



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

[Rogov B.V.](#), [Bragin M.D.](#)

On the convergence of the method of iterative approximate factorization of difference operators of high-order accurate bicomact scheme for nonstationary three-dimensional hyperbolic equations

Recommended form of bibliographic references: Rogov B.V., Bragin M.D. On the convergence of the method of iterative approximate factorization of difference operators of high-order accurate bicomact scheme for nonstationary three-dimensional hyperbolic equations // Keldysh Institute Preprints. 2018. No. 132. 16 p. doi:[10.20948/prepr-2018-132-e](https://doi.org/10.20948/prepr-2018-132-e)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-132&lg=e>

KELDYSH INSTITUTE OF APPLIED MATHEMATICS

Russian Academy of Sciences

B.V.Rogov, M.D.Bragin

**On the convergence of the method
of iterative approximate factorization
of difference operators of high-order
accurate bicomact scheme
for nonstationary three-dimensional
hyperbolic equations**

Moscow — 2018

Boris Vadimovich Rogov, Mikhail Dmitrievich Bragin

On the convergence of the method of iterative approximate factorization of difference operators of high-order accurate bicomact scheme for nonstationary three-dimensional hyperbolic equations

The convergence of the iterative approximate factorization method for operators of the bicomact scheme for numerical solution of hyperbolic equations is investigated. In this method, iterations are made in order to eliminate the approximate factorization error from the numerical solution. Iterations convergence is studied in case of the bicomact scheme of fourth order in space for the non-stationary three-dimensional linear advection equation with constant positive coefficients. It is proved that iterations converge for all positive Courant numbers in all three space dimensions. A theoretical estimate for convergence rate of iterations is obtained. This estimate is confirmed in a numerical experiment for different relations between Courant numbers.

Key words: bicomact schemes, hyperbolic equations, iterative approximate factorization method, iterations convergence

Рогов Б.В., Брагин М.Д.

О сходимости метода итерируемой приближенной факторизации операторов высокоточной бикомпактной схемы для нестационарных трехмерных уравнений гиперболического типа

В работе исследована сходимость метода итерируемой приближенной факторизации операторов бикомпактной схемы для численного решения нестационарных трехмерных уравнений гиперболического типа. В этом методе итерации проводятся с целью исключения из численного решения ошибки, связанной с приближенной факторизацией. Сходимость итераций изучена применительно к бикомпактной схеме четвертого порядка пространственной аппроксимации для нестационарного трехмерного линейного уравнения переноса с постоянными положительными коэффициентами. Доказано, что итерации сходятся при всех положительных числах Куранта по всем трем пространственным направлениям. Получена теоретическая оценка для скорости сходимости итераций. Проведен численный эксперимент, который подтвердил эту оценку при различных соотношениях между числами Куранта.

Ключевые слова: бикомпактные схемы, уравнения гиперболического типа, метод итерируемой приближенной факторизации, сходимость итераций

This research was supported by the Russian Foundation for Basic Research, projects nos. 18-01-00857-a and 18-31-00045-mol_a.

Introduction

Numerous models used to solve problems in physics and engineering rely on hyperbolic equations. They describe transport processes, wave propagation, etc. Recently, much attention has been given to the construction and development of compact schemes for the numerical solution of hyperbolic equations [1]. This interest is motivated by stricter requirements imposed on the reliability, accuracy (including spectral one), and efficiency of computations. Compact schemes satisfy all these requirements (see [1]). When considered on the same stencil, they provide a higher order of accuracy (see [1]) than classical schemes constructed using finite difference approximations for derivatives involved in differential equations. Moreover, schemes constructed on a compact stencil can be implemented by applying efficient methods (see [1]).

In recent years, a special class of compact schemes, namely, bcompact schemes [3–10] has been actively developed as applied to the numerical solution of hyperbolic equations of the first order in space variables. Inheriting positive properties of compact schemes, they possess the best spectral resolution [5–7] as compared with conventional compact schemes [1] of the same or even higher order of accuracy. Bcompact schemes are constructed by applying the method of lines with the use of the finite-volume method. First, semidiscrete bcompact schemes are derived, and then the ordinary differential equations (ODEs) in these schemes are integrated with respect to time usually by applying the trapezoidal rule [5] or A- and L-stable diagonally implicit Runge–Kutta methods [2–4, 6–10]. An important difference between bcompact and other compact schemes is that the spatial stencil in each spatial direction in the former consists of only two integer nodes. It is due to this feature that bcompact schemes received their name. Their stencil can involve auxiliary fractional nodes lying symmetrically about the stencil center; with the help of these nodes, the order of the spatial approximation of a bcompact scheme can be increased up to any even order greater than two [6]. The difference order of a bcompact scheme with respect to a space variable is defined as the difference between the total number of nodes in the stencil and the total number of difference equations in the scheme and is equal to unity, i.e., coincides with the order of the approximated differential equation in the same variable. Semidiscrete bcompact schemes can be obtained using two approaches, namely, by involving differential consequences of the original differential equation [3–5, 8–10] and using the finite-volume method or by applying collocation polynomials [6, 7]. The former approach to the construction of bcompact schemes implies that they are symmetric, conservative, and preserve a high order of accuracy in space on highly nonuniform grids [3, 4].

Another important advantage of bcompact schemes over traditional three-point compact schemes of the same even order of accuracy in space is concerned with the computation of discontinuous solutions to hyperbolic equations. It was shown in [10] that a bcompact scheme of fourth-order accuracy in space generates local nonmonotonocities of small amplitude near a contact discontinuity, whereas a

traditional fourth-order accurate compact scheme generates extended nonmonotonicity zones of large amplitude near a discontinuity.

An efficient method for implementing two- and three-dimensional (2D and 3D) bicomact schemes for solving systems of nonstationary inhomogeneous quasilinear hyperbolic equations was proposed in [8, 9]. The method relies on an approximate factorization of the operators of a multidimensional bicomact scheme and involves iterations performed to eliminate the error of the approximate factorization from the numerical solution. Note that, first, the factorization of the multidimensional operators of a bicomact scheme is possible due to its symmetry with respect to the space variables and the commutation of the one-dimensional difference operators. Second, the maximum order of accuracy in time that can be obtained by applying the classical approximate factorization is equal to two (see the introductions in [8, 9]), while the iterative approximate factorization method [8, 9] preserves the order of accuracy in time in the factorized scheme equal to the order of accuracy in the unfactorized scheme. The iterative approximate factorization method in the 2D case was described in [8], while in [9] it was considered in the 2D and 3D cases. The convergence of iterations of the method was proved in detail in [9] as applied to the nonstationary 2D scalar homogeneous linear advection equation with positive constant coefficients. However, the convergence of the iterations in the 3D case remained unstudied.

In this paper, the convergence of the iterative approximate factorization method in the 3D case is proved as applied to the linear advection equation with positive constant coefficients. The proof technique differs from that used in [9]. For this reason, before presenting the new proof technique in the 3D case, we describe its counterpart in the 2D case. The consideration of the 2D case is also motivated by the subsequent comparison of the convergence rate estimates obtained in the 2D and 3D cases.

