



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Чикиткин А.В., Рогов Б.В.

Два варианта параллельной
реализации высокоточных
бикompактных схем для
многомерного
неоднородного уравнения
переноса

Рекомендуемая форма библиографической ссылки: Чикиткин А.В., Рогов Б.В. Два варианта параллельной реализации высокоточных бикompактных схем для многомерного неоднородного уравнения переноса // Препринты ИПМ им. М.В.Келдыша. 2018. № 177. 24 с. doi:[10.20948/prepr-2018-177](https://doi.org/10.20948/prepr-2018-177)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-177>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

А.В.Чикиткин, Б.В.Рогов

**Два варианта параллельной реализации
высокоточных бикомпактных схем
для многомерного неоднородного
уравнения переноса**

Москва — 2018

Чикиткин А.В., Rogov Б.В.

Два варианта параллельной реализации высокоточных бикомпактных схем для многомерного неоднородного уравнения переноса

В работе проведено сравнение эффективности двух параллельных алгоритмов решения уравнений высокоточных бикомпактных схем для многомерного неоднородного уравнения переноса. Первый из них есть пространственный маршевый алгоритм счета нефакторизованных схем, а второй основан на приближенной факторизации многомерных схем. В последнем алгоритме используются итерации для сохранения высокого (выше второго) порядка точности бикомпактных схем по времени. Доказана сходимость этих итераций для нестационарного двухмерного и трехмерного линейного неоднородного уравнения переноса с постоянными положительными коэффициентами. Модельные расчёты показывают, что алгоритм на основе факторизованных схем является предпочтительным с точки зрения параллельной реализации.

Ключевые слова: многомерное неоднородное линейное уравнение переноса, бикомпактные схемы, параллельные алгоритмы, метод итерируемой приближенной факторизации

Aleksandr Viktorovich Chikitkin, Boris Vadimovich Rogov

Two variants of parallel implementation of high-order accurate bicomact schemes for multi-dimensional inhomogeneous transport equation

In this paper, we compare the efficiency of two parallel algorithms for solution of the equations of multidimensional high-order accurate bicomact schemes for a multidimensional inhomogeneous transport equation. The first one is a space-marching algorithm for computing non-factorized schemes, and the second one is based on the approximate factorization of multidimensional schemes. The latter algorithm uses iterations to preserve the high (higher than second) order of accuracy of bicomact schemes in time. The convergence of these iterations is proved for a nonstationary two-dimensional and three-dimensional linear inhomogeneous transport equation with constant positive coefficients. Model computations show that the factorization scheme is preferable from the point of view of parallel implementation.

Key words: multidimensional inhomogeneous transport equation, bicomact schemes, parallel algorithms, iterative approximate factorization method

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 18-01-00857-а.

1. Введение

Необходимость решения линейного неоднородного уравнения переноса возникает в задачах переноса излучения и переноса незаряженных частиц (нейтронов). Подобные задачи возникают во многих областях науки и техники от расчетов атмосферной радиации [1] и термоядерных мишеней [2] до расчета активной зоны ядерного реактора [3].

Решение уравнения переноса является функцией распределения в фазовом пространстве и по физическому смыслу должно быть неотрицательно. Во многих случаях приходится решать это уравнение в сильно гетерогенной среде с большим количеством контактных разрывов (облака в атмосфере или различные материалы в активной зоне реактора). Поэтому к дискретной схеме для расчета уравнения переноса предъявляются соответствующие требования. Во-первых, схема должна обладать консервативностью и хорошим разрешением, чтобы адекватно учитывать разрыв коэффициентов и источников уравнения. Эта цель лучше всего достигается при аппроксимации определяющего уравнения на сеточном шаблоне, расположенном в рамках одной расчетной ячейки. Во-вторых, она должна обеспечивать положительность численного решения.

Схема, удовлетворяющая указанным требованиям, была предложена в работе [4] для одномерного (1D) однородного линейного уравнения переноса. Она построена методом прямых на минимальном пространственном шаблоне, состоящем из двух узлов, и названа бикompактной схемой. В полудискретной форме этой схемы четвертый порядок пространственной аппроксимации достигнут за счет добавления в число искомых функций вспомогательной зависимой переменной – первообразной основной искомой функции. Для этой зависимой переменной выводится дополнительное дискретное уравнение, являющееся аппроксимацией интегрального следствия основного дифференциального уравнения. Поскольку в схему [4] входила только конечная разность первообразной функции, то, начиная с работы [5], бикompактная схема стала записываться в более удобной для вычислений форме: конечные разности первообразной функции были заменены интегральными средними искомой функции по ячейкам сетки. В дальнейшем бикompактные схемы были построены для двухмерного (2D) и трехмерного (3D) нестационарного неоднородного линейного уравнения переноса [6,7], для систем нестационарных многомерных квазилинейных уравнений гиперболического типа [8]. Отметим, что в бикompактных схемах [8, 9] для квазилинейных уравнений гиперболического типа повышение порядка пространственной аппроксимации обеспечивается не с помощью вспомогательных значений интегральных средних искомой функции по ячейкам сетки, а с помощью вспомогательных значений искомой функции в полуцелых узлах пространственной сетки. Для нахождения этих значений выводятся дополнительные разностные уравнения, которые являются аппроксимацией следствий основных дифференциальных уравнений. Для интегрирования

уравнений полудискретных бикомпактных схем по времени обычно использовались А- и L-устойчивые диагонально неявные методы Рунге-Кутты (DIRK методы) [4-8]. Важнейшими достоинствами бикомпактных схем при нахождении гладких решений являются сохранение порядка точности на произвольной неравномерной пространственной сетке, экономичность маршевого метода решений уравнений схемы по каждой пространственной переменной. На основе высокоточных бикомпактных схем построены гибридные схемы для расчета разрывных решений уравнений гиперболического типа [6, 8].

Для эффективного расчета нестационарного многомерного неоднородного уравнения переноса в работе [6] предложен параллельный алгоритм бегущего счета разностных уравнений высокоточных бикомпактных схем. В настоящей работе рассматривается другой параллельный алгоритм решения системы разностных уравнений высокоточных бикомпактных схем. Он основан на приближенной факторизации многомерных разностных операторов. В этом алгоритме используются итерации для сохранения высокого (выше второго) порядка точности схем по времени [9, 10]. В работе проведено сравнение эффективности двух параллельных алгоритмов счета уравнений многомерных бикомпактных схем – алгоритма бегущего счета нефакторизованных схем и алгоритма счета факторизованных схем.

2. Бикомпактные схемы

Основные содержательные моменты способа построения многомерных бикомпактных схем [6, 7] без ограничения общности могут быть рассмотрены на примере трехмерного нестационарного уравнения переноса

$$L_3(u) = 0, \quad L_3(u) \equiv \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + c \frac{\partial u}{\partial z} + \sigma u - q(x, y, z, t) \quad (1)$$

и его двухмерного аналога

$$L_2(u) = 0, \quad L_2(u) \equiv \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + \sigma u - q(x, y, t) \quad (2)$$

в предположении, что a , b , c и σ – постоянные положительные параметры.

Полудискретные схемы для уравнения переноса выводятся методом прямых путем дискретизации пространственных производных на минимальном шаблоне. В одномерном (1D) случае пространственный шаблон состоит из двух узлов x_{j+1}, x_j , в двухмерном (2D) случае – из четырех узлов (x_j, y_k) , (x_{j+1}, y_k) , (x_j, y_{k+1}) , (x_{j+1}, y_{k+1}) , а в трехмерном (3D) случае – из восьми узлов $(x_\alpha, y_\beta, z_\gamma)$, $\alpha = j, j+1$, $\beta = k, k+1$, $\gamma = l, l+1$. Благодаря тому, что в каждом пространственном направлении шаблон является двухточечным, схемы названы *бикомпактными*. Вывод схемы рассмотрим для 3D случая. Поскольку он будет проводиться на

минимальном пространственном шаблоне, то он справедлив в общем случае неравномерных декартовых сеток с шагами $h_x \equiv h_{x,j+1/2} = x_{j+1} - x_j$, $h_y \equiv h_{y,k+1/2} = y_{k+1} - y_k$, $h_z \equiv h_{z,l+1/2} = z_{l+1} - z_l$.

Для получения полудискретной схемы повышенного порядка аппроксимации по пространственным переменным наряду со значениями искомой функции u в узлах шаблона

$$u_{\alpha,\beta,\gamma}, \quad \alpha = j, j+1, \quad \beta = k, k+1, \quad \gamma = l, l+1 \quad (3)$$

вводятся вспомогательные величины:

1) интегральные средние по 12 ребрам 3D ячейки $G_{3D} = [x_j, x_{j+1}] \times [y_k, y_{k+1}] \times [z_l, z_{l+1}]$

$$\begin{aligned} \bar{u}_{j+1/2,\beta,\gamma}^x &= \frac{1}{h_x} \int_{x_j}^{x_{j+1}} u(x, y_\beta, z_\gamma, t) dx, \quad \bar{u}_{\alpha,k+1/2,\gamma}^y = \frac{1}{h_y} \int_{y_k}^{y_{k+1}} u(x_\alpha, y, z_\gamma, t) dy, \\ \bar{u}_{\alpha,\beta,l+1/2}^z &= \frac{1}{h_z} \int_{z_l}^{z_{l+1}} u(x_\alpha, y_\beta, z, t) dz, \quad \alpha = j, j+1, \quad \beta = k, k+1, \quad \gamma = l, l+1, \end{aligned} \quad (4)$$

2) интегральные средние по 6 граням ячейки

$$\begin{aligned} \bar{u}_{j+1/2,k+1/2,\gamma}^{xy} &= \frac{1}{h_x h_y} \int_{x_j}^{x_{j+1}} \int_{y_k}^{y_{k+1}} u(x, y, z_\gamma, t) dy dx, \quad \bar{u}_{j+1/2,\beta,l+1/2}^{xz} = \frac{1}{h_x h_z} \int_{x_j}^{x_{j+1}} \int_{z_l}^{z_{l+1}} u(x, y_\beta, z, t) dz dx, \\ \bar{u}_{\alpha,k+1/2,l+1/2}^{yz} &= \frac{1}{h_y h_z} \int_{y_k}^{y_{k+1}} \int_{z_l}^{z_{l+1}} u(x_\alpha, y, z, t) dz dy, \quad \alpha = j, j+1, \quad \beta = k, k+1, \quad \gamma = l, l+1, \end{aligned} \quad (5)$$

