Additive two-dimensional splitting schemes for solving 3D suspension transport problems on optimal boundary-adaptive grids with uniform spacing’s in the vertical direction

A Sukhinov¹ and V Sidoryakina²*

¹Don State Technical University, 1 Gagarina sq., Rostov-on-Don, 344000, Russia
²Taganrog Institute of A.P. Chekhov (branch) RGEU (RINE), Taganrog, Russia

Abstract. The article considers 3D matter transport model in dissolved and suspended forms (impurities) in coastal marine systems. The initial boundary value problem numerical solution is carried out on the basis of local 2D splitting schemes. In this case, special attention is paid to the high-quality computational grid construction. As a rule, in the proposed methods, Cartesian grids are used, a variety of which are grids with boundary adaptation. The technology of constructing 3D vertically uniform boundary-adaptive grids on the basis of surface 2D grids, which is created using the procedure of minimizing the generalized Dirichlet functional, is presented. Previously, this approach has shown its effectiveness in constructing 2D non-degenerate regular grids containing the minimum number of cells (convex quadrangles) for test Z-shape regions, such as the "Maltese cross" and others, as well as in the 2D hydrophysics problems numerical solution of coastal systems.

1 Introduction

The article discusses 3D diffusion-convection-sedimentation model of suspended matter in the aquatic environment, taking into account the coastal system features. Convective diffusion transfer operators in horizontal and vertical directions for the suspension transport problems have significantly different physical and spectral properties [1-4]. In addition, real coastal systems have complex coastline shape and bottom topography. This geometry of the computational domain motivates the boundary-adaptive grid construction [5 – 6].

To construct effective algorithms for the suspended matter distribution operational forecast (reservoir pollution) in coastal systems that involve the multiprocessor computing systems use, it is advisable to use locally two dimensional splitting schemes (LTDS) [7]. Earlier it was shown for spatially 3D initial-boundary value problems for parabolic type equations that the LTS use makes it possible to obtain low-cost algorithms for the number of arithmetic operations for solving parabolic grid equations, which are economical in terms of the operations number and the time required for exchanges between processors [8].

* Corresponding author: cvv9@mail.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
To ensure regularity and exchanges simplicity, it is desirable to build 3D region boundary-adaptive grid as a Cartesian product of optimal 2D boundary-adaptive grid [9, 10], built on the surface of the reservoir and uniform along the vertical direction. This is most easily achieved by replacing the Cartesian coordinate $\theta$ varying everywhere in a closed interval $[0; 1]$.

After changing the variables and constructing the grid, additive scheme for splitting the original problem along geometric directions is constructed. A chain of two problems is formulated, the first of which contains the 2D diffusion transfer operator and the convective transfer operator written in symmetric form, and the second problem contains the convective transfer operator in the vertical direction taking into account gravitational settling symmetrical form and the diffusion type operator. Then discretization is performed by the finite volume method.

### 2 Continuous 3D diffusion-convection-aggregation model

Let use rectangular Cartesian coordinate system $Oxyz$, where is the axis $Ox$ passes on the surface of the undisturbed water surface and is directed towards the sea. Let be $h=H+\eta$ is the total water depth [m]; $H$ is the depth at undisturbed surface of the reservoir, [m]; $\eta$ is the elevation of the free surface relative to the undisturbed surface, [m].

![Fig. 1. Introduction of the coordinate system.](image)

In the model, carry out the transition from $z$ – coordinate system to the $\theta$ – coordinate system, for which we use the Cartesian coordinate system in the horizontal plane and as the vertical variable - the dimensionless variable $\theta, \theta \in [0, 1]$.

In the $\theta$ – coordinate system, the water column is divided into the same number of layers at each point regardless of the depth, therefore, when using the «new» coordinate system, some problems associated with adding and subtracting layers are solved [11, 12].

At $\theta$ transition to $\theta$ – coordinate system use the formula:

$$\theta = a - \frac{(a-b)(z-\eta)}{h}, x_o=x, y_o=y,$$  

(1)

where $\theta=a=0$ on the free surface of the reservoir (upper boundary), $\theta=b=1$ at the bottom.

Further, instead of expressions (1), we are using formulas
\[ \theta = \frac{z - \eta}{h} \]  
\[ x'_o = x, y'_o = y. \]  

Let in water volume \( V = \{0 \leq x \leq L, 0 \leq y \leq L, 0 \leq \theta \leq 1\} \) impurity particles that at the point \((x, y, \theta)\) and at the moment of time \(t\) have concentration \(c = c(x, y, \theta, t)\), [mg/l]; \(t\) – temporary variable, \(t \in [0,T]\) [sec].