Convergence analysis in the two-dimensional case

For the 2D scalar homogeneous linear advection equation

$$L_2(u) \equiv \partial_t u + a \partial_x u + b \partial_y u = 0, \quad a = \text{const} > 0, \quad b = \text{const} > 0, \quad (1)$$

the equations of a semidiscrete bicomact scheme of fourth-order accuracy in space can be obtained by integration over the 2D cell $[x_j, x_{j+1}] \times [y_k, y_{k+1}]$ and the subsequent approximation of the original equation (1) and its three independent differential consequences

$$\partial_x L_2(u) = 0, \quad \partial_y L_2(u) = 0, \quad \partial_x \partial_y L_2(u) = 0,$$

where the symbols ∂_t , ∂_x , and ∂_y denote $\partial/\partial t$, $\partial/\partial x$, and $\partial/\partial y$, respectively. The equations of the semidiscrete bicomact scheme have the form

$$\begin{aligned}
A_0^y A_0^x (du/dt)_C + (aA_0^y \Lambda_1^x + b\Lambda_1^y A_0^x) u_C &= 0, \\
A_0^y \Lambda_1^x (du/dt)_C + (aA_0^y \Lambda_2^x + b\Lambda_1^y \Lambda_1^x) u_C &= 0, \\
\Lambda_1^y A_0^x (du/dt)_C + (a\Lambda_1^y \Lambda_1^x + b\Lambda_2^y A_0^x) u_C &= 0, \\
\Lambda_1^y \Lambda_1^x (du/dt)_C + (a\Lambda_1^y \Lambda_2^x + b\Lambda_2^y \Lambda_1^x) \hat{u}_C &= 0,
\end{aligned} \tag{2}$$

where $C = (j+1/2, k+1/2)$ is a multi-index. The one-dimensional difference operators A_0^d , Λ_1^d , Λ_2^d ($d = x, y$) in Eqs. (2) for an arbitrary grid function u are given by the formulas

$$\begin{aligned}
A_0^x u_{j+1/2, k'} &= \frac{u_{j, k'} + 4u_{j+1/2, k'} + u_{j+1, k'}}{6}, & \Lambda_1^x u_{j+1/2, k'} &= \frac{u_{j+1, k'} - u_{j, k'}}{h_{x, j+1/2}}, \\
\Lambda_2^x u_{j+1/2, k'} &= \frac{4(u_{j, k'} - 2u_{j+1/2, k'} + u_{j+1, k'})}{h_{x, j+1/2}^2}, & k' \in \{k, k+1/2, k+1\}, \\
A_0^y u_{j', k+1/2} &= \frac{u_{j', k} + 4u_{j', k+1/2} + u_{j', k+1}}{6}, & \Lambda_1^y u_{j', k+1/2} &= \frac{u_{j', k+1} - u_{j', k}}{h_{y, k+1/2}}, \\
\Lambda_2^y u_{j', k+1/2} &= \frac{4(u_{j', k} - 2u_{j', k+1/2} + u_{j', k+1})}{h_{y, k+1/2}^2}, & j' \in \{j, j+1/2, j+1\},
\end{aligned} \tag{3}$$

where $h_{x, j+1/2} = x_{j+1} - x_j$ and $h_{y, k+1/2} = y_{k+1} - y_k$ are the stepsizes in the space variables x and y . For notational brevity, $h_{x, j+1/2}$ and $h_{y, k+1/2}$ are hereafter designated as h_x and h_y . Note that the operators A_0^x , Λ_1^x , and Λ_2^x are pairwise commuting with the operators A_0^y , Λ_1^y , and Λ_2^y , which can be seen from their definition (3).

As a time stepping method in the fully discrete bicomcompact scheme, we use the A-stable single-stage trapezoidal rule or A- and L-stable diagonally implicit multistage Runge–Kutta methods of high-order accuracy. At each stage, these time stepping methods are reduced to integrating the equations of the semidiscrete bicomcompact scheme by the implicit Euler method with a different stage time step. Accordingly, without loss of generality, we consider the baseline bicomcompact scheme

$$\begin{aligned}
(A_0^y A_0^x + \tau a A_0^y \Lambda_1^x + \tau b \Lambda_1^y A_0^x) \hat{u}_C &= A_0^y A_0^x u_C, \\
(A_0^y \Lambda_1^x + \tau a A_0^y \Lambda_2^x + \tau b \Lambda_1^y \Lambda_1^x) \hat{u}_C &= A_0^y \Lambda_1^x u_C, \\
(\Lambda_1^y A_0^x + \tau a \Lambda_1^y \Lambda_1^x + \tau b \Lambda_2^y A_0^x) \hat{u}_C &= \Lambda_1^y A_0^x u_C, \\
(\Lambda_1^y \Lambda_1^x + \tau a \Lambda_1^y \Lambda_2^x + \tau b \Lambda_2^y \Lambda_1^x) \hat{u}_C &= \Lambda_1^y \Lambda_1^x u_C,
\end{aligned} \tag{4}$$

where τ is the time step, $\hat{u} = u^{n+1}$, $u = u^n$, and n is the time level number. Scheme (4) is obtained by approximating the time derivatives in the semidiscrete bcompact scheme (2) by the implicit Euler method.

The baseline bcompact scheme (4) can be written in the matrix-vector form

$$\mathbf{M}\mathbf{v} = \boldsymbol{\psi},$$

$$\mathbf{M} = \begin{bmatrix} 4/9 & 1/9 + 2\kappa_x/3 & 1/9 + 2\kappa_y/3 & 1/36 + \kappa_x/6 + \kappa_y/6 \\ -16\kappa_x/3 & 2/3 + 8\kappa_x/3 & -4\kappa_x/3 & 1/6 + 2\kappa_x/3 + \kappa_y \\ -16\kappa_y/3 & -4\kappa_y/3 & 2/3 + 8\kappa_y/3 & 1/6 + \kappa_x + 2\kappa_y/3 \\ 0 & -8\kappa_y & -8\kappa_x & 1 + 4\kappa_x + 4\kappa_y \end{bmatrix},$$

$$\mathbf{v} = \begin{pmatrix} \hat{u}_{j+1/2, k+1/2} \\ \hat{u}_{j+1, k+1/2} \\ \hat{u}_{j+1/2, k+1} \\ \hat{u}_{j+1, k+1} \end{pmatrix}, \quad \boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix},$$

where $\boldsymbol{\psi}$ is a given column vector and $\kappa_x = a\tau/h_x$ and $\kappa_y = b\tau/h_y$ are Courant numbers.