3) интегральное среднее по этой ячейке

$$\bar{u}_{j+1/2,k+1/2,l+1/2}^{xyz} = \frac{1}{h_x h_y h_z} \int_{x_j}^{x_{j+1}} \int_{y_k}^{y_{k+1}} \int_{z_l}^{z_{l+1}} u(x, y, z, t) dz dy dx. \quad (6)$$

В формулах (3)–(6) величины $u_{j+1,k+1,l+1}$, $\bar{u}_{j+1/2,k+1,l+1}^x$, $\bar{u}_{j+1,k+1/2,l+1}^y$, $\bar{u}_{j+1,k+1,l+1/2}^z$, $\bar{u}_{j+1/2,k+1/2,l+1}^{xy}$, $\bar{u}_{j+1/2,k+1,l+1/2}^{xz}$, $\bar{u}_{j+1,k+1/2,l+1/2}^{yz}$, $\bar{u}_{j+1/2,k+1/2,l+1/2}^{xyz}$ являются неизвестными функциями времени при $a > 0$, $b > 0$, $c > 0$ (другие знаки рассматриваются аналогично), а другие величины известны либо из расчетов в соседних ячейках, либо из начальных и граничных условий. Уравнения для этих переменных могут быть получены интегрированием по пространственной ячейке G_{3D} исходного уравнения (1), а также семи независимых его дифференциальных следствий:

$$\begin{aligned} \frac{\partial L_3(u)}{\partial x} = 0, \quad \frac{\partial L_3(u)}{\partial y} = 0, \quad \frac{\partial L_3(u)}{\partial z} = 0, \\ \frac{\partial^2 L_3(u)}{\partial x \partial y} = 0, \quad \frac{\partial^2 L_3(u)}{\partial x \partial z} = 0, \quad \frac{\partial^2 L_3(u)}{\partial y \partial z} = 0, \quad \frac{\partial^3 L_3(u)}{\partial x \partial y \partial z} = 0. \end{aligned} \quad (7)$$

Результаты интегрирования уравнений (1), (7) по ячейке G_{3D} можно представить в компактной форме

$$\begin{aligned} A^z A^y A^x (u_t)_C + (aA^z A^y \Lambda_1^x + bA^z \Lambda_1^y A^x + c\Lambda_1^z A^y A^x + \sigma A^z A^y A^x) u_C &= A^z A^y A^x q_C, \\ A^z A^y \Lambda_1^x (u_t)_C + (aA^z A^y \Lambda_2^x + bA^z \Lambda_1^y \Lambda_1^x + c\Lambda_1^z A^y \Lambda_1^x + \sigma A^z A^y \Lambda_1^x) u_C &= A^z A^y \Lambda_1^x q_C, \\ A^z \Lambda_1^y A^x (u_t)_C + (aA^z \Lambda_1^y \Lambda_1^x + bA^z \Lambda_2^y A^x + c\Lambda_1^z \Lambda_1^y A^x + \sigma A^z \Lambda_1^y A^x) u_C &= A^z \Lambda_1^y A^x q_C, \\ A^z \Lambda_1^y \Lambda_1^x (u_t)_C + (aA^z \Lambda_1^y \Lambda_2^x + bA^z \Lambda_2^y \Lambda_1^x + c\Lambda_1^z \Lambda_1^y \Lambda_1^x + \sigma A^z \Lambda_1^y \Lambda_1^x) u_C &= A^z \Lambda_1^y \Lambda_1^x q_C, \\ \Lambda_1^z A^y A^x (u_t)_C + (a\Lambda_1^z A^y \Lambda_1^x + b\Lambda_1^z \Lambda_1^y A^x + c\Lambda_2^z A^y A^x + \sigma \Lambda_1^z A^y A^x) u_C &= \Lambda_1^z A^y A^x q_C, \\ \Lambda_1^z A^y \Lambda_1^x (u_t)_C + (a\Lambda_1^z A^y \Lambda_2^x + b\Lambda_1^z \Lambda_1^y \Lambda_1^x + c\Lambda_2^z A^y \Lambda_1^x + \sigma \Lambda_1^z A^y \Lambda_1^x) u_C &= \Lambda_1^z A^y \Lambda_1^x q_C, \\ \Lambda_1^z \Lambda_1^y A^x (u_t)_C + (a\Lambda_1^z \Lambda_1^y \Lambda_1^x + b\Lambda_1^z \Lambda_2^y A^x + c\Lambda_2^z \Lambda_1^y A^x + \sigma \Lambda_1^z \Lambda_1^y A^x) u_C &= \Lambda_1^z \Lambda_1^y A^x q_C, \\ \Lambda_1^z \Lambda_1^y \Lambda_1^x (u_t)_C + (a\Lambda_1^z \Lambda_1^y \Lambda_2^x + b\Lambda_1^z \Lambda_2^y \Lambda_1^x + c\Lambda_2^z \Lambda_1^y \Lambda_1^x + \sigma \Lambda_1^z \Lambda_1^y \Lambda_1^x) u_C &= \Lambda_1^z \Lambda_1^y \Lambda_1^x q_C, \end{aligned} \quad (8)$$

если ввести следующие операторы:

$$\begin{aligned} A^x u_{j+1/2, \beta, \gamma} &= \bar{u}_{j+1/2, \beta, \gamma}^x, \quad A^y u_{\alpha, k+1/2, \gamma} = \bar{u}_{\alpha, k+1/2, \gamma}^y, \quad A^z u_{\alpha, \beta, l+1/2} = \bar{u}_{\alpha, \beta, l+1/2}^z, \\ \Lambda_1^x u_{j+1/2, \beta, \gamma} &= (u_{j+1, \beta, \gamma} - u_{j, \beta, \gamma}) / h_x, \\ \Lambda_1^y u_{\alpha, k+1/2, \gamma} &= (u_{\alpha, k+1, \gamma} - u_{\alpha, k, \gamma}) / h_y, \\ \Lambda_1^z u_{\alpha, \beta, l+1/2} &= (u_{\alpha, \beta, l+1} - u_{\alpha, \beta, l}) / h_z, \\ \Lambda_2^x u_{j+1/2, \beta, \gamma} &= 6(u_{j+1, \beta, \gamma} - 2\bar{u}_{j+1/2, \beta, \gamma}^x + u_{j, \beta, \gamma}) / h_x^2, \\ \Lambda_2^y u_{\alpha, k+1/2, \gamma} &= 6(u_{\alpha, k+1, \gamma} - 2\bar{u}_{\alpha, k+1/2, \gamma}^y + u_{\alpha, k, \gamma}) / h_y^2, \\ \Lambda_2^z u_{\alpha, \beta, l+1/2} &= 6(u_{\alpha, \beta, l+1} - 2\bar{u}_{\alpha, \beta, l+1/2}^z + u_{\alpha, \beta, l}) / h_z^2. \end{aligned} \quad (9)$$

В уравнениях (8) $u_t \equiv du/dt$, $C = (j+1/2, k+1/2, l+1/2)$ – мультииндекс. Операторы $A^r, \Lambda_1^r, \Lambda_2^r$ ($r = x, y, z$) имеют смысл операторов осреднения, первой и второй разностных производных соответственно.

Первое уравнение в (8) есть результат точного интегрирования основного уравнения (1), а остальные уравнения в (8) получены из уравнений (7) с использованием квадратурных формул Эйлера–Маклорена

$$\begin{aligned}
\int_{x_j}^{x_{j+1}} u dx &= \frac{h_x}{2} (u_{j+1} + u_j) - \frac{h_x^2}{12} (\partial_x u_{j+1} - \partial_x u_j) + O(h_x^5), \quad \partial_x u \equiv \frac{\partial u}{\partial x}, \\
\int_{y_k}^{y_{k+1}} u dy &= \frac{h_y}{2} (u_{k+1} + u_k) - \frac{h_y^2}{12} (\partial_y u_{k+1} - \partial_y u_k) + O(h_y^5), \quad \partial_y u \equiv \frac{\partial u}{\partial y}, \\
\int_{z_l}^{z_{l+1}} u dz &= \frac{h_z}{2} (u_{l+1} + u_l) - \frac{h_z^2}{12} (\partial_z u_{l+1} - \partial_z u_l) + O(h_z^5), \quad \partial_z u \equiv \frac{\partial u}{\partial z}
\end{aligned} \tag{10}$$

и имеют четвертый порядок аппроксимации по пространственным переменным.

Уравнения полудискретной 2D бикompактной схемы могут быть получены интегрированием по пространственной ячейке $G_{2D} = [x_j, x_{j+1}] \times [y_k, y_{k+1}]$ исходного уравнения (2), а также трех независимых его дифференциальных следствий:

$$\frac{\partial L_2(u)}{\partial x} = 0, \quad \frac{\partial L_2(u)}{\partial y} = 0, \quad \frac{\partial^2 L_2(u)}{\partial x \partial y} = 0. \tag{11}$$

Полудискретная форма бикompактной схемы для уравнения (2) состоит из четырех обыкновенных дифференциальных уравнений (ОДУ)

$$\begin{aligned}
A^y A^x (u_t)_C + (a A^y \Lambda_1^x + b \Lambda_1^y A^x + \sigma A^y A^x) u_C &= A^y A^x q_C, \\
A^y \Lambda_1^x (u_t)_C + (a A^y \Lambda_2^x + b \Lambda_1^y \Lambda_1^x + \sigma A^y \Lambda_1^x) u_C &= A^y \Lambda_1^x q_C, \\
\Lambda_1^y A^x (u_t)_C + (a \Lambda_1^y \Lambda_1^x + b \Lambda_2^y A^x + \sigma \Lambda_1^y A^x) u_C &= \Lambda_1^y A^x q_C, \\
\Lambda_1^y \Lambda_1^x (u_t)_C + (a \Lambda_1^y \Lambda_2^x + b \Lambda_2^y \Lambda_1^x + \sigma \Lambda_1^y \Lambda_1^x) u_C &= \Lambda_1^y \Lambda_1^x q_C
\end{aligned} \tag{12}$$

для определения четырех неизвестных величин $u_{j+1,k+1}$, $\bar{u}_{j+1/2,k+1}^x$, $\bar{u}_{j+1,k+1/2}^y$, $\bar{u}_{j+1/2,k+1/2}^{xy}$ при $a > 0$, $b > 0$ (другие знаки рассматриваются аналогично). В уравнениях (12) мультииндекс C равен $(j+1/2, k+1/2)$.

