

# Блочные алгоритмы решения сеточных уравнений Zheng/Chen/Zhang

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

<sup>1</sup> ИСОИ РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН,  
443001, Россия, г. Самара, ул. Молодогвардейская, д. 151,

<sup>2</sup> Самарский национальный исследовательский университет имени академика С.П. Королёва,  
443086, Россия, г. Самара, Московское шоссе, д. 34

## Аннотация

Работа посвящена синтезу блочных алгоритмов FDTD-метода для расчетов по неявной разностной схеме Zheng/Chen/Zhang. Существенное внимание уделяется экспериментальному исследованию построенных алгоритмов, выявлению особенностей организации блочных вычислений по неявным сеточным уравнениям. Эффективность предложенных подходов подтверждена шестикратным ускорением вычислений.

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

**Цитирование:** Головашкин, Д.Л. Блочные алгоритмы решения сеточных уравнений Zheng/Chen/Zhang / Д.Л. Головашкин, Н.Д. Морунов, Л.В. Яблокова // Компьютерная оптика. – 2021. – Т. 45, № 3. – С. 461-468. – DOI: 10.18287/2412-6179-CO-837.

**Citation:** Golovashkin DL, Morunov ND, Yablokova LV. Block algorithms to solve Zheng/Chen/Zhang's finite-difference equations. Computer Optics 2021; 45(3): 461-468. DOI: 10.18287/2412-6179-CO-837.

## Введение

Развитие вычислительной оптики и электродинамики в XXI веке неразрывно связано с FDTD-методом (Finite-Difference Time-Domain), широко применяемым [1, 2] для моделирования разнообразных оптических и электродинамических явлений в рамках строгой теории дифракции. С точки зрения теории разностных схем принято различать два подхода внутри FDTD-метода, основанных на разработке явных и неявных схем. К достоинствам первого следует отнести простоту понимания и реализации, важную для исследователей, далеких от вычислительной математики (ведь метод предназначен для физиков); второго – абсолютную устойчивость, столь ценимую специалистами-математиками. Длительное время упомянутое разделение по специализации сдерживало развитие неявных разностных схем; так, между публикациями первых работоспособных явной [3] и неявной [4] схем прошло целых 33 года. Со временем необходимость проведения исследований в тех областях оптики и электродинамики, где ограничения, налагаемые критерием Куранта, представляются чрезмерными [1], обусловила развитие математического аппарата неявных схем для моделирования распространения излучения: в средах с частотной дисперсией [5], плазме [6], тонких ферромагнитных пленках [7] и др.

Суждение о высокой вычислительной сложности FDTD-метода в полной мере относится и к его неявным схемам. Практически сразу с момента их появления основным способом преодоления данной проблемы являлась организация вычислений на суперкомпьютерах: от легендарных Стау в начале века [8] до использующих графические процессоры в настоящее время [9]. Вместе с тем получают актуальность другие, более доступные широкому кругу исследова-

телей способы сокращения длительности вычислений по FDTD-методу, в частности, разработка и реализация блочных алгоритмов. Для явных схем FDTD-метода такие алгоритмы появились относительно недавно [10, 11], хотя сам прием составления блочных алгоритмов для снижения вычислительной сложности имеет длительную историю, пусть и в другой предметной области – матричных вычислениях [12].

Задачей авторов настоящей работы была демонстрация возможности синтеза различных блочных алгоритмов для неявной разностной схемы FDTD-метода, чего ранее в вычислительной практике не делалось. В начале нового направления исследований принято решение ограничиться одномерным случаем классической схемы Zheng/Chen/Zhang [13], который при успехе далее можно развивать до многомерного.

## 1. Векторные алгоритмы решения сеточных уравнений Zheng/Chen/Zhang

Ориентируясь на материал фундаментальной монографии [1], разностную схему Zheng/Chen/Zhang в одномерном случае можно представить следующим образом. Этап I – Переход со слоя  $n$  на слой  $n + 1/2$ :

$$\epsilon_0 \frac{E_{x_k}^{n+1/2} - E_{x_k}^n}{h_t / 2} = - \frac{H_{y_{k+1/2}}^n - H_{y_{k-1/2}}^n}{h_z}; \quad (1)$$

$$\mu_0 \frac{H_{y_{k+1/2}}^{n+1/2} - H_{y_{k+1/2}}^n}{h_t / 2} = - \frac{E_{x_{k+1}}^n - E_{x_k}^n}{h_z}. \quad (2)$$

Этап II – переход со слоя  $n + 1/2$  на слой  $n + 1$ :

$$\epsilon_0 \frac{E_{x_k}^{n+1} - E_{x_k}^{n+1/2}}{h_t / 2} = - \frac{H_{y_{k+1/2}}^{n+1} - H_{y_{k-1/2}}^{n+1}}{h_z}; \quad (3)$$

$$\mu_0 \frac{H_{y_{k+1/2}}^{n+1} - H_{y_{k+1/2}}^{n+1/2}}{h_t / 2} = - \frac{E_{x_{k+1}}^{n+1} - E_{x_k}^{n+1}}{h_z}. \quad (4)$$

Диэлектрическая и магнитная проницаемости для простоты приняты равными 1 и в выражениях (1–4) не фигурируют,  $\epsilon_0$ ,  $\mu_0$  – электрическая и магнитная постоянные. Сеточная проекция  $E_{x_k}^n$  напряженности электрического поля на ось  $X$  определена в узлах

$$\{(t_n, z_k): t_n = n \cdot h_t, n = 0, 1/2, 1, 3/2, \dots, N; h_t = T/N; z_k = (k-1) \cdot h_z, k = 1, 2, \dots, K; h_z = L/(K-1)\},$$

а сеточная проекция  $H_{y_{k+1/2}}^n$  напряженности магнитного поля на ось  $Y$  определена в узлах

$$\{(t_n, z_{k+1/2}): t_n = n \cdot h_t, n = 0, 1/2, 1, 3/2, \dots, N; z_{k+1/2} = (k-1/2) \cdot h_z, k = 1, 2, \dots, K-1\}.$$

Значения  $T$  и  $L$  задают протяженность вычислительной области по времени и пространству,  $h_t$  и  $h_z$  – соответствующие шаги дискретизации,  $n$  и  $k$  – индексы дискретизации.