The operators on the left-hand sides of Eqs. (4) can be approximately factorized as

$$\begin{aligned} A_0^y A_0^x + a\tau A_0^y \Lambda_1^x + b\tau \Lambda_1^y A_0^x &= B_1^y(b\tau) B_1^x(a\tau) - \tau^2 ab \Lambda_1^y \Lambda_1^x, \\ A_0^y \Lambda_1^x + a\tau A_0^y \Lambda_2^x + b\tau \Lambda_1^y \Lambda_1^x &= B_1^y(b\tau) B_2^x(a\tau) - \tau^2 ab \Lambda_1^y \Lambda_2^x, \\ \Lambda_1^y A_0^x + a\tau \Lambda_1^y \Lambda_1^x + b\tau \Lambda_2^y A_0^x &= B_2^y(b\tau) B_1^x(a\tau) - \tau^2 ab \Lambda_2^y \Lambda_1^x, \\ \Lambda_1^y \Lambda_1^x + a\tau \Lambda_1^y \Lambda_2^x + b\tau \Lambda_2^y \Lambda_1^x &= B_2^y(b\tau) B_2^x(a\tau) - \tau^2 ab \Lambda_2^y \Lambda_2^x, \end{aligned} \quad (5)$$

where $B_1^d(r) = A_0^d + r\Lambda_1^d$, $B_2^d(r) = \Lambda_1^d + r\Lambda_2^d$ for $d = x, y$.

An iterative algorithm for solving the equations of baseline scheme (4) relies on approximate factorization (5) of the difference operators of the bcompact scheme and is given by the formulas

$$\begin{aligned} B_1^y(b\tau) B_1^x(a\tau) \hat{u}^{(s+1)} &= A_0^y A_0^x u + \tau^2 ab \Lambda_1^y \Lambda_1^x \hat{u}^{(s)}, \\ B_1^y(b\tau) B_2^x(a\tau) \hat{u}^{(s+1)} &= A_0^y \Lambda_1^x u + \tau^2 ab \Lambda_1^y \Lambda_2^x \hat{u}^{(s)}, \\ B_2^y(b\tau) B_1^x(a\tau) \hat{u}^{(s+1)} &= \Lambda_1^y A_0^x u + \tau^2 ab \Lambda_2^y \Lambda_1^x \hat{u}^{(s)}, \\ B_2^y(b\tau) B_2^x(a\tau) \hat{u}^{(s+1)} &= \Lambda_1^y \Lambda_1^x u + \tau^2 ab \Lambda_2^y \Lambda_2^x \hat{u}^{(s)}, \end{aligned}$$

where s is the iteration number.

This algorithm can be written in the matrix-vector form

$$\mathbf{M}_y \mathbf{M}_x \mathbf{v}^{(s+1)} = \mathbf{M}_r \mathbf{v}^{(s)} + \boldsymbol{\psi},$$

$$\mathbf{M}_y = \begin{bmatrix} 2/3 & 0 & 1/6 + \kappa_y & 0 \\ 0 & 2/3 & 0 & 1/6 + \kappa_y \\ -8\kappa_y & 0 & 1 + 4\kappa_y & 0 \\ 0 & -8\kappa_y & 0 & 1 + 4\kappa_y \end{bmatrix}, \mathbf{M}_x = \begin{bmatrix} 2/3 & 1/6 + \kappa_x & 0 & 0 \\ -8\kappa_x & 1 + 4\kappa_x & 0 & 0 \\ 0 & 0 & 2/3 & 1/6 + \kappa_x \\ 0 & 0 & -8\kappa_x & 1 + 4\kappa_x \end{bmatrix}, \quad (6)$$

$$\mathbf{M}_r = \kappa_x \kappa_y \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -8 & 4 \\ 0 & -8 & 0 & 4 \\ 64 & -32 & -32 & 16 \end{bmatrix},$$

where $\mathbf{M}_y \mathbf{M}_x - \mathbf{M}_r = \mathbf{M}$.

The matrices $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_r$ can be represented in the form of Kronecker matrix products

$$\mathbf{M}_x = \mathbf{E} \otimes \mathbf{P}_x, \quad \mathbf{M}_y = \mathbf{P}_y \otimes \mathbf{E}, \quad \mathbf{M}_r = \kappa_x \kappa_y \mathbf{R} \otimes \mathbf{R} = (\mathbf{P}_y - \mathbf{Q}) \otimes (\mathbf{P}_x - \mathbf{Q}), \quad (7)$$

where

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{P}_d = \mathbf{Q} + \kappa_d \mathbf{R}, \quad \mathbf{Q} = \begin{bmatrix} 2/3 & 1/6 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -8 & 4 \end{bmatrix}.$$

The matrix product $\mathbf{M}_y, \mathbf{M}_x$ are transformed as follows:

$$\mathbf{M}_y \mathbf{M}_x = (\mathbf{P}_y \otimes \mathbf{E})(\mathbf{E} \otimes \mathbf{P}_x) = \mathbf{P}_y \otimes \mathbf{P}_x. \quad (8)$$

The iterative method (6) is written in the recurrence form

$$\mathbf{v}^{(s+1)} = \mathbf{S} \mathbf{v}^{(s)} + \boldsymbol{\xi}. \quad (9)$$

where the matrix of a single step of the method is given by

$$\mathbf{S} = (\mathbf{M}_y \mathbf{M}_x)^{-1} \mathbf{M}_r. \quad (10)$$

In view of formulas (7) and (8), the step matrix (10) of the iterative method can be written as a Kronecker product of two matrices, each related to only one spatial direction:

$$\begin{aligned} \mathbf{S} &= (\mathbf{P}_y \otimes \mathbf{P}_x)^{-1} [(\mathbf{P}_y - \mathbf{Q}) \otimes (\mathbf{P}_x - \mathbf{Q})] = (\mathbf{P}_y^{-1} \otimes \mathbf{P}_x^{-1}) [(\mathbf{P}_y - \mathbf{Q}) \otimes (\mathbf{P}_x - \mathbf{Q})] = \\ &= (\mathbf{E} - \mathbf{P}_y^{-1} \mathbf{Q}) \otimes (\mathbf{E} - \mathbf{P}_x^{-1} \mathbf{Q}) = (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x), \end{aligned} \quad (11)$$

where $\mathbf{T}_d = \mathbf{P}_d^{-1} \mathbf{Q}$, $d = x, y$.

Due to representation (11) for the matrix \mathbf{S} , its eigenvalues $\lambda(\mathbf{S})$ are equal to the product of the eigenvalues λ_x, λ_y of the matrices $\mathbf{E}-\mathbf{T}_y$ and $\mathbf{E}-\mathbf{T}_x$:

$$\lambda(\mathbf{S}) = \lambda_y \lambda_x, \quad (12)$$

where

$$\lambda_d = \frac{12\kappa_d^2 + (3 \pm i\sqrt{3})\kappa_d}{12\kappa_d^2 + 6\kappa_d + 1}, \quad d = x, y. \quad (13)$$

The moduli of the complex conjugate numbers λ_d (13) are identical and given by

$$|\lambda_d| = \frac{2\sqrt{3}\kappa_d}{\sqrt{12\kappa_d^2 + 6\kappa_d + 1}}.$$

Therefore, by virtue of (12), the spectral radius $\rho(\mathbf{S})$ of the step matrix \mathbf{S} is

$$\rho(\mathbf{S}) = \frac{12\kappa_y \kappa_x}{\sqrt{(12\kappa_y^2 + 6\kappa_y + 1)(12\kappa_x^2 + 6\kappa_x + 1)}}. \quad (14)$$

Since the spectral radius of the stationary iterative process (9) is less than unity for any $\kappa_y > 0, \kappa_x > 0$, the iterations always converge.