Для интегрирования по времени ОДУ (8) и (12) предлагается использовать A - и L -устойчивые диагонально неявные методы Рунге-Кутты (DIRK методы). Эти методы обеспечивают абсолютную устойчивость полностью дискретных бикompактных схем. Кроме того, DIRK методы являются экономичными по сравнению с полностью неявными методами Рунге-Кутты (РК). Вычисления на каждой стадии DIRK метода сводятся к решению уравнений полностью дискретной базовой бикompактной схемы со своим стадийным начальным условием и шагом по времени. Базовая бикompактная схема получается путем интегрирования полудискретной схемы по неявному методу Эйлера. Эта схема для 2D уравнения переноса (2) состоит из четырех разностных уравнений и имеет следующий вид:

$$\begin{aligned}
& \left((1 + \sigma\tau)A^y A^x + \tau a A^y \Lambda_1^x + \tau b \Lambda_1^y A^x \right) \hat{u}_C = A^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)A^y \Lambda_1^x + \tau a A^y \Lambda_2^x + \tau b \Lambda_1^y \Lambda_1^x \right) \hat{u}_C = A^y \Lambda_1^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^y A^x + \tau a \Lambda_1^y \Lambda_1^x + \tau b \Lambda_2^y A^x \right) \hat{u}_C = \Lambda_1^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^y \Lambda_1^x + \tau a \Lambda_1^y \Lambda_2^x + \tau b \Lambda_2^y \Lambda_1^x \right) \hat{u}_C = \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C),
\end{aligned} \tag{13}$$

где τ — шаг по времени, $\hat{u} = u^{n+1}$, $u = u^n$, n — номер временного слоя. Для 3D уравнения переноса (1) базовая бикомпактная схема состоит из восьми разностных уравнений

$$\begin{aligned}
& \left((1 + \sigma\tau)A^z A^y A^x + a\tau A^z A^y \Lambda_1^x + b\tau A^z \Lambda_1^y A^x + c\tau \Lambda_1^z A^y A^x \right) \hat{u}_C = A^z A^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)A^z A^y \Lambda_1^x + a\tau A^z A^y \Lambda_2^x + b\tau A^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_1^z A^y \Lambda_1^x \right) \hat{u}_C = A^z A^y \Lambda_1^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)A^z \Lambda_1^y A^x + a\tau A^z \Lambda_1^y \Lambda_1^x + b\tau A^z \Lambda_2^y A^x + c\tau \Lambda_1^z \Lambda_1^y A^x \right) \hat{u}_C = A^z \Lambda_1^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)A^z \Lambda_1^y \Lambda_1^x + a\tau A^z \Lambda_1^y \Lambda_2^x + b\tau A^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x \right) \hat{u}_C = A^z \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^z A^y A^x + a\tau \Lambda_1^z A^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_1^y A^x + c\tau \Lambda_2^z A^y A^x \right) \hat{u}_C = \Lambda_1^z A^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^z A^y \Lambda_1^x + a\tau \Lambda_1^z A^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_2^z A^y \Lambda_1^x \right) \hat{u}_C = \Lambda_1^z A^y \Lambda_1^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^z \Lambda_1^y A^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_2^y A^x + c\tau \Lambda_2^z \Lambda_1^y A^x \right) \hat{u}_C = \Lambda_1^z \Lambda_1^y A^x (u_C + \tau \hat{q}_C), \\
& \left((1 + \sigma\tau)\Lambda_1^z \Lambda_1^y \Lambda_1^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_2^z \Lambda_1^y \Lambda_1^x \right) \hat{u}_C = \Lambda_1^z \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C).
\end{aligned} \tag{14}$$

3. Метод итерируемой приближенной факторизации

С увеличением размерности задачи d число уравнений в каждой ячейке растёт как 2^d . При решении методом бегущего счёта в каждой ячейке решается система размерности 2^d с помощью прямого метода Гаусса, который имеет сложность $C \times (2^d)^3 = C \times 2^{3d}$. Уже при $d=3$ это приводит к сильному росту вычислительных затрат по сравнению с одномерным случаем. Среди способов, которые позволяют избежать такого роста, можно упомянуть расщепление по направлениям (local one-dimensional scheme – LOD scheme), метод переменных направлений и методы приближенной факторизации линейных систем разностных уравнений. Большинство этих способов приводят к падению порядка точности до второго.

Разностные уравнения многомерных бикомпактных схем также могут быть приближенно факторизованы. Для сохранения высокого порядка схем по времени строится итерационная процедура на основе приближенной факторизации. Ниже эта процедура излагается для 2D и 3D бикомпактных схем, предназначенных для численного решения неоднородного уравнения переноса.

3.1. Двухмерная схема

Рассмотрим базовую схему (13), когда система ОДУ (12) интегрируется неявным методом Эйлера. Введем следующие операторы:

$$\begin{aligned} B_1^r(s) &= A^r + \tau C_1^r(s), \quad B_2^r(s) = \Lambda_1^r + \tau C_2^r(s), \quad r = x, y, \\ C_1^r(s) &= \frac{1}{2} \sigma A^r + s \Lambda_1^r, \quad C_2^r(s) = \frac{1}{2} \sigma \Lambda_1^r + s \Lambda_2^r. \end{aligned} \quad (15)$$

Операторы, стоящие в левых частях уравнений (13), допускают следующее представление:

$$\begin{aligned} A^y A^x (1 + \sigma \tau) + \tau a A^y \Lambda_1^x + \tau b \Lambda_1^y A^x &= B_1^y(b) B_1^x(a) - \tau^2 C_1^y(b) C_1^x(a), \\ A^y \Lambda_1^x (1 + \sigma \tau) + \tau a A^y \Lambda_2^x + \tau b \Lambda_1^y \Lambda_1^x &= B_1^y(b) B_2^x(a) - \tau^2 C_1^y(b) C_2^x(a), \\ \Lambda_1^y A^x (1 + \sigma \tau) + \tau a \Lambda_1^y \Lambda_1^x + \tau b \Lambda_2^y A^x &= B_2^y(b) B_1^x(a) - \tau^2 C_2^y(b) C_1^x(a), \\ \Lambda_1^y \Lambda_1^x (1 + \sigma \tau) + \tau a \Lambda_1^y \Lambda_2^x + \tau b \Lambda_2^y \Lambda_1^x &= B_2^y(b) B_2^x(a) - \tau^2 C_2^y(b) C_2^x(a). \end{aligned} \quad (16)$$

С учетом равенств (16) уравнения (13) можно записать таким образом:

$$\begin{aligned} B_1^y(b) B_1^x(a) \hat{u}_C &= A^y A^x (u_C + \tau \hat{q}_C) + \tau^2 C_1^y(b) C_1^x(a) \hat{u}_C, \\ B_1^y(b) B_2^x(a) \hat{u}_C &= A^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 C_1^y(b) C_2^x(a) \hat{u}_C, \\ B_2^y(b) B_1^x(a) \hat{u}_C &= \Lambda_1^y A^x (u_C + \tau \hat{q}_C) + \tau^2 C_2^y(b) C_1^x(a) \hat{u}_C, \\ B_2^y(b) B_2^x(a) \hat{u}_C &= \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 C_2^y(b) C_2^x(a) \hat{u}_C. \end{aligned} \quad (17)$$

Из структуры системы уравнений (17) видно, что её можно приближенно факторизовать при малых τ , отбросив последние члены в правых частях. Это обстоятельство позволяет использовать для решения системы (17) следующий итерационный метод:

$$\begin{aligned} B_1^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= A^y A^x (u_C + \tau \hat{q}_C) + \tau^2 C_1^y(b) C_1^x(a) \hat{u}_C^{(i)}, \\ B_1^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= A^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 C_1^y(b) C_2^x(a) \hat{u}_C^{(i)}, \\ B_2^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^y A^x (u_C + \tau \hat{q}_C) + \tau^2 C_2^y(b) C_1^x(a) \hat{u}_C^{(i)}, \\ B_2^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 C_2^y(b) C_2^x(a) \hat{u}_C^{(i)}, \end{aligned} \quad (18)$$

где i – номер итерации. Начальное приближение $\hat{u}^{(0)}$ можно взять нулевым. Решение уравнений (18) относительно $\hat{u}^{(i+1)}$ можно найти, сначала решая две системы уравнений для сеточных функций v_1, v_2 :

$$\begin{cases}
B_1^y(b)v_1 = A^y A^x(u_c + \tau \hat{q}_c) + \tau^2 C_1^y(b)C_1^x(a)\hat{u}_c^{(i)}, \\
B_2^y(b)v_1 = \Lambda_1^y A^x(u_c + \tau \hat{q}_c) + \tau^2 C_2^y(b)C_1^x(a)\hat{u}_c^{(i)}, \\
B_1^y(b)v_2 = A^y \Lambda_1^x(u_c + \tau \hat{q}_c) + \tau^2 C_1^y(b)C_2^x(a)\hat{u}_c^{(i)}, \\
B_2^y(b)v_2 = \Lambda_1^y \Lambda_1^x(u_c + \tau \hat{q}_c) + \tau^2 C_2^y(b)C_2^x(a)\hat{u}_c^{(i)},
\end{cases} \quad (19)$$

а затем – систему уравнений

$$\begin{cases}
B_1^x(a)\hat{u}^{(i+1)} = v_1, \\
B_2^x(a)\hat{u}^{(i+1)} = v_2.
\end{cases} \quad (20)$$

Граничные условия для задач (19) относительно v_1, v_2 вычисляются из выражений (20).