В качестве начального условия примем отсутствие поля при  $t=0$  соответственно,  $E_{x_k}^0$  и  $H_{y_{k+1/2}}^0$  положим равными 0 при любых  $k$ . На правой границе установим электрическую стенку  $E_{x_K}^n = 0$ ; на левой – «жесткий» источник [1]  $E_{x_1}^n = \sin(2\pi c h_t n)$ , где  $c$  – скорость света в свободном пространстве. При выбранном способе наложения сеточной области необходимости в задании магнитного поля на ее краях не возникает.

Отметим сходства и отличия схемы Zheng/Chen/Zhang из [1] от классической схемы Yee [3]. В обоих случаях для определения каждой сеточной проекции напряженности полей используются свои узлы и их подмножества не пересекаются. Вместе с тем таких узлов в схеме Zheng/Chen/Zhang задается вдвое больше – в отличие от Yee здесь обе сеточные проекции вычисляются как на слое  $n$ , так и на слое  $n+1/2$ . Указанное обстоятельство определяет двухэтапный переход от слоя  $n$  к слою  $n+1$ . Сеточные уравнения первого этапа (1), (2) следует признать явными (как в Yee), уравнения (3), (4) второго этапа очевидно следует отнести к неявным (в отличие от Yee). Последнее обуславливает как преимущество схемы Zheng/Chen/Zhang – абсолютную устойчивость, так и ее недостаток – необходимость решения систем линейных алгебраических уравнений в ходе производства вычислений.

Определяя вид системы, появляющейся на этапе II, дважды подставим (4) в (3), выразив из (4)  $H_{y_{k+1/2}}^{n+1}$  и  $H_{y_{k-1/2}}^{n+1}$ , получив в итоге сеточное уравнение

$$\begin{aligned} & -\frac{h_t^2}{4h_z^2 \epsilon_0 \mu_0} E_{x_{k-1}}^{n+1} + \left(1 + \frac{h_t^2}{2h_z^2 \epsilon_0 \mu_0}\right) E_{x_k}^{n+1} - \frac{h_t^2}{4h_z^2 \epsilon_0 \mu_0} E_{x_{k+1}}^{n+1} = \\ & = E_{x_k}^{n+1/2} - \frac{h_t}{2h_z \epsilon_0} (H_{y_{k+1/2}}^{n+1/2} - H_{y_{k-1/2}}^{n+1/2}). \end{aligned} \quad (5)$$

Варьируя  $k$  от 2 до  $K-1$ , получаем систему трехдиагонального вида, после решения которой этап II завершается подстановкой найденных значений се-

точной напряженности электрического поля на слое  $n+1$  в формулу (4).

Признавая вслед за авторами работы [9] уместность метода Томаса (известен как метод прогонки в отечественной литературе) для решения набора таких систем в многомерном случае, отметим следующую особенность рассматриваемого одномерного. На этапе II возникает лишь одна система и векторизация вычислений по ней при использовании метода Томаса невозможна, как впрочем и для любого другого прямого метода [14]. Следовательно, решение (5) необходимо проводить итерационными методами, например Якоби или Гаусса–Зейделя. И хотя в вычислительной математике их считают «простыми», однако ранее для синтеза векторных [14], а теперь и блочных [15] алгоритмов принято использовать именно их; возможно, именно в силу упомянутой «простоты», допускающей их эффективную векторизацию и удачный переход к блочности. Сходимость обоих методов в данном случае обеспечивается в соответствии с теоремами 3.1.1 и 3.2.1 из [14], строгим диагональным преобладанием матрицы системы, образованной левой частью (5).

Запишем следующий векторный алгоритм на языке Фортран (в отличие от Си допускающем векторную нотацию), реализующий вычисления по рассматриваемой разностной схеме с использованием метода Якоби.

#### Алгоритм 1.

```

01 do n=1,N
02   Exx=Ex;
03   Ex(2:Nz-1)=Ex(2:Nz-1)-c4*(Hy(2:Nz-1)-
                                Hy(1:Nz-2));
04   Hy=Hy-c1*(Exx(2:Nz)-Exx(1:Nz-1));
05   Ex(1)=sin(c5*n);
06   b=c6*(c4*(Hy(1:Nz-2)-Hy(2:Nz-1))+
                                Ex(2:Nz-1));
07   do g=1,M
08     Ex(2:Nz-1)=c2*(Ex(1:Nz-2)+Ex(3:Nz))+b;
09   end do
10   Hy=Hy-c1*(Ex(2:Nz)-Ex(1:Nz-1));
11 end do

```

В теле цикла 01–11 производится переход со слоя  $n$  на слой  $n+1$  в два этапа: I, строчки 02–04, и II, строчки 06–10. При этом вычисления по формуле (1) производятся в строчке 03; по (2) – в строчке 04; по (5) – в 07–09, где в цикле реализуются  $M$  итераций метода Якоби для решения системы; по (4) – в строчке 10. В строчке 02 производится дублирование вектора  $E_x$  в  $E_{xx}$  с целью предотвращения преждевременного затирания значений  $E_{x_k}^n$ , необходимых при реализации (2). В 05 задается «жесткий источник», в 06 формируется правая часть системы. Константы  $c1$ ,  $c4$ ,  $c3$ ,  $c2$ ,  $c6$  и  $c5$  определяются до алгоритма как  $h_t/(2h_z \mu_0)$ ,  $h_t/(2h_z \epsilon_0)$ ,  $c1*c4$ ,  $c3/(1+2*c3)$ ;  $1/(1+2*c3)$  и  $2\pi c h_t$  соответственно.

