Solving the diffusion-convection problem using MPI parallel computing technology

A M Atayan

Department of Mathematics and Informatics, Don State Technical University, 1 Gagarina sq., Rostov-on-Don, 344003, Russia

E-mail: atayan24@mail.ru

Abstract. The aim of the work is to build a software package for the distributed solution of the problem of matter transfer in a reservoir. A parallel implementation of the grid domain decomposition methods for computationally complex diffusion-convection problems, taking into account the architecture and parameters of a multiprocessor computing system located on the basis of the infrastructure facility of the University of Science and Technology «Sirius», is considered. An algorithm for parallel solution of the formulated hydrophysics problem using MPI technology has been developed. The analysis of the efficiency of the developed parallel algorithms is carried out and the study of the obtained solutions of computationally complex problems of aquatic ecology is carried out.

1. Introduction

Human activities and natural processes have a detrimental effect on the hydrosphere, causing pollution of natural waters due to the ingress of various chemicals into streams, rivers, lakes, seas and oceans. Preventive measures include control of untreated wastewater discharges and comprehensive environmental monitoring, which can be carried out on the basis of mathematical modeling. Working not with the object itself (phenomenon, process), but with its model makes it possible to study its properties and behavior in any conceivable situations painlessly, relatively quickly and without significant costs and large impact on the modeled object. It becomes necessary to predict changes in the ecological state of shallow water bodies due to the occurrence of natural and man-made phenomena in them based on methods and means of mathematical modeling in coastal systems with different spatial resolution from tens of meters to several kilometers.

An important problem related to the ecology of water systems is the prediction of the spread of pollutants in the air and water environments. Mathematical modeling of matter transfer makes it possible to study dynamics and trends of phenomena occurring in shallow water bodies and river systems (aquatic ecosystems) [1]. On this basis it becomes possible to predict the consequences of anthropogenic interference in the aquatic ecosystem. The development of algorithms and programs is necessary for the numerical implementation of hydrodynamic nonlinear modeling problems for finding current fields in coastal systems, studying hydrophysical processes in river sections with complex morphometry of channels and floodplains [2]. The development of software and analytical tools makes it possible to solve an important scientific and
practical problem associated with assessing the impact of the planned construction of technological facilities on the flooding of floodplain areas and erosion of the river channel [3].

In the field of mathematical modeling of water pollutants movement, as well as in the development of numerical methods for solving the problems posed, a situation has arisen in which the most of the studies consider individual phenomena and do not cover them in a complex [4–6]. Therefore, to solve the problems that meet the set task, it is necessary to develop and theoretically study new algorithms and programs for solving model problems, including the equations of aerodynamics and hydrodynamics, satisfying the basic laws of conservation of matter, taking into account the multicomponent nature of the medium [7, 8].

Currently, there is no universal method for constructing optimal three-dimensional computational grids [9]. Methods for constructing three-dimensional unstructured grids for solving problems with discontinuous coefficients are described in detail in [10–12]. In [13–15], to solve this class of problems, it was proposed to use the grid-characteristic method.

To build operational forecasts of the pollution impact that have entered the coastal system with river flows and from the air by settling on the surface of the reservoir, on the production and destruction processes in it, it becomes necessary to use high-performance computing systems, parallel algorithms and programs that numerically implement the mathematical models of the studied processes of aerodynamics and hydrophysics. This allows us to reduce the calculation time by three or more orders of magnitude, as it is strictly limited by regulatory documents.

One of the parallelization methods for a large amount of computational work arising in the numerical solution of nonlinear hydrophysics problems is the use of MPI technology, which allows data exchange between processes performing the same task. With parallel implementation, methods for decomposition of grid regions have been developed to solve computationally complex diffusion-convection problems, considering the architecture and parameters of a multiprocessor computer system.

2. Problem setup

The problem of transport of pollutants in the two-dimensional case can be represented by the equation of diffusion-convection-reaction (1) [16]:

$$c' + uc'_x + vc'_y = \left( \mu c'_x \right)_x + \left( \mu c'_y \right)_y + f$$

with boundary conditions (1):

$$c'_x(x, y, t) = \alpha_n c + \beta_n,$$

where \(c\) is the concentration of pollutants; \(u, v\) are the components of the velocity vector, \(f\) is a function describing the intensity and distribution of sources, \(\mu\) is the coefficient of diffusion (turbulent) exchange.