Convergence analysis in the three-dimensional case

For the 3D homogeneous linear advection equation

$$L_3(u) \equiv \partial_t u + a\partial_x u + b\partial_y u + c\partial_z u = 0, \quad a = \text{const} > 0, \quad b = \text{const} > 0, \quad c = \text{const} > 0, \quad (15)$$

the equations of a semidiscrete bcompact scheme of fourth-order accuracy in space can be obtained by integration over the 3D cell $[x_j, x_{j+1}] \times [y_k, y_{k+1}] \times [z_l, z_{l+1}]$ and the subsequent approximation of the original equation (15) and its seven independent differential consequences

$$\begin{aligned} \partial_x L_3(u) &= 0, \quad \partial_y L_3(u) = 0, \quad \partial_z L_3(u) = 0, \quad \partial_x \partial_y L_3(u) = 0, \\ \partial_x \partial_z L_3(u) &= 0, \quad \partial_y \partial_z L_3(u) = 0, \quad \partial_x \partial_y \partial_z L_3(u) = 0. \end{aligned}$$

The equations of the semidiscrete bcompact scheme are given by

$$\begin{aligned}
& A_0^z A_0^y A_0^x (du/dt)_C + \left(a A_0^z A_0^y \Lambda_1^x + b A_0^z \Lambda_1^y A_0^x + c \Lambda_1^z A_0^y A_0^x \right) u_C = 0, \\
& A_0^z A_0^y \Lambda_1^x (du/dt)_C + \left(a A_0^z A_0^y \Lambda_2^x + b A_0^z \Lambda_1^y \Lambda_1^x + c \Lambda_1^z A_0^y \Lambda_1^x \right) u_C = 0, \\
& A_0^z \Lambda_1^y A_0^x (du/dt)_C + \left(a A_0^z \Lambda_1^y \Lambda_1^x + b A_0^z \Lambda_2^y A_0^x + c \Lambda_1^z \Lambda_1^y A_0^x \right) u_C = 0, \\
& \Lambda_1^z A_0^y A_0^x (du/dt)_C + \left(a \Lambda_1^z A_0^y \Lambda_1^x + b \Lambda_1^z \Lambda_1^y A_0^x + c \Lambda_2^z A_0^y A_0^x \right) u_C = 0, \\
& A_0^z \Lambda_1^y \Lambda_1^x (du/dt)_C + \left(a A_0^z \Lambda_1^y \Lambda_2^x + b A_0^z \Lambda_2^y \Lambda_1^x + c \Lambda_1^z \Lambda_1^y \Lambda_1^x \right) u_C = 0, \\
& \Lambda_1^z A_0^y \Lambda_1^x (du/dt)_C + \left(a \Lambda_1^z A_0^y \Lambda_2^x + b \Lambda_1^z \Lambda_1^y \Lambda_1^x + c \Lambda_2^z A_0^y \Lambda_1^x \right) u_C = 0, \\
& \Lambda_1^z \Lambda_1^y A_0^x (du/dt)_C + \left(a \Lambda_1^z \Lambda_1^y \Lambda_1^x + b \Lambda_1^z \Lambda_2^y A_0^x + c \Lambda_2^z \Lambda_1^y A_0^x \right) u_C = 0, \\
& \Lambda_1^z \Lambda_1^y \Lambda_1^x (du/dt)_C + \left(a \Lambda_1^z \Lambda_1^y \Lambda_2^x + b \Lambda_1^z \Lambda_2^y \Lambda_1^x + c \Lambda_2^z \Lambda_1^y \Lambda_1^x \right) u_C = 0,
\end{aligned}$$

where $C = (j+1/2, k+1/2, l+1/2)$ is a multi-index. The one-dimensional difference operators A_0^d , Λ_1^d , Λ_2^d ($d = x, y, z$) are similar to their counterparts (3). For example, the operator A_0^z is given by the formula

$$A_0^z u_{j',k',l+1/2} = \frac{u_{j',k',l} + 4u_{j',k',l+1/2} + u_{j',k',l+1}}{6}, \quad j' \in \{j, j+1/2, j+1\}, \quad k' \in \{k, k+1/2, k+1\}.$$

The fully discrete baseline bicomcompact scheme has the form

$$\begin{aligned}
& \left(A_0^z A_0^y A_0^x + a\tau A_0^z A_0^y \Lambda_1^x + b\tau A_0^z \Lambda_1^y A_0^x + c\tau \Lambda_1^z A_0^y A_0^x \right) \hat{u}_C = A_0^z A_0^y A_0^x u_C, \\
& \left(A_0^z A_0^y \Lambda_1^x + a\tau A_0^z A_0^y \Lambda_2^x + b\tau A_0^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_1^z A_0^y \Lambda_1^x \right) \hat{u}_C = A_0^z A_0^y \Lambda_1^x u_C, \\
& \left(A_0^z \Lambda_1^y A_0^x + a\tau A_0^z \Lambda_1^y \Lambda_1^x + b\tau A_0^z \Lambda_2^y A_0^x + c\tau \Lambda_1^z \Lambda_1^y A_0^x \right) \hat{u}_C = A_0^z \Lambda_1^y A_0^x u_C, \\
& \left(\Lambda_1^z A_0^y A_0^x + a\tau \Lambda_1^z A_0^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_1^y A_0^x + c\tau \Lambda_2^z A_0^y A_0^x \right) \hat{u}_C = \Lambda_1^z A_0^y A_0^x u_C, \\
& \left(A_0^z \Lambda_1^y \Lambda_1^x + a\tau A_0^z \Lambda_1^y \Lambda_2^x + b\tau A_0^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x \right) \hat{u}_C = A_0^z \Lambda_1^y \Lambda_1^x u_C, \\
& \left(\Lambda_1^z A_0^y \Lambda_1^x + a\tau \Lambda_1^z A_0^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_2^z A_0^y \Lambda_1^x \right) \hat{u}_C = \Lambda_1^z A_0^y \Lambda_1^x u_C, \\
& \left(\Lambda_1^z \Lambda_1^y A_0^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_2^y A_0^x + c\tau \Lambda_2^z \Lambda_1^y A_0^x \right) \hat{u}_C = \Lambda_1^z \Lambda_1^y A_0^x u_C, \\
& \left(\Lambda_1^z \Lambda_1^y \Lambda_1^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_2^z \Lambda_1^y \Lambda_1^x \right) \hat{u}_C = \Lambda_1^z \Lambda_1^y \Lambda_1^x u_C.
\end{aligned} \tag{16}$$