The equation describing the behavior of the particles will look like this:

\[
\frac{\partial c}{\partial t} + \nabla (uc) + \nabla (vc) = a - b \frac{\partial}{\partial \theta} \left( \frac{w + w_g}{h} \right) c + \mu \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) + \frac{a - b}{h} \frac{\partial}{\partial \theta} \left( \frac{\partial c}{\partial \theta} \right) + F. \]  

Here \(u, v, w\) are the components of the fluid velocity vector \(\vec{U}\), [m/sec]; \(w_g\) is the hydraulic size or particle settling speed, [m/s]; \(\mu, \nu\) are the coefficients of horizontal and vertical diffusion of particles, [m²/sec]; \(F\) is the function that describes the intensity of the distribution of sources of suspended matter.

The terms on the left-hand side (except for the time derivative) of Eq. (3) describe the convection of particles: their transfer under the action of fluid flow and gravity. The terms on the right-hand side describe suspension diffusion. The vertical diffusion coefficient is chosen to be different from the horizontal diffusion coefficient. In combination with the boundary and initial conditions for the particle concentration function, the solution to equation (3) makes it possible to determine the fluxes and concentrations of suspended matter.

As a water volume \(V\), consider 3D cylinder, the upper base of which lies on the free surface, and the lower base is a part of the bottom surface. The of this cylinder is parallel to the Oz axis and moves at some distance from the coastline along some depth line of relatively small value- for example, for the Taganrog Bay and the Azov Sea, it may be taken 0.5 m. Let be \(\Gamma = S \cup S_{top} \cup S_{bottom}\) is the boundary of the region \(V\), \(S\) is the lateral boundary surface, and \(S_{top}, S_{bottom}\) are the parts of the free surface and the bottom surface, respectively.

The lateral surface \(S\) of the region \(V\) is formed as a result of the movement of the generatrix, along a closed line, which is a contour approximating the coastline along the isobath of relatively shallow depth. Consider the domain of the Eq. (3) \(\Psi = \{0 < t \leq T\}\), \(\Psi = V \cup \Gamma\).

Let us add the initial and boundary conditions to Eq. (3) assuming that the deposition of particles to the bottom is irreversible:

- initial conditions at \(t=0\)
  \[
  c(x, y, \theta, 0) \equiv c_0(x, y, \theta); \]  

- boundary conditions at the lateral boundary \(S\) at any time \(t\)
  \[
  \frac{\partial c}{\partial n} = 0, \text{ if } (U_r, \nu) \leq 0, \]  
  \[
  \frac{\partial c}{\partial n} = \frac{u_c}{\mu}, \text{ if } (U_r, \nu) \geq 0, \]
where \( \mathbf{n} \) is the outward normal to the boundary of the region \( S \), \( \mathbf{U}_r \) is the vector of the velocity of fluid movement at the boundary \( S \), \( \mathbf{u}_r \) is the projection of the velocity vector to the direction of the normal at the boundary of the region \( S \);

- boundary conditions on the water free surface \( S_{\text{top}} \)

\[
\frac{\partial c}{\partial \theta} = 0; \\
\tag{7}
\]

- bottom boundary conditions \( S_{\text{bottom}} \)

\[
\frac{\partial c}{\partial \theta} = \frac{w_c}{v};
\tag{8}
\]

Assuming that the bottom surface is not «strongly» curved.

In [13, 14], the conditions for the correctness of the problem of transport of suspensions were investigated in the case of a multicomponent particle size distribution of particles, and therefore problem (3) – (8), as its particular case under the smoothness conditions of the solution function

\[
c(x,y,\theta,t) \in C(Q_r) \cap C(\overline{Q}_r), \text{grad}c \in C(\overline{Q}_r)
\]

and the required smoothness of the region boundary.

### 3 Technology for constructing three-dimensional boundary-adaptive grids description

On the undisturbed surface of the reservoir \( S_{\text{top}} \) in plane \( (x,y) \) the grid is being built \( \omega_{12} = \{(x_i,y_j), i=1,N_x, j=1,N_y\} \), by the given coordinates of the boundary nodes. One-dimensional grid \( \omega_3 = \{k=1,\ldots,N_o, h_0=1,\ldots,N_o\} \) built by volume \( V \) and has uniform steps within the area. The desired grid \( \Omega \) find as the Cartesian product of a 2D grid \( \omega_{12} \) and one-dimensional grid \( \omega_3 \), t.e. \( \Omega = \omega_{12} \times \omega_3 \).

Let’s take a closer look at the construction of boundary-adaptive 2D grids.