Одномерные уравнения (19) требуется решать на координатных линиях $x = x_{j+1/2}$, $j = \overline{1, N_x}$ (N_x – число ячеек сетки по направлению x):

$$\begin{cases}
B_1^y(b)v_{1,j+1/2,k+1/2} = A^y A^x(u_c + \tau \hat{q}_c) + \tau^2 C_1^y(b)C_1^x(a)\hat{u}_c^{(i)}, \\
B_2^y(b)v_{1,j+1/2,k+1/2} = \Lambda_1^y A^x(u_c + \tau \hat{q}_c) + \tau^2 C_2^y(b)C_1^x(a)\hat{u}_c^{(i)}, \\
B_1^y(b)v_{2,j+1/2,k+1/2} = A^y \Lambda_1^x(u_c + \tau \hat{q}_c) + \tau^2 C_1^y(b)C_2^x(a)\hat{u}_c^{(i)}, \\
B_2^y(b)v_{2,j+1/2,k+1/2} = \Lambda_1^y \Lambda_1^x(u_c + \tau \hat{q}_c) + \tau^2 C_2^y(b)C_2^x(a)\hat{u}_c^{(i)}.
\end{cases} \quad (21)$$

Одномерные уравнения (20) требуется решать на координатных линиях $y = y_{k+1/2}$

$$\begin{cases}
B_1^x(a)(\hat{u}^y)^{(i+1)}_{j+1/2,k+1/2} = \bar{v}_{1,j+1/2,k+1/2}^y, \\
B_2^x(a)(\hat{u}^y)^{(i+1)}_{j+1/2,k+1/2} = \bar{v}_{2,j+1/2,k+1/2}^y
\end{cases} \quad (22)$$

и линиях $y = y_{k+1}$, $k = \overline{1, N_y}$ (N_y – число ячеек сетки по направлению y)

$$\begin{cases}
B_1^x(a)\hat{u}_{j+1/2,k+1}^{(i+1)} = v_{1,j+1/2,k+1}, \\
B_2^x(a)\hat{u}_{j+1/2,k+1}^{(i+1)} = v_{2,j+1/2,k+1}.
\end{cases} \quad (23)$$

Уравнения (22) следуют из уравнений (20), если к ним применить оператор осреднения A^y .

В итоге, на каждом временном слое нужно решить $2(N_x + N_y)$ одномерных задач. Каждая одномерная задача решается методом бегущего счета. Факторизованная схема имеет два преимущества перед исходной (нефакторизованной) схемой. Во-первых, при решении одномерных задач обращаются матрицы размерности 2×2 вместо матриц размерности $2^d \times 2^d$. Во-вторых, одномерные задачи вдоль разных сеточных линий на каждой итерации можно решать независимо, поэтому алгоритм может быть эффективно

распараллелен, в отличие от исходной схемы с более сложной зависимостью данных.

Рассмотрим вопрос о сходимости итерационного метода (18). Это метод можно записать в матрично-векторном виде:

$$\mathbf{M}_y \mathbf{M}_x \hat{\mathbf{v}}^{(i+1)} = \mathbf{N} \hat{\mathbf{v}}^{(i)} + \boldsymbol{\psi}, \quad \mathbf{v} = \begin{pmatrix} \bar{u}_{j+1/2, k+1/2}^{xy} \\ \bar{u}_{j+1, k+1/2}^y \\ \bar{u}_{j+1/2, k+1}^x \\ u_{j+1, k+1} \end{pmatrix}, \quad \boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix},$$

$$\mathbf{M}_y = \begin{bmatrix} \theta & 0 & \kappa_y & 0 \\ 0 & \theta & 0 & \kappa_y \\ -12\kappa_y & 0 & \theta + 6\kappa_y & 0 \\ 0 & -12\kappa_y & 0 & \theta + 6\kappa_y \end{bmatrix}, \quad \mathbf{M}_x = \begin{bmatrix} \theta & \kappa_x & 0 & 0 \\ -12\kappa_x & \theta + 6\kappa_x & 0 & 0 \\ 0 & 0 & \theta & \kappa_x \\ 0 & 0 & -12\kappa_x & \theta + 6\kappa_x \end{bmatrix}, \quad (24)$$

$$\mathbf{N} = \begin{bmatrix} (\theta-1)^2 & (\theta-1)\kappa_x & (\theta-1)\kappa_y & \kappa_x \kappa_y \\ -12(\theta-1)\kappa_x & (\theta-1)(\theta-1+6\kappa_x) & -12\kappa_x \kappa_y & \kappa_y(\theta-1+6\kappa_x) \\ -12(\theta-1)\kappa_y & -12\kappa_x \kappa_y & (\theta-1)(\theta-1+6\kappa_y) & \kappa_x(\theta-1+6\kappa_y) \\ 144\kappa_x \kappa_y & -12\kappa_y(\theta-1+6\kappa_x) & -12\kappa_x(\theta-1+6\kappa_y) & (\theta-1+6\kappa_x)(\theta-1+6\kappa_y) \end{bmatrix},$$

где $\theta = 1 + (\sigma\tau)/2$; $\kappa_x = a\tau/h_x$, $\kappa_y = b\tau/h_y$ – числа Куранта; $\boldsymbol{\psi}$ – вектор-столбец, включающий известные значения величин с предыдущего временного слоя, значения, известные из граничных условий или уже вычисленные в предыдущих ячейках, а также значения известной источниковой функции q .

Матрицы $\mathbf{M}_x, \mathbf{M}_y, \mathbf{N}$ можно представить в виде следующих кронекеровых произведений матриц:

$$\mathbf{M}_x = \mathbf{E} \otimes \mathbf{P}_x, \quad \mathbf{M}_y = \mathbf{P}_y \otimes \mathbf{E}, \quad \mathbf{N} = (\mathbf{P}_y - \mathbf{E}) \otimes (\mathbf{P}_x - \mathbf{E}), \quad (25)$$

где

$$\mathbf{P}_r = \theta \mathbf{E} + \kappa_r \mathbf{R}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -12 & 6 \end{bmatrix}, \quad r = x, y. \quad (26)$$

Произведение матриц $\mathbf{M}_y, \mathbf{M}_x$ можно преобразовать следующим образом:

$$\mathbf{M}_y \mathbf{M}_x = (\mathbf{P}_y \otimes \mathbf{E})(\mathbf{E} \otimes \mathbf{P}_x) = \mathbf{P}_y \otimes \mathbf{P}_x. \quad (27)$$

Запишем итерационный метод (24) в рекуррентном виде

$$\mathbf{v}^{(i+1)} = \mathbf{S} \mathbf{v}^{(i)} + \boldsymbol{\xi}, \quad (28)$$

где матрица шага метода

$$\mathbf{S} = (\mathbf{M}_y \mathbf{M}_x)^{-1} \mathbf{N}. \quad (29)$$

С учетом формул (25), (27) матрица шага (29) итерационного метода может быть записана как кронекерово произведение двух матриц, каждая из которых связана только с одним пространственным направлением:

$$\begin{aligned} \mathbf{S} &= (\mathbf{P}_y \otimes \mathbf{P}_x)^{-1} [(\mathbf{P}_y - \mathbf{E}) \otimes (\mathbf{P}_x - \mathbf{E})] = (\mathbf{P}_y^{-1} \otimes \mathbf{P}_x^{-1}) [(\mathbf{P}_y - \mathbf{E}) \otimes (\mathbf{P}_x - \mathbf{E})] = \\ &= (\mathbf{E} - \mathbf{P}_y^{-1}) \otimes (\mathbf{E} - \mathbf{P}_x^{-1}) = (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x), \end{aligned} \quad (30)$$

где $\mathbf{T}_r = \mathbf{P}_r^{-1}$, $r = x, y$.

Благодаря представлению (30) для матрицы \mathbf{S} , ее собственные значения $\lambda(\mathbf{S})$ равны произведению собственных значений λ_y, λ_x матриц $\mathbf{E} - \mathbf{T}_y, \mathbf{E} - \mathbf{T}_x$

$$\lambda(\mathbf{S}) = \lambda_y \lambda_x, \quad (31)$$

где

$$\lambda_r = \frac{12\kappa_r^2 + (6\theta - 3 \pm i\sqrt{3})\kappa_r + \theta(\theta - 1)}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}, \quad r = x, y. \quad (32)$$

Модули комплексно сопряженных чисел λ_r (32) одинаковы и равны

$$\begin{aligned} |\lambda_r| &= \frac{\sqrt{(12\kappa_r^2 + 3(2\theta - 1)\kappa_r + \theta(\theta - 1))^2 + 3\kappa_r^2}}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2} = \\ &= \sqrt{1 - \frac{6\kappa_r + 2\theta - 1}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}} < 1 \quad \forall \kappa_r > 0, \theta \geq 1, \end{aligned} \quad (33)$$

поэтому в силу равенства (31) спектральный радиус $\rho(\mathbf{S})$ матрицы шага \mathbf{S} равен

$$\rho(\mathbf{S}) = |\lambda_y| |\lambda_x| < 1 \quad \forall \kappa_x > 0, \kappa_y > 0, \theta \geq 1. \quad (34)$$

Поскольку спектральный радиус стационарного итерационного процесса (27) меньше единицы при любых $\kappa_y > 0, \kappa_x > 0, \theta \geq 1$, то итерации всегда сходятся.

Отметим, что ранее в работах [9, 10] дано доказательство сходимости метода итерируемой приближенной факторизации бикомпактных схем для нестационарного 2D линейного *однородного* уравнения переноса.

