Processing the overlay of geometry segments in solving hydrophysics problems by the finite difference method

Vladimir Litvinov[1,2*], *Natalya* Gracheva[1,2], and *Nelli* Rudenko[1,2]

[1]Don State Technical University, Rostov-on-Don, 344000, Russia
[2]Azov-Black Sea Engineering Institute of Don State Agrarian University, Zernograd, 347740, Russia

Abstract. The article deals with issues related to increasing the efficiency of working with data on the geometry of the computational domain when solving hydrophysics problems using the finite difference method. The model problem is a system of equations of the pollutant distribution, including the oil and its refined products, in the computational domain – Azov Sea. To describe the computational domain, a model of a two-dimensional computational grid is used, which is used in the implementation of numerical calculations. Class diagrams are presented for describing the geometry of the object under study, as well as its constituent segments. In order to improve the performance of calculations, an algorithm for combining geometry segments was developed, in which the original algorithm was divided into separate fragments by introducing a number of conditional structures. As a result of experimental data processing, regression equations were obtained that describe the dependence of the algorithm execution time on the number of joins. The developed algorithm and class library make it possible to work with the description of the geometry of the object under study as a set of parameterized primitives and reduce the time spent on the formation of the description of the computational domain by up to 27%.

1 Introduction

At present, the assessment of the hydrodynamic impact on bank protection structures and coastal structures installed on the bottom surface of river systems is the most important urgent task. When constructively transforming the morphometry of the riverbed and floodplain, one should take into account the dynamics of the processes of bank formation, and study the formation of the bottom profile in the coastal zone of the reservoir under the influence of currents. To determine the dynamics and trends of phenomena that occur in channel systems and predict possible interference with the ecosystem, it becomes necessary to develop algorithms and programs for numerically solving the problems of matter transfer in channel systems, including channel processes in river sections with complex channel and floodplain morphometry. With their use, an important scientific and practical task of assessing the

[*] Corresponding author: litvinovvn@rambler.ru

impact of the planned construction on the flooding of floodplain territories and erosion of the riverbed will be solved.

To predict the state of shallow water bodies, when designing coastal engineering structures, mathematical models are built that take into account the unique features of the studied water body - climatic factors, hydrodynamic regimes and the complex geometry of the computational area. Among the works of Russian scientists devoted to the study and forecast of aquatic ecosystems, one can single out the works of Marchuk G.I. [1], Matishov G.G. [2, 3], Sukhinov A.I. [4, 5], Tyutyunov Yu.V.[6], Yakusheva E.V. [7], Ilyicheva V.G. and others. Leading foreign research centers and organizations are developing models, software systems and information systems for monitoring and predicting the state of water bodies: Sweden's Meteorological and Hydrological Institute; Center for Water Re-search; National Oceanic and Atmospheric Administration; Center for Ecology and Hydrology [8, 9].

Existing software systems that allow modeling hydrodynamic processes (SALMO, CHARISMA, Mars3d, CHTDM, CARDINAL, PHOENICS, Ecointegrator, etc.) do not take into account the spatially inhomogeneous movement of the aquatic environment, do not have the necessary accuracy for modeling the vortex structures of flows, are not conservative, do not take into account the complex shape of the bottom and coast relief, evaporation, river runoff, salinity, temperature and other factors, and also show instability with significant depth differences and changes in the density of the aquatic environment [10–13].

2 Material and methods

Let's consider the complete 2D system of equations of the pollutant distribution, including the oil and its refined products, in the computational domain – Azov Sea – with the side surface σ, the water undisturbed surface Σ_0 in the Cartesian coordinate system for the axes Oy, Ox, directed to the north and east, respectively. The model is based on the researches [3, 4, 14-16] and has the following form:

$$\frac{\partial S_i}{\partial t}+u\frac{\partial S_i}{\partial x}+v\frac{\partial S_i}{\partial y}+\sigma_i S_i=\mu_i\left(\frac{\partial^2 S_i}{\partial x^2}+\frac{\partial^2 S_i}{\partial y^2}\right)+f(\mathrm{x},\mathrm{y},\mathrm{t}), \quad (1)$$