The operators on the left-hand sides of Eqs. (16) can be approximately factorized as

$$\begin{aligned}
A_0^z A_0^y A_0^x + a\tau A_0^z A_0^y \Lambda_1^x + b\tau A_0^z \Lambda_1^y A_0^x + c\tau \Lambda_1^z A_0^y A_0^x &= B_1^z(c\tau) B_1^y(b\tau) B_1^x(a\tau) - \tau^2 I_1, \\
A_0^z A_0^y \Lambda_1^x + a\tau A_0^z A_0^y \Lambda_2^x + b\tau A_0^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_1^z A_0^y \Lambda_1^x &= B_1^z(c\tau) B_1^y(b\tau) B_2^x(a\tau) - \tau^2 I_2, \\
A_0^z \Lambda_1^y A_0^x + a\tau A_0^z \Lambda_1^y \Lambda_1^x + b\tau A_0^z \Lambda_2^y A_0^x + c\tau \Lambda_1^z \Lambda_1^y A_0^x &= B_1^z(c\tau) B_2^y(b\tau) B_1^x(a\tau) - \tau^2 I_3, \\
\Lambda_1^z A_0^y A_0^x + a\tau \Lambda_1^z A_0^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_1^y A_0^x + c\tau \Lambda_2^z A_0^y A_0^x &= B_2^z(c\tau) B_1^y(b\tau) B_1^x(a\tau) - \tau^2 I_4, \\
A_0^z \Lambda_1^y \Lambda_1^x + a\tau A_0^z \Lambda_1^y \Lambda_2^x + b\tau A_0^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x &= B_1^z(c\tau) B_2^y(b\tau) B_2^x(a\tau) - \tau^2 I_5, \\
\Lambda_1^z A_0^y \Lambda_1^x + a\tau \Lambda_1^z A_0^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + c\tau \Lambda_2^z A_0^y \Lambda_1^x &= B_2^z(c\tau) B_1^y(b\tau) B_2^x(a\tau) - \tau^2 I_6, \\
\Lambda_1^z \Lambda_1^y A_0^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_1^x + b\tau \Lambda_1^z \Lambda_2^y A_0^x + c\tau \Lambda_2^z \Lambda_1^y A_0^x &= B_2^z(c\tau) B_2^y(b\tau) B_1^x(a\tau) - \tau^2 I_7, \\
\Lambda_1^z \Lambda_1^y \Lambda_1^x + a\tau \Lambda_1^z \Lambda_1^y \Lambda_2^x + b\tau \Lambda_1^z \Lambda_2^y \Lambda_1^x + c\tau \Lambda_2^z \Lambda_1^y \Lambda_1^x &= B_2^z(c\tau) B_2^y(b\tau) B_2^x(a\tau) - \tau^2 I_8,
\end{aligned} \tag{17}$$

where

$$\begin{aligned}
B_1^d(r) &= A_0^d + r\Lambda_1^d, \quad B_2^d(r) = \Lambda_1^d + r\Lambda_2^d, \quad d = x, y, z, \\
I_1 &= abA_0^z \Lambda_1^y \Lambda_1^x + ac\Lambda_1^z A_0^y \Lambda_1^x + bc\Lambda_1^z \Lambda_1^y A_0^x + \tau abc\Lambda_1^z \Lambda_1^y \Lambda_1^x, \\
I_2 &= abA_0^z \Lambda_1^y \Lambda_2^x + ac\Lambda_1^z A_0^y \Lambda_2^x + bc\Lambda_1^z \Lambda_1^y \Lambda_1^x + \tau abc\Lambda_1^z \Lambda_1^y \Lambda_2^x, \\
I_3 &= abA_0^z \Lambda_2^y \Lambda_1^x + ac\Lambda_1^z \Lambda_1^y \Lambda_1^x + bc\Lambda_1^z \Lambda_2^y A_0^x + \tau abc\Lambda_1^z \Lambda_2^y \Lambda_1^x, \\
I_4 &= ab\Lambda_1^z \Lambda_1^y \Lambda_1^x + ac\Lambda_2^z A_0^y \Lambda_1^x + bc\Lambda_2^z \Lambda_1^y A_0^x + \tau abc\Lambda_2^z \Lambda_1^y \Lambda_1^x, \\
I_5 &= abA_0^z \Lambda_2^y \Lambda_2^x + ac\Lambda_1^z \Lambda_1^y \Lambda_2^x + bc\Lambda_1^z \Lambda_2^y \Lambda_1^x + \tau abc\Lambda_1^z \Lambda_2^y \Lambda_2^x, \\
I_6 &= ab\Lambda_1^z \Lambda_1^y \Lambda_2^x + ac\Lambda_2^z A_0^y \Lambda_2^x + bc\Lambda_2^z \Lambda_1^y \Lambda_1^x + \tau abc\Lambda_2^z \Lambda_1^y \Lambda_2^x, \\
I_7 &= ab\Lambda_1^z \Lambda_2^y \Lambda_1^x + ac\Lambda_2^z \Lambda_1^y \Lambda_1^x + bc\Lambda_2^z \Lambda_2^y A_0^x + \tau abc\Lambda_2^z \Lambda_2^y \Lambda_1^x, \\
I_8 &= ab\Lambda_1^z \Lambda_2^y \Lambda_2^x + ac\Lambda_2^z \Lambda_1^y \Lambda_2^x + bc\Lambda_2^z \Lambda_2^y \Lambda_1^x + \tau abc\Lambda_2^z \Lambda_2^y \Lambda_2^x.
\end{aligned}$$

An iterative algorithm for solving the equations of the baseline scheme relies on the approximate factorization (17) of the difference operators and is given by the formulas

$$\begin{aligned}
B_1^z(c\tau) B_1^y(b\tau) B_1^x(a\tau) \hat{u}_C^{(s+1)} &= A_0^z A_0^y A_0^x u_C + \tau^2 I_1 \hat{u}_C^{(s)}, \\
B_1^z(c\tau) B_1^y(b\tau) B_2^x(a\tau) \hat{u}_C^{(s+1)} &= A_0^z A_0^y \Lambda_1^x u_C + \tau^2 I_2 \hat{u}_C^{(s)}, \\
B_1^z(c\tau) B_2^y(b\tau) B_1^x(a\tau) \hat{u}_C^{(s+1)} &= A_0^z \Lambda_1^y A_0^x u_C + \tau^2 I_3 \hat{u}_C^{(s)}, \\
B_2^z(c\tau) B_1^y(b\tau) B_1^x(a\tau) \hat{u}_C^{(s+1)} &= \Lambda_1^z A_0^y A_0^x u_C + \tau^2 I_4 \hat{u}_C^{(s)}, \\
B_1^z(c\tau) B_2^y(b\tau) B_2^x(a\tau) \hat{u}_C^{(s+1)} &= A_0^z \Lambda_1^y \Lambda_1^x u_C + \tau^2 I_5 \hat{u}_C^{(s)}, \\
B_2^z(c\tau) B_1^y(b\tau) B_2^x(a\tau) \hat{u}_C^{(s+1)} &= \Lambda_1^z A_0^y \Lambda_1^x u_C + \tau^2 I_6 \hat{u}_C^{(s)}, \\
B_2^z(c\tau) B_2^y(b\tau) B_1^x(a\tau) \hat{u}_C^{(s+1)} &= \Lambda_1^z \Lambda_1^y A_0^x u_C + \tau^2 I_7 \hat{u}_C^{(s)}, \\
B_2^z(c\tau) B_2^y(b\tau) B_2^x(a\tau) \hat{u}_C^{(s+1)} &= \Lambda_1^z \Lambda_1^y \Lambda_1^x u_C + \tau^2 I_8 \hat{u}_C^{(s)},
\end{aligned} \tag{18}$$

where s is the iteration number.