На основе Алгоритма 1 предложим Алгоритм 2, также реализующий метод Якоби, однако несколько иначе. Пусть вместо цикла 07–09 будет записано:

```

07 do g=1,M/2
08   Exx(2:Nz-1)=c2*(Ex(1:Nz-2)+Ex(3:Nz))+b;
09   Ex(2:Nz-1)=c2*(Exx(1:Nz-2)+Exx(3:Nz))+b;
10 end do

```

Тогда строки 10–11 Алгоритма 1 получают номера 11–12 в Алгоритме 2. В ходе нечетных итераций по  $g$ , строка 08, приближенное решение системы будет записываться в вектор  $E_{xx}$  (который ранее уже был введен, пусть и с другой целью), в ходе четных, строка 09, – в  $E_x$ . Предложенная модификация, как будет показано в ходе экспериментального исследования, позволит сэкономить память и ускорить вычисления.

При векторизации метода Гаусса–Зейделя принято [14] использовать его вариант, основанный на красно-черном упорядочивании. Тогда для обсуждаемой разностной схемы можно составить третий векторный алгоритм (Алгоритм 3), отличающийся от первого небольшим фрагментом. Вместо строки 08, в нем будет записано две следующих:

```

08 Ex(3:Nz-2:2)=c2*(Ex(2:Nz-3:2)+Ex(4:Nz-1:2))+
    b(2:Nz-3:2);
09 Ex(2:Nz-1:2)=c2*(Ex(1:Nz-2:2)+Ex(3:Nz:2))+
    b(1:Nz-2:2);

```

В первой реализуется расчет итерационного приближения в «красных» (нечетных) позициях массива через использование значений на предыдущей итерации, во второй – в «черных» (четных) позициях через использование только что найденных значений на текущей итерации. Соответственно, строки 09–11 Алгоритма 1 получают номера 10–12 в Алгоритме 3. Следует различать итерационные приближения (в методах Якоби или Гаусса–Зейделя) для решения (5), реализуемого в цикле по  $s$ , и переход на следующий временной слой по формулам (1, 2, 5) и (4), в цикле по  $n$ .

## 2. Блочные алгоритмы решения сеточных уравнений Zheng/Chen/Zhang с использованием метода Якоби

Поясняя смысл и особенности блочных алгоритмов, предостережем читателя от путаницы в терминологии. Далее речь пойдет не о «блочных методах» из [14], которые предназначены для работы с блочными матрицами (таких в нашем изложении нет), а о приеме организации вычислительного процесса, позволяющем сократить длительность расчетов за счет снижения коммуникационных издержек. В классических работах по теории разностных схем, вышедших как в период становления этой теории [16], так и значительно позже [17], вычислительная сложность численного метода определяется как количество скалярных арифметических операций по нему и только. Однако в соседней предметной области, вычислитель-

ной линейной алгебре, самое серьезное внимание (и это отражено в фундаментальных монографиях [12, 18]) всегда уделялось учету коммуникаций внутри иерархической структуры памяти ЭВМ и синтезу алгоритмов (их и называют блочными), предназначенных для минимизации таких коммуникаций. Основной идеей блочности является максимально полное использование данных, загруженных в верхний уровень иерархической памяти. Один из смыслов настоящей работы ее авторы видят в распространение практики составления блочных алгоритмов на теорию разностных схем.

Обращаясь к решению сеточных уравнений (1, 2, 5, 4), зададимся целью перейти от Алгоритма 2 к новому путем фрагментации сеточной области (рис. 1), допускающей раздельное решение (5) по методу Якоби для каждого фрагмента (блока). В этом случае значения  $E_{x_k}^{n+1}$  формируются не одновременно для всех  $k$  (как в строчках 07–10 Алгоритма 2), а лишь для определенного интервала. Такого, что все необходимые данные помещаются в верхний уровень иерархической памяти.

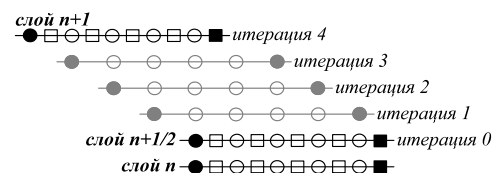


Рис. 1. Блок сеточной области при использовании метода Якоби

Так, на рис. 1 изображены узлы сеточной области и значения итерационных приближений, составляющие определенный блок. Левее – их узлы и значения, относящиеся к предыдущему блоку, правее – к следующему. Представлен частный случай для ширины блока  $dl=4$  и  $M=4$ . Черные окружности обозначают узлы, на которых определены сеточные проекции электрического поля, границы квадратов – магнитного. Серым цветом выделены значения (кроме нулевого и последнего) итерационных приближений. Круги и квадраты соответствуют узлам и значениям из соседних блоков. Отметим, что нулевое итерационное приближение составляется из значений  $E_{x_k}^{n+1/2}$ , а последнее задает  $E_{x_k}^{n+1}$ .

Смещение границ блока на одну позицию влево при переходе на следующую итерацию обусловлено необходимостью соблюдения информационных зависимостей на пространстве итерации циклической конструкции 07–10 Алгоритма 2. Хранение значений на четных и нечетных итерациях в различных массивах ( $E_x$  и  $E_{xx}$ ) также необходимо для удовлетворения упомянутых информационных зависимостей. В работе [15] предлагается соблюдать зависимости за счет хранения каждого итерационного приближения в отдельном массиве, что приводит к значительному увеличению объема занимаемой памяти. Следующий блочный алгоритм лишен этого недостатка.

