S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

УДК 519.6
DOI 10.33048/semi.2019.16.xxx MSC ??X??

AN AUTOMATIC 3D MESH MODEL GENERATOR ACCOUNTING FOR BATHYMETRIC DATA FOR TSUNAMI WAVE SIMULATIONS

CHUKHMANOV N.V., BORISENKO O.N., KOZELKOV A.S., SMOLKINA D.N., CHERENKOVA M.V., KUZMENKO M.V., GINIYATULLINA A.G., TIMAEVA T.E.

Abstract. Numerical simulations of the processes of occurrence and propagation of tsunamis constitute a relevant area of scientific research and are currently used, for example, in zoning and prediction of tsunami occurrence. As tsunami flows have a complex three-dimensional structure, to take into account all their features, it is necessary to use numerical simulations based on the system of three-dimensional Navier-Stokes equations.

The most labor-consuming of the entire simulation process, as a rule, is the stage of generating a computational mesh. At the same time, it is very important, since generating a high-quality mesh model directly controls the overall accuracy of task calculations. Due to the variability of the domain in spatial directions and the characteristics of the waves themselves, correct numerical simulations require special-type computational meshes with a uniform distribution of cells in the directions of wave propagation deep into water and a higher resolution at the air/water interface. To generate such meshes, an automatic generator for tsunami simulations has been implemented.

The paper describes the algorithms and formulas used in the implementation of the automatic generator of polyhedron mesh models for simulations of landslide tsunami waves. This generator enables automatic preparation of mesh models for simulating the process of tsunami propagation in order to assess the risk of tsunami occurrence in various water areas.

Keywords: polyhedron mesh, mesh model, mesh generator, tsunami, unstructured meshes, arbitrary polyhedrons.

1. Introduction

Currently, simulations of tsunami waves are mainly performed using models based on the shallow-water theory [1, 2, 3].

This theory has been and continues to be used for simulations of historical and prognostic tsunamis, so the bibliography here is huge. The system of shallow-water equations is obtained by integration over depth of full three-dimensional fluid dynamics equations (Euler equations), which excludes the vertical coordinate and essentially reduces the three-dimensional system to two-dimensional. Such a system has proven itself in modeling the propagation of tsunami waves, but it is not able to reproduce the complex structure of their three-dimensional flow. In addition, it is necessary to take into account the dispersion of waves both on long routes and, for landslide and volcanic tsunamis, the size of their focus, which is comparable to the ocean depth. This can be done using the non-linear Boussinesq equations [4, 5], which in the case of tsunamis are two-dimensional and are also not able to reproduce the flow structure.

At the same time, tsunamis have a number of physical properties manifested only in the three-dimensional case, which must be taken into account when zoning and predicting tsunamis, for example, the movement of liquid in the tsunami focus when landslides, rock fragments, celestial bodies enter water, or when converting bottom vibrations into displacements of the water surface. Considering the interaction with coastal infrastructure, the stage of wave transformation in the shelf zone during its collapse, rolling ashore and moving over land also has three-dimensional properties. Special models and algorithms are required to simulate these processes.

To take into account all the features of the three-dimensional structure of tsunamis, it is necessary to use numerical simulations based on the solution of the Navier-Stokes system of three-dimensional equations for a viscous liquid [6, 7, 8, 9, 10, 11]. In the general case, this system has no analytical solution, so all solutions are obtained numerically. The problems of discretization of the Navier-Stokes equations and their numerical solution belong to the key issues of mathematical modeling [12, 13, 14]. At present, the Navier-Stokes equations begin to be actively used in tsunami simulations [15, 16].

Mesh model generation is one of the most time-consuming stages in the mathematical modeling of tsunamis, which often takes a significant part of the time, while the accuracy of the resulting solution depends directly on the mesh quality. Different types of computational meshes are used for simulations of different processes:

- adaptive Cartesian template-based polyhedral grids consisting of truncated hexagons and polygonal prisms [17, 18];
- meshes consisting of tetrahedrons and triangular prisms near solids [19];
- polyhedron meshes of nonconvex cells with an arbitrary number of faces obtained by merging tetrahedrons [20];
- block-structured meshes [21].

When solving the Navier-Stokes equations in simulations of flows occurring in various industrial and scientific applications, it is necessary to generate mesh models bounded by the geometric dimensions of the region under consideration. The experience of using the Navier-Stokes equations for the numerical solution of various classes of problems shows that generating a mesh model can take more than 70 percent of the time from setting up the task to obtaining the final result.

The features of numerical simulations of tsunami propagation impose certain requirements on the mesh models. First of all, this is due to the mutliscale nature of the domain and parameters of simulated waves. The size of the mesh model cells at the wave source is no more than several tens of meters, while the domain itself

can be as large as hundreds of kilometers. Thus, the cell sizes at the wave source, as a rule, differ several times from the cell sizes in the region of tsunami propagation. To improve the level of detail of mesh models, control volumes with specified cell sizes can be used, which allows avoiding a significant increase in the total number of cells in the task.

Thus, it becomes necessary to create a method for generating a computational mesh for modeling tsunami waves in water bodies taking into account their topology based on bathymetric data. The developed methodology includes the steps of preparing a closed mesh of triangles based on bathymetric charts for the domain and generating a computational mesh from the resulting mesh of triangles.

To describe the motion of tsunami waves in the open ocean, where the wave has a small amplitude, the cells must be as small as several centimeters, while the ocean can be several kilometers deep. As a result, generating meshes with so many cells results in mesh models composed of hundreds of millions or billions of cells. One of the solutions to achieve the required cell size in the simulated region of wave motion with an acceptable number of cells in the mesh model as a whole is to generate a mesh refined at the water/air interface.