where u, v are components of the water flow velocity vector; σ_i is the decomposition coefficient of i-th impurity; f is a chemical and biological source (the runoff); μ_i are diffusion coefficients in the horizontal direction; S_i is the concentration of i-th impurity, $i=\overline{1,6}$: 1 is the mercury (Hg); 2 is the plumbum (Pb); 3 is the manganese (Mn); 4 is the iron (Fe^{+2}); 5 is the phytoplankton (*Sceletonema costatum* diatoms); 6 are oil and petroleum products.

The model includes pollutants whose concentration in the Azov Sea, according to the analysis of scientific publications and environmental databases, exceeds the maximum permissible concentration (MPC) [10]. This model takes into account the water flow movement; microturbulent diffusion; interaction of pollutants and plankton; biogenic, temperature and oxygen regimes; the influence of salinity.

Let's define the initial conditions:

$$S_i(\mathrm{x},\mathrm{y},0)=\mathrm{S}_{0i}(\mathrm{x},\mathrm{y}),\ i=\overline{1,6}. \quad (2)$$

The computational domain G it is a closed area bounded by the undisturbed water surface Σ_0 and the coastline σ for $0 < t \le T_0$. $\Sigma = \Sigma_0 \cup \sigma$, Σ is a piecewise smooth boundary of the domain G.

Let $\mathbf{U_n}$ is the component of the water flow velocity vector, normal to the boundary Σ, $\mathbf{n}$ is the external normal vector to the Σ. Then the boundary conditions for the model (1) will have the form:

$$S_i = 0 \ on\, \sigma,\ if\ U_n < 0;\ \frac{\partial S_i}{\partial \mathbf{n}} = 0\ on\ \sigma,\ if\ U_n \ge 0;$$

$$S_i(\mathrm{x},\mathrm{y},0) = \mathrm{S}_{0i}(\mathrm{x},\mathrm{y}),\ i = \overline{1,6}. \tag{3}$$

We took into account the water exchange with the Black Sea, the flow of the Don River, the complex shape of the coastline, and the bottom relief at setting the boundary conditions.

Let's take the calculation area G with the following parameters: l_x is the characteristic dimension along the axis Ox, l_y – along the axis Oy. Let us compare the specified area with a uniform computational grid of the following form:

$$\begin{gathered} W = \{x_i = ih_x, y_j = jh_y;\ i = \overline{0, n_x - 1}, j = \overline{0, n_y - 1}; \\ (n_x - 1)h_x = l_x, (n_y - 1)h_y = l_y\}, \end{gathered} \tag{4}$$

where h_x, h_y are steps of the computational grid in the corresponding spatial directions; n_x , n_y are number of nodes of the computational grid in the corresponding spatial directions.

Let us represent the set of nodes of the computational grid in the form

$$Q = \{q_{i,j}, i = \overline{0, n_x - 1}, j = \overline{0, n_y - 1}\},\ q_{i,j} = \langle x_i, y_j \rangle, \tag{5}$$

where $q_{i,j}$ is the grid node.

Number of grid nodes N_Q calculated by the formula

$$N_Q = n_x \cdot n_y. \tag{6}$$

3 Calculation

Modeling of hydrophysical processes is carried out by the finite difference method [13-17], while software implementation of the description of the geometric shape of the object under study and boundary conditions is required. In the case of studying the process of spreading pollutants over the water surface, it is necessary to take into account the geometry of objects of complex shape, which include coastal engineering structures that are part of the coastline. The solution of this problem can cause significant difficulties.

The generally accepted approach to describing the geometry of an object of complex shape under study is the approach in which the geometric parameters of the object are specified directly by using one of the geometry description formats: STL, COLLADA, IGES, STEP, VRML/X3D. The listed formats are convenient for transferring data between applications, however, when modeling by the finite difference method, they require complex transformations.