The computational domain is inscribed in a rectangle. To discretize the mathematical model (1) - (2), a uniform grid is introduced (3) [17]:

$$w_n = \{t^n = n\tau, x_i = ih_x, y_j = jh_y; n = 0, N_x; i = 0, N_x; N_x, N_y, \tau = T, N_x h_x = l_x, N_y h_y = l_y \},$$

where \(\tau\) is the time step; \(h_x, h_y\) are the space steps; \(l_x, l_y\) are the characteristic dimensions of the computational domain; \(N_x, N_y\) are the space boundaries; \(N_t\) is the upper time limit.

To approximate (1) by the time coordinate, schemes with weights were used (4):

$$\frac{\hat{c} - c}{\tau} + uc'_x + vc'_y = \left( \mu c'_x \right)_x + \left( \mu c'_y \right)_y + f,$$
where $a, \hat{a}$ are values of the unknown function at $n$ and $n+1$ time layers; 
$\tau = \sigma \hat{a} + (1 - \sigma)c, \sigma \in [0,1]$ is weight of the scheme.

Let $O_{i,j}$ denote the filling function of the cell $(i,j)$. The values of the substance’s concentration field
are calculated at the vertices of the cell. The vertices of cell $(i,j), (i-1,j), (i,j-1), (i-1,j-1)$. To describe the geometry of the computational domain, coefficients
$q_0, q_1, q_2, q_3, q_4$ are introduced, which describe the degree of filling of the control areas lying in the vicinity
of the cell. The $q_0$ value characterizes the degree of filling of the area (5):

$$
D_0 : x \in \left(x_{i-1/2,j}, x_{i+1/2,j}\right), y \in \left(y_{j-1/2}, y_{j+1/2}\right),
q_1 - D_1 : x \in \left(x_{i-1/2,j}, y \in \left(y_{j-1/2}, y_{j+1/2}\right),
q_2 - D_2 : x \in \left(x_{i-1/2,j}, x \in \left(y_{j-1/2}, y_{j+1/2}\right),
q_3 - D_3 : x \in \left(x_{i+1/2,j}, y \in \left(y_{j-1/2}, y_{j+1/2}\right),
q_4 - D_4 : x \in \left(x_{i+1/2,j}, x \in \left(y_{j-1/2}, y_{j+1/2}\right).\right)
$$

$\Omega_m$ is the filled parts of the areas $D_m$, where $m = 1, 4$. The fillings of the control areas $q_m$ with
the functions of the filling of the cells are related by the following relations (6):

$$
\left(\frac{q_m}{\Omega_m}\right)_{i,j} = \frac{S_{\Omega_m}}{S_{\Omega_m}}, \left(\frac{q_0}{\Omega_m}\right)_{i,j} = \frac{o_{i,j} + o_{i+1,j} + o_{i+1,j+1} + o_{i,j+1}}{4}, \left(\frac{q_1}{\Omega_m}\right)_{i,j} = \frac{o_{i,j} + o_{i+1,j+1}}{2},
\left(\frac{q_2}{\Omega_m}\right)_{i,j} = \frac{o_{i,j} + o_{i,j+1}}{2}, \left(\frac{q_3}{\Omega_m}\right)_{i,j} = \frac{o_{i+1,j} + o_{i,j+1}}{2}, \left(\frac{q_4}{\Omega_m}\right)_{i,j} = \frac{o_{i,j} + o_{i+1,j}}{2}.
$$