In the plane \( (x,y) \), grid is built \( \omega_{12} \) according to the specified coordinates of the boundary nodes. The grid area is divided \( S_{\text{top}} \) into elementary sub-areas in the form of quadrangles. Then we pass from variables \( x,y \) to variables \( \xi, \eta \), specified in the domain \( \Xi \), in which the elementary sub-domains will become «standard». Grid \( \omega_3 = \{i=1,\ldots,N_x, j=1,\ldots,N_y\} \) is built on the plane \( (\xi,\eta) \), which is determined by functions \( x=x(\xi,\eta), y=y(\xi,\eta) \) that map a rectangle \( 1 \leq \xi \leq N_x, 1 \leq \eta \leq N_y \) in a plane \( (\xi,\eta) \) to a curvilinear quadrangle in a plane \( (x,y) \).

For transformations

\[
(\xi,\eta) \rightarrow (x,y), x=x(\xi,\eta), y=y(\xi,\eta),
\]
\[(x,y) \rightarrow (\xi, \eta); \xi = \xi(x,y), \eta = \eta(x,y),\]

it must be required nonzero Jacobians

\[J(\xi, \eta) = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} = \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}, \quad (9)\]

\[J(x,y) = \frac{\partial \xi}{\partial x} \frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y} \frac{\partial \eta}{\partial x}. \quad (10)\]

For convenience of perception, it is advisable to present a local consideration of the grids by considering the corresponding cells \( \omega_{ij}, \omega_{ij} \). Set the nodes of the cell \( \omega_{ij} \) from coordinates

\[x_{ij}, y_{ij}, \quad (x_{ij}, y_{ij}, x_{ij+1}, y_{ij+1}, x_{ij+1}, y_{ij+1}), \quad (11)\]

The nodes with coordinates \( x_{ij}, y_{ij}, \quad (x_{ij}, y_{ij}, x_{ij+1}, y_{ij+1}, x_{ij+1}, y_{ij+1}), \quad (12)\)

should be traversed counterclockwise (when looking along the direction of free fall acceleration).

Continuous transformation \((x,y) \rightarrow (\xi, \eta)\) is determined at each specific cell by the formulas

\[x = x(\xi, \eta) = x_{ij} + (x_{ij} - x_{ij}), x_{ij} + (x_{ij} - x_{ij}) \eta + (x_{ij} - x_{ij} - y_{ij} + y_{ij}), \xi, \]

\[y = y(\xi, \eta) = y_{ij} + (y_{ij} - y_{ij}), y_{ij} + (y_{ij} - y_{ij} - y_{ij} + y_{ij}), \xi \eta. \quad (11)\]

The cells are convex quadrangles, where each of the corners is less than \( \pi \), which ensures that the Jacobian is positive (9)

\[J(\xi, \eta) = \left( x_{ij} - x_{ij}, y_{ij} - y_{ij}, y_{ij} - y_{ij} - y_{ij} + y_{ij}, \xi \right), \]

\[- \left( x_{ij} - x_{ij}, y_{ij} - y_{ij}, y_{ij} - y_{ij} - y_{ij} + y_{ij}, \eta \right). \quad (12)\]

and introduces restrictions when choosing a cell shape.

By numbering the vertices of the cell from 1 to 4 clockwise starting from the node \( ij \), denote the doubled areas of triangles

\[J_{1}, J_{2}, J_{3}, J_{4}; \quad J_{0} = \frac{1}{4} \left( J_{1} + J_{2} + J_{3} + J_{4} \right). \quad (12)\]

Obviously, the condition for the positiveness of the Jacobian \( J \), is equivalent to the positivity of the quantities \( J_{1}, J_{2}, J_{3}, J_{4} \). At the boundary of the constructed convex grids, at least one of the indicated values will be equal to zero.

The procedure for constructing grids is related to considering the Dirichlet functional
\[ I = \int J \, g \, dx \, dy = \int J \, g \, d \xi \, d \eta, \quad (13) \]

\[ g = g_{11} + g_{22}, \quad g_{11} = \left( \frac{\partial \xi}{\partial x} \right)^2 + \left( \frac{\partial \xi}{\partial y} \right)^2 \quad \text{and} \quad g_{22} = \left( \frac{\partial \eta}{\partial x} \right)^2 + \left( \frac{\partial \eta}{\partial y} \right)^2 \]

are the components of the metric coordinate transformation tensor.

Euler’s equations for functional (13) can be written as a system of nonlinear equations for the sought functions \( x = x(\xi, \eta), y = y(\xi, \eta) \).

