уравнения.

Таким образом, целью данной работы является синтез и исследование блочных алгоритмов разностного решения одномерного уравнения переноса. Для достижения данной цели решаются следующие задачи:

1. Построение разностной схемы пригодной для использования в ходе вычислений по блочному алгоритму.
2. Синтез блочного алгоритма на основе построенной схемы.
3. Исследование свойств новой схемы и нового алгоритма.

2. Смешанная разностная схема со сдвигом

В основу данной работы было положено уравнение переноса [7].

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = F(x, t) \tag{1}$$

Для численного решения уравнения переноса (1) составим новую смешанную разностную схему на основе комбинаций известной явной (2) [8] и неявной (3) [8] конечно-разностной схемы для краевой задачи математической физики. Для этого наложим на область вычислительного эксперимента

$$D = \{(x, t) : 0 \leq x \leq L; 0 \leq t \leq T\}, \tag{2}$$

следующую сеточную область [9]:

$$D_h = \left\{ (x_k, t_n) : x_k = k \cdot h, k = \overline{0, N_x}, h = \frac{L}{N_x - 1}; t_n = \tau \cdot n, n = \overline{0, N_t}, \tau = \frac{T}{N_t} \right\}, \tag{3}$$

где h – шаг по пространству, τ – шаг по времени, N_x – дискретизация по пространству, N_t – дискретизация по времени.

Новая схема выглядит следующим образом:

$$\begin{cases} \frac{u_k^n - u_k^{n-1}}{\tau} + a \frac{u_k^{n-1} - u_{k-1}^{n-1}}{h} = f_k^{n-1}; \\ u_k^0 = \varphi_k; \\ u_0^n = \varphi^n. \end{cases} \tag{4}$$

$$\begin{cases} \frac{u_k^n - u_k^{n-1}}{\tau} + a \frac{u_k^n - u_{k-1}^n}{h} = f_k^{n-1}; \\ u_k^0 = \varphi_k; \\ u_0^n = \varphi^n. \end{cases} \tag{5}$$

Где (4) решается для вычисления узлов на нечетных слоях сеточной области по времени для нечетных узлов по пространству и на четных слоях сеточной области по времени для четных узлов по пространству, а (5) для вычисления узлов на нечетных слоях сеточной области по времени для нечетных узлов по пространству и на четных слоях сеточной области по времени для четных узлов по пространству.

Поясним предложенную схему графически. Пусть на нечетных слоях сеточной области по времени для нечетных узлов по пространству для вычисления значений сеточной функции применяется явный дифференциальный шаблон, рисунок 1, на четных – неявный, рисунок 2. Для четных слоев сеточной области по времени на четных узлах по пространству применяется явный дифференциальный шаблон, рисунок 3, на нечетных – неявный, рисунок 4.

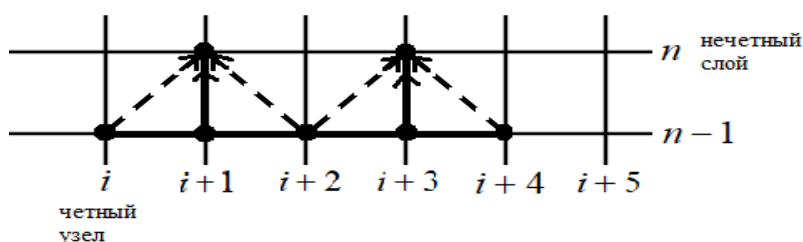


Рисунок 1. Явный дифференциальный шаблон на нечетных слоях сеточной области по времени для нечетных узлов по пространству.

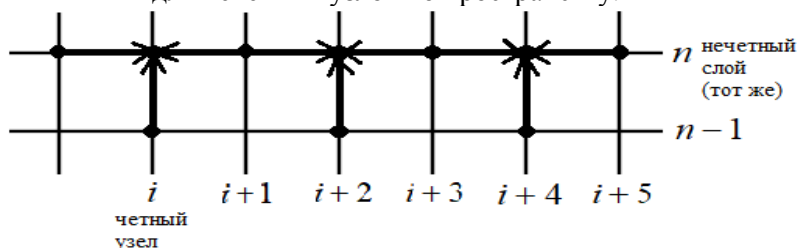


Рисунок 2. Неявный дифференциальный шаблон на нечетных слоях сеточной области по времени для нечетных узлов по пространству.

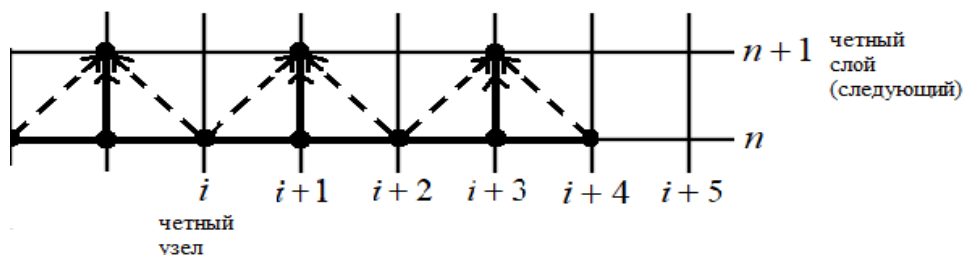


Рисунок 3. Явный дифференциальный шаблон на четных слоях сеточной области по времени для четных узлов по пространству.