3.2. Трехмерная схема

Рассмотрим базовую бикомпактную схему (14), когда система ОДУ (8) интегрируется неявным методом Эйлера. Введем следующие операторы:

$$\begin{aligned}
B_1^r(s) &= A^r + \tau C_1^r(s), \quad B_2^r(s) = \Lambda_1^r + \tau C_2^r(s), \quad r = x, y, z, \\
C_1^r(s) &= \frac{1}{3} \sigma A^r + s \Lambda_1^r, \quad C_2^r(s) = \frac{1}{3} \sigma \Lambda_1^r + s \Lambda_2^r.
\end{aligned} \tag{35}$$

Операторы, стоящие в левых частях уравнений (14), допускают приближенную факторизацию

$$\begin{aligned}
(1 + \sigma \tau) A^z A^y A^x + a \tau A^z A^y \Lambda_1^x + b \tau A^z \Lambda_1^y A^x + c \tau \Lambda_1^z A^y A^x &= B_1^z(c) B_1^y(b) B_1^x(a) - \tau^2 I_1, \\
(1 + \sigma \tau) A^z A^y \Lambda_1^x + a \tau A^z A^y \Lambda_2^x + b \tau A^z \Lambda_1^y \Lambda_1^x + c \tau \Lambda_1^z A^y \Lambda_1^x &= B_1^z(c) B_1^y(b) B_2^x(a) - \tau^2 I_2, \\
(1 + \sigma \tau) A^z \Lambda_1^y A^x + a \tau A^z \Lambda_1^y \Lambda_1^x + b \tau A^z \Lambda_2^y A^x + c \tau \Lambda_1^z \Lambda_1^y A^x &= B_1^z(c) B_2^y(b) B_1^x(a) - \tau^2 I_3, \\
(1 + \sigma \tau) A^z \Lambda_1^y \Lambda_1^x + a \tau A^z \Lambda_1^y \Lambda_2^x + b \tau A^z \Lambda_2^y \Lambda_1^x + c \tau \Lambda_1^z \Lambda_1^y \Lambda_1^x &= B_1^z(c) B_2^y(b) B_2^x(a) - \tau^2 I_4, \\
(1 + \sigma \tau) \Lambda_1^z A^y A^x + a \tau \Lambda_1^z A^y \Lambda_1^x + b \tau \Lambda_1^z \Lambda_1^y A^x + c \tau \Lambda_2^z A^y A^x &= B_2^z(c) B_1^y(b) B_1^x(a) - \tau^2 I_5, \\
(1 + \sigma \tau) \Lambda_1^z A^y \Lambda_1^x + a \tau \Lambda_1^z A^y \Lambda_2^x + b \tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + c \tau \Lambda_2^z A^y \Lambda_1^x &= B_2^z(c) B_1^y(b) B_2^x(a) - \tau^2 I_6, \\
(1 + \sigma \tau) \Lambda_1^z \Lambda_1^y A^x + a \tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + b \tau \Lambda_1^z \Lambda_2^y A^x + c \tau \Lambda_2^z \Lambda_1^y A^x &= B_2^z(c) B_2^y(b) B_1^x(a) - \tau^2 I_7, \\
(1 + \sigma \tau) \Lambda_1^z \Lambda_1^y \Lambda_1^x + a \tau \Lambda_1^z \Lambda_1^y \Lambda_2^x + b \tau \Lambda_1^z \Lambda_2^y \Lambda_1^x + c \tau \Lambda_2^z \Lambda_1^y \Lambda_1^x &= B_2^z(c) B_2^y(b) B_2^x(a) - \tau^2 I_8,
\end{aligned} \tag{36}$$

где

$$\begin{aligned}
I_1 &= A^z C_1^y C_1^x + C_1^z A^y C_1^x + C_1^z C_1^y A^x + \tau C_1^z C_1^y C_1^x, \\
I_2 &= A^z C_1^y C_2^x + C_1^z A^y C_2^x + C_1^z C_1^y C_1^x + \tau C_1^z C_1^y C_2^x, \\
I_3 &= A^z C_2^y C_1^x + C_1^z C_1^y C_1^x + C_1^z C_2^y A^x + \tau C_1^z C_2^y C_1^x, \\
I_4 &= A^z C_2^y C_2^x + C_1^z C_1^y C_2^x + C_1^z C_2^y C_1^x + \tau C_1^z C_2^y C_2^x, \\
I_5 &= C_1^z C_1^y C_1^x + C_2^z A^y C_1^x + C_2^z C_1^y A^x + \tau C_2^z C_1^y C_1^x, \\
I_6 &= C_1^z C_1^y C_2^x + C_2^z A^y C_2^x + C_2^z C_1^y C_1^x + \tau C_2^z C_1^y C_2^x, \\
I_7 &= C_1^z C_2^y C_1^x + C_2^z C_1^y C_1^x + C_2^z C_2^y A^x + \tau C_2^z C_2^y C_1^x, \\
I_8 &= C_1^z C_2^y C_2^x + C_2^z C_1^y C_2^x + C_2^z C_2^y C_1^x + \tau C_2^z C_2^y C_2^x.
\end{aligned}$$

Итерационный алгоритм решения уравнений базовой схемы, основанный на приближенной факторизации (36) разностных операторов, дается формулами

$$\begin{aligned}
B_1^z(c) B_1^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= A^z A^y A^x (u_C + \tau \hat{q}_C) + \tau^2 I_1 \hat{u}_C^{(i)}, \\
B_1^z(c) B_1^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= A^z A^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 I_2 \hat{u}_C^{(i)}, \\
B_1^z(c) B_2^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= A^z \Lambda_1^y A^x (u_C + \tau \hat{q}_C) + \tau^2 I_3 \hat{u}_C^{(i)}, \\
B_1^z(c) B_2^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= A^z \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 I_4 \hat{u}_C^{(i)}, \\
B_2^z(c) B_1^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^z A^y A^x (u_C + \tau \hat{q}_C) + \tau^2 I_5 \hat{u}_C^{(i)}, \\
B_2^z(c) B_1^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^z A^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 I_6 \hat{u}_C^{(i)}, \\
B_2^z(c) B_2^y(b) B_1^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^z \Lambda_1^y A^x (u_C + \tau \hat{q}_C) + \tau^2 I_7 \hat{u}_C^{(i)}, \\
B_2^z(c) B_2^y(b) B_2^x(a) \hat{u}_C^{(i+1)} &= \Lambda_1^z \Lambda_1^y \Lambda_1^x (u_C + \tau \hat{q}_C) + \tau^2 I_8 \hat{u}_C^{(i)},
\end{aligned} \tag{37}$$

где i – номер итерации.

Алгоритм (37) можно записать в следующей матрично-векторной форме:

$$\mathbf{M}_z \mathbf{M}_y \mathbf{M}_x \hat{\mathbf{v}}^{(i+1)} = \mathbf{N} \hat{\mathbf{v}}^{(i)} + \boldsymbol{\Psi}, \quad \mathbf{v} = \begin{pmatrix} \bar{u}_{j+1/2, k+1/2, l+1/2}^{xyz} \\ \bar{u}_{j+1, k+1/2, l+1/2}^{yz} \\ \bar{u}_{j+1/2, k+1, l+1/2}^{xz} \\ \bar{u}_{j+1, k+1, l+1/2}^z \\ \bar{u}_{j+1/2, k+1/2, l+1}^{xy} \\ \bar{u}_{j+1, k+1/2, l+1}^y \\ \bar{u}_{j+1/2, k+1, l+1}^x \\ u_{j+1, k+1, l+1} \end{pmatrix}, \quad \boldsymbol{\Psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \end{pmatrix}, \quad (38)$$

$$\mathbf{M}_z = \mathbf{P}_z \otimes \mathbf{E} \otimes \mathbf{E}, \quad \mathbf{M}_y = \mathbf{E} \otimes \mathbf{P}_y \otimes \mathbf{E}, \quad \mathbf{M}_x = \mathbf{E} \otimes \mathbf{E} \otimes \mathbf{P}_x, \quad (39)$$

$$\mathbf{N} = \mathbf{E} \otimes (\mathbf{P}_y - \mathbf{E}) \otimes (\mathbf{P}_x - \mathbf{E}) + (\mathbf{P}_z - \mathbf{E}) \otimes \mathbf{E} \otimes (\mathbf{P}_x - \mathbf{E}) + (\mathbf{P}_z - \mathbf{E}) \otimes (\mathbf{P}_y - \mathbf{E}) \otimes \mathbf{E} + (\mathbf{P}_z - \mathbf{E}) \otimes (\mathbf{P}_y - \mathbf{E}) \otimes (\mathbf{P}_x - \mathbf{E}), \quad (40)$$

где $\boldsymbol{\Psi}$ – известный вектор-столбец, а матрицы \mathbf{P}_r ($r = x, y, z$) и \mathbf{E} определяются следующими формулами:

$$\mathbf{P}_r = \theta \mathbf{E} + \kappa_r \mathbf{R}, \quad \theta = 1 + \frac{1}{3} \sigma \tau, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -12 & 6 \end{bmatrix}. \quad (41)$$

В формулах (41) величины $\kappa_x = a\tau/h_x$, $\kappa_y = b\tau/h_y$, $\kappa_z = c\tau/h_z$ – числа Куранта.

Произведение матриц, стоящих в левой части уравнения (38), можно преобразовать к такому виду:

$$\mathbf{M}_z \mathbf{M}_y \mathbf{M}_x = \mathbf{P}_z \otimes \mathbf{P}_y \otimes \mathbf{P}_x.$$

Запишем итерационный метод (38) в рекуррентном виде

$$\hat{\mathbf{v}}^{(i+1)} = \mathbf{S} \hat{\mathbf{v}}^{(i)} + \boldsymbol{\xi}, \quad (42)$$

где матрица шага метода

$$\mathbf{S} = (\mathbf{M}_z \mathbf{M}_y \mathbf{M}_x)^{-1} \mathbf{N} = (\mathbf{P}_z \otimes \mathbf{P}_y \otimes \mathbf{P}_x)^{-1} \mathbf{N} = (\mathbf{P}_z^{-1} \otimes \mathbf{P}_y^{-1} \otimes \mathbf{P}_x^{-1}) \mathbf{N}. \quad (43)$$

С учетом формул (40), (43) матрица шага \mathbf{S} итерационного метода может быть преобразована к следующему виду:

$$\begin{aligned} \mathbf{S} = & \mathbf{T}_z \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x) + (\mathbf{E} - \mathbf{T}_z) \otimes \mathbf{T}_y \otimes (\mathbf{E} - \mathbf{T}_x) + \\ & + (\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes \mathbf{T}_x + (\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x), \end{aligned} \quad (44)$$

где $\mathbf{T}_r = \mathbf{P}_r^{-1}$, $r = x, y, z$.