Discrete analog of the functional \( I \) is the functional obtained by summing over all four corners (triangles formed when a quadrilateral cell is divided by its diagonals) of all grid cells, i.e.

\[
I^h = \frac{1}{4} \sum_{i,j} \sum_{m,n} \left[ (x_{i,j} - x_{i,m})^2 + (y_{i,j} - y_{i,n})^2 + (x_{i,j} - x_{m,j})^2 + (y_{i,j} - y_{n,j})^2 \right] J^{-1}_{i,j} +
\]

\[
+ \left[ (x_{i,j} - x_{i,m})^2 + (y_{i,j} - y_{i,n})^2 + (x_{i,j} - x_{m,j})^2 + (y_{i,j} - y_{n,j})^2 \right] J^{-1}_{i,j} +
\]

\[
+ \left[ (x_{i,j} - x_{i,m})^2 + (y_{i,j} - y_{i,n})^2 + (x_{i,j} - x_{m,j})^2 + (y_{i,j} - y_{n,j})^2 \right] J^{-1}_{i,j} +
\]

\[
+ \left[ (x_{i,j} - x_{i,m})^2 + (y_{i,j} - y_{i,n})^2 + (x_{i,j} - x_{m,j})^2 + (y_{i,j} - y_{n,j})^2 \right] J^{-1}_{i,j}, \quad (14)
\]

Expression (14) can be obtained by replacing \( x(\xi, \eta), y(\xi, \eta) \) in (13) by \( x'(\xi, \eta), y'(\xi, \eta) \) and approximation by quadrature formulas of integrals over cells in the plane \( \xi, \eta \).

System solution of the form

\[
\frac{\partial I^h}{\partial x_{i,j}} = 0, \quad \frac{\partial I^h}{\partial y_{i,j}} = 0, \quad (15)
\]

gives at least one solution - a convex mesh. To find it, we minimize the function \( I^h \).

When the grid is built on \( m \) step of iterations, then at the \( (m+1) \)-th step it is determined from the solution of two linear equations written for each internal node [15]:

\[
\begin{align*}
\tau \frac{\partial J_{i,j}^{(m)}}{\partial x_{i,j}} + \frac{\partial}{\partial x_{i,j}} J_{i,j}^{(m)} (x_{i,j}^{(m)} - x_{i,j}^{(m)}) - \frac{\partial}{\partial x_{i,j}} J_{i,j}^{(m)} (x_{i,j}^{(m)} - y_{i,j}^{(m)}) &= 0, \\
\tau \frac{\partial J_{i,j}^{(m)}}{\partial y_{i,j}} + \frac{\partial}{\partial y_{i,j}} J_{i,j}^{(m)} (x_{i,j}^{(m)} - x_{i,j}^{(m)}) - \frac{\partial}{\partial y_{i,j}} J_{i,j}^{(m)} (y_{i,j}^{(m)} - y_{i,j}^{(m)}) &= 0,
\end{align*}
\]

(16)

\( \tau \) is the iteration parameter, \( \tau = 1 \).

Note that a necessary condition for the minimum of functional is a consequence of the conformity of the transformation under consideration.

The minimization \( I^h \) on a set of convex grids was carried out by means of a quasi-Newtonian procedure in the computational domain \( (\xi, \eta) \). To do this, it is advisable to use the well-proven gradient methods tested and described in [11] in relation to the Azov Sea and the Taganrog Bay.

A change in the sign in the denominators of the integrands of the functionals (13), (14) of the Jacobian \( J \), leads to discontinuities of the second kind, which turns out to be critical.
for second-order methods (quasi-Newtonian procedure). To get out of this situation, near the
degeneracy boundary \((J=0)\) the coordinate transformation is replaced \(J\) by

\[
J' = \max(J, \varepsilon) = (J - \varepsilon)
\]

\(\varepsilon\) - small enough number, \(\varepsilon > 0\).

It is important to note that when generating the grid using the above procedure, an initial
approximation in the form of a convex grid is given. If this is not the case, then the initial
grid, a convex grid, is determined by sequentially solving optimization problems.

Note that since cells with numbers containing \(i=1,N, j=1,N\) are boundary, then the
position of the nodes is not optimized for them. Therefore, further consideration of the
problem is carried out on the nodes \(i=2,N-1, j=2,N-1\).