When constructing a mesh model, it is necessary to take into account bathymetric data of real bodies of water, that is, the distribution of their bottom topology heights. For this purpose, based on bathymetry data, one creates a triangular surface mesh by triangulation at bathymetric points and uses it then as an input mesh for generating the computational model.

The approach used previously [10] for constructing a volume mesh consisted of several stages:

- constructing a mesh for the most of the model using a block-structured mesh generator;
- constructing an unstructured mesh in a part of the mesh model in the bottom topology region using a trimming generator for more accurate approximation of the surface and detailing of the landslide area;
- combining the constructed models into one for further calculations

This approach produced acceptable results, but it had significant drawbacks:

- preparation of the geometry and mesh model using the block structured mesh generator was carried out by hand, which required a lot of time and effort of the user;
- when constructing an unstructured mesh by the trimming generator [Borisenko et al., 2018] in the bottom topology region, cell sizes were adapted based on the minimum cell size setting in the region, which, together with the multiscale nature of the model, led to an uncontrolled number of computational cells;
- the constructed meshes were to be matched and a single mesh was to be constructed from the fragments obtained.

The purpose of this article is to present a developed methodology for constructing a computational grid for modeling tsunami waves in water areas, taking into account the surface topography based on bathymetric data. This methodology includes stages for preparing a closed mesh of triangles based on bathymetric maps to define the modeling area, as well as the automated construction of the computational grid from the generated triangular mesh.

2. Preparation of a closed mesh of triangles based on bathymetric maps for forming the modeling area

The methodology for constructing a grid model includes stages for creating a closed computational area based on bathymetric maps, filling it with a surface grid of triangles, and then constructing a special type of volumetric grid based on that. Bathymetry consists of a set of coordinates with associated height values relative to the surface of the world Ocean. In the case of water bodies, this height determines the depth at the surface point. To obtain the bathymetry of the selected water body, the special program is used, which provides information about the terrain of the area defined by ranges of latitude and longitude. Figure 1 shows the program window with the selected water area.

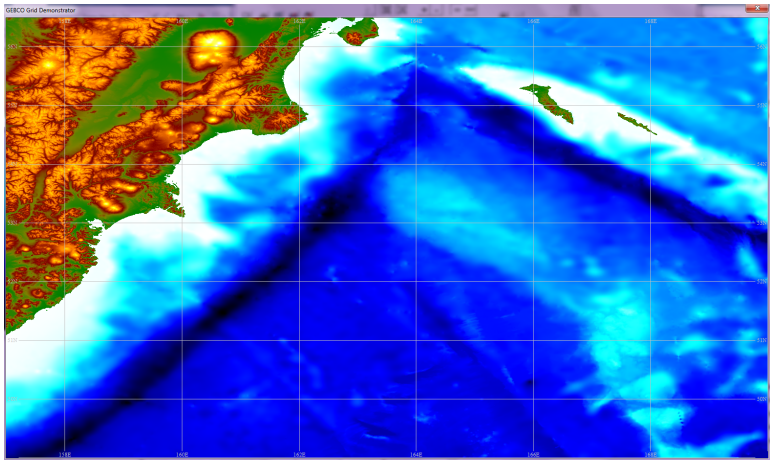

Figure 1. Bathymetry of the water area

For the specified water area, a topographic map is created, consisting of a regular uniform grid where the elevations from the bathymetric map are known at the nodes. Based on this map, a closed computational domain is then constructed along with a surface mesh of triangles, which will be used as initial data for creating a specialized computational grid for tsunami modeling, as illustrated in Figure 2 and Figure 3.

3. Building a computational grid based on a closed surface mesh of triangles

The process of volume mesh generation consists of the following stages:

- checking the initial triangular mesh for geometric and topological errors;
- recovering/obtaining characteristic curves of the model from the surface mesh;
- constructing a Cartesian template mesh with varying sizes in each spatial direction;
- trimming the template mesh cells with the surface mesh triangles;
- improving the quality of the trimmed cells.

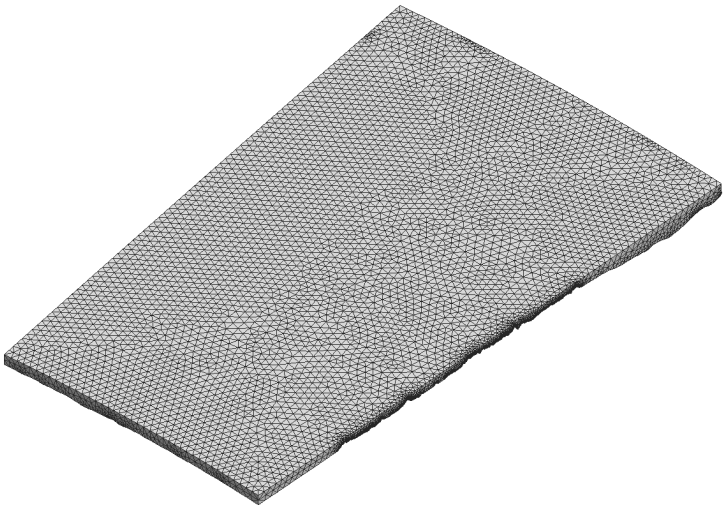

FIGURE 2. Surface grid by bathymetry (external boundary)

Because the surface mesh supplied to the generator must be of satisfactory quality, it is checked for errors, such as open contours and intersections, before generating the volume mesh. If the surface mesh is of poor quality, the volume mesh is not generated.