**Алгоритм 4.**

```

01 do n=1,N
02   { ЛЕВАЯ ТРАПЕЦИЯ }
03   do q=2,p-1
04     w1=(q-1)*dl+2; wr=q*dl+1;
05     Exx(w1:wr)=Ex(w1:wr);
06     Ex(w1:wr)=Ex(w1:wr)-c4*(Hy(w1:wr)-
                                Hy(w1-1:wr-1));
07     Hy(w1-1:wr-1)=Hy(w1-1:wr-1)-
                                c1*(Exx(w1:wr)-Exx(w1-1:wr-1));
08     b(w1-2:wr-2)=c6*(c4*(Hy(w1-2:wr-2)-
                                Hy(w1-1:wr-1))+Ex(w1-1:wr-1));
09     do g=1,M/2
10       w1=w1-1; wr=wr-1; Exx(w1:wr)=
c2*(Ex(w1-1:wr-1)+Ex(w1+1:wr+1))+b(w1-1:wr-1);
11       w1=w1-1; wr=wr-1; Ex(w1:wr)=
c2*(Exx(w1-1:wr-1)+Exx(w1+1:wr+1))+b(w1-1:wr-1);
12     end do
13     Hy(w1-1:wr-1)=Hy(w1-1:wr-1)-
                                c1*(Ex(w1:wr)-Ex(w1-1:wr-1));
14   end do
15   { ПРАВАЯ ТРАПЕЦИЯ }
16 end do
    
```

Вычисления в центральных блоках, имеющих форму, близкую к параллелепипеду, расписаны подробно (строки 03 – 14); первый и последний блок по форме напоминают трапецию, о вычислениях в них для краткости лишь упоминается (строки 02 и 15). Сравнивая Алгоритмы 2 и 4, отметим основные отличия.

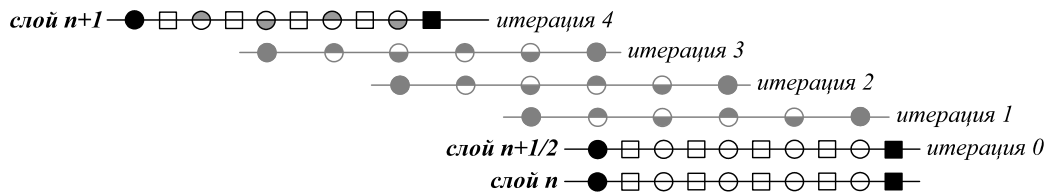


Рис. 2. Блок сеточной области при использовании метода Гаусса–Зейделя

На рис. 2 черные окружности обозначают узлы, на которых определены сеточные проекции электрического поля, границы квадратов – магнитного. Серым цветом выделены значения (кроме нулевого) итерационных приближений. Круги и квадраты соответствуют узлам и значениям из соседних блоков. Закрашенные сверху окружности – «черные» позиции массивов, снизу – «красные».

Действительно, при переходе к новому итерационному приближению (за исключением первого) границы блока теперь сдвигаются на две позиции влево (строчка 12 Алгоритма 6), что вызвано необходимостью согласования позиций: «красные» должны быть под «красными», «черные» – под «черными». Начинается блок всегда с «черной» позиции, а заканчивается «красной»; это обусловлено состоянием вычислительного процесса, при котором значения на текущей итерации слева от блока уже вычислены, а справа еще нет. При работе с «красными» позициями используются значения, найденные только на предыдущей итерации, с «черными» – на текущей.

Векторные операции по формулам (1, 2, 5) и (4) теперь проводятся не для всей сеточной области, а для заданного фрагмента с границами w1 (левая граница) и wr (правая). Границы фрагмента (блока) смещаются влево с увеличением номера итерации (начало строк 10 и 11). Сами блоки перебираются слева направо в цикле по q (строки 03 – 14), где q – номер блока. Общее количество блоков определяется до выполнения обсуждаемых алгоритмов по правилу  $p = K / dl$ .

Можно избежать хранения массивов Exx и b на всей сеточной области, ограничившись лишь той их частью, что необходима при работе с текущим блоком. Ведь для начала вычислений в каждом блоке необходимы лишь массивы Ex и Hy, а Exx и b формируются далее с их использованием. Оформим это в Алгоритме 5, характеризующемся в силу указанного обстоятельства вдвое меньшими (ведь  $dl \ll K$ ) затратами памяти по сравнению с Алгоритмами 2 и 4.

**3. Блочные алгоритмы решения сеточных уравнений Zheng/Chen/Zhang с использованием метода Гаусса–Зейделя**

Метод Гаусса–Зейделя не только выгодно отличается от метода Якоби лучшей сходимостью (в большинстве случаев), но и служит основой для построения других, более развитых итерационных методов [14]. Декомпозиция сеточной области на блоки при его применении несколько изменится (рис. 2).

Перечисленные особенности, а также необходимость в двух векторных операциях при переходе к следующему итерационному приближению (вместо одной в методе Якоби) обуславливают отличия Алгоритма 6, реализующего блочный вариант разностного решения с использованием метода Гаусса–Зейделя, от Алгоритма 4.

**Алгоритм 6.**

```

01 do n=1,N
02   { ЛЕВАЯ ТРАПЕЦИЯ }
03   do q=2,p-1
04     w1=(q-1)*dl+2; wr=q*dl+1;
05     Exx(w1+1:wr+1)=Ex(w1+1:wr+1);
06     Ex(w1+1:wr+1)=Ex(w1+1:wr+1)-
                                c4*(Hy(w1+1:wr+1)-Hy(w1:wr));
07     Hy(w1:wr)=Hy(w1:wr)-c1*(Exx(w1+1:wr+1)-
                                Exx(w1:wr));
08     b(w1-1:wr-1)=c6*(c4*(Hy(w1-1:wr-1)-
                                Hy(w1:wr))+Ex(w1:wr));
09   end do
    
```

```

10      Ex(wl+1:wr:2)=c2*(Ex(wl:wr-1:2)+
      Ex(wl+2:wr+1:2))+b(wl:wr-1:2);
11      Ex(wl:wr-1:2)=c2*(Ex(wl-1:wr-2:2)+
      Ex(wl+1:wr:2))+b(wl-1:wr-2:2);
12      wl=wl-2; wr=wr-2;
13  end do
14      Hy(wl+1:wr+1)=Hy(wl+1:wr+1)-
      c1*(Ex(wl+2:wr+2)-Ex(wl+1:wr+1));
15  end do
16  { ПРАВАЯ ТРАПЕЦИЯ }
17 end do

```

Как и в предыдущем параграфе, здесь тоже уместен разговор о построении нового алгоритма (Алгоритма 7), отличающегося от только что представленного двукратным сокращением затрат памяти за счет хранения только тех частей массивов  $E_{xx}$  и  $b$ , что необходимы в ходе вычислений внутри текущего блока.