For a clear presentation of further reasoning, we represent (14) in the form

\[
I = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \left(\frac{1}{4} \sum_{i=2}^{N} \sum_{j=2}^{N} \left( (x_{i,j+1} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2 + (x_{i,j+1} - y_{i+1,j})^2 \right) \right) +
\]

\[
+ \frac{1}{4} \sum_{i=2}^{N} \sum_{j=2}^{N} \left( (x_{i,j} - x_{i,j+1})^2 + (y_{i,j+1} - y_{i,j})^2 + (x_{i,j+1} - y_{i,j+1})^2 \right)
\]

\[
+ \frac{1}{4} \sum_{i=2}^{N} \sum_{j=2}^{N} \left( (x_{i+1,j} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2 + (x_{i,j+1} - y_{i,j+1})^2 \right)
\]

\[
+ \frac{1}{4} \sum_{i=2}^{N} \sum_{j=2}^{N} \left( (x_{i,j} - x_{i,j+1})^2 + (y_{i,j+1} - y_{i,j})^2 + (x_{i,j+1} - y_{i,j+1})^2 \right)
\]

(17)

Let introduce the notation

\[
f(J) = \begin{cases} 
\frac{1}{J_k}, & J_k \geq \varepsilon, \\
\frac{1 + \left(1 - \frac{1}{J_k}\right)^2}{\varepsilon}, & J_k \leq \varepsilon, k = 1,4. 
\end{cases}
\]

(18)

Taking into account (18), expression (17) can be compactly represented in the following form

\[
I' = \sum_{i=2}^{N} \sum_{j=2}^{N} \left( g_1 f(J_1) + g_2 f(J_2) + g_3 f(J_3) + g_4 f(J_4) \right),
\]

(19)

where

\[
J_{1,ij} = (x_{i+1,j} - x_{i,j}) (y_{i+1,j} - y_{i,j}) (x_{i+1,j+1} - x_{i,j+1}),(y_{i+1,j+1} - y_{i,j+1})
\]

\[
J_{2,ij} = (x_{i+1,j} - x_{i,j}) (y_{i+1,j} - y_{i,j}) (y_{i+1,j+1} - y_{i,j+1}) (x_{i+1,j+1} - x_{i,j+1})
\]

\[
J_{3,ij} = (x_{i+1,j} - x_{i,j}) (x_{i+1,j+1} - x_{i,j+1}) (y_{i+1,j+1} - y_{i,j+1}) (x_{i+1,j+1} - x_{i,j+1})
\]

\[
J_{4,ij} = (x_{i+1,j} - x_{i,j}) (y_{i+1,j} - y_{i,j}) (x_{i+1,j+1} - x_{i,j+1}) (y_{i+1,j+1} - y_{i,j+1})
\]

(20)
\( g_{i,j} = \frac{1}{4} \left[ (x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2 \right] \),
\( g_{2,i,j} = \frac{1}{4} \left[ (x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2 \right] \),
\( g_{3,i,j} = \frac{1}{4} \left[ (x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2 \right] \),
\( g_{4,i,j} = \frac{1}{4} \left[ (x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2 \right] \).

The solution of the system of algebraic equations of the form (15) for \( I^v \), representing the necessary condition for the minimum of the functional \( I^v \), gives at least one solution (convex grid).

Further, for minimization \( I^v \) used the same quasi-Newtonian procedure as in minimizing the function \( I^h \).

To determine the components of the gradient of a discrete functional \( I^v \) represented locally at a point \((x_{i,j}, y_{i,j})\), the derivatives were required:

- for expressions \( J_1, J_2, J_3, J_4, g_1, g_2, g_3, g_4 \) on variable \( x \)

\[
\left( \frac{\partial J_1}{\partial x_{i,j}} \right) = - (y_{i,j+1} - y_{i,j}) + (y_{i+1,j} - y_{i,j}) = y_{i+1,j} - y_{i,j}, \quad \left( \frac{\partial g_1}{\partial x_{i,j}} \right) = \frac{1}{2} \left[ (-x_{i+1,j} + x_{i,j}) + (x_{i,j} - x_{i-1,j}) \right],
\]
\[
\left( \frac{\partial J_2}{\partial x_{i,j}} \right) = - (y_{i,j} - y_{i,j+1}) - (y_{i+1,j} - y_{i,j}) = y_{i,j+1} - y_{i,j}, \quad \left( \frac{\partial g_2}{\partial x_{i,j}} \right) = \frac{1}{2} \left[ (-x_{i+1,j} + x_{i,j}) + (x_{i,j} - x_{i-1,j}) \right],
\]
\[
\left( \frac{\partial J_3}{\partial x_{i,j}} \right) = (y_{i,j} - y_{i,j}) - (y_{i,j} - y_{i,j+1}) = y_{i,j} - y_{i,j+1}, \quad \left( \frac{\partial g_3}{\partial x_{i,j}} \right) = \frac{1}{2} \left[ (-x_{i+1,j} + x_{i,j}) + (x_{i,j} - x_{i-1,j}) \right],
\]
\[
\left( \frac{\partial J_4}{\partial x_{i,j}} \right) = (y_{i,j} - y_{i,j+1}) + (y_{i,j} - y_{i,j+1}) = y_{i,j} - y_{i,j+1}, \quad \left( \frac{\partial g_4}{\partial x_{i,j}} \right) = \frac{1}{2} \left[ (-x_{i+1,j} + x_{i,j}) + (x_{i,j} - x_{i-1,j}) \right],
\]