Algorithm (18) can be written in the matrix-vector form

$$\mathbf{M}_z \mathbf{M}_y \mathbf{M}_x \mathbf{v}^{(s+1)} = \mathbf{M}_r \mathbf{v}^{(s)} + \boldsymbol{\psi}, \quad \mathbf{v} = \begin{pmatrix} \hat{u}_{j+1/2, k+1/2, l+1/2} \\ \hat{u}_{j+1, k+1/2, l+1/2} \\ \hat{u}_{j+1/2, k+1, l+1/2} \\ \hat{u}_{j+1, k+1, l+1/2} \\ \hat{u}_{j+1/2, k+1/2, l+1} \\ \hat{u}_{j+1, k+1/2, l+1} \\ \hat{u}_{j+1/2, k+1, l+1} \\ \hat{u}_{j+1, k+1, l+1} \end{pmatrix}, \quad \boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \end{pmatrix}, \quad (19)$$

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

$$\mathbf{M}_r = \kappa_y \kappa_x \mathbf{Q} \otimes \mathbf{R} \otimes \mathbf{R} + \kappa_z \kappa_x \mathbf{R} \otimes \mathbf{Q} \otimes \mathbf{R} + \kappa_z \kappa_y \mathbf{R} \otimes \mathbf{R} \otimes \mathbf{Q} + \kappa_z \kappa_y \kappa_x \mathbf{R} \otimes \mathbf{R} \otimes \mathbf{R},$$

where $\boldsymbol{\psi}$ is a given column vector, $\kappa_x, \kappa_y, \kappa_z = c\tau/h_z$ are Courant numbers, and the matrices \mathbf{P}_d ($d = x, y, z$), \mathbf{E} , \mathbf{Q} , and \mathbf{R} are given by the formulas

$$\mathbf{P}_d = \mathbf{Q} + \kappa_d \mathbf{R}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 2/3 & 1/6 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ -8 & 4 \end{bmatrix}.$$

The matrix product on the left-hand side of (19) can be brought to the form

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

and the matrix \mathbf{M}_r on the right-hand side of (19) can be written as

$$\begin{aligned} \mathbf{M}_r = & \mathbf{Q} \otimes (\mathbf{P}_y - \mathbf{Q}) \otimes (\mathbf{P}_x - \mathbf{Q}) + (\mathbf{P}_z - \mathbf{Q}) \otimes \mathbf{Q} \otimes (\mathbf{P}_x - \mathbf{Q}) + \\ & + (\mathbf{P}_z - \mathbf{Q}) \otimes (\mathbf{P}_y - \mathbf{Q}) \otimes \mathbf{Q} + (\mathbf{P}_z - \mathbf{Q}) \otimes (\mathbf{P}_y - \mathbf{Q}) \otimes (\mathbf{P}_x - \mathbf{Q}). \end{aligned} \quad (20)$$

Iterative method (19) is written in the recurrence form

$$\mathbf{v}^{(s+1)} = \mathbf{S} \mathbf{v}^{(s)} + \boldsymbol{\xi}, \quad (21)$$

where the step matrix of the method is

$$\mathbf{S} = (\mathbf{M}_z \mathbf{M}_y \mathbf{M}_x)^{-1} \mathbf{M}_r = (\mathbf{P}_z \otimes \mathbf{P}_y \otimes \mathbf{P}_x)^{-1} \mathbf{M}_r = (\mathbf{P}_z^{-1} \otimes \mathbf{P}_y^{-1} \otimes \mathbf{P}_x^{-1}) \mathbf{M}_r. \quad (22)$$

In view of formulas (20) and (22), the step matrix \mathbf{S} of the iterative method can be brought into the form

$$\begin{aligned} \mathbf{S} = & \mathbf{T}_z \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x) + (\mathbf{E} - \mathbf{T}_z) \otimes \mathbf{T}_y \otimes (\mathbf{E} - \mathbf{T}_x) + \\ & + (\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes \mathbf{T}_x + (\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x), \end{aligned} \quad (23)$$

where $\mathbf{T}_d = \mathbf{P}_d^{-1} \mathbf{Q}$, $d = x, y, z$.

Formula (23) shows that \mathbf{S} is the sum of four 8×8 matrices, each being a Kronecker product of three 2×2 matrices. In turn, the column eigenvectors of the 8×8 matrices on the right-hand side of (23) are Kronecker products of the column eigenvectors of 2×2 matrices [11, p.596]. Since the column eigenvectors of the 2×2 matrices \mathbf{T}_d and $\mathbf{E} - \mathbf{T}_d$ ($d = x, y, z$) coincide, the column eigenvectors of all four 8×8 matrices on the right-hand side of (23) coincide as well. Therefore, the eigenvalues $\lambda(\mathbf{S})$ of the step matrix \mathbf{S} are equal to the sum of the eigenvalues of four 8×8 matrices on the right-hand side of (23):

$$\begin{aligned} \lambda(\mathbf{S}) = & \lambda(\mathbf{T}_z \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x)) + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes \mathbf{T}_y \otimes (\mathbf{E} - \mathbf{T}_x)) + \\ & + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes \mathbf{T}_x) + \lambda((\mathbf{E} - \mathbf{T}_z) \otimes (\mathbf{E} - \mathbf{T}_y) \otimes (\mathbf{E} - \mathbf{T}_x)) = \\ & = (1 - \lambda_z) \lambda_y \lambda_x + \lambda_z (1 - \lambda_y) \lambda_x + \lambda_z \lambda_y (1 - \lambda_x) + \lambda_z \lambda_y \lambda_x, \end{aligned} \quad (24)$$

where λ_d are the eigenvalues of the matrix $\mathbf{E} - \mathbf{T}_d$ ($d = x, y, z$). The last equality in (24) was written using the properties of eigenvalues of a matrix that is a Kronecker matrix product. Since the matrix $\mathbf{E} - \mathbf{T}_d$ has two complex conjugate eigenvalues

$$\lambda_d = \frac{12\kappa_d^2 + (3 \pm i\sqrt{3})\kappa_d}{12\kappa_d^2 + 6\kappa_d + 1}, \quad (25)$$

formula (24) determines eight different eigenvalues of \mathbf{S} .