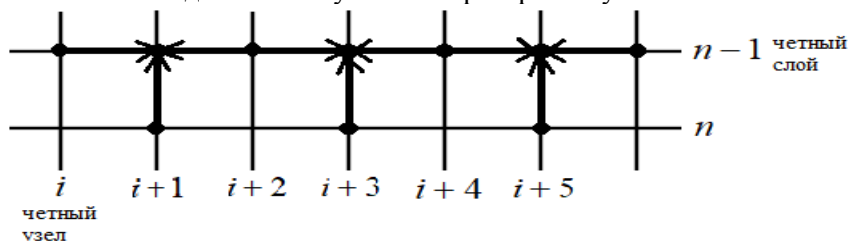


Рисунок 4. Неявный дифференциальный шаблон на четных слоях сеточной области по времени для четных узлов по пространству.

Таким образом, новая разностная схема по-разному применяет два простейших дифференциальных шаблона, явный и неявный, на разных слоях по пространству и времени. Вследствие чего автор называет ее смешанной разностной схемой со сдвигом.

3. Блочный алгоритм

Далее будет представлен блочный алгоритм разностного решения, уравнения математической физики – уравнения переноса, на основе смешанной разностной схемы со сдвигом из предыдущего пункта.

Алгоритмическая реализация данного блочного алгоритма на языке MATLAB выглядит следующим образом.

```

01   for t=1:delta:Nt
02       kr=n+1;
03       for j=2:2:delta
04           U(3:2:kr)=c1*U(3:2:kr)+c2*U(2:2:kr-1);

```

```

05         U(2:2:kr)=c3*U(2:2:kr)+c4*U(1:2:kr-1);
06         U(2:2:kr-1)=c1*U(2:2:kr-1)+c2*U(1:2:kr-2);
07         U(3:2:kr-1)=c3*U(3:2:kr-1)+c4*U(2:2:kr-2);
08         kr=kr-2;
09     end
10     for k=2:p-1
11         kl=(k-1)*n+2; kr=kl+n-1;
12         for j=2:2:delta
13             U(kl+1:2:kr)=c1*U(kl+1:2:kr)+c2*U(kl:2:kr-1);
14             U(kl:2:kr)=c3*U(kl:2:kr)+c4*U(kl-1:2:kr-1);
15             U(kl:2:kr-1)=c1*U(kl:2:kr-1)+c2*U(kl-1:2:kr-2);
16             U(kl-1:2:kr-1)=c3*U(kl-1:2:kr-1)+c4*U(kl-2:2:kr-2);
17             kl=kl-2; kr=kr-2;
18         end
19     end
20     kl=(p-1)*n+2;
21     for j=2:2:delta
22         U(kl+1:2:Nx)=c1*U(kl+1:2:Nx)+c2*U(kl:2:Nx-1);
23         U(kl:2:Nx)=c3*U(kl:2:Nx)+c4*U(kl-1:2:Nx-1);
24         U(kl:2:Nx)=c1*U(kl:2:Nx)+c2*U(kl-1:2:Nx-1);
25         U(kl-1:2:Nx)=c3*U(kl-1:2:Nx)+c4*U(kl-2:2:Nx-1);
26         kl=kl-2;
27     end
28 end

```

Внешний цикл (строки 01, 28) необходим для перебора всех временных слоев сеточной области, однако сейчас в его теле перебираются не пары слоев, а блоки высотой d . Первый вложенный цикл (строки 03, 09) отвечает за действия внутри первого треугольного блока. Переменная kr в нем ограничивает блок справа, ее значение уменьшается с увеличением номера временного слоя внутри блока - j . Третий вложенный цикл (строки 21, 27) связан с вычислением внутри последнего блока, также треугольного. Переменная kl в нем ограничивает блок слева. Во втором цикле (строки 10, 19) перебираются блоки в форме параллелограммов, общим количеством $p-2$. Каждому (с номером k) из них соответствует вложенная циклическая конструкция (строки 12, 18). Переменные kl , kr по-прежнему ограничивают левую и правую границы блоков соответственно.

В телах вложенных циклов (строки 04-07, 13-16 и 22-25) производятся вычисления внутри блока по смешанной разностной схеме со сдвигом. Сдвигу соответствует наклон сторон параллелограмма, обеспечивающий возможность вычисления значений сеточной функции в левых узлах каждого не треугольного блока на любом временном слое по неявному шаблону. Значения в правых узлах находятся по явному. При этом отпадает необходимость в задании дополнительного массива для хранения промежуточных значений сеточных функций. Ширина блока m всегда выбирается четной в силу смешанного характера схемы. Общее число блоков на одном блочном временном слое определяется как $p = \text{fix}((1-2)/m) + 1$. Если количество блочных временных слов N/d оказывается не целым, то алгоритм нуждается в небольшой доработке, связанной с отдельной организацией вычислений на верхнем блочном слое, высота которого меньше d .