After the input surface mesh is checked, characteristic curves are reconstructed on it. The characteristic curves are mapped from the features of the geometric model (non-smooth surface joints, sharp geometry edges). The stage of detecting the model features [22] is performed to describe sharp edges and/or surface details that need to be preserved when generating the volume mesh. The generator takes into account three types of characteristic curves: characteristic curves of the model created in the graphical interface (before starting the generator), boundaries of selected regions, and geometry features detected automatically at the mesh generation stage.

After the model features are reconstructed, a Cartesian template mesh is created with dimensions specified for each spatial direction. To create a mesh, one must set partition laws for each spatial direction, the size of the regular cell of the region (the maximum cell size), and the number of transition layer cells to balance the levels of the template mesh. At this stage, the entire domain is filled with parallelepipeds constructed in accordance with the specified dimensions and partition laws. An example of a computational mesh condensed at the water/air interface is shown in Figure 4.

The mesh fragment enclosed in the blue box in the figure is zoomed-in in Figure 5. Figure 6 shows the mesh lines in this fragment. Figure 6 indicates that the size of the mesh cells increases as a geometric progression, starting from the water/air interface.

Since the mesh is partitioned in each direction separately, in fact, the task of constructing a partition in accordance with the specified law is one-dimensional. Consider the partition along the axis, assuming that this direction defines the depth of the water area.

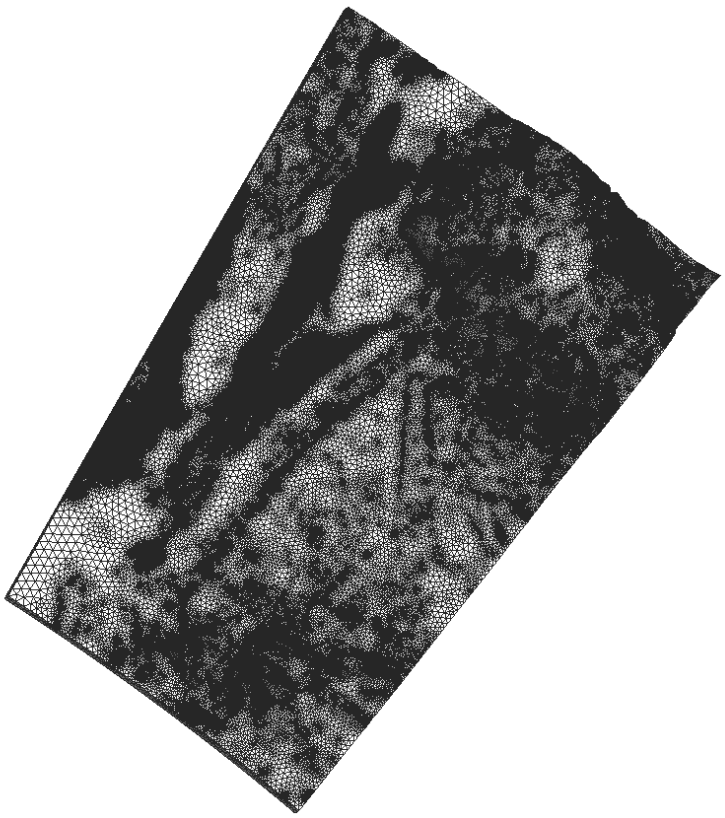

FIGURE 3. Surface grid by bathymetry (waterbody relief)

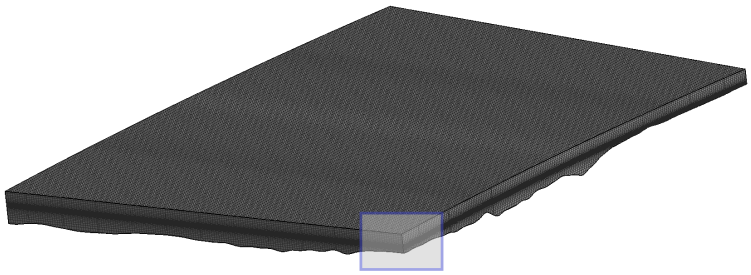

FIGURE 4. An example of the computational mesh condensed at the water/air interface

At the first stage of constructing the template mesh, the input surface mesh is immersed in the overall box to determine the mesh generation limits in each spatial direction.

Denote the mesh generation limits along the z axis as $[z_{\min}; z_{\max}]$

The input parameters for generating a geometric-progression mesh are:

- starting point $z = z_c, z_c \in [z_{\min}, z_{max}]$;