Solving optimization engineering problems requires not only a more convenient way to describe the geometry, but also the possibility of simply varying various parameters, which requires detailed work with the original geometry model and building on its basis a whole set of geometries with small changes.

The approach proposed in the article and the algorithm implemented on its basis make it possible to work with the geometry of the object under study as with a set of two-dimensional geometric primitives that can be superimposed on each other. The absence of dependencies of the geometric dimensions of the primitives from each other allows you to quickly and accurately describe the object and proceed directly to modeling. A distinctive feature of the proposed approach is that geometric primitives can be of two types: a continuous medium and a cutout. In the developed class library, they are heirs of the abstract class $Geometry2DPrimitive$ (Fig. 1), containing data on the coordinates of the offset of primitives relative to the geometry placement point ($\_dS0$), the type of primitives ($primitiveType$) and the "cut" property flag ($\_isCavity$). Behavior details are described in derived classes. In particular, the class $Geometry2DPrimitiveRectangle$ (Fig. 1) models a rectangle.

When applying the computational grid, the problem arises of correctly determining the characteristics of each node, i.e. it is necessary to determine whether the node is internal, boundary or fictitious. In the general case, the algorithm for solving this problem has asymptotic complexity $O(n^2)$. The algorithm proposed by us allows us to reduce it by one order due to the transition to the description of the geometric segments of the body under study, located on the secant straight lines of the computational grid (geometry segments).

We represent the geometry segment as a tuple $<\_x_1,\ \_x_2,\ \_gridSegmentType,\ \_isFillBottom,\ \_isCavity>$, where $\_x_1$, $\_x_2$ – start and end coordinates of geometry segment, $\_gridSegmentType$ – geometry segment type (0 – boundary segment, 1 – internal segment), $\_isFillBottom$ – filling of the space under the segment (0 – empty from below, 1 – filled from below), $\_isCavity$ – flag of belonging of a segment to a geometric primitive that is a cutout (0 – continuous medium, 1 – cutout).

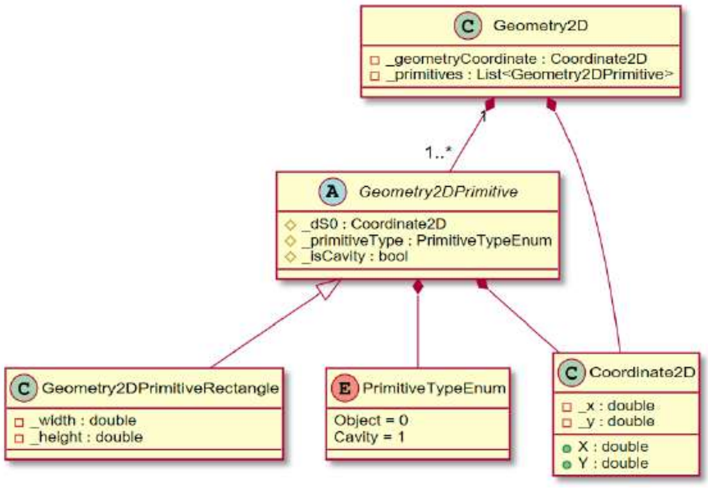

Fig. 1. Class diagram for describing the geometry of the object under study.

To work with a segment in an object-oriented programming style, a universal class has been developed $GeometryPrimitiveSegment<T>$, where T – the data type of the start and end coordinates of the segment. For the convenience of working with the geometry segment type, an enumeration has been created *GridSegmentTypeEnum* from two elements: *Boundary* – boundary segment and Internal – internal segment (Fig. 2).

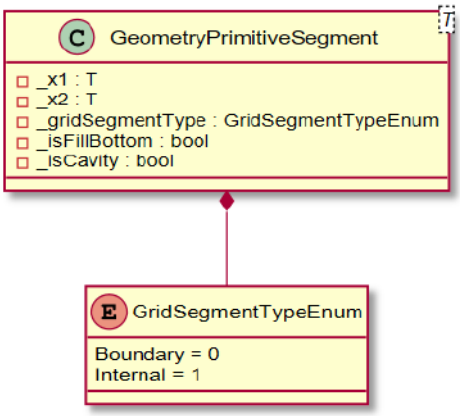

Fig. 2. Geometry segment class diagram.