#### 4. Экспериментальное исследование построенных алгоритмов

Как известно, единственным научным критерием истины является практика. Эксперименты проводились на вычислительной системе, оснащенной процессором Intel Core i3-4150 3,5 ГГц (кэш L1 = 64 Кб, L2 = 512 Кб, L3 = 3 Мб), материнской платой Gigabyte GA-Z97M-DS3H (частота системной шины DMI 5000 МГц), ОЗУ с 12 Гб памяти (DIMM DDR3-1333 МГц) и работающей под управлением операционной системы Ubuntu 18.04.4. Использовался компилятор GCC 7.5.0, ключ компиляции -O3.

Предваряя представление экспериментальных данных, отметим, что рассмотрению подлежала исключительно длительность вычислений по алгоритмам при равных параметрах сеточной области. Проблемы устойчивости, аппроксимации, корректности и сходимости не рассматривались, как выходящие за границы предлагаемой работы (для схемы Zheng/Chen/Zhang все это подробно изложено в [1]). Новизной здесь характеризуются лишь авторские блочные алгоритмы, перечисленные характеристики которых полностью совпадают с соответствующими характеристиками не блочных. В ходе программной реализации каждого блочного алгоритма проводилась проверка совпадения разностных решений в блочном и неблочном случаях. Правильно реализованным признавался лишь тот алгоритм, где такое совпадение было полным.

Целью первой серии экспериментов был подбор сеточной области, обеспечивающей, с одной стороны, достаточно длительные вычисления по неблочным алгоритмам – случайные системные события не должны значимо влиять на результаты; с другой стороны, нет смысла в избыточно большом времени расчетов – ускорение блочных алгоритмов не зависит от числа временных слоев сеточной области. Выбор был остановлен на следующих значениях:  $N=50$ ,  $K=1e8$ .

Количество итераций  $M=16$  в методах Якоби и Гаусса–Зейделя обеспечивало приемлемую точность разностного решения на заданной сеточной области. Результаты первой серии экспериментов представлены в табл. 1.

Табл. 1. Результаты первой серии экспериментов

Алгоритм	Длительность вычислений, с	Занимаемая память, Гб
1	268,06	от 1,5 до 1,9
2	168,72	1,5
3	232,39	1,5

Как видно, Алгоритм 2 характеризуется меньшим временем расчетов по сравнению с Алгоритмом 1 за счет лучшей работы с памятью. При вычислениях по Алгоритму 1 объем выделяемой памяти непрерывно варьировался в указанных пределах (табл. 1), превышая ожидаемое значение в 1,5 Гб (четыре массива длиной  $K$  действительных чисел одинарной точности). Очевидно, это было предусмотрено при компиляции для предотвращения возможности преждевременного затирания значений массива  $E_x$  в строке 08 Алгоритма 1, для чего массив автоматически дублировался. Во втором Алгоритме необходимости в таком дублировании нет. Поэтому блочные Алгоритмы 4 и 5 строились на его основе.

Отметим, что расчеты по Алгоритму 3 производились заметно дольше, чем по Алгоритму 2 при тех же затратах памяти и одинаковом количестве скалярных арифметических операций. Причинами тому могут служить особенности производства векторных операций в цикле по  $g$  третьего Алгоритма: таких операций две на одну итерацию метода Гаусса–Зейделя (вместо одной на одну итерацию Якоби) и при составлении участвующих в них векторов шаг выборки 2 (вместо шага 1 у Якоби).

В ходе второй серии экспериментов исследуем зависимость длительности расчетов по двум блочным алгоритмам, основанным на методе Якоби, от ширины блока  $dl$  (табл. 2). Остальные параметры остаются прежними.

Табл. 2. Результаты второй серии экспериментов

Алгоритм	Ширина блока	Объем блока	Длительность вычислений, с	Занимаемая память, Мб	Ускорение
4	4e1	896 байт	53,79	1526	3,14
	4e2	6,5 Кб	38,07		4,43
	4e3	62,75 Кб	40,73		4,14
	4e4	625,25 Кб	46,65		3,62
	4e5	6,12 Мб	135,80		1,25
	4e6	61,04 Мб	167,14		1,01
5	4e1	896 байт	48,41	765	3,49
	4e2	6,5 Кб	29,32	765	5,75
	4e3	62,75 Кб	31,81	765	5,30
	4e4	625,25 Кб	38,59	766	4,37
	4e5	6,12 Мб	130,46	775	1,29
	4e6	61,04 Мб	164,89	859	1,02

Под ускорением здесь понимается отношение времени вычислений по Алгоритму 2 к длительности

расчетов по блочным Алгоритмам 4 и 5. Если объем занимаемой памяти в табл. 1 и 2 измерялся по показаниям системного монитора, то объем блока оценивался теоретически с учетом его ширины и количества задействованных массивов. Ширина блока выбиралась из следующих соображений: вычисления в правой и левой трапеции (строки 02 и 16 Алгоритма 4) удобно, хотя и не обязательно, проводить при соблюдении условия  $dl > M$ , когда используется метод Якоби, и  $dl > 2M$ , когда метод Гаусса–Зейделя. Для корректности будущего сравнения ориентируемся на последнее неравенство, выбирая минимальную ширину блока равной 40.