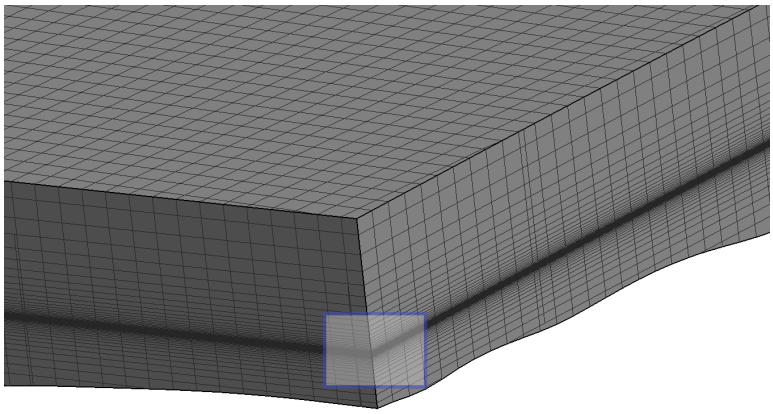

FIGURE 5. The zoomed-in mesh fragment

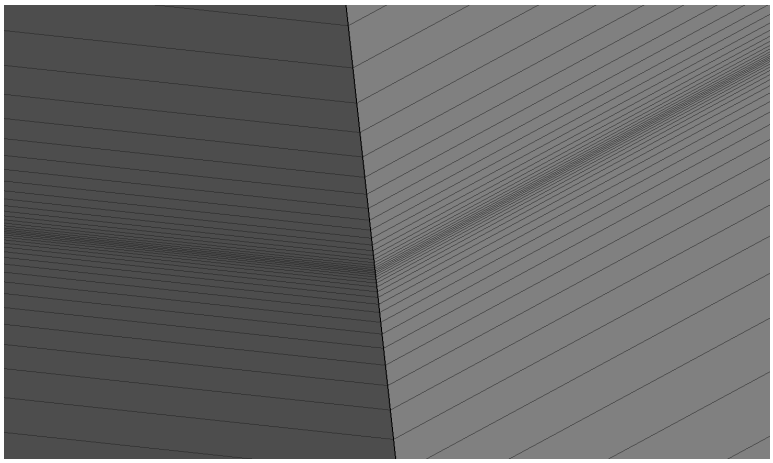

FIGURE 6. Mesh lines at the water/air interface

- growth rate $r > 1$;
- the first cell size $C_0 > 0$.

The coordinate of the starting point typically defines the interface. The cell size at the j-th level of partitioning is calculated using formula (1):

$$C_j = C_0 r^j \tag{1}$$

where C_0 – is the size of the first cell, C_j – is the cell size at the j-th level of partitioning, r – is the progression growth rate, j – is the partitioning step number.

The cell size at the j-th level of partitioning is calculated using formula (2):

$$H_j = \frac{C_0 - C_j r}{1 - r} \tag{2}$$

Determine the dimensions of the region along the z axis according to formula (3):

$$H^g = z_{\max} - z_{\min} \tag{3}$$

where $z_{\max}$ – is the value of the maximum coordinate of the region in the z direction, $z_{\min}$ – is the value of the minimum coordinate of the region in the z direction.

The one-dimensional space along the z axis can be divided into two half-spaces relative to the starting point of the z_c, partition, which we conventionally call "upper"and "lower"half-spaces. In the formulas, all quantities related to the "upper"half-space are indicated by the subscript "+"to the right of the quantity, and those related to the "lower"half-space are indicated by the subscript "–".

Then, the dimensions of the region in the upper and lower half-spaces relative to the starting point are calculated using formulas (4) and (5) respectively:

$$H^g_+ = z_{\max} - z_c \tag{4}$$

$$H^g_- = z_c - z_{\min} \tag{5}$$

where z_c – is the starting point of z partitioning.

For the template generator to work correctly, the dimensions of the region must be adjusted in accordance with the specified mesh generation parameters.

To determine the dimensions of the "upper"and "lower"half-spaces, formula (5) is transformed into formula (6):

$$H^g_{+/-} = \frac{C_0 - C^{\max}_{+/-} r}{1 - r} \tag{6}$$

where $C^{\max}_{+/-}$ is the maximum cell size in the half-space.

From formula (6) one can determine the maximum cell size in each half-space based on the dimensions of the input surface mesh using formula (7):

$$C^{\max}_{+/-} = \frac{C_0 - H^g_{+/-}(1 - r)}{r} \tag{7}$$

where $H^g_{+/-}$ is calculated using formulas (4) и (5) for the respective half-spaces.

On the other hand, the maximum cell size, based on the geometric progression, is calculated using formula (8):

$$\bar{C}^{\max}_{+/-} = C_0 r^{L_{+/-}} \tag{8}$$

where $L_{+/-}$ – is the number of partitioning levels to reach the value of $\bar{C}^{\max}_{+/-}$.

$\bar{C}^{\max}_{+/-}$ and $C^{\max}_{+/-}$ need to be matched. To do this, one needs to determine the number of levels $L_{+/-}$ to achieve $C^{\max}_{+/-}$ calculated using formula (8).

Substituting the value of the maximum cell size from formula (7) into (8) we convert the latter to (9):

$$r^{L_{+/-}} = \frac{C^{\max}_{+/-}}{C_0} \tag{9}$$

Next, (10) – (12) are carried out to calculate $L^{corr}_{+/-}$, which is the number of levels of geometric progression adjusted to the geometry dimensions:

$$L_{+/-} \ln r = \ln C^{\max}_{+/-} - \ln C_0 \tag{10}$$

$$L_{+/-} \ln r = \ln C^{\max}_{+/-} - \ln C_0 \tag{11}$$

$$L_{+/-}^{corr} = \frac{\ln C_{+/-}^{\max} - \ln C_0}{\ln r} \tag{12}$$

The corrected size of the maximum cell in each half-space $\left[C_{+/-}^{\max}\right]^{corr}$ is then determined using formula (13):

$$\left[C_{+/-}^{\max}\right]^{corr} = C_0 r^{L_{+/-}^{corr}} \tag{13}$$

The region dimensions adjusted relative to the maximum cell size in each half-space $H_{+/-}^{corr}$ are calculated by formula (14):

$$H_{+/-}^{corr} = \frac{C_0 - \left[C_{+/-}^{\max}\right]^{corr} r}{1 - r} \tag{14}$$

The maximum values of the coordinates of the “upper” and “lower” half-spaces are calculated using formulas (15) and (16), respectively:

$$z_{corr}^{+} = z_c + H_{+}^{corr} \tag{15}$$

$$z_{corr}^{-} = z_c - H_{-}^{corr} \tag{16}$$

where H_{+}^{corr} are the adjusted dimensions of the “upper” half-space, H_{-}^{corr} are the adjusted dimensions of the “lower” half-space.