Как видно из формулы (44), матрица \mathbf{S} есть сумма четырех матриц размера 8×8 , каждая из которых является кронекеровым произведением трех матриц размера 2×2 . В свою очередь, собственные векторы-столбцы матриц размера 8×8 в правой части (44) являются кронекеровыми произведениями собственных векторов-столбцов матриц размера 2×2 [13, с.596]. Поскольку собственные векторы-столбцы матриц \mathbf{T}_r и $\mathbf{E} - \mathbf{T}_r$ размера 2×2 совпадают, то совпадают и собственные векторы-столбцы всех четырех матриц размера 8×8 в правой части (44). Вследствие этого собственные значения $\lambda(\mathbf{S})$ матрицы шага \mathbf{S} равны сумме собственных значений четырех матриц размера 8×8 в правой части (44)

$$\begin{aligned} \lambda(\mathbf{S}) = & \lambda(\mathbf{T}_z \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x)) + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes \mathbf{T}_y \otimes (\mathbf{E} - \mathbf{T}_x)) + \\ & + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes \mathbf{T}_x) + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x)) = \\ & = (1 - \lambda_z)\lambda_y\lambda_x + \lambda_z(1 - \lambda_y)\lambda_x + \lambda_z\lambda_y(1 - \lambda_x) + \lambda_z\lambda_y\lambda_x, \end{aligned} \quad (45)$$

где λ_r – собственные значения матрицы $\mathbf{E} - \mathbf{T}_r$. При написании последнего равенства в формуле (45) было использовано свойство собственных значений матрицы, являющейся кронекеровым произведением матриц. Поскольку у матрицы $\mathbf{E} - \mathbf{T}_r$ имеются два комплексно сопряженных собственных значения

$$\lambda_r = \frac{12\kappa_r^2 + (6\theta - 3 \pm i\sqrt{3})\kappa_r + \theta(\theta - 1)}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}, \quad (46)$$

то число различных собственных значений матрицы \mathbf{S} , определяемых формулой (45), равно восьми. Из (46) следуют формулы

$$\begin{aligned} \operatorname{Re}(\lambda_r) = & \frac{12\kappa_r^2 + (6\theta - 3)\kappa_r + \theta(\theta - 1)}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2} = 1 - \frac{3\kappa_r + \theta}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}, \\ \operatorname{Im}(\lambda_r) = & \pm \frac{\sqrt{3}\kappa_r}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}, \\ |\lambda_r|^2 = & \frac{(12\kappa_r^2 + (6\theta - 3)\kappa_r + \theta(\theta - 1))^2 + 3\kappa_r^2}{(12\kappa_r^2 + 6\theta\kappa_r + \theta^2)^2} = 1 - \frac{6\kappa_r + 2\theta - 1}{12\kappa_r^2 + 6\theta\kappa_r + \theta^2}. \end{aligned} \quad (47)$$

Вычислим квадрат модуля собственного значения $\lambda(\mathbf{S})$

$$\begin{aligned}
|\lambda(\mathbf{S})|^2 &= \lambda(\mathbf{S})\bar{\lambda}(\mathbf{S}) = (1-4\operatorname{Re}\lambda_z)|\lambda_y|^2|\lambda_x|^2 + (1-4\operatorname{Re}\lambda_y)|\lambda_z|^2|\lambda_x|^2 + \\
&+ (1-4\operatorname{Re}\lambda_x)|\lambda_z|^2|\lambda_y|^2 + 4|\lambda_z|^2|\lambda_y|^2|\lambda_x|^2 + 2(\operatorname{Re}\lambda_z\operatorname{Re}\lambda_y + \operatorname{Im}\lambda_z\operatorname{Im}\lambda_y)|\lambda_x|^2 + \\
&+ 2(\operatorname{Re}\lambda_z\operatorname{Re}\lambda_x + \operatorname{Im}\lambda_z\operatorname{Im}\lambda_x)|\lambda_y|^2 + 2(\operatorname{Re}\lambda_y\operatorname{Re}\lambda_x + \operatorname{Im}\lambda_y\operatorname{Im}\lambda_x)|\lambda_z|^2.
\end{aligned} \quad (48)$$

Используя формулы (47), (48), найдем квадрат спектрального радиуса матрицы \mathbf{S}

$$\begin{aligned}
\rho^2(\mathbf{S}) &= \max |\lambda(\mathbf{S})|^2 = 1 - \frac{f(\kappa_z, \kappa_y, \kappa_x)}{(12\kappa_z^2 + 6\theta\kappa_z + \theta^2)(12\kappa_y^2 + 6\theta\kappa_y + \theta^2)(12\kappa_x^2 + 6\theta\kappa_x + \theta^2)}, \\
f(\kappa_z, \kappa_y, \kappa_x) &= -144\kappa_z\kappa_y\kappa_x(\kappa_z + \kappa_y + \kappa_x) - \\
&- 72\theta \left[3\kappa_z\kappa_y\kappa_x + (\kappa_y + \kappa_x)\kappa_z^2 + (\kappa_z + \kappa_x)\kappa_y^2 + (\kappa_z + \kappa_y)\kappa_x^2 \right] - \\
&- 12 \left[(2\theta^2 - 1)(\kappa_z^2 + \kappa_y^2 + \kappa_x^2) + (7\theta^2 - 2\theta - 2)(\kappa_z\kappa_y + \kappa_z\kappa_x + \kappa_y\kappa_x) \right] - \\
&- 6(4\theta^3 - 2\theta^2 - 3\theta + 2)(\kappa_z + \kappa_y + \kappa_x) - (3\theta - 2)(2\theta^3 - 3\theta + 2).
\end{aligned} \quad (49)$$

Из формулы (49) следует, что спектральный радиус матрицы шага стационарного итерационного процесса (42) меньше единицы при любых $\kappa_z > 0$, $\kappa_y > 0$, $\kappa_x > 0$, $\theta \geq 1$, поэтому итерации всегда сходятся.

Отметим, что ранее в работе [10] дано доказательство сходимости метода итерируемой приближенной факторизации бикомпактных схем для нестационарного 3D линейного *однородного* уравнения переноса.

4. Параллельная реализация вычислительных алгоритмов

Несмотря на то что бикомпактные схемы являются неявными, благодаря логической простоте они могут быть эффективно решены методом бегущего счёта, который допускает эффективное распараллеливание. Параллельный алгоритм бегущего счёта является методом волновой обработки данных и использует конвейерную загрузку процессоров. Алгоритм может быть реализован и на системах с общей памятью, и на системах с распределённой памятью. Стоит подчеркнуть, что компактность шаблона бикомпактной схемы приводит к минимальному объёму данных, которыми должны обмениваться процессоры в случае распределённой памяти.

Кроме того, факторизованный вариант схемы также может быть распараллелен с большей эффективностью, поскольку одномерные задачи вдоль каждой сеточной линии могут решаться независимо.

В данной главе описаны параллельные реализации исходной и факторизованной схемы. Эффективность параллельной обработки и масштабируемость обоих алгоритмов сравниваются на решении тестовой

начально-краевой задачи для двухмерного нестационарного уравнения переноса с жёстким источниковым членом.

4.1. Параллельный алгоритм бегущего счёта

Рассмотрим параллельную реализацию алгоритма бикompактной схемы для скалярного неоднородного уравнения переноса на системах с общей памятью с использованием технологии OpenMP. Последовательность вычислений в параллельном алгоритме схематично изображена на рис. 1.

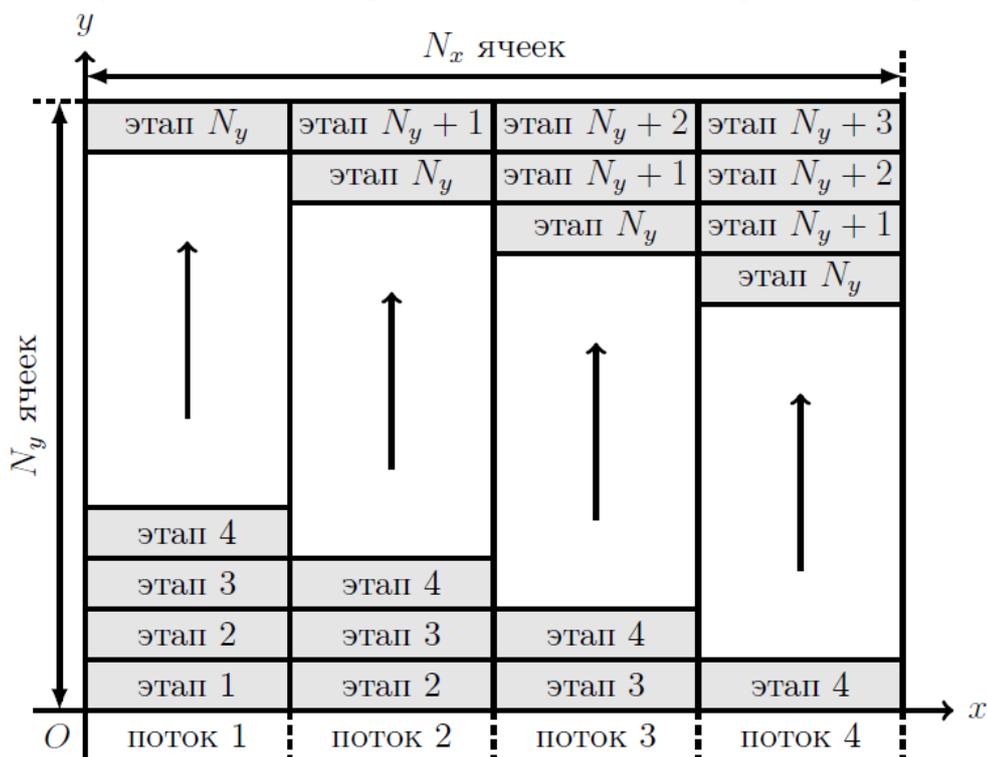


Рис. 1. Схема параллельного бегущего счёта для четырех потоков

Пусть расчётная область состоит из $N_x \times N_y$ ячеек, число потоков равно p . Разобьём расчётную сетку по направлению x на p частей. Будем считать, что N_x / p – целое число. Процесс расчёта на одном шаге по времени внутри каждой стадии метода Рунге-Кутты представим в виде последовательности этапов. На каждом этапе каждый из задействованных потоков выполняет вычисления в блоке сетки размера $(N_x / p) \times 1$, вычисления выполняются всеми потоками параллельно и независимо. Следующий этап начинается, когда все потоки выполнят все действия на предыдущем этапе, т.е. в конце каждого этапа необходим барьер для потоков. Последовательность этапов, изображённая на рис. 1, гарантирует, что на каждом этапе все граничные значения в каждом блоке, нужные для решения бегущим счётом, либо известны из граничных условий, либо уже вычислены ранее.