A discrete analogue of equation (4) can be written in the following form (7):

$$
(q_0)_{i,j} \hat{a}_{i,j} - a_{i,j} + (g_1)_{i,j} u_{i+1/2,j} \frac{\tilde{a}_{i+1/2,j} - \tilde{a}_{i,j}}{2h_x} + (q_2)_{i,j} v_{i,j+1/2} \frac{\tilde{a}_{i,j+1/2} - \tilde{a}_{i,j}}{2h_y} + (q_3)_{i,j} v_{i,j-1/2} \frac{\tilde{a}_{i,j-1/2} - \tilde{a}_{i,j}}{2h_y} +
(q_4)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} - (q_1)_{i,j} \mu_{i+1/2,j} \frac{\tilde{a}_{i+1/2,j} - \tilde{a}_{i,j}}{h_x} + (q_1)_{i,j} \mu_{i+1/2,j} \frac{\tilde{a}_{i+1/2,j} - \tilde{a}_{i,j}}{h_x} - (q_2)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} - (q_3)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} - (q_4)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} -
\hat{a}_{i,j} \tilde{a}_{i,j} - a_{i,j} - (q_1)_{i,j} \mu_{i+1/2,j} \frac{\tilde{a}_{i+1/2,j} - \tilde{a}_{i,j}}{h_x} + (q_1)_{i,j} \mu_{i+1/2,j} \frac{\tilde{a}_{i+1/2,j} - \tilde{a}_{i,j}}{h_x} - (q_2)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} - (q_3)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y} - (q_4)_{i,j} \mu_{i,j} \frac{\tilde{a}_{i,j} - \tilde{a}_{i,j-1}}{h_y}.
$$

where $q_1, i = 0, 4$ is filling factors of control volumes [2].

Discrete analog (7) of the equation (1) describes not only the change in the impurity concentration due
to sources at the interface, but the complex geometry of the computational domain.

3. Algorithm

The software has been developed based on a multiprocessor computing system (MSC) intended for
mathematical modeling of the transport of substances in the aquatic environment.

The following parameters are set in the software module: dimensions of the computational grid (int $Nx$
= 100; int $Ny$ = 100); steps in spatial coordinates (double $hx = 1$; double $hy = 1$) and in time (double $tau$
= 0.001); time interval (double $lt = 10$); calculation error of grid equations (double $eps = 1e-8$); coefficients
of the boundary conditions (double $alphax = 0$; double $alphay = 0$; double $betax = 0$; double $betay = 0$);
implicit schema weight (double $sigma = 0.5$).
The program module allocates memory for the following one-dimensional floating point arrays for:
- concentration field \( C[N] \);
- turbulent exchange coefficient field \( mu[N] \);
- a function describing the intensity and distribution of sources \( f[N] \);
- components of the velocity vector \( u[N], v[N] \);
- function that describes the filling of cells \( O[N] \);
- coefficients of grid equations on the current time layer \( A[N], B1[N], B2[N], B3[N], B4[N] \);
- coefficients of the grid equations on the previous time layer \( B5[N], B6[N], B7[N], B8[N], B9[N] \);
- the right-hand sides of the grid equations \( F[N] \).

The following variables are generated in the program module, in which are stored:
- number of nodes \( N = N_x \ast N_y \);
- loop counters \( i, j \);
- time value \( t \);
- numbers of nodes \( m, m_x, m_y, m_1, m_2, m_3, m_4 \);
- the filling factor of the control areas \( q_0, q_1, q_2, q_3, q_4 \).

3.1 Computational scheme

Consider the computational scheme of the algorithm.

1. Start of the algorithm.
2. Initialization and calling of MPI functions.
3. Call the function for calculating the number of processors \( \text{world.size} \) and the processor number \( \text{world.rank} \). The beginning of the sector is calculated \( N1 = \text{world.rank} \ast (N_y - 2) / \text{world.size} + 1 \) and the length of sector \( N2 = ((\text{world.rank} + 1) \ast (N_y - 2) / \text{world.size} + 2) - N1 \) for domain decomposition.
4. Generate variables and arrays.
5. Loop counters \( i, j \); variables-nodes of the grid \( m, m_x, m_y, m_1, m_2, m_3, m_4 \); error \( \varepsilon \leftarrow 1 \ast 10^{-8} \); coefficients of boundary conditions \( \alpha_x \leftarrow 0, \alpha_y \leftarrow 0, \beta_x \leftarrow 0, \beta_y \leftarrow 0 \); implicit scheme weight \( \sigma \leftarrow 0.5 \).
6. Zero the arrays, set the initial conditions and distributions.
7. Decomposition of the computational domain.
8. Set arrays of duplicates of smaller size for decomposition \( C_{par}, f_{par}, mu_{par}, u_{par}, v_{par}, O_{par} \).
   Start of loops \( i, j (i = 0, N_x, j = 0, N_y) \).
9. \( m \leftarrow j + N2 \ast i, m_x \leftarrow j + N1 + i \ast N_y, C_{par} \leftarrow C_{m}, f_{par} \leftarrow f_{m}, O_{par} \leftarrow O_{m}, \mu_{par} \leftarrow \mu_{m}, u_{par} \leftarrow u_{m}, v_{par} \leftarrow v_{m} \).
   End of loops \( i, j \).
8. Start of the time loop.
9. Construction of grid equations.