Let's introduce the notation: k_{11} – coordinate of the beginning of the first segment, k_{12} – coordinate of the end of the first segment, k_{21} – coordinate of the beginning of the second segment, k_{22} – the coordinate of the end of the second segment.

Let's introduce boolean variables: $A = k_{11} < k_{21}$; $B = k_{11} == k_{21}$; $C = k_{11} > k_{21}$; $D = k_{12} < k_{22}$; $E = k_{12} == k_{22}$; $F = k_{12} > k_{22}$; $G = k_{11} == k_{22}$; $H = k_{12} == k_{21}$; $I = k_{12} < k_{21}$; $J = k_{11} > k_{22}$; K – type of the first segment; L – type of the second segment; M – filling of space under the first segment; N – filling of the space under the second segment; O – the second segment is a cutout.

The introduced logical variables allow using the algebra of logic to describe all possible combinations of overlapping geometry segments:

1) end-to-end, the second segment is located to the right of the first: H ;
2) end-to-end, the second segment is located to the left of the first: G;
3) full overlay: $B \wedge E$;
4) second segment inside the first: $A \wedge F$;
5) the second segment overlaps the first: $C \wedge D$;
6) the left sides of both segments are aligned, the length of the second segment is less than the length of the first: $B \wedge F$;
7) the left sides of both segments are aligned, the length of the second segment is greater than the length of the first: $B \wedge D$;
8) alignment on the right, the length of the second segment is less than the length of the first: $A \wedge E$;
9) alignment on the right, the length of the second segment is greater than the length of the first: $C \wedge E$;
10) overlap on the left: $C \wedge F$;
11) overlap on the right: $A \wedge D \wedge \bar{I}$;
12) there is no intersection, the second segment is located to the right of the first: I ;
13) there is no intersection, the second segment is located to the left of the first: J .

Along with the option of overlapping segments, the model must take into account all possible combinations of segment types and infills. By filling we mean the presence of a layer of material on the previous cut (filling from below) or on the next one (filling from above). The presence of filling at the top and bottom indicates that the segment is internal. This implies the requirement to check the correctness of the data: the inner layer cannot be filled from below.

Four combinations of geometry segment types are possible: boundary – boundary $(\overline{K} \wedge \overline{L})$, boundary – internal $(\overline{K} \wedge L)$, internal – boundary $(K \wedge \overline{L})$, internal – internal $(K \wedge L)$.

Similarly, it is required to take into account four options for the combination of fillings of the merged geometry segments: top – top $(\overline{M} \wedge \overline{N})$, top – bottom $(\overline{M} \wedge N)$, bottom – top $(M \wedge \overline{N})$, bottom – bottom $(M \wedge N)$.

It is assumed that the first segment must necessarily be an object, therefore, it is only necessary to check the second object for the set "Cut" property, i.e. there are two options – no $(\overline{O})$, yes (O).

All possible combinations of groups of segments were found in the basis of the original calculation algorithm, the structural linear function. In order to calculate the results of the calculations of the original algorithm, they were divided into separate fragments by introducing conditional structures.

Here is a fragment of an improved algorithm for combining geometry segments:

1) Variable initialization k_{11}, k_{12}, k_{21}, k_{22}.
2) Calculation of the values of boolean variables A, B, C, D, E, F, G, H, I, J, K, L, M, N, O.
3) Checking the correctness of the data. If the result of evaluating the expression $L \wedge \overline{N} \vee K \wedge \overline{M}$ is true, so there is an error in the input and an exception is thrown.
4) Search for an overlay option in accordance with the logical expression given in Table 1.
5) If the overlay option is not found, then there is an error in the input data and an exception is thrown. Further actions are performed for the found overlay option. For example, overlay option No. 1 is described. For other options, the actions are similar and differ only in combinations of logical conditions.
6) If the first segment is boundary $(\overline{K})$, then go to step 7, otherwise go to step 14.
7) If the second segment is boundary $(\overline{L})$, then go to step 8, otherwise go to step 11.
8) Checking conditions $\overline{M} \wedge \overline{N} \wedge \overline{O}$, $\overline{M} \wedge \overline{N} \wedge O$, $\overline{M} \wedge N \wedge \overline{O}$, $\overline{M} \wedge N \wedge O$, $M \wedge \overline{N} \wedge \overline{O}$, $M \wedge \overline{N} \wedge O$, $M \wedge N \wedge \overline{O}$, $M \wedge N \wedge O$ and creating the resulting segments for the true expression.
9) Return created result segments.
10) End.
11) Check conditions $\overline{M} \wedge N \wedge \overline{O}$, $\overline{M} \wedge N \wedge O$, $M \wedge N \wedge \overline{O}$, $M \wedge N \wedge O$ and creating the resulting segments for the true expression.
12) Return created result segments.
13) End.
14) If the second segment is boundary $(\overline{L})$, then go to step 15, otherwise go to step 18.
15) Check conditions $M \wedge \overline{N} \wedge \overline{O}$, $M \wedge \overline{N} \wedge O$, $M \wedge N \wedge \overline{O}$, $M \wedge N \wedge O$ and creating the resulting segments for the true expression.
16) Return created result segments.
17) End.

18) Check conditions $M \wedge N \wedge \overline{O}$, $M \wedge N \wedge O$ and creating the resulting segments for the true expression.
19) Return created result segments.

4 Results and discussion

Solving the problem of the distribution of pollutants over the water surface requires a detailed description of the coastline geometry. When modeling the process of spreading pollutants, it is necessary to take into account not only the natural geometry of the coastline of a reservoir, but also the presence of artificial structures and buildings.

A team of authors has developed an algorithm for describing artificial structures as a set of geometric primitives, the combination of which, together with a natural coastline, makes it possible to describe the complex geometry of the coastline for the problem described above.

Comparison of the performance of the conventional and proposed algorithms was performed on a personal computer with an Intel Core i5-6600 3.3 GHz processor and 32 GB of DDR4 RAM. The initial data were randomly generated objects of geometry segments, which were then combined. The comparison results are shown in Table 1

Table 1. Performance comparison results of the original and improved algorithms for merging geometry segments.

Number of associations N_m, $\times 10^6$	**2**	**4**	**6**	**8**	**10**
Initial algorithm T_1, sec	0.075	0.164	0.234	0.295	0.376
Advanced algorithm T_2, sec	0.055	0.149	0.196	0.253	0.308

As a result of experimental data processing, regression equations were obtained that describe the dependence of the algorithm execution time on the number of joins: $T_1 = 360N_m + 141$ ($R^2 = 0.995$) – for the original algorithm; $T_2 = 312N_m + 10$ ($R^2 = 0.996$) – for the improved algorithm. Smaller values of the coefficients of the regression equation for the improved algorithm indicate its greater computational efficiency.

The proposed algorithm allows increasing performance by 14% to 27% and has a parallelization resource, which allows it to be used in building software systems for solving problems of mathematical physics and computer-aided design systems.проектирования.

5 Conclusions

When modeling the process of spreading pollutants over the water surface, the computational domain was described in the form of a two-dimensional computational grid. The description of the geometry of the coastline takes into account the geometric characteristics of artificial structures.

The article presents class diagrams for describing the geometry of the object under study, as well as its constituent segments. In order to improve the performance of calculations, an algorithm for combining geometry segments was developed, in which the original algorithm was divided into separate fragments by introducing a number of conditional structures.

The use of the modified algorithm makes it possible to reduce the calculation time when combining geometry segments by up to 27% and has a parallelization resource, which allows it to be used in the construction of software systems for solving problems of mathematical physics and computer-aided design systems.