Обращаясь к четвертому столбцу табл. 2, наблюдаем U-образную зависимость длительности вычислений от ширины блока для обоих рассматриваемых алгоритмов. Действительно, при росте значений  $dl$  время расчетов сначала сокращается в силу уменьшения числа блоков и, как следствие, коммуникационных издержек между оперативной памятью и кэш-памятью процессора. С дальнейшим увеличением  $dl$  объем блока выходит за размеры кэш-памяти и эффективность алгоритмов резко падает. Например, при  $dl = 4e4$  (блок еще умещается в кэш) ускорения Алгоритмов 4 и 5 составляют 3,62 и 4,37 соответственно; при  $dl = 4e5$  (объем блока вдвое превосходит емкость уровня  $L3$ ) ускорения резко падают до 1,25 и 1,29; для  $dl = 4e5$  (емкость  $L3$  превышена в 20 раз) эффект от блочности исчезает совсем. Достаточно высокие ускорения при минимальном значении  $dl$  (3,14 и 3,49) объясняются существенной величиной этого значения. Выбрав  $dl$  небольшим (что возможно, но приведет к некоторому усложнению алгоритмов), будем наблюдать и небольшие ускорения, как это демонстрируется в [19] на примере явных разностных схем. Проведя дополнительные эксперименты, авторы установили, что при  $dl = 2e5$  (объем блока совпадает с емкостью  $L3$ ) величины ускорений достаточно велики и составляют 3,47 и 3,94 раза. Учитывая все изложенное, причиной ускорения вычислений однозначно следует считать блочность исследуемых алгоритмов.

Сравнивая блочные алгоритмы между собой, отметим преимущество пятого, наилучшее ускорение которого 5,75; в то время как максимальное ускорение четвертого всего 4,43 раза. Различие, очевидно, вызвано сокращением занимаемой памяти вдвое при переходе от четвертого Алгоритма к пятому. Отметим также увеличение занимаемой памяти у пятого Алгоритма при больших блоках. Испытав алгоритмы для других значениях  $dl$ , авторы нашли наибольшее ускорение Алгоритма 5 равным 6,12 при  $dl = 2000$ . Возвращаясь к общему взгляду на теорию разностных схем, отметим несостоятельность оценки длительности вычислений исключительно по количеству арифметических операций. Количество таких операций в Алгоритмах 2 и 5 одинаково, однако время расчетов отличается в 6 раз. Наибольший вклад в длитель-

ность, как видно из анализа экспериментальных результатов, оказывают не арифметические операции, а работа с памятью, и при построении новых разностных схем важно учитывать это обстоятельство.

Третья серия экспериментов (табл. 3) посвящена исследованию эффективности Алгоритмов 6 и 7, использующих метод Гаусса–Зейделя для решения сеточных уравнений Zheng/Chen/Zhang.

Табл. 3. Результаты третьей серии экспериментов

Алгоритм	Ширина блока	Объем блока	Длительность вычислений, с	Занимаемая память, Мб	Ускорение
6	4e1	896 байт	76,65	1526	3,03
	4e2	6,5 Кб	76,48		3,04
	4e3	62,75 Кб	79,31		2,93
	4e4	625,25 Кб	83,15		2,8
	4e5	6,12 Мб	100,03		2,33
	4e6	61,04 Мб	231,70		1
7	4e1	896 байт	76,45	765	3,04
	4e2	6,5 Кб	72,34	765	3,21
	4e3	62,75 Кб	72,90	765	3,19
	4e4	625,25 Кб	74,57	766	3,12
	4e5	6,12 Мб	98,98	773	2,35
	4e6	61,04 Мб	232,34	841	1

Здесь под ускорением понимается отношение времени вычислений по Алгоритму 3 к длительности расчетов по блочным Алгоритмам 6 и 7.

Отметим характерные отличия результатов из табл. 3 от данных, полученных в ходе предыдущей серии экспериментов.

1. Вид зависимости времени расчетов от ширины блока остался прежним, однако левая ветка U-образного графика выражена слабее. Это связано либо с уменьшением максимально достижимого ускорения (ведь сами значения ускорений при  $dl = 4e1$  в табл. 2 и 3 достаточно близки); либо с особенностями другой зависимости – длительности вычислений от  $M$ , которая в данной работе не исследуется. Безусловно,  $M$  является блочным параметром, задающим высоту блока и влияющим на ускорение; однако любые его изменения скажутся и на точности разностного решения. Поэтому авторы настоящей работы не рекомендуют играть величиной  $M$ , как блочным параметром. Такова отличительная особенность предложенных блочных алгоритмов для неявных разностных схем от алгоритмов для явных (например, из [10, 11]), где можно произвольно варьировать любые параметры блока.

2. При незначительном выходе за емкость кэш-памяти процессора ( $dl = 4e5$ ) происходит снижение ускорения, однако не такое резкое, как в табл. 2. Возможно, это объясняется меньшим (2 из 4) по сравнению с Алгоритмами 4 и 5 (там 3 из 4) количеством массивов, участвующих в итерационном процессе – к этому процессу относится большинство арифметических операций внутри блока. Ес-

ли это так, то теоретическую оценку объема блока в табл. 2 следует умножить на  $3/4$  а в табл. 3 – на  $1/2$ . К сожалению, прямое обращение к кэш-памяти процессора при выполнении программы на языке высокого уровня (впрочем, и на низком ситуация немногим лучше) невозможно, и подтвердить высказанное предположение в ходе прямого эксперимента затруднительно.

3. Значения ускорений для Алгоритмов 6 и 7 при одинаковой ширине блоков различаются не столь существенно (максимальное отличие в 1,15 раз для  $d1=4e4$ ), как для Алгоритмов 4 и 5 (максимальное отличие в 1,3 раза для  $d1=4e2$ ), хотя утверждение о двукратной экономии памяти при переходе к алгоритму с большим номером верно для обеих пар алгоритмов. Впрочем, двукратное сокращение объема занимаемой памяти имеет самостоятельную ценность.

4. Наконец, максимально достигнутое ускорение в 3,21 раза (Алгоритм 7 при  $d1=4e2$ ), хотя и существенно (например, в [10, 11] трехкратное ускорение для блочного алгоритма, связанного со схемой Yee, является наибольшим при задействовании одного вычислительного процесса), однако значительно ниже аналогичной характеристики Алгоритма 5.

Причинами последних двух обстоятельств повидимости являются уже отмеченные при анализе первой серии экспериментов различия Алгоритмов 2 и 3.