- on variable \( y \)

\[
\left( \frac{\partial J_1}{\partial y_{i,j}} \right) = (x_{i,j+1} - x_{i,j}) + (x_{i+1,j} - x_{i,j}) = x_{i+1,j} - x_{i,j}, \quad \left( \frac{\partial g_1}{\partial y_{i,j}} \right) = \frac{1}{2} \left[ (-y_{i,j+1} + y_{i,j}) + (y_{i,j} - y_{i+1,j}) \right],
\]
\[
\left( \frac{\partial J_2}{\partial y_{i,j}} \right) = (x_{i,j} - x_{i,j+1}) + (x_{i,j+1} - x_{i,j}) = x_{i,j+1} - x_{i,j}, \quad \left( \frac{\partial g_2}{\partial y_{i,j}} \right) = \frac{1}{2} \left[ (-y_{i,j+1} + y_{i,j}) + (y_{i,j} - y_{i+1,j}) \right],
\]
\[
\left( \frac{\partial J_3}{\partial y_{i,j}} \right) = (x_{i,j} - x_{i,j+1}) - (x_{i,j} - x_{i,j+1}) = x_{i,j+1} - x_{i,j}, \quad \left( \frac{\partial g_3}{\partial y_{i,j}} \right) = \frac{1}{2} \left[ (-y_{i,j+1} + y_{i,j}) + (y_{i,j} - y_{i+1,j}) \right],
\]
\[
\left( \frac{\partial J_4}{\partial y_{i,j}} \right) = (x_{i,j} - x_{i,j+1}) - (x_{i,j} - x_{i,j+1}) = x_{i,j+1} - x_{i,j}, \quad \left( \frac{\partial g_4}{\partial y_{i,j}} \right) = \frac{1}{2} \left[ (-y_{i,j+1} + y_{i,j}) + (y_{i,j} - y_{i+1,j}) \right],
\]

- for function \( f(J_i) \)
\[ \left( f'_i \right)_k = \begin{cases} \frac{1}{(J'_i)^{1/2}} & J'_i \geq \varepsilon, \\ 1 + 2 \left( \frac{1 - J_i}{\varepsilon} \right)^2, & J_i \leq \varepsilon, k = 1,4. \end{cases} \]  

(24)

Taking into account the obtained expressions (20) – (24), find the components of the gradient of the functional \( I^w = \{ g_x f(J_i) + g_y f(J_j) + g_z f(J_3) \} \), corresponding to Cartesian coordinates \((x_i, y_j, z_k)\) of the node \((i, j)\):

\[
\frac{\partial I^w}{\partial x}_{i,j} = \sum_{k=0}^3 \frac{\partial g_x}{\partial x} f(J_k),
\quad \frac{\partial I^w}{\partial y}_{i,j} = \sum_{k=0}^3 \frac{\partial g_y}{\partial y} f(J_k),
\quad \frac{\partial I^w}{\partial z}_{i,j} = \sum_{k=0}^3 \frac{\partial g_z}{\partial z} f(J_k).
\]

### 4 Numerical experiments results on constructing optimal boundary-adaptive grids

The described algorithms for constructing 3D boundary-adaptive grids were implemented in software and tested on a number of test areas. The areas under consideration are \(6 \times 15 \times 6\) (area 1) and \(9 \times 12 \times 6\) (area 2). In fig. 2 (a) and 2 (b) show the initial approximations for mappings to the 3D Z-shaped regions 1 and 2, respectively, with the same sets of boundary, but different choices of corner nodes. In fig. 2 (c) and 2 (d) show the results of grid building 1, 2, after reaching the convexity of all cells of the 2D grid. Figures 2 (e) and 2 (f) show the results of grid generation for regions 1 and 2, where the 2D grid is constructed based on the minimization of the Dirichlet functional using a quasi-Newtonian procedure.
Fig. 2. Results of building grids for test area.

5 Method for constructing local 2D additive scheme

Consider the original problem (3) – (8). Let \( u_1 \equiv u, \ u_2 \equiv v, \ u_3 \equiv w + w_g \).

Let represent the spatially three-dimensional operator of convective transfer in symmetric form [16]:

\[
L_x c = \frac{\partial (uc)}{\partial x} + \frac{\partial (vc)}{\partial y} + \frac{a - b}{h} \frac{\partial ((w + w_g)c)}{\partial \theta} = \frac{1}{2} \left( \frac{\partial (uc)}{\partial x} + u_c \frac{\partial c}{\partial x} \right) + \frac{1}{2} \left( \frac{\partial (vc)}{\partial y} + v_c \frac{\partial c}{\partial y} \right) + \frac{1}{2} \frac{a - b}{h} \left( \frac{\partial (w + w_g)}{\partial \theta} + (w + w_g) \frac{\partial c}{\partial \theta} \right). \tag{25}
\]

Let’s introduce the notation

\[
L_{c12} c = \frac{1}{2} \left( \frac{\partial (uc)}{\partial x} + u_1 \frac{\partial c}{\partial x} \right) + \frac{1}{2} \left( \frac{\partial (u_2 c)}{\partial y} + u_2 \frac{\partial c}{\partial y} \right), \quad L_{c3} c = \frac{a - b}{2h} \left( \frac{\partial (u_3 c)}{\partial \theta} + u_3 \frac{\partial c}{\partial \theta} \right). \tag{26}
\]

Obviously, equality is met: \( L_x c = L_{c12} c + L_{c3} c. \)

On the time interval \( 0 \leq t \leq T \) construct a uniform grid \( \omega_t \) with step \( \tau \), i.e. set of points

\( \omega_t = \{ t_n = n \tau, \ n = 0, 1, ..., N, \ N \tau = T \}. \)

We also introduce the notation (consider, that the operators of diffusion transfer are defined)

\[
D_c = \mu \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) + \left( \frac{a - b}{h} \right) \frac{\partial}{\partial \theta} \left( \nu \frac{\partial c}{\partial \theta} \right), \tag{27}
\]

\[
D_{c12} c = \mu \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right), \quad D_3 c = \left( \frac{a - b}{2h} \right) \frac{\partial}{\partial \theta} \left( \nu \frac{\partial c}{\partial \theta} \right). \tag{28}
\]

Expression (27) taking into account (28) can be written in the form: \( D_c = D_{c12} c + D_3 c. \)

Taking into account the entered operators \( L_{c12}, \ D_{c12}, \ L_3, \ D_3 \), equation (15) will be written in the form
\[
\frac{\partial c}{\partial t} + L_{c_{12}}c + L_{c_3}c = D_{c_{12}}c + D_{c_3}c + f(x, y, \theta, t), \quad (x, y, \theta) \in V, \quad t \in [0, T].
\] (29)

For simplicity, we will consider the case when the source function may be represented as the sum of two smooth enough functions: 
\[
f(x, y, \theta, t) = f_1(x, y, \theta, t) + f_2(x, y, \theta, t).
\]

On the entered time grid \( \omega_t \) construct chain of two-dimensional-one-dimensional problems related by the initial and final data – the values of the grid functions \( c^{(1)} \) and \( c^{(2)} \):
\[
\frac{\partial c_{n+1}^{(1)}}{\partial t} + L_{c_{12}}c_{n+1}^{(1)} = D_{c_{12}}c_{n+1}^{(1)} + f_1(x, y, \theta, t), \quad t_n < t \leq t_{n+1}, \quad (30)
\]

As the boundary conditions for equation (30), boundary conditions (5), (6) on \( S \) – lateral boundary surface area \( V \)
\[
c^{(1)}(x, y, \theta, 0) = c_0(x, y, \theta),
\]
\[
c_{k+1}^{(1)}(x, y, \theta, t_k) = c_k^{(2)}(x, y, \theta, t_k), \quad k = 1, N_t - 1
\] (31)

and
\[
\frac{\partial c_{n+1}^{(2)}}{\partial t} + L_{c_3}c_{n+1}^{(2)} = D_{c_3}c_{n+1}^{(2)} + f_2(x, y, \theta, t), \quad t_n < t \leq t_{n+1},
\]
\[
\frac{\partial c_{k+1}^{(2)}}{\partial \theta} = -\frac{w_g}{v} c_{k+1}^{(2)},
\]
\[
c_{k+1}^{(2)}(x, y, \theta, t_k) = c_k^{(1)}(x, y, \theta, t_{k+1}), \quad k = 1, N_t - 1.
\] (32)

Next, we approximate each of the problems - two-dimensional (30) – (32) and one-dimensional (33) – (35) on the grid \( \Omega = \omega_{c_{12}} \times \omega_3 \) implicit schemes by the finite volume method.

With respect to the obtained difference operators \( \Lambda_{c_{12}}, \Delta_{c_{12}}, \Lambda_3, \Delta_3 \), which are discrete analogs of the operators \( L_{c_{12}}, D_{c_{12}}, L_3, D_3 \) accordingly, with the appropriate approximation of the boundary conditions, it can be argued that in the Hilbert space of grid functions \( M_{c_{12}} \equiv \Lambda_{c_{12}} - \Delta_{c_{12}} > 0, \quad M_3 \equiv \Lambda_3 - \Delta_3 > 0 \), which guarantees the stability of the difference splitting scheme.

Difference problem with operator \( M_{c_{12}} \) can be solved using the alternating triangular method, and the discrete problem with the operator \( M_3 \) – by the sweep method, possibly its variants for strongly varying coefficients \( v(\theta), \frac{a-b}{h} \) or \( \left( \frac{a-b}{h} \right)^2 \).

Based on the results presented in [9], we may distinguish 3 cases of solving grid equations on a multiprocessor computer system containing \( p \) processors:

1. \( p \leq N \), where \( N \) is the characteristic number of nodes in the horizontal direction;
2. \( N < p \leq N^2 \);
3. $N^2 < p \leq \alpha \cdot N_\theta \cdot N^2$, where $N_\theta$ is the characteristic number of nodes in the direction $\theta$, $\alpha$—some positive coefficient significantly less than 1.

It turns out that the constructed LDS is preferable for a moderate and large number of processors in comparison with a full approximation scheme and a splitting scheme in spatial variables using one-dimensional difference problems (local one-dimensional problems).

6 Conclusion

A model of 3D diffusion-sedimentation of suspended matter in an aqueous medium was considered, taking into account the proven system, for which economical algorithms for the operational forecast of impurity propagation were built on the basis of a local two-dimensional splitting scheme. In this case, special attention is paid to the construction of a high-quality computational grid. The technology of constructing three-dimensional vertically uniform boundary-adaptive grids based on surface two-dimensional grids, when creating a generalized Dirichlet functional, is presented. The results of numerical test experiments on the construction of three-dimensional optimal boundary-adaptive grids are presented.

Acknowledgments

This paper was supported by the Russian Foundation for Basic Research (RFBR) grant No. 20-01-00421

References

1. Sukhinov A I, Chistyakov A E, Protsenko E A, Sidoryakina V V, Protsenko S V 2020 *Mathematical Models and Computer Simulations* 12(2) 232-245
2. Sukhinov A I, Chistyakov A E, Protsenko E A, Sidoryakina V V, Protsenko S V 2019 *Matem Mod* 31(8) 79–100
3. Sidoryakina V V, Sukhinov A I 2017 *Comput Math Phys* 57(6) 978–994
4. Sukhinov A I, Sukhinov A A 2005 *Proceedings of Parallel CFD 2004 Conference, Las Palmas de Gran Canaria, Spain* (Amsterdam-Berlin-London-New York-Tokyo: ELSEVIER) p 223
5. Sukhinov A I, Sidoryakina V V, Sukhinov A A 2017 *Vestnik of Don State Technical University* 17(1) 5-17
6. Ivanenko S A 1988 *Zh Vychisl Math and Math fiz* 28:10 1498-1506
7. Noelle S, Rosenbaum W, Rumpf M 2001 *Int Ser Numer Math* 141 775–784
8. Sukhinov A I, Vasil’ev V S 1996 *Izv Vyssh Uchebn Zaved Mat* 3 58–67
9. Sukhinov A I 2005 *Two-dimensional splitting schemes and some of their applications* (M: Max PRESS: Moscow State University Publishing House) p 408
10. Sukhinov A, Chistyakov A, Sidoryakina V 2017 *Matec Web of Conferences* 132 UNSP 04003
11. Vasiliev V S, Sukhinov A I 2003 *Matem Modeling* 15(10) 17-34
12. Sukhinov A I, Chistyakov A E, Sidoryakina V V 2018 *Parallel Computational Technologies PCT 2018 Communications in Computer and Information Science* 910
13. Sukhinov A I, Sukhinov A A, Sidoryakina V V 2020 *Journal of Physics: Conference
14. Sukhinov A I, Sidoryakina V V 2018 Vestnik of Don State Technical University 18(4) 350-361

15. Marchuk G I, Agoshkov V I 1981 Introduction to projection-grid methods (Moscow: Science The main edition of physics and mathematics) p 416

16. Vabishchevich P N, Samarskii A A 2002 Comput Math Math Phys 42(9) 1317–1330