3. Ускорение вычислений по блочному алгоритму

Предложенный блочный алгоритм нуждается в тщательном исследовании. Здесь же автор начинает его исследование.

При помощи метода вычислительного эксперимента найдем такие параметры блочности δ – высота блока, n – ширина блока при которых время вычислений будет наименьшим. В

эксперименте 1 будем искать лучший блочный параметр n . В эксперименте 2 будем искать лучший блочный параметр δ .

В эксперименте 1, после достижения отметки $n=500$, увеличение ширины блока – n увеличивает время выполнения блочного алгоритма. Таким образом, самый лучший результат дал блок с шириной $n=500$.

В эксперименте 2, увеличение высоты блока – δ уменьшает время выполнения блочного алгоритма на основе смешанной разностной схемы со смещением до тех пор, пока $\delta = 40$. После достижения отметки $\delta = 40$ время выполнения начинает увеличиваться. Таким образом, самый лучший результат дал блок с высотой $\delta = 40$.

Эксперименты проводились на следующей аппаратной и системной базе: процессор Intel Core: i7-4700, работающий на частоте 3,4 GHz., оперативная память 32 Гб, операционная система CentOS 7.2, компилятор gfortran 5.4.

В ходе экспериментального исследования программной реализации блочного алгоритма продемонстрирована его эффективность. В частности, при параметрах $N_t = 120, N_x = 250 \times 10^6, \delta = 10, n = 500$ ускорение по сравнению с тривиальным вариантом на основе смешанной разностной схемы смещением составило 1.5 раза, а по сравнению с тривиальным вариантом на основе явной разностной схемы 2 раза.

4. Заключение

В ходе выполнения работы был синтезирован и исследован блочный алгоритм разностного решения одномерного уравнения переноса.

При этом:

1) Построена смешанная разностная схема со смещением пригодная для использования в ходе вычислений по блочному алгоритму. Указанная схема предположительно характеризуется абсолютной устойчивостью и условной сходимостью при $h_t / h_x \rightarrow 0$.

2) Синтезирован блочный алгоритм на основе построенной схемы. Особенностью данного алгоритма является отличие в линейных размерах блоков по горизонтали и вертикали от известных алмазных алгоритмов. Указанное отличие позволяет гибко управлять вычислительным процессом.

3) Продемонстрирована эффективность алгоритма, и ускорение, которое дает данный алгоритм.

5. Литература

- [1] Марчук, Г.И. Проблемы вычислительной техники и фундаментальные исследования / Г.И. Марчук, В.Е. Котов. – М.: Наука, 1979. – 176 с.
- [2] Деммель, Дж. Вычислительная линейная алгебра. – М.: Мир, 2001. – 77 с.
- [3] 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 ма-тематики им. М.В. Келдыша РАН, 2015. – 16 с.
- [4] Takeshi, M. Automatic Parameter Tuning of Three-Dimensional Tiled FDTD Kernel // High Performance Computing for Computational Science (VECPAR). – 2014. – Vol. 8969. – P. 284-297.
- [5] Perpelkina, A.Yu. Diamond Torre Algorithm for High-Performance Wave Modeling / A.Yu. Perpelkina, V.D. Levchenko // Keldysh Institute Preprints. – 2015. – Vol. 18. – P. 20.
- [6] Головашкин, Д.Л. Блочный алгоритм совместного разностного решения уравнений Даламбера и Максвелла // Оптический журнал. – 2003. – Т. 70, № 8. – С. 70-74.
- [7] Самарский, А.А. Теория разностных схем. – М.: Наука, 1977. – 288 с.
- [8] Самарский, А.А. Введение в теорию разностных схем. – М.: Наука, 1971. – 553 с.
- [9] Калиткин, Н.Н. Численные методы. – М.: Наука, 1987. – 512 с.

The synthesis of the block algorithm for the difference solution of transport equation

Z.R. Sunagatov¹

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. In the modern world computers often spend much less time on calculating arithmetic operations than on communication between different areas of memory. Moreover, the time of these communications determines the total computing time. The block algorithm idea takes into this feature of computers. The use of block algorithm for solving differential equations has begun recently. Unfortunately, in the literature studied by the author there was no mention of block algorithms for the difference solution of transport equation. Therefore, in this work, the idea of block algorithm was first used to solve this equation. In the course of this work, a block algorithm for the differential solution of a one-dimensional transport equation based on a mixed difference scheme with shift was synthesized and investigated. The created scheme is suitable for use in the course of computations by a block algorithm. During the experimental study of the software implementation of the block algorithm, its effectiveness was studied. Acceleration compared to the trivial version based on a mixed difference scheme with an shift was 1.5 times. And compared to the trivial version based on an explicit difference scheme was 2 time.