Start of loops \(i,j\) \((i = 1, N_i - 1, j = 1, N_j - 1)\). The loop \(i\) is external, \(j\) is nested.

Each pair of numbers \(i,j\) is assigned a number (element number \(i, j\) in a one-dimensional array) according to the formula \(m_{ij} = i + j \cdot N_j\).

Based on the above formula, calculates the center element number of the grid equation template, and the values of the elements numbers in the vicinity of the center of template are found:

\[
m_i = m_0 + 1, m_i = m_0 - 1, \quad m_i = m_0 + N_x, m_i = m_0 - N_x, m_{i,j} = m_0 - N_x - 1.
\]

Calculate the coefficients characterizing the degree of filling of the control areas:

\[
q_1 = \frac{(O_{m_0} + O_{m_0})}{2}, \quad q_2 = \frac{(O_{m_0} + O_{m_0})}{2}, \quad q_3 = \frac{(O_{m_0} + O_{m_0})}{2}, \quad q_4 = \frac{(q_1 + q_2)}{2}.
\]

Calculate the coefficients of the grid equations for the nodes in the vicinity of the center of template.

First, calculate the values of the coefficients in the vicinity of the center of the template without taking into account the schema weight:

\[
B1_{m_0} = \left(\frac{u_{m_0} + u_{m_0}}{4h_x} + \frac{\mu_{m_0} + \mu_{m_0}}{2h_x^2}\right) q_1, \quad B2_{m_0} = \left(\frac{u_{m_0} + u_{m_0}}{4h_x} + \frac{\mu_{m_0} + \mu_{m_0}}{2h_x^2}\right) q_2,
\]

\[
B3_{m_0} = \left(\frac{v_{m_0} + v_{m_0}}{4h_y} + \frac{\mu_{m_0} + \mu_{m_0}}{2h_y^2}\right) q_3, \quad B4_{m_0} = \left(\frac{v_{m_0} + v_{m_0}}{4h_y} + \frac{\mu_{m_0} + \mu_{m_0}}{2h_y^2}\right) q_4.
\]

Calculate the coefficients that are in the vicinity of the center of the template on the previous time layer:

\[
B6_{m_0} = -(1 - \sigma)B1_{m_0}, \quad B7_{m_0} = -(1 - \sigma)B2_{m_0}, \quad B8_{m_0} = -(1 - \sigma)B3_{m_0}, \quad B9_{m_0} = -(1 - \sigma)B4_{m_0}.
\]

Consideration of the scheme weight for the coefficients in the vicinity of the center of template on the current time layer:

\[
B1_{m_0} = \sigma B1_{m_0}, \quad B2_{m_0} = \sigma B2_{m_0}, \quad B3_{m_0} = \sigma B3_{m_0}, \quad B4_{m_0} = \sigma B4_{m_0}.
\]

Calculate the coefficients for the nodes that are in the center of template at the current time layer:

\[
A_{m_0} = q_0 \cdot h + B1_{m_0} + B2_{m_0} + B3_{m_0} + B4_{m_0} - \sigma(\mu_{m_0} + \mu_{m_0} + \mu_{m_0} + \mu_{m_0})\mu_{m_0},
\]

and on the previous time layer:

\[
B5_{m_0} = q_0 \cdot h - B6_{m_0} + B7_{m_0} + B8_{m_0} + B9_{m_0} - \sigma(\mu_{m_0} + \mu_{m_0} + \mu_{m_0} + \mu_{m_0})\mu_{m_0}.
\]