The moduli of the complex conjugate numbers λ_d (25) are identical and given by

$$|\lambda_d| = \frac{2\sqrt{3}\kappa_d}{\sqrt{12\kappa_d^2 + 6\kappa_d + 1}}. \quad (26)$$

The squared modulus of the eigenvalue $\lambda(\mathbf{S})$ is calculated as

$$\begin{aligned} |\lambda(\mathbf{S})|^2 = & \lambda(\mathbf{S}) \bar{\lambda}(\mathbf{S}) = (1 - 4 \operatorname{Re} \lambda_z) |\lambda_y|^2 |\lambda_x|^2 + (1 - 4 \operatorname{Re} \lambda_y) |\lambda_z|^2 |\lambda_x|^2 + \\ & + (1 - 4 \operatorname{Re} \lambda_x) |\lambda_z|^2 |\lambda_y|^2 + 4 |\lambda_z|^2 |\lambda_y|^2 |\lambda_x|^2 + 2(\operatorname{Re} \lambda_z \operatorname{Re} \lambda_y + \operatorname{Im} \lambda_z \operatorname{Im} \lambda_y) |\lambda_x|^2 + \\ & + 2(\operatorname{Re} \lambda_z \operatorname{Re} \lambda_x + \operatorname{Im} \lambda_z \operatorname{Im} \lambda_x) |\lambda_y|^2 + 2(\operatorname{Re} \lambda_y \operatorname{Re} \lambda_x + \operatorname{Im} \lambda_y \operatorname{Im} \lambda_x) |\lambda_z|^2. \end{aligned} \quad (27)$$

The squared spectral radius of the matrix \mathbf{S} is found using formulas (25)–(27):

$$\begin{aligned} \rho^2(\mathbf{S}) &= \max |\lambda(\mathbf{S})|^2 = \\ &= \frac{1728\kappa_z\kappa_y\kappa_x \left[\kappa_z\kappa_y\kappa_x + \frac{1}{2}(\kappa_z\kappa_y + \kappa_z\kappa_x + \kappa_y\kappa_x) + \frac{1}{6}(\kappa_z + \kappa_y + \kappa_x) \right] + 144(\kappa_z^2\kappa_y^2 + \kappa_z^2\kappa_x^2 + \kappa_y^2\kappa_x^2)}{(12\kappa_z^2 + 6\kappa_z + 1)(12\kappa_y^2 + 6\kappa_y + 1)(12\kappa_x^2 + 6\kappa_x + 1)}. \end{aligned} \quad (28)$$

It follows from (28) that the spectral radius of the step matrix of the stationary iterative process (21) is less than unity for any $\kappa_z > 0$, $\kappa_y > 0$, $\kappa_x > 0$; therefore, the iterations always converge.

The results presented in Fig. 1 show that the iterations in the 3D case converge more slowly than in the 2D case.

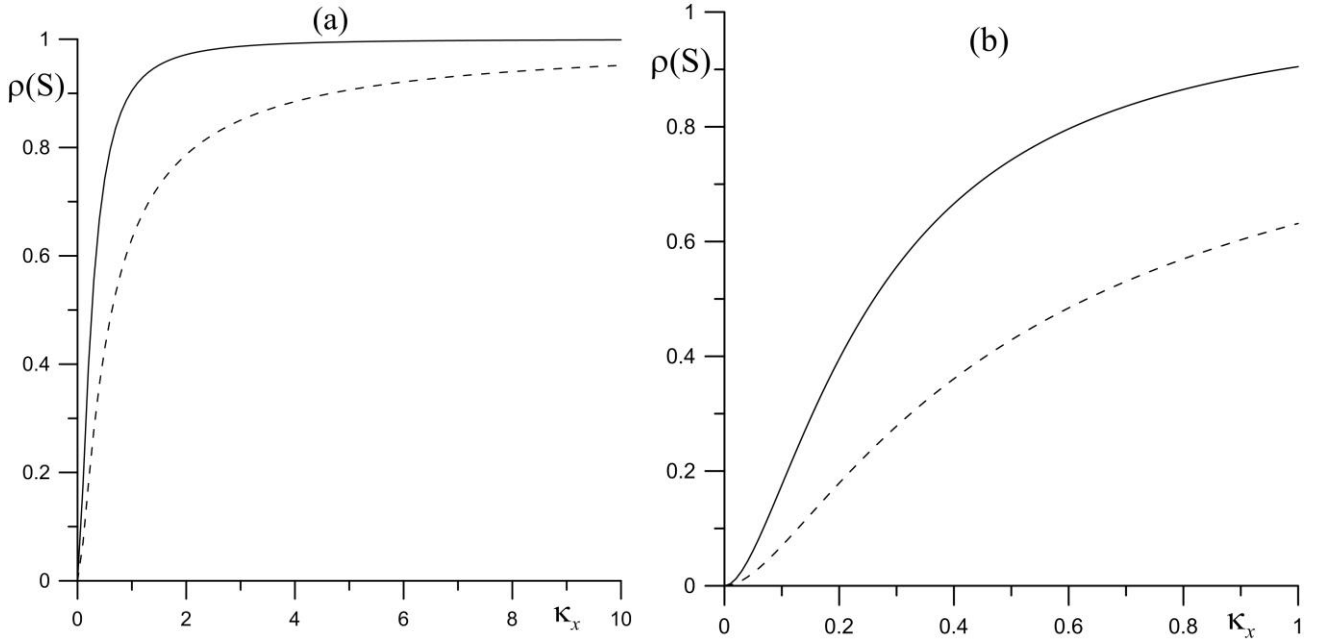


Fig. 1. (a) Spectral radius of the step matrix \mathbf{S} as a function of Courant number in the case $\kappa_x = \kappa_y = \kappa_z$; and (b) a zoomed fragment. The solid and dashed curves depict the 3D and 2D cases, respectively.

Numerical experiment

The convergence of method (19) was numerically investigated for the column

$$\begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \end{pmatrix} \text{ computed assuming that the exact solution is } \begin{pmatrix} \hat{u}_{j+1/2,k+1/2,l+1/2} \\ \hat{u}_{j+1,k+1/2,l+1/2} \\ \hat{u}_{j+1/2,k+1,l+1/2} \\ \hat{u}_{j+1,k+1,l+1/2} \\ \hat{u}_{j+1/2,k+1/2,l+1} \\ \hat{u}_{j+1,k+1/2,l+1} \\ \hat{u}_{j+1/2,k+1,l+1} \\ \hat{u}_{j+1,k+1,l+1} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \text{ As an initial}$$

$$\text{approximation, we used } \begin{pmatrix} \hat{u}_{j+1/2,k+1/2,l+1/2} \\ \hat{u}_{j+1,k+1/2,l+1/2} \\ \hat{u}_{j+1/2,k+1,l+1/2} \\ \hat{u}_{j+1,k+1,l+1/2} \\ \hat{u}_{j+1/2,k+1/2,l+1} \\ \hat{u}_{j+1,k+1/2,l+1} \\ \hat{u}_{j+1/2,k+1,l+1} \\ \hat{u}_{j+1,k+1,l+1} \end{pmatrix}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

A decrease in the error of the numerical solution in the norm $\|\cdot\|_{L_2}$ was studied for various relations between the Courant numbers $\kappa_x, \kappa_y, \kappa_z$. The results of the study are shown in Fig. 2. It can be seen that the convergence rate of the iterative process agrees with estimate (28) obtained for the spectral radius of the step matrix.

Conclusions

For a bicomact scheme of fourth-order accuracy in space, the convergence of the iterative approximate factorization of its operators was analyzed as applied to the 3D homogeneous linear advection equation with positive constant coefficients. It was proved that convergence occurs for all positive Courant numbers in all three spatial directions.

The convergence rate of the iterative approximate factorization method was theoretically estimated. A numerical experiment was performed, which confirmed this estimate for various relations between the Courant numbers.

The convergence rate of the iterative approximate factorization method was compared in the 2D and 3D cases. It was shown that the convergence in the 3D case is slower than in the 2D case.