As a result, the adjusted dimensions of the region are calculated by formula (17):

$$H_{corr}^{g} = z_{corr}^{+} - z_{corr}^{-} \tag{17}$$

where z_{corr}^{+}, z_{corr}^{-} are the adjusted maximum coordinates of the “upper” and “lower” half-spaces, respectively.

For this partition, local indexing is introduced from the starting point of the partition so that the indexes of the quantities are positive in the "upper"half-space and negative in the "lower"half-space.

The coordinates of the mesh node at the k–th level are calculated using the mesh indexes by formula (18):

$$z_k = z_c + \mathrm{sgn}(k) C_0 \sum_{n=1}^{|k|} r^{n-1} \tag{18}$$

where $\mathrm{sgn}(k)$ is the sign defining the position of the coordinate in the half-spaces.

An additional condition that determines the parameters of the template mesh is control regions, in which a certain cell size is specified.

The principle of constructing a mesh in control regions is that the size of the cell of the template mesh that falls into the control region as a whole or in part cannot be larger than the size specified for this region. The cell inside the control region is split to the desired size in the region. At the moment, it is possible to set control regions of the following types: rectangular parallelepiped, sphere, cone, cylinder, polyhedron and their thin-walled variants, as well as regions in the form of an arbitrary closed triangulation loaded as an stl-file.

After selecting the region, one needs to set parameters that correspond to the type of its geometric shape, and the local cell size.

Figures 7 and 8 show the result of volume mesh generation without setting the control region and with it.

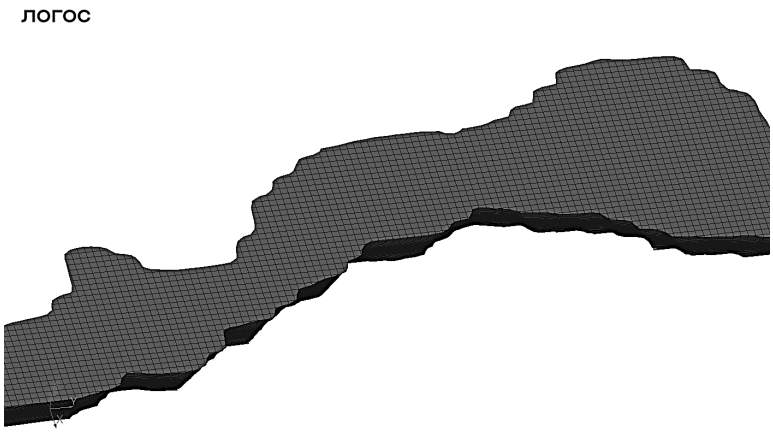

FIGURE 7. The result of volume mesh generation without setting the control region

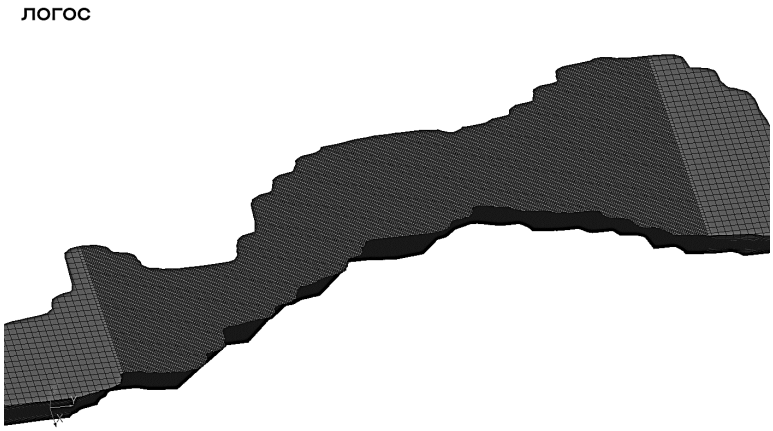

FIGURE 8. The result of volume mesh generation with the control region in the form of a rectangular parallelepiped

The next step is to trim the cells of the template mesh with surface mesh triangles. As a result of this stage, a polyhedron mesh is created, in which each edge in the cell separates only two faces, while the faces and cells can be convex or non-convex, and the cells are arbitrary polyhedrons. Regions of the model surface that do not contain the characteristic features of the model are approximated by simple-shape polyhedrons. Examples of such cells are shown in Figure 9.

Model surface regions containing characteristic features are approximated by complex-shape polyhedron cells to improve the accuracy of model approximation. Examples of such cells are shown in Figure 12.

The next step in mesh generation is improvement of the quality of the trimmed cells. Within the framework of the implemented specialized generator, cell quality

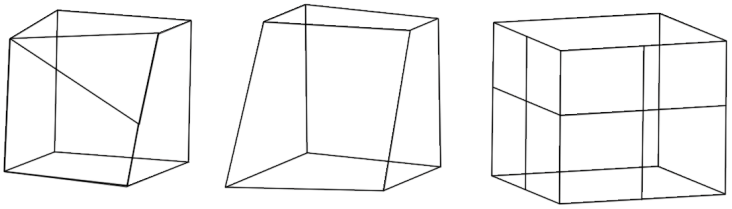

FIGURE 9. Examples of polyhedron cells in regions without characteristic features

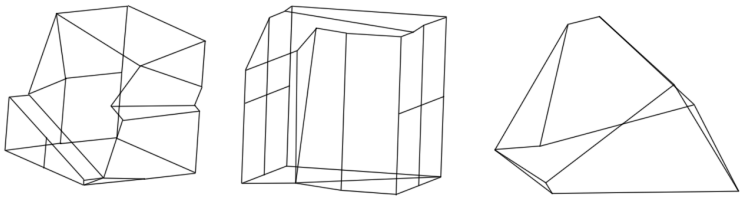

FIGURE 10. Examples of polyhedron cells in regions with characteristic features

improvement algorithms [23] were adapted to take into account the structure of the template mesh used to simulate tsunami waves.