Calculate the values of the right-hand sides of the grid equations:

\[
F_{m_0} = q_0 \cdot h - |q_1 - q_2| h - |q_3 - q_4| h + B5_{m_0} C_{m_0} + B6_{m_0} C_{m_0} + B7_{m_0} C_{m_0} + B8_{m_0} C_{m_0} + B9_{m_0} C_{m_0}.
\]

End of loops \(i, j\).

10. Apply parallel computations to solve grid equations by the method for SLAE.
11. Increase the time \(t \leftarrow t + h\).
12. End of the time loop (if \(t \leq t + h / 2\), to return to step 8).
13. Data output.
14. End of the algorithm.

3.2 Parallel algorithm for solving SLAE (Seidel method)

The function sets the parameters: loop counters \(i, j\); the number of iterations \(it\); the maximum value of difference modulus between the value of the pollutant concentration function on previous and current layers \(max\); the number of blocks for receiving and sending \(kol_del\).
The function generates following variables, in which are stored: start address of the data being sent vec_in; start address of received message vec_out; number of processors world_size; processor number world_rank.

The beginning of sector is calculated $N1 \leftarrow \text{world_rank} \cdot \left( \frac{N_y - 2}{\text{world_size}} \right)$ and the length of sector $N2 \leftarrow \left( \left( \text{world_rank} + 1 \right) \cdot \left( \frac{N_y - 2}{\text{world_size}} \right) \right) - N1$; block number of the sent data num_sek; start of sector st; end of sector fin, block size for receiving and sending ob.

1. Start of the algorithm.
2. Generate variables and arrays.
3. Start of loops $i, j \left( i = 1, N_x - 1, j = 1, N2 - 1 \right)$. The loop $i$ is external, $j$ is nested.

$$m_0 \leftarrow i \cdot N2 + j, m_i \leftarrow m_0 + N2, m_i \leftarrow m_0 - N2, m_j \leftarrow m_0 + 1, m_j \leftarrow m_0 - 1.$$ 

Save the concentration in the specified node $r \leftarrow C_{m_0}$.

Calculate a new concentration value at this node:

$$C_{m_i} \leftarrow \left( F_{m_i} + B1_{m_i} C_{m_i} + B2_{m_i} C_{m_i} + B3_{m_i} C_{m_i} + B4_{m_i} C_{m_i} \right) / A_{m_i}.$$ 

Calculate the maximum value of difference modulus between the value of the pollutant concentration function on previous and current layers $r \leftarrow r - C_{m_0}$.

If $\max < |r|$, then $\max \leftarrow |r|$.

End of loops $i, j$.
4. Pipeline operation MPI_Recv, MPI_Send.

Start of external loop $i \left( i = 0, N_x - 1 \right)$.

$$\text{num_sek} \leftarrow (i + 1) \cdot \text{kol_del} / (N_x - 2), \text{st} \leftarrow (\text{num_sek} \cdot (N_y - 2) / \text{kol_del}) + 1, \text{fin} \leftarrow (\text{num_sek} + 1) \cdot (N_y - 2) / \text{kol_del}, \text{ob} \leftarrow \text{fin} - \text{st} + 1.$$ 

Receive MPI_Recv if the block is not the first.

Start of nested loop $ii \left( ii = 0, \text{ob} \right)$.

$$m_0 \leftarrow (\text{st} + ii) \cdot N2, C_{m_0} \leftarrow \text{vec_{in}_{ii}}.$$ 

End of nested loop $ii$.