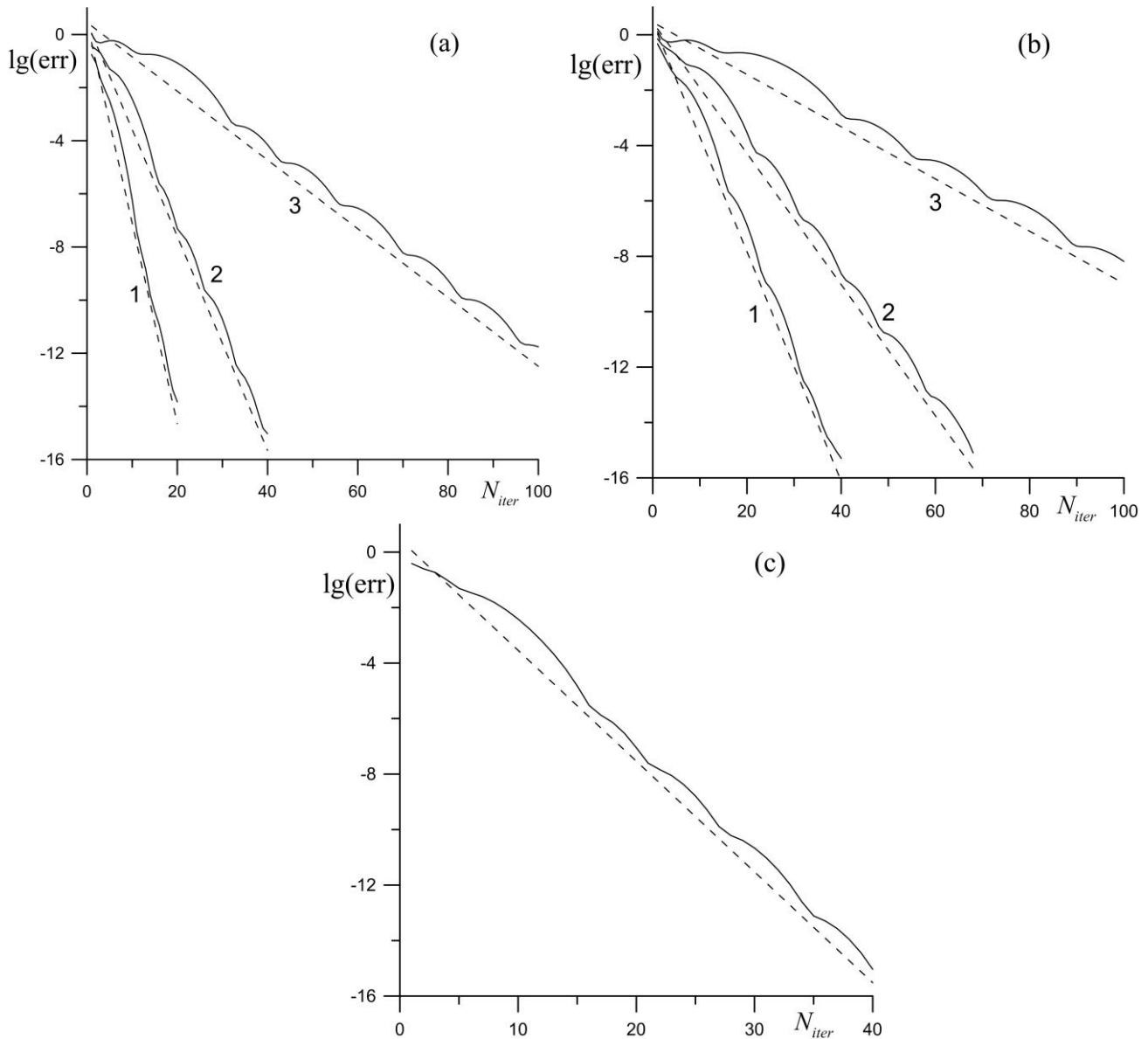


Fig. 2. Error in the numerical solution against the number of iterations. The solid curves depict iterative process (19), and the dashed curves show an ideal iterative process with an error suppression coefficient equal to $\rho(\mathbf{S})$ at every iteration step. (a) $\kappa_x = \kappa_y = \kappa_z$: (1) $\kappa_x = 0.1$, (2) $\kappa_x = 0.2$, (3) $\kappa_x = 0.5$; (b) $\kappa_x = \kappa_y, \kappa_z = 1.0$: (1) $\kappa_x = 0.1$, (2) $\kappa_x = 0.2$, (3) $\kappa_x = 0.5$; and (c) $\kappa_x = 0.1, \kappa_y = 0.2, \kappa_z = 0.4$.

References

1. Tolstykh A.I., High accuracy compact and multioperators approximations for partial differential equations. Moscow: Nauka, 2015.
2. Rogov B.V., Mikhailovskaya M.N. Fourth-order accurate bicomact schemes for hyperbolic equations // Dokl. Math. 2010. V. 81, No. 1. P. 146–150.
3. Mikhailovskaya M.N., Rogov B.V. Monotone compact running schemes for systems of hyperbolic equations // Comput. Math. Math. Phys. 2012. V. 52, No. 4. P. 578–600.

4. Rogov B.V. High-order accurate monotone compact running scheme for multidimensional hyperbolic equations // *Comput. Math. Math. Phys.* 2013. V. 53, No. 2. P. 205–214.
5. Rogov B.V., Bragin M.D. On spectral-like resolution properties of fourth-order accurate symmetric bcompact schemes // *Dokl. Math.* 2017. V. 96, No. 1. P. 339–343.
6. Chikitkin A.V., Rogov B.V. A sixth-order bcompact scheme with spectral-like resolution for hyperbolic equations // *Dokl. Math.* 2017. V. 96, No. 2. P. 480–485.
7. Chikitkin A.V., Rogov B.V. Optimized symmetric bcompact scheme of the sixth order of approximation with low dispersion for hyperbolic equations // *Dokl. Math.* 2018. V. 97, No. 1. P. 90–94.
8. Bragin M.D., Rogov B.V. Iterative approximate factorization for difference operators of high-order bcompact schemes for multidimensional nonhomogeneous hyperbolic systems// *Dokl. Math.* 2017. V. 95, No. 2. P. 140–143.
URL: <https://link.springer.com/article/10.1134/S1064562417020107>
9. Bragin M.D., Rogov B.V. Iterative approximate factorization of difference operators of high-order accurate bcompact schemes for multidimensional nonhomogeneous quasilinear hyperbolic systems // *Comput. Math. Math. Phys.* 2018. V. 58, No. 3. P. 295–306.
10. Bragin M.D., Rogov B.V. On the exact dimensional splitting for a scalar quasilinear hyperbolic conservation law // In: Klingenberg C., Westdickenberg M. (eds) *Theory, Numerics and Applications of Hyperbolic Problems I. HYP 2016.* – Cham: Springer, 2018. – P.215-225. URL: https://doi.org/10.1007/978-3-319-91545-6_17.
11. Watkins D.S. *Fundamentals of matrix computations.* New York: Wiley Interscience, 2002.

Contents

Introduction	3
Convergence analysis in the two-dimensional case	4
Convergence analysis in the three-dimensional case	8
Numerical experiment	14
Conclusions	14
References	15