Обсуждая развитие предложенного подхода к составлению блочных алгоритмов решения сеточных уравнений Zheng/Chen/Zhang на многомерный случай, следует упомянуть возможность синтеза волновых алгоритмов [20], хорошо зарекомендовавших себя для двумерных явных разностных схем [19]. Кроме того, безусловный интерес представляет реализация блочных алгоритмов решения упомянутых сеточных уравнений на графических процессорах, как это сделано в работе [21] для явной схемы Yee. Задачами такой реализации будет не только ускорение вычислений, но и преодоление ограничения на размер видеопамати графического ускорителя.

### Заключение

Продемонстрирована возможность построения блочных алгоритмов FDTD-метода для организации вычислений по неявной разностной схеме Zheng/Chen/Zhang, характеризующейся абсолютной устойчивостью. Экспериментально подтверждена эффективность предложенных блочных алгоритмов, обеспечивающих ускорение расчетов до шести раз на той же аппаратной и программной базе. Изложены соображения по развитию исследований в выбранной предметной области.

### Благодарности

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

### Литература

1. **Taflove, A.** Computational electrodynamics: The finite-difference time-domain method / A. Taflove, S. Hagness. – Boston: Artech House Publishers, 2005. – 1006 p.
2. **Taflove, A.** Advances in FDTD computational electrodynamics: Photonics and nanotechnology / A. Taflove, A. Oskooi, S.G. Johnson. – Boston: Artech House Publishers, 2013. – 623 p.
3. **Yee, K.S.** Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media / K.S. Yee // IEEE Transactions on Antennas and Propagation. – 1966. – Vol. AP-14. – P. 302-307. – DOI: 10.1109/TAP.1966.1138693.
4. **Namiki, T.** A new FDTD algorithm based on alternating-direction implicit method / T. Namiki // IEEE Transactions on Microwave Theory and Techniques. – 1999. – Vol. 47. – P. 2003-2007.
5. **Xie, G.** A unified 3-D ADI-FDTD algorithm with one-step leapfrog approach for modeling frequency-dependent dispersive media / G. Xie, Z. Huang, M. Fang, X. Wu // International Journal of Numerical Modelling: Electronic Networks, Devices and Fields. – 2019. – Vol. 33, Issue 2. – P. 184940-184949. – DOI: 10.1002/jnm.266610.
6. **Wanjuan, S.** Analysis of electromagnetic wave propagation and scattering characteristics of plasma sheath via high order ADE-ADI FDTD / S. Wanjuan, Z. Hou // Journal of Electromagnetic Waves and Applications. – 2016. – Vol. 30, Issue 10. – P. 1321-1333. – DOI: 10.1080/09205071.2016.1198727.
7. **Yao, Z.** 3D ADI-FDTD modeling of platform reduction with thin film ferromagnetic material / Z. Yao, Y.E. Wang // 2016 IEEE International Symposium on Antennas and Propagation (APSURSI). – 2016. – P. 2019-2020. – DOI: 10.1109/APS.2016.7696716.
8. **Jordan, H.** Experience with ADI-FDTD techniques on the Cray MTA supercomputer / H. Jordan, S. Bokhari, S. Staker, J. Sauer, M. ElHelbawy, M. Picket-May // Proceedings of SPIE. – 2001. – Vol. 4528. – P. 68-76. – DOI: 10.1117/12.434878.
9. **Liu, S.** A multi-GPU accelerated parallel domain decomposition one-step leapfrog ADI-FDTD / S. Liu, B. Zou, L. Zhang, S. Ren // IEEE Antennas and Wireless Propagation Letters. – 2020. – Vol. 19, Issue 5. – P. 816-820. – DOI: 10.1109/LAWP.2020.2981123.
10. **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.
11. **Minami, T.** Automatic parameter tuning of three-dimensional tiled FDTD kernel / T. Minami, M. Hibino, T. Hiraishi, T. Iwashita, H. Nakashima // High Performance Computing for Computational Science VECPAR 2014. – 2014. – P. 284-297. – DOI: 10.1007/978-3-319-17353-5\_24.
12. **Golub, G.H.** Matrix computations / G.H. Golub, C.F. Van Loan. – Baltimore, London: Johns Hopkins University Press, 1996. – 694 p.
13. **Zhen, F.** Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method / F. Zhen, Z. Chen, J. Zhang // IEEE Transactions on Microwave Theory and Techniques. – 2000. – Vol. 48, Issue 9. – P. 1550-1558.
14. **Ортега, Дж.** Введение в параллельные и векторные методы решения линейных систем / Дж. Ортега; пер. с англ. – М.: Мир, 1991. – 368 с.
15. **Zhou, X.** Tiling optimizations for stencil computations / X. Zhou [Electronical Resource]. – 2013. – URL:

- [https://www.ideals.illinois.edu/bitstream/handle/2142/44340/Xing\\_Zhou.pdf](https://www.ideals.illinois.edu/bitstream/handle/2142/44340/Xing_Zhou.pdf) (request date 15.11.2020).
16. Самарский, А.А. Теория разностных схем / А.А. Самарский. – М.: Наука, 1977. – 657 с.
  17. Самарский, А.А. Вычислительная теплопередача / А.А. Самарский, П.Н. Вабищевич. – М.: Едиториал УРСС, 2003. – 784 с.
  18. Деммель, Дж. Вычислительная линейная алгебра / Дж. Деммель; пер. с англ. – М.: Мир, 2001. – 435 с.
  19. 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.
  20. 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.
  21. Закиров, А.В. Алгоритм DiamondTorge и высокопроизводительная реализация FDTD метода для суперкомпьютеров с графическими ускорителями / А.В. Закиров, В.Д. Левченко, А.Ю. Перепёлкина, Я. Земпо // Труды международной конференции Суперкомпьютерные дни в России. – 2016. – С. 80-94.

#### Сведения об авторах

**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 прикладных математики и физики Самарского национального исследовательского университета имени академика С. П. Королева. Область научных интересов: FDTD-метод, векторные и матричные вычисления, математическое моделирование оптических элементов и устройств нанопотоники. E-mail: [dimitriy@ipsiras.ru](mailto:dimitriy@ipsiras.ru).