Start of nested loop $j \left( j = 1, N2 - 1 \right)$.

$$m_0 \leftarrow i \cdot N2 + j, m_i \leftarrow m_0 + N2, m_i \leftarrow m_0 - N2, m_j \leftarrow m_0 + 1, m_j \leftarrow m_0 - N_x,$$

$$C_{m_i} \leftarrow \left( F_{m_i} + B1_{m_i} C_{m_i} + B2_{m_i} C_{m_i} + B3_{m_i} C_{m_i} + B4_{m_i} C_{m_i} \right) / A_{m_i}.$$ 

Start of nested loop $ii \left( ii = 0, \text{ob} \right)$.

$$m_0 \leftarrow (\text{st} + ii) \cdot N2 + N2 - 2, \text{vec_{out}_{ii}} \leftarrow C_{m_0}.$$ 

End of nested loop $ii$.

Passing MPI_Send to the second block.
End of nested loop $j$.
End of external loop $i$.

Start of loop $ii \left( ii = 1, N_x - 1 \right)$.

$$m_0 \leftarrow ii \cdot N2 + 1, \text{vec_{out}_{ii}} \leftarrow C_{m_0}.$$
End of loop $ii$.
Sending MPI_Send if the block is not the first.
Receive MPI_Recv if the block is not the last.
Start of loop $ii\{ii = 1,N_x - 1\}$.

$$m_0 \leftarrow ii \cdot N2 + N2 - 1, C_{m_0} \leftarrow \text{vec}_i n_{i_0}.$$ 

End of loop $ii$.

5. Increasing the iteration counter $it = it + 1$.
6. End of loop (if $\max > \varepsilon$, to return to step 3).
7. Saving data.
8. End of the algorithm.

The function returns the concentration field of the substance - array $C$.

3.3 Parallel algorithm for solving SLAE (Jacobi method)
The function sets the parameters: loop counters $i, j$; the number of iterations $it$; the maximum value of difference modulus between the value of the pollutant concentration function on previous and current layers $\max$; the number of blocks for receiving and sending $kol_del$.

The function generates following variables, in which are stored: start address of the data being sent $\text{vec}_i n$; start address of received message $\text{vec}_o ut$; number of processors $world_size$; processor number $world_rank$.

The beginning of sector is calculated

$$N1 \leftarrow world\_\text{rank} \cdot (N_x - 2) / world\_\text{size} \quad \text{and the length of sector}$$

$$N2 \leftarrow (\text{world}_\text{rank} + 1) \cdot (N_x - 2) / \text{world}_\text{size} + 2 - N1.$$ 

1. Start of the algorithm.
2. Generate variables and arrays.
3. Start of loops $i, j \ (i = 1,N_x - 1, \ j = 1,N2 - 1)$. The loop $i$ is external, $j$ is nested.

$$m_0 \leftarrow i + N2, m_i \leftarrow m_0 + N2, m_t \leftarrow m_0 - N2, m_s \leftarrow m_0 + 1, m_0 \leftarrow m_0 - 1.$$ 

Calculation of the uniform norm of the correction vector:

$$r_{m_0} \leftarrow F_{m_0} - A_{m_0} \cdot C_{m_0} + B1_{m_0} C_{m_0} + B2_{m_0} C_{m_0} + B3_{m_0} C_{m_0} + B4_{m_0} C_{m_0}.$$ 

If $\max < \left| r_{m_0} \right|$, then $\max \leftarrow \left| r_{m_0} \right|$.

End of loops $i, j$.
4. Sending and receiving MPI_Recv, MPI_Send.
Start of loops $i, j \ (i = 0,N_x - 1, \ j = 0,N2 - 1)$. The loop $i$ is external, $j$ is nested.

$$m_0 \leftarrow i \cdot N2 + j, C_{m_0} \leftarrow C_{m_0} + r_{m_0} / A_{m_0}.$$ 

End of loops $i, j$.

Sending and receiving in the left direction. If the processor is not the first, then send MPI_Send.

Start of loop $ii\{ii = 1,N_x - 1\}$.

$$m_0 \leftarrow ii \cdot N2 + 1, \text{vec}_i n_{ii} \leftarrow C_{m_0}.$$ 

End of loop $ii$.
If the processor is not the last one to receive MPI_Recv.
Start of loop $ii\{ii = 1,N_x - 1\}$.
4. Materials and methods

4.1 Description of input parameters

Modeling of the transport of substances was carried out on grids with dimensions of 100x100, 200x200, 1000x1000, 2000x2000, 5000x5000, 10000x10000 computational nodes taking into account the complex geometry of the computational domain, while the parameters were set as follows: the dimensions of the computational domain in spatial coordinates \( l_x = 100 \text{ m}, l_y = 100 \text{ m} \) by time \( \tau = 0.1 \text{ s} \); the horizontal component is 4 m/s, the vertical component is 3 m/s, the turbulent exchange coefficient is 5 m^2/s.