4. An example of constructing a computational mesh

To demonstrate the performance of the mesh generator, a mesh model was created for one of the water areas of the Sea of Japan. A surface mesh was prepared in advance based on the information about the bathymetry of the simulated area and written to an stl-file.

The mesh generation parameters are shown in Figure 11. The mesh is partitioned regularly in two spatial directions and geometrically in the third one.

To refine cells in some part of the geometry, a control region in the form of a box is defined. Parameters and the general view of the region are shown in Figure **??**.

Figures 13 and 14 show the view of the input surface mesh and the resulting volume mesh, respectively, from the same angle (from the water surface side).

Figures 15 and 16 show the same view of the bottom topography on the input surface mesh and the resulting volume mesh, respectively. Figure 17 shows that part of cells were further refined during construction to obtain smaller-size cells. The area of refinement corresponds to the created control region.

Figure 15 shows that the surface mesh near the bottom surface is quite detailed. Using a standard trimming generator would result in the formation of a significant number of adaptive computational cells near the bottom surface, which is redundant for modeling tsunami waves.

Therefore, in the special mesh generator for tsunami wave simulations, the cell size in the template mesh is not adapted to that in the triangular mesh, and the cell size is determined only by the user settings. Figure 16 demonstrates that the mesh is not adapted to the size of the triangles. The area of refinement corresponds to the created control region.

Mesh generators

Parameter	Value
Cell growth rate from surface	Fast
Number of cells	1
Cell growth rate inside the ...	Fast
Number of cells	1
Split along axis X	
Law	Regular partitioning
Relative size	☐
Relative to base, %	200
Cell size	2000
Split along axis Y	
Law	Regular partitioning
Relative size	☐
Relative to base, %	200
Cell size	2000
Split along axis Z	
Law	Geometric progression
Relative size	☐
Relative to base, %	0,1
First cell size	1
Progression growth rate	1,2
Starting point of progr...	0

FIGURE 11. Template mesh parameters

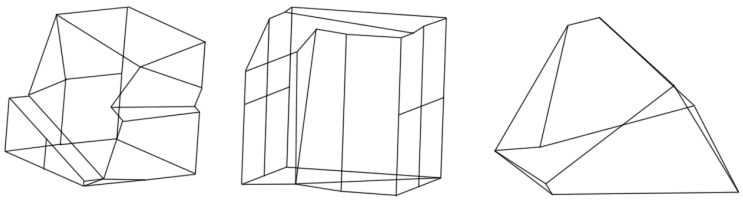

FIGURE 12. Parameters of the control region and its view

Figures 17 and 18 show fragments of the input surface mesh and the volume mesh in the control region, respectively.

To view the mesh structure in the third spatial direction, let us draw a section plane as shown in Figure 19. In the section plane in Figure 20, one can see that

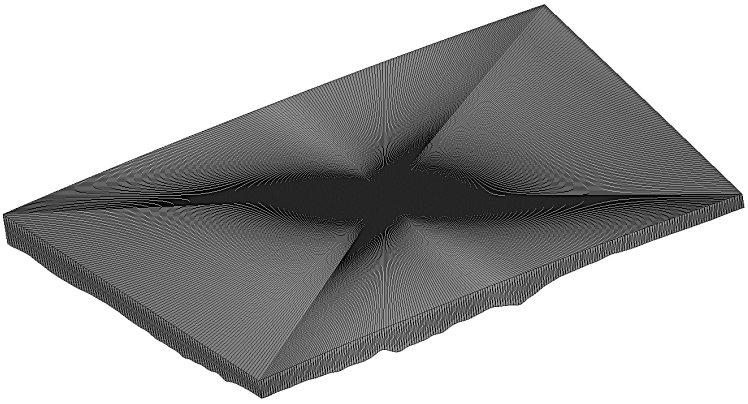

FIGURE 13. The input surface mesh created based on the bathymetry data (water surface view)

FIGURE 14. The resulting volume mesh with the control region in the form of a box (water surface view)

mesh lines are distributed non-uniformly getting condensed towards the water/air interface.

The generated mesh contained 3,3 million cells, the mesh generation time on 8-core processor CPU @ 3.00GHz 64 GB RAM was about 2 minutes with 1.5 GB memory consumption.

5. Using computational meshes for tsunami simulations

Computational meshes constructed by the generator presented in the paper have been used for numerical simulations of tsunami propagation by the Logos software package [8] based on historical data. Consider the case of tsunami propagation in Lituya Bay on the Alaska Peninsula (1958).