Acknowledgments

The study was carried out with the financial support of the Russian Science Foundation within the framework of the scientific project No. 21-71-20050.

References

1. G.I. Marchuk, B.A. Kagan, Dynamics of ocean tides (Gidrometeoizdat, Leningrad, 1983).
2. G.G. Matishov, V.G. Ilyichev, Report of the Academy of Sciences, **406(2)**, 249–251 (2006).
3. G.G. Matishov, S.V. Berdnikov, A.P. Zhichkin, et. all., Atlas of climate change in large marine ecosystems of the Northern Hemisphere (1878-2013). Region 1. Seas of the Eastern Arctic. Region 2. Black, Azov and Caspian seas. SSC RAS Publishing House, Rostov-on-Don (2014)
4. A.I. Sukhinov, A.E. Chistyakov, A.V. Nikitina, Y.V. Belova, V.V. Sumbaev, A.A. Semenyakina, Supercomputer Modeling of Hydrochemical Condition of Shallow Waters in Summer Taking into Account the Influence of the Environment. In: Sokolinsky, L., Zymbler, M. (eds) Parallel Computational Technologies. PCT 2018. Communications in Computer and Information Science, Springer, Cham., **910**, 336–351 (2018). DOI: 10.1007/978-3-319-99673-8_24.
5. V.A. Gushchin, A.I. Sukhinov, A.V. Nikitina, et al., A Model of Transport and Transformation of Biogenic Elements in the Coastal System and Its Numerical Implementation. Comput. In: Math. and Math. Phys., **58**, 1316–1333 (2018). DOI: 10.1134/S0965542518080092.
6. Y. Tyutyunov, L. Titova, Simple models for studying complex spatiotemporal patterns of animal behavior. In: Deep Sea Research Part II Topical Studies in Oceanography, **140**, 193-202. (2017). DOI: 10.1016/j.dsr2.2016.08.010.
7. E. Yakushev, G. Mikhailovsky, Mathematical modelling of the influence of marine biota on the carbon dioxide ocean-atmosphere exchange in high latitudes. In: Third International Symposium on Air-Water Gas Transfer, pp.37–48. AEON Verlag & Studio, Hanau. (1995).
8. J. Oliger, A. Sundstorm, Journal on Applied Mathematics, **35**, 19–45 (1978). DOI: 10.1137/0135035.
9. P. Marchesiello, J.C. McWilliams, A.F. Shchepetkin, Ocean Modelling, **3**, 1-20. (2001). DOI: 10.1016/S1463-5003(00)00013-5.
10. N.E. Voltsinger, K.A. Klevanny, E.N. Pelinovsky, Long-wave dynamics of the coastal zone (Gidrometeoizdat, Leningrad, 1989)
11. A.A. Androsov, N.E. Voltsinger, Straits of the oceans. General approach to modeling (Science, St. Petersburg, 2005)
12. E. Alekseenko, B. Roux, A.I. Sukhinov, et al., Computers and Fluids, **77**, 24–35 (2013). DOI: 10.1016/J.COMPFLUID.2013.02.003
13. A.V. Nikitina, L.V. Kravchenko, I. Semenov, Y.V. Belova, A.A. Semenyakina, MATEC Web of Conference, **22**, 04025. (2018). DOI: 10.1051/matecconf/201822604025
14. O.M. Belotserkovsky, A.M. Oparin, A.M. Chechetkin, Turbulence. New approaches, (Nauka, Moscow, 2003)

15. A.A. Samarsky, P.N. Vabishchevich, Numerical methods for solving convection-diffusion problems (URSS, Moscow, 2009)
16. A.N. Konovalov, The steepest descent method with an adaptive alternating-triangular preconditioner. In: Differential equations? **40(7)**, 953-963 (2004)
17. A.I. Sukhinov, A.E. Chistyakov, A.V. Shishenya, et al., Math Models Comput Simul, **10**, 648–658 (2018). DOI: 10.1134/S2070048218050125.