The initial distribution was set by the function (8):

\[
C(x, y) = \begin{cases}
\sin(\pi (x - 10)/10)\sin(\pi (y - 10)/10), & x \in D \\
0, & x \notin D, D: \{x \in [10, 20], y \in [10, 20]\},
\end{cases}
\]

computational domain \( \Omega = \Sigma \setminus \sigma \), where \( \Sigma: \{x \in [0, l_x], y \in [0, l_y]\}, \sigma: \{x \in [30, 50], y \in [45, 50]\} \).

4.2 Description of a multiprocessor computing system

To solve the computationally laborious problem of the diffusion-convection-reaction type (1), (2), a high-performance system was used, which is a cluster located at the University of Science and Technology «Sirius». The architecture of the computing system used is shown in figure 1.
The cluster includes 24 processors (computing nodes) of the Intel (R) - Xeon (R) Gold 5118 CPU 2.30GHz family. Storage system of each node is 3.84TB SSD drive.

5. Computational experiment

A number of experiments were carried out on the basis of the developed software module focused on multiprocessor computing systems. Figure 2 shows the results of calculations for modeling substances transport on each computer when working on a multiprocessor computer system with distributed memory. The size of the computational grid is 100x100 nodes. The computational domain was decomposed into four parts, with the width of the adjacent domain equal to two nodes along the Ox direction.

Figure 2. The dynamics of the propagation of substances on a grid with dimensions of 100 by 100, taking into account the geometry of a complex region (the time interval is 8, 11 seconds).
Computational experiments were carried out sequentially, starting from one processor and ending with the maximum possible number of processors. Data on the operating time of software components for the Jacobi and Seidel methods are presented in table 1.

**Table 1.** Results of the operation of the algorithm using MPI technology for a different number of processors using different methods.

|          | Yakobi | Seidel |
|----------|--------|--------|
| 1        | 657,204| 507,823|
| 2        | 337,442| 259,984|
| 4        | 175,488| 127,683|
| 8        | 93,6307| 69,0433|
| 12       | 66,4512| 49,7603|
| 16       | 50,848 | 38,2223|
| 20       | 43,2453| 32,4241|
| 24       | 39,846 | 29,993 |

Figure 3 shows a comparative analysis of developed algorithms' acceleration, depending on the number of calculators involved and the size of the computational grid. The maximum number of used computers is 24.

Figure 3. Dependence of acceleration on the number of computers.

The figure shows that the Jacobi method and the Seidel method showed similar efficiency, reducing the computation time by more than 15 times.
6. Conclusions
As a result, the software package has been implemented makes it possible to calculate the problem of matter transfer in a shallow reservoir on various computational grids. The parallel algorithm implemented in the software package is focused on a multiprocessor computing system and can significantly reduce the operating time with a large amount of input data. The presented software complex can be used to study transfer processes in natural and technological systems.

Acknowledgments
The reported study was funded by RFBR according to the research project 20-31-90105 and the author expresses gratitude to University of Science and Technology «Sirius» for the provided computing resource.