Можно грубо оценить эффективность распараллеливания, пренебрегая потерями из-за синхронизации потоков. Обозначим время расчёта в одной ячейке T_1 . Тогда время работы последовательного алгоритма на одном шаге по времени равно $N_x N_y T_1$. Число этапов в параллельном методе равно $N_y + p - 1$, каждый поток рассчитывает N_x / p ячеек, следовательно, время расчёта равно $T_1(N_y + p - 1)N_x$, а ускорение S равно:

$$S = \frac{N_x N_y T_1}{T_1(N_y + p - 1)N_x / p} = \frac{p}{1 + (p - 1) / N_y}. \quad (50)$$

Эта формула показывает, что даже если отношение p / N_y не очень мало, то теоретическое ускорение близко к идеальному.

Следует отметить, что размер блока может подбираться в зависимости от размера кэша и других параметров вычислительного устройства для улучшения масштабируемости вычислений.

Для проверки масштабируемости параллельного алгоритма была рассчитана тестовая начально-краевая задача:

$$\begin{aligned} u_t + au_x + bu_y + \sigma u - q &= 0, (x, y) \in [0, 1] \times [0, 1], t \in [0, T], \\ u(x, y, 0) &= u^0(x, y), u(0, y, t) = 0, u(x, 0, t) = 0 \end{aligned} \quad (51)$$

с начальным условием в виде финитного гладкого импульса

$$u^0(x, y) = \begin{cases} (1 - (4x - 1)^2)^7 (1 - (4y - 1)^2)^6, & |4x - 1| \leq 1, |4y - 1| \leq 1, \\ 0, & |4x - 1| > 1, |4y - 1| > 1. \end{cases} \quad (52)$$

с $a = b = 1, T = 0.5, \sigma = 1$, с источником $q = \sigma u^0(x - at, y - bt)$ и точным решением $u^{ex}(x, y, t) = u^0(x - at, y - bt)$ на сетке из 2048×2048 ячеек. Для интегрирования по времени использовался метод ESDIRK75 5-го порядка точности из [11].

Тестовые расчёты были проведены на вычислительном узле системы МВС-10П МСЦ РАН, состоящем из двух 8-ядерных процессоров Xeon E5-2690, а также на сопроцессоре Intel Xeon Phi 5110p лаборатории математического моделирования нелинейных процессов в газовых средах МФТИ. Сопроцессор Intel Xeon Phi 5110p содержит 61 физическое ядро, на каждом из которых может выполняться 4 потока.

На рис. 2 показана зависимость времени расчёта одного шага по времени от числа потоков в логарифмической шкале. Ускорение алгоритма в обоих случаях близко к идеальному. На процессоре Xeon E5-2690 на 16 потоках ускорение равно 11, эффективность распараллеливания примерно 70%, на сопроцессоре Intel Xeon Phi 5110p на 60 потоках ускорение равно 47, эффективность распараллеливания около 80%. На основании этих данных можно сделать вывод, что параллельный алгоритм бегущего счёта хорошо

масштабируется на системах с общей памятью. Кроме того, простая логическая структура алгоритма позволяет легко использовать векторизацию, что потенциально может ускорить счёт в 4 раза на процессорах Xeon и в 8 раз на процессорах Xeon Phi.

Описанная схема алгоритма почти без изменений распространяется на трехмерный случай: расчётная сетка разбивается, например, по переменной z на двухмерные слои, на каждом из которых вычисления выполняются в порядке, изображённом на рис. 1. Отличие будет только в том, что в каждой ячейке будет решаться линейная система из 8 уравнений.

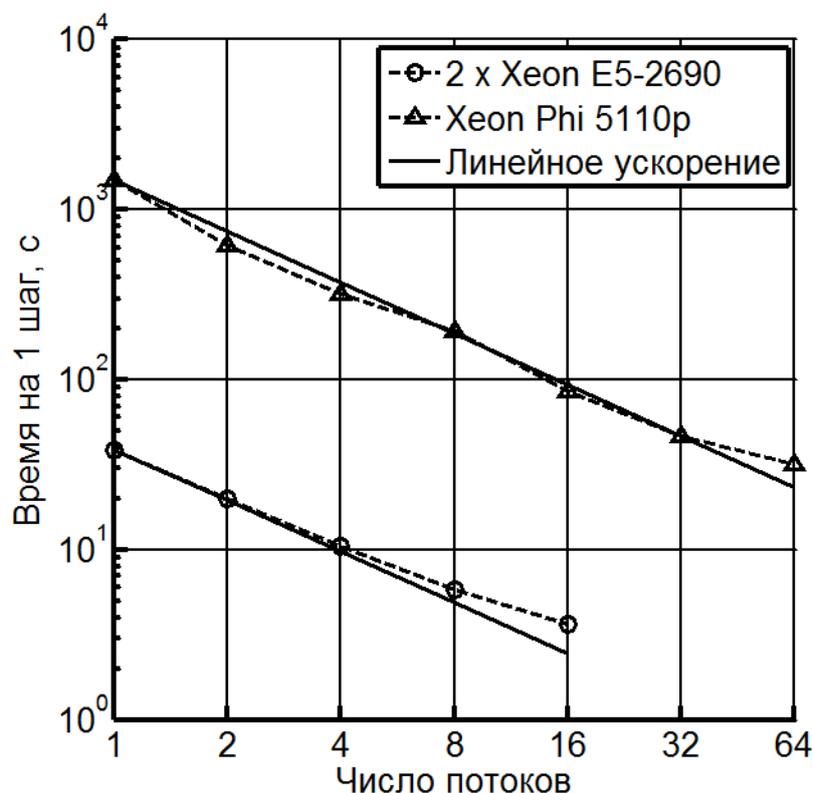


Рис. 2. Зависимость времени на один шаг от числа потоков

4.2. Параллельный алгоритм счета по факторизованной схеме

На каждой итерации в факторизованной 2D схеме можно независимо решать методом бегущего счета одномерные задачи (21) вдоль каждого слоя ячеек $j = \text{const}, j = \overline{1, N_x}$, а одномерные задачи (22), (23) – вдоль слоя ячеек $k = \text{const}, k = \overline{1, N_y}$. Алгоритмически эти операции представляют собой одномерные циклы. На системах с общей памятью их можно сделать параллельными с помощью технологии OpenMP добавлением простой директивы `!$omp parallel do`. Если число ячеек по каждому направлению кратно числу потоков, или если не кратно, но значительно больше числа потоков, то можно ожидать хорошей балансировки нагрузки и хорошей масштабируемости алгоритма.

Масштабируемость параллельного алгоритма факторизованной схемы была изучена на тестовой задаче (51)-(52). На практике для подобных задач используются относительно небольшие сетки с числом ячеек $\leq 10^2$ по каждому направлению, что в трёхмерном случае даёт сетку с числом ячеек 10^6 . Поэтому важно сравнить масштабируемость параллельного алгоритма бегущего счёта и параллельного алгоритма факторизованной схемы на таких сетках: в последовательном случае вычисления по факторизованной схеме занимают больше времени, чем бегущий счёт, из-за большого числа итераций; но параллельный алгоритм бегущего счёта плохо масштабируется, когда отношение числа потоков к числу ячеек по одному направлению p / N_x велико.

В факторизованной схеме критерий останова итераций был таким: на каждом шаге s метода Рунге-Кутты, при решении системы ОДУ метода прямых с правой частью f , разность величин $k^s = f(t + c_s \tau, u^n + \tau \sum_{q=1}^s a_{sq} k_q)$ на текущей (i)-ой и предыдущей ($i-1$)-ой итерациях должна быть меньше по модулю заданного параметра tol :

$$\max_{j,k} |k_{jk}^{s,(i)} - k_{jk}^{s,(i-1)}| < tol, \quad (53)$$

где c_s, a_{sq} – коэффициенты метода Рунге-Кутты (см. Приложение). В качестве начального приближения использовались нулевые значения: k_{jk}^s .

Подобные особенности алгоритмов особенно важны при использовании современных вычислительных архитектур, в которых число потоков становится большим, а производительность отдельного ядра (вычислителя) относительно небольшая. Например, процессоры Intel последнего поколения Xeon Phi имеют от 61 до 72 ядер на процессор, причем на каждом ядре может выполняться до 4 потоков. При этом такие процессоры обладают небольшой производительностью и небольшой памятью, по сравнению с процессорами с меньшим числом ядер (8-12). Поэтому для того, чтобы в полной мере использовать возможности таких процессоров, нужны алгоритмы, которые хорошо масштабируются на задачах небольшого размера.

На рис. 3 показана зависимость времени расчёта задачи от числа потоков для алгоритма бегущего счёта и алгоритма факторизованной схемы при расчёте на 12-ядерном процессоре Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz. Задача (104)-(105) решалась на сетке 48×48 ячеек, число шагов по времени – 48, использовался метод SDIRK54 [12]. Параметр $tol = 10^{-2}$, среднее число итераций для достижения заданной точности равно 3.65. При этом факторизованная схема даёт такую же ошибку, как и обычная схема $\|Err\|_{\infty} = 4.0 \times 10^{-4}$.

Как и следовало ожидать, алгоритм бегущего счёта плохо масштабируется на такой мелкой сетке, т.к. отношение числа потоков к числу

ячеек по каждому направлению достаточно велико (0.25 для 12 потоков). Эффективность параллельного исполнения составляет 40% на 12 потоках.

Алгоритм факторизованной схемы показывает хорошую масштабируемость до 12 потоков, эффективность равна 74%. Последовательный алгоритм факторизованной схемы в 1.6 раз медленнее последовательного алгоритма бегущего счёта. Но, благодаря хорошей масштабируемости, на 12 потоках факторизованная схема считает быстрее, чем бегущий счёт.