**Морунов Никита Дмитриевич**, аспирант кафедры прикладных математики и физики Самарского национального исследовательского университета имени академика С. П. Королева. Область научных интересов: разностные схемы, FDTD-метод, векторные и матричные вычисления. E-mail: [niknixad@mail.ru](mailto:niknixad@mail.ru).

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

ГРНТИ: 27.41.19

Поступила в редакцию 16 ноября 2020 г. Окончательный вариант – 23 декабря 2020 г.



---

# Block algorithms to solve Zheng/Chen/Zhang's finite-difference equations

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

<sup>1</sup>IPSI RAS – Branch of the FSRC “Crystallography and Photonics” RAS,  
443001, Samara, Russia, Molodogvardeyskaya 151,

<sup>2</sup>Samara National Research University, 443086, Samara, Russia, Moskovskoye Shosse 34

## Abstract

This paper is devoted to the design of multiblock algorithms of the FDTD-method intended for computations based on a Zheng-Chen-Zhang implicit finite-difference scheme. Special emphasis is placed on experimental research of the designed algorithms and detecting specific features of the multiblock computing based on implicit finite-difference equations. The efficiency of the proposed approaches is proved by a six-fold speed-up of computations.

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

**Citation:** Golovashkin DL, Morunov ND, Yablokova LV. Block algorithms to solve Zheng/Chen/Zhang's finite-difference equations. *Computer Optics* 2021; 45(3): 461-468. DOI: 10.18287/2412-6179-CO-837.

**Acknowledgements:** This work was supported by the Russian Foundation for Basic Research under grant 19-07-00423 A.

## References

- [1] Taflove A. Computational electrodynamics: The finite-difference time-domain method. Boston: Artech House Publishers; 2005.
  - [2] Taflove A, Oskooi A, Johnson SG. Advances in FDTD computational electrodynamics: Photonics and nanotechnology. Boston: Artech House Publishers; 2013.
  - [3] Yee KS. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans Antennas Propag* 1966; AP-14: 302-307. DOI: 10.1109/TAP.1966.1138693.
  - [4] Namiki T. A new FDTD algorithm based on alternating-direction implicit method. *IEEE Trans Microw Theory Tech* 1999; 47: 2003-2007.
  - [5] Xie G, Huang Z, Fang M, Wu X. A unified 3-D ADI-FDTD algorithm with one-step leapfrog approach for modeling frequency-dependent dispersive media. *Int J Numer Model* 2019; 33(2): 184940-184949. DOI: 10.1002/jnm.266610.
  - [6] Wanjun S, Hou Z. Analysis of electromagnetic wave propagation and scattering characteristics of plasma sheath via high order ADE-ADI FDTD. *J Electromagn Waves Appl* 2016; 30(10): 1321-1333. DOI: 10.1080/09205071.2016.1198727.
  - [7] Yao Z, Wang YE. 3D ADI-FDTD modeling of platform reduction with thin film ferromagnetic material. *IEEE International Symposium on Antennas and Propagation (AP-SURSI)* 2016: 2019-2020. DOI: 10.1109/APS.2016.7696716.
  - [8] Jordan H, Bokhari S, Staker S, Sauer J, ElHelbawy M, Piket-May M. Experience with ADI-FDTD techniques on the Cray MTA supercomputer. *Proc SPIE* 2001; 4528: 68-76. DOI: 10.1117/12.434878.
  - [9] Liu S, Zou B, Zhang L, Ren S. A multi-GPU accelerated parallel domain decomposition one-step leapfrog ADI-FDTD. *IEEE Antennas Wirel Propag Lett* 2020; 19(5): 816-820. DOI: 10.1109/LAWP.2020.2981123.
  - [10] Orozco D, Guang G. Mapping the FDTD application to many-core chip architectures. *International Conference on Parallel Processing (ICPP '09)* 2009: 309-316. DOI: 10.1109/ICPP.2009.44.
  - [11] Minami T, Hibino M, Hiraishi T, Iwashita T, Nakashima H. Automatic parameter tuning of three-dimensional tiled FDTD kernel. *High Performance Computing for Computational Science (VECPAR 2014)* 2014: 284-297. DOI: 10.1007/978-3-319-17353-5\_24.
  - [12] Golub GH, Van Loan CF. Matrix computations. Baltimore, London: Johns Hopkins University Press; 1996.
  - [13] Zhen F, Chen Z, Zhang J. Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method. *IEEE Trans Microw Theory Tech* 2000; 48(9): 1550-1558.
  - [14] Ortega J. Introduction to Parallel and Vector Solution of Linear Systems. New York: Plenum Press; 1988.
  - [15] Zhou X. Tiling optimizations for stencil computations. Source: ([https://www.ideals.illinois.edu/bitstream/handle/2142/44340/Xing\\_Zhou.pdf](https://www.ideals.illinois.edu/bitstream/handle/2142/44340/Xing_Zhou.pdf)).
  - [16] Samarskii AA. The theory of difference schemes. New York: Marcel Dekker Inc; 2001.
  - [17] Samarskii AA, Vabishchevich PN. Computational heat transfer. Chichester: Wiley; 1995.
  - [18] Demmel J. Applied numerical linear algebra. Philadelphia: SIAM; 1997.
  - [19] 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.
  - [20] Wolfe M. Loops skewing: The wavefront method revisited. *Int J Parallel Program* 1986; 15(4): 279-293. DOI: 10.1007/BF01407876.
  - [21] Zakirov AV, Levchenko VD, Perepelkina AY, Zempo Y. DiamondTorre algorithm and high-performance implementation of the FDTD-method for supercomputers with graphics accelerators [In Russian]. *Proc "Supercomputer days in Russia '16"* 2016: 80-94.
-

---

*Authors' information*

**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 and Physics 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).

**Nikita Dmitrievich Morunov** study as graduate student of the Applied Mathematics and Physics department at Samara University. Research interests: finite-difference approximations, FDTD-method, vector and matrix computations. E-mail: [niknixad@mail.ru](mailto:niknixad@mail.ru).

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

---

*Received November 16, 2020. The final version – December 23, 2020.*

---