Figure 21 shows the landslide location in Lituya Bay [24], and the area of damage [25]. The numbers along the shore of the bay indicate the height above

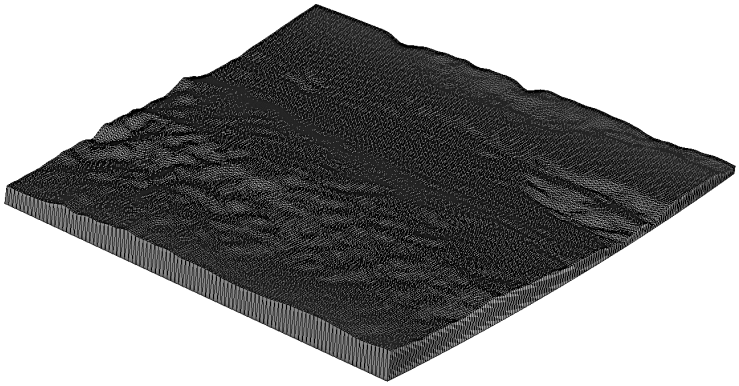

FIGURE 15. The input surface mesh created based on the bottom bathymetry data

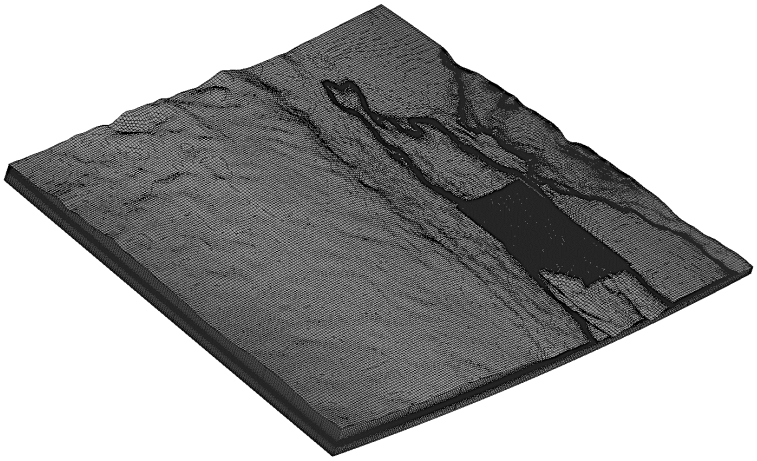

FIGURE 16. The resulting volume mesh with the control region in the form of a box resolving the bottom topology

sea level of the edge of the damaged land area and correspond to the height of the past wave.

To simulate the landslide and the resulting wave propagation, a three-dimensional mesh model was constructed, as shown in Figure 22. To simulate the wave propagation more accurately, the mesh was constructed using geometric progression with refinement towards the interface. To resolve the nature of the landslide movement and the run-up of the tsunami wave over the opposite shore in greater detail, an additional refinement block in the region of descent and run-up was constructed, which is marked with a frame in the figure.

The constructed mesh contained about 11,6 million cells, which is sufficient for numerical simulation of this problem. The mesh generation time on 8-core processor CPU @ 3.00GHz 64 GB RAM was about 8 minutes with memory consumption of 7.4 GB.

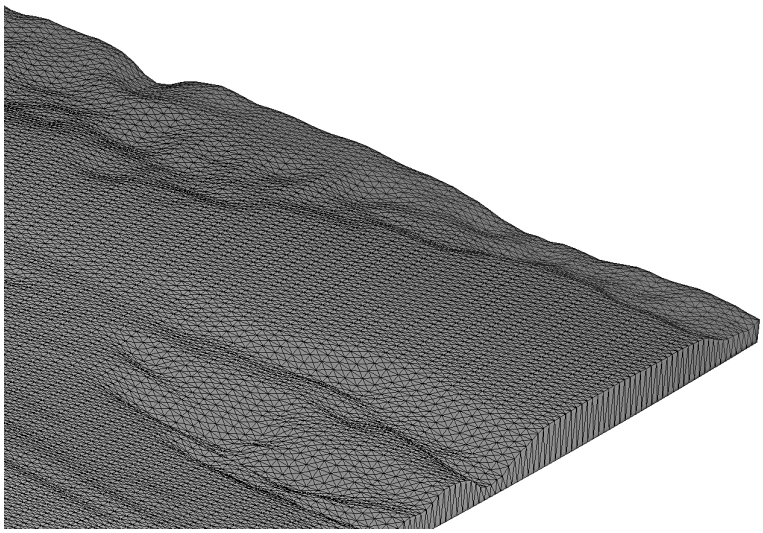

FIGURE 17. A fragment of the input surface mesh in the control region

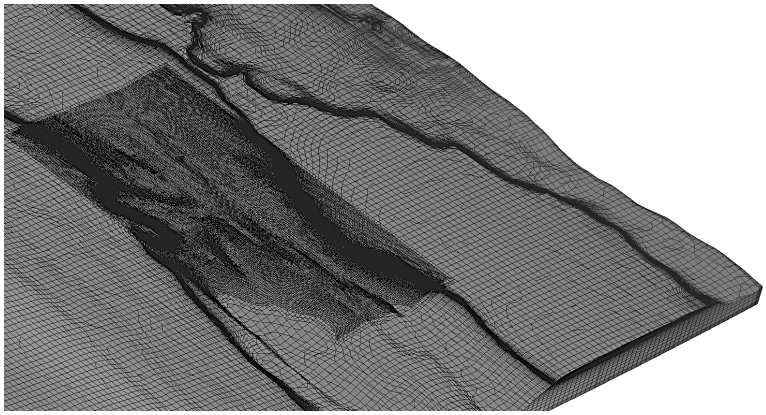

FIGURE 18. A fragment of the volume mesh in the control region

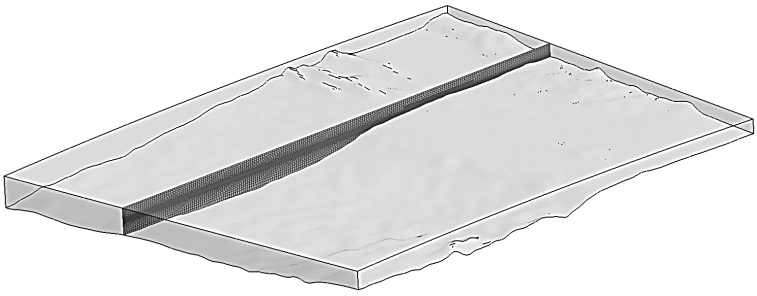

FIGURE 19. A general view of the section plane

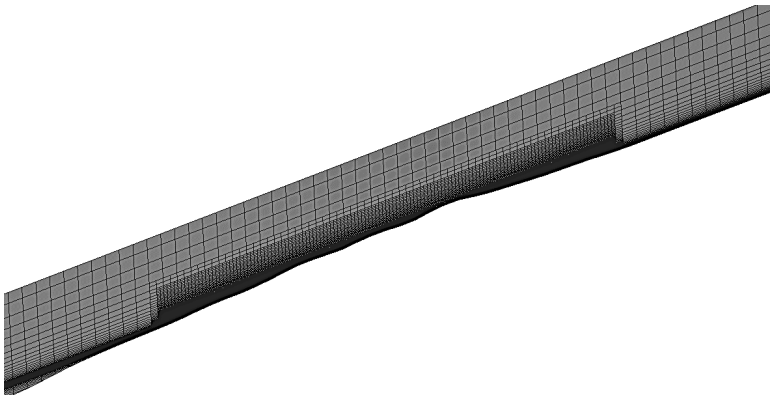

FIGURE 20. A section plane of the resulting mesh in the control region

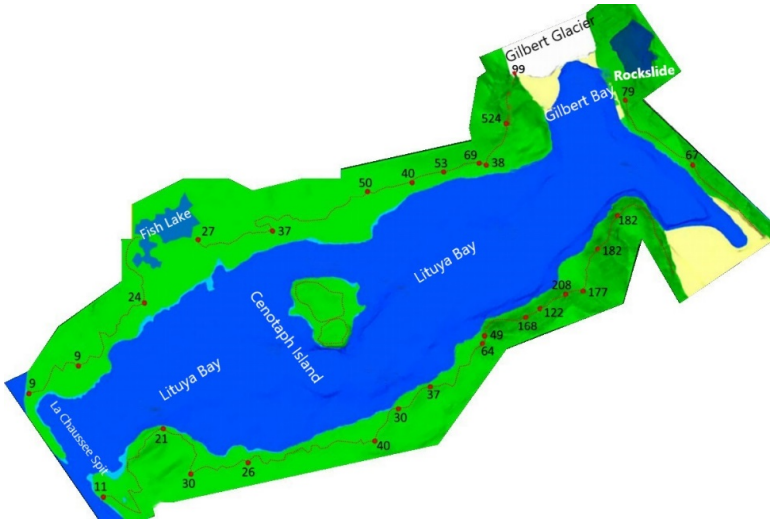

FIGURE 21. Lituya Bay and the area of damage

Due to the features of the problem, the most important parameter of the mesh to provide high-quality simulation is its cell size at the interface and in the direction of wave propagation [26], as well as the structure of mesh lines in the direction of increasing depth.

To perform a calculation, the mesh must be topologically correct, so the constructed mesh was checked for the presence of closed cells, free edges, degenerate cells, etc. As a result, none of these checks revealed bad cells.

Paper [8] presents comparisons of the results obtained by solving the Navier Stokes equations on the computational meshes constructed by the presented generator and observations.

Figure 23 illustrates the process of landsliding and tsunami formation. The figure shows that the landslide produces a huge splash of water up to 360 m high.

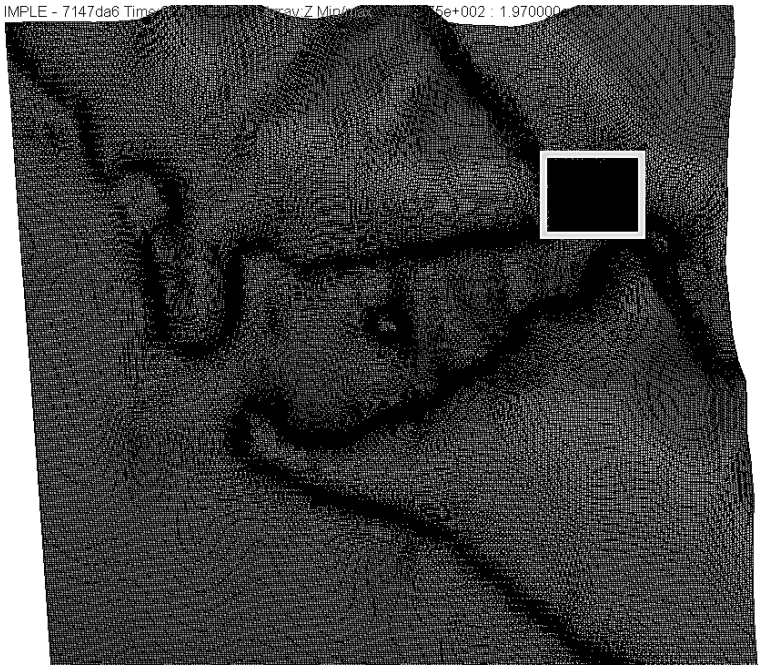

FIGURE 22. The mesh model for Lituya Bay