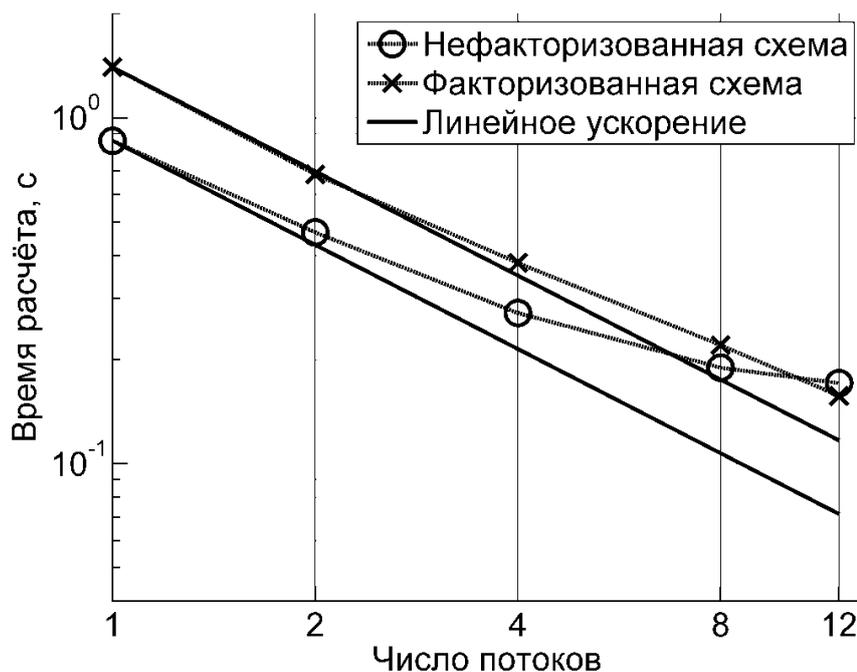


Рис. 3. Зависимость времени расчёта от числа потоков для исходной и факторизованной схемы

На следующем рис. 4 показана зависимость времени расчёта задачи (51), (52) на сетке 60×60 ячеек, число шагов по времени равно 60 от числа потоков для алгоритма бегущего счёта и алгоритма факторизованной схемы при расчёте на процессоре Intel Xeon Phi 5110p, который содержит 61 физическое ядро, на каждом из которых может исполняться до 4 потоков. Параметр $tol = 10^{-2}$, среднее число итераций для достижения заданной точности равно 2.77. При этом факторизованная схема даёт такую же ошибку, как и обычная схема $\|Err\|_{\infty} = 1.6 \times 10^{-4}$.

В этом примере отношение максимального числа потоков к числу ячеек по каждому направлению равно 1, поэтому алгоритм бегущего счёта масштабируется только до 12 потоков. Эффективность составляет 50% на 12 потоках и 4% на 60 потоках. Это означает, что параллельный алгоритм бегущего счёта не позволяет использовать всю производительность процессора Xeon Phi при решении подобных задач.

Алгоритм факторизованной схемы показывает удовлетворительную масштабируемость, эффективность равна 50% на 60 потоках. На 1 ядре

алгоритм факторизованной схемы в 1.6 раз медленнее последовательного алгоритма бегущего счёта. На 60 ядрах время расчёта по факторизованной схеме в 3 раза меньше, чем наименьшее время расчёта для бегущего счёта (на 12 ядрах).

Данные модельные расчёты показывают, что с точки зрения параллельной реализации более предпочтительной является факторизованная схема: во-первых, факторизованный последовательный алгоритм легко распараллеливается, во-вторых, на сетках среднего размера (которые чаще всего используются на практике) расчёт по факторизованной схеме занимает меньше времени. Стоит отметить, что при решении трехмерных задач выигрыш будет ещё больше.

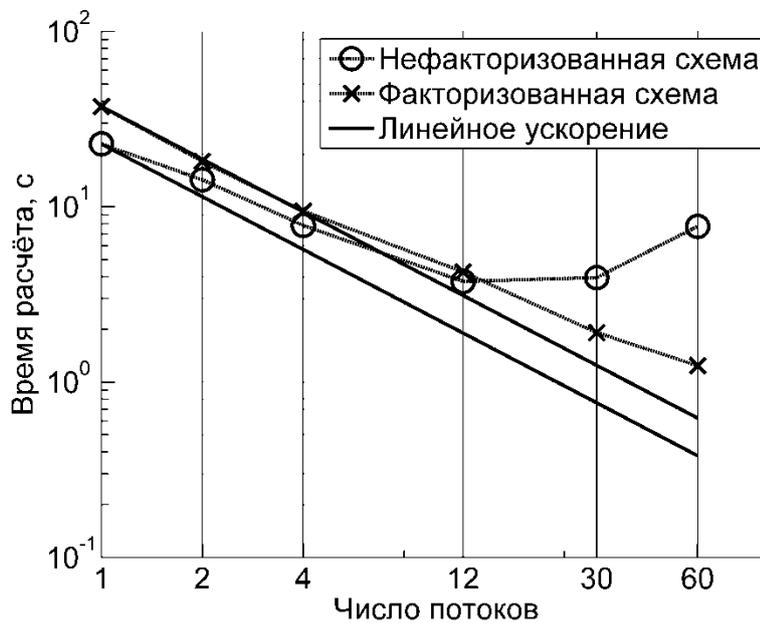


Рис. 4. Зависимость времени расчёта от числа потоков для исходной и факторизованной схем

5. Заключение

В работе проведено сравнение эффективности двух параллельных алгоритмов решения уравнений высокоточных бикомпактных схем для многомерного неоднородного уравнения переноса. Первый из них есть маршевый по пространству алгоритм счета нефакторизованных схем, а второй основан на приближенной факторизации многомерных схем. В последнем алгоритме используются итерации для сохранения высокого (выше второго) порядка точности бикомпактных схем по времени. Доказана сходимость этих итераций для нестационарного двухмерного и трехмерного линейного неоднородного уравнения переноса с постоянными положительными коэффициентами. Модельные расчёты показывают, что алгоритм на основе факторизованных бикомпактных схем является предпочтительным с точки зрения параллельной реализации.

Приложение. Коэффициенты диагонально неявных методов Рунге-Кутты

Таблица коэффициентов Бутчера DIRK метода четвертого порядка с пятью неявными стадиями SDIRK54 [12]:

		1/4	1/4				
		0	-1/4	1/4			
c	A	1/2	1/8	1/8	1/4		
	$b^T =$	1	-3/2	3/4	3/2	1/4	
		1	0	1/6	2/3	-1/12	1/4
			0	1/6	2/3	-1/12	1/4

Таблица коэффициентов Бутчера DIRK метода пятого порядка с шестью неявными стадиями и одной явной стадией ESDIRK75 [11]:

		0	0						
		0.4	0.2	0.2					
		0.6	0.25	0.15	0.2				
c	A	1	0.6214285714	-0.9642857142	1.1428571428	0.2			
	$b^T =$	0.8	0.2657743915	0.4927153439	-0.3035089947	0.1450192592	0.2		
		0.2	0.0448367885	-0.2368271451	0.2540183114	-0.1402888244	0.0782608695	0.2	
		1	0.0659722222	0.1736111111	0.1736111111	-0.1340277777	0.2604166666	0.2604166666	0.2
			0.0659722222	0.1736111111	0.1736111111	-0.1340277777	0.2604166666	0.2604166666	0.2

Библиографический список

1. Шильков А.В., Герцев М.Н., Аристова Е.Н., Шилькова С.В. Методика эталонных «line-by-line» расчетов атмосферной радиации // Компьютерные исследования и моделирование. 2012. Т. 4, № 3. С. 553-562.
2. Розанов В.Б., Вергунова Г.А., Аристова Е.Н. и др. Взаимодействие лазерного излучения с малоплотным структурированным абсорбером // ЖЭТФ. 2016. Т. 149. № 2. С. 294–319.
3. Байдин Д.Ф., Аристова Е.Н. Параллельный код QuDiff для расчета критических параметров реактора на быстрых нейтронах в трехмерной гексагональной геометрии // Мат. моделирование. 2016. Т. 28, № 1. С. 107–116.
4. Рогов Б.В., Михайловская М.Н. Бикомпактные схемы четвертого порядка аппроксимации для гиперболических уравнений // Докл. АН. 2010. Т.430, №4. С.470-474.
5. Аристова Е.Н., Рогов Б.В. О реализации граничных условий в бикомпактных схемах для линейного уравнения переноса // Мат. моделирование. 2012. Т. 24, № 10. С. 3-14.
6. Аристова Е.Н., Рогов Б.В., Чикиткин А.В. Оптимальная монотонизация высокоточной бикомпактной схемы для нестационарного многомерного

уравнения переноса // Ж. вычисл. матем. и матем. физ. 2016. Т. 56, № 6. С. 973–988.

7. Чикиткин А.В., Рогов Б.В., Аристова Е.Н. Высокоточные бикомпактные схемы для многомерного неоднородного уравнения переноса и их эффективная параллельная реализация // Докл. АН. 2016. Т. 470. № 2. С.144–149.

8. Брагин М.Д., Рогов Б.В. Гибридные бикомпактные схемы с минимальной диссипацией для уравнений гиперболического типа // Ж. вычисл. матем. и матем. физ. 2016. Т. 56, № 6. С. 958-972.

9. Брагин М.Д., Рогов Б.В. Метод итерируемой приближенной факторизации операторов высокоточной бикомпактной схемы для систем многомерных неоднородных квазилинейных уравнений гиперболического типа // Ж. вычисл. матем. и матем. физ. 2018. Т. 58, № 3. С. 313-325.

10. Рогов Б.В., Брагин М.Д. О сходимости метода итерируемой приближенной факторизации операторов высокоточной бикомпактной схемы для нестационарных трехмерных уравнений гиперболического типа // Препринты ИПМ им. М.В.Келдыша. 2018. № 132. 16 с. doi:10.20948/prepr-2018-132. URL: <http://library.keldysh.ru/preprint.asp?id=2018-132>

11. Скворцов Л.М. Диагонально неявные FSAL-методы Рунге-Кутты для жестких и дифференциально-алгебраических систем // Мат. моделирование. 2002. Т. 14, № 2. С. 3-17.

12. Скворцов Л.М. Диагонально-неявные методы Рунге-Кутты для жестких задач // Ж. вычисл. матем. и матем. физ. 2006. Т. 46, № 12. С. 2209–2222.

13. Уоткинс Д. Основы матричных вычислений. Пер. с англ. М.: БИНОМ. Лаборатория знаний, 2006. 664 с.

Оглавление

1. Введение	3
2. Бикомпактные схемы	4
3. Метод итерируемой приближенной факторизации	8
3.1. Двухмерная схема	9
3.2. Трехмерная схема	12
4. Параллельная реализация вычислительных алгоритмов	16
4.1. Параллельный алгоритм бегущего счёта	17
4.2. Параллельный алгоритм счета по факторизованной схеме	19
5. Заключение	22
Приложение. Коэффициенты диагонально неявных методов Рунге-Кутты	23
Библиографический список	23