References
[1] Tishkin V F, Gasilov V A, Zmitrenko N V, Kuchugov P A, Ladonkina M E and Poveschenko Yu A 2020 Modern methods of mathematical modeling of the development of hydrodynamic instabilities and turbulent mixing (Mathematical modeling) 32:8 pp. 57–90 (in Russian)
[2] Sukhinov A I, Chistyakov A E, Protsenko E A, Sidoryakina V V and Protsenko S V 2020 Accounting method of filling cells for the hydrodynamics problems solution with complex geometry of the computational domain (Mathematical modeling) 12:2 pp. 232-245 DOI: 10.1134/S2070048220020155 (in Russian)
[3] Sukhinov A I, Chistyakov A E, Shishenya A V and Timofeeva E F 2018 Predictive Modeling of Coastal Hydrophysical Processes in Multiple-Processor Systems Based on Explicit Schemes (Mathematical Models and Computer Simulations) 10:5 pp. 648-658 DOI: 10.1134/S2070048218050125
[4] Gushchin V A, Nikitina A V, Semenyakina A A, Sukhinov A I and Chistyakov A E 2018 A model of transport and transformation of biogenic elements in the coastal system and its numerical implementation (Computational Mathematics and Mathematical Physics) 58:8 pp. 1316-1333 DOI: 10.1134/S0965542518080092
[5] Sukhinov A I, Khachunts D S and Chistyakov A E 2015 A mathematical model of pollutant propagation in near-ground atmospheric layer of a coastal region and its software implementation (Computational Mathematics and Mathematical Physics) 55:77 pp. 1216-1231 DOI: 10.1134/S096554251507012X
[6] Sukhinov A I, Nikitina A V, Semenyakina A A and Chistyakov A E 2016 Complex of models, explicit regularized schemes of high-order of accuracy and applications for predictive modeling of aftermath of emergency oil spill (CEUR Workshop Proceedings) 1576 pp. 308-319
[7] Chetverushkin B N, Saveliev A V and Saveliev V I 2018 A quasi-gasdynamic model for the description of magnetogasdynamic phenomena (Computational Mathematics and Mathematical Physics) 58:8 pp. 1384–1394 DOI: 10.1134/S0965542518080055
[8] Chetverushkin B N 2013 Resolution limits of continuous media models and their mathematical formulations (Mathematical Models and Computer Simulations) 5:3 pp. 266–279 DOI: 10.1134/S2070048213030034
[9] Vabishchevich P N and Samarskii A A 2002 Monotone finite-difference schemes on triangular grids for convection-diffusion problems (Computational Mathematics and Mathematical Physics) 42:9 pp. 1317–1330
[10] Krasnov M M, Kuchugov P A, Ladonkina M E and Tishkin V F 2017 Discontinuous Galerkin method on three-dimensional tetrahedral grids: Using the operator programming method (Mathematical Models and Computer Simulations) 9:5 pp. 529–543
[11] Milyukova O Yu and Tishkin V F 2016 A multigrid method for a heat equation with discontinuous coefficients with a special choice of grids (Mathematical Models and Computer Simulations) 8:2 pp. 118–128.

[12] Gasilov V A, Gasilova I V, Klochkova L V, Poveshchenko Yu A and Tishkin V F 2015 Difference schemes based on the support operator method for fluids dynamics problems in a collector containing gas hydrates (Computational Mathematics and Mathematical Physics) 55:8 pp.1310–1323.

[13] Vasyukov A V and Petrov I B 2018 Grid-characteristic method on tetrahedral unstructured meshes with large topological inhomogeneities (Computational Mathematics and Mathematical Physics) 58:8 pp. 1259–1269.

[14] Petrov I B, Favorskaya A V and Khokhlov N I 2017 Grid-characteristic method on embedded hierarchical grids and its application in the study of seismic waves (Computational Mathematics and Mathematical Physics) 57:11 pp. 1771–1777

[15] Golubev V I, Petrov I B and Khokhlov N I 2016 Compact grid-characteristic schemes of higher orders for 3D linear transport equation (Mathematical Models and Computer Simulations) 8:5 pp. 577–584

[16] Sukhinov A, Chistyakov A, Nikitina A, Semenyakina A, Korovin I and Schaefer G 2016 Modelling of oil spill spread (5th International Conference on Informatics, Electronics and Vision, ICIEV) pp. 1134-1139 DOI: 10.1109/ICIEV.2016.7760176

[17] Khokhlov N I and Petrov I B 2019 Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems (Proceedings of ISP RAS) 31:6 pp.237–252 (in Russian)

[18] 2020 MPI. Introductory course URL: http://www.ssd.sscc.ru/old/kraeva/MPI.html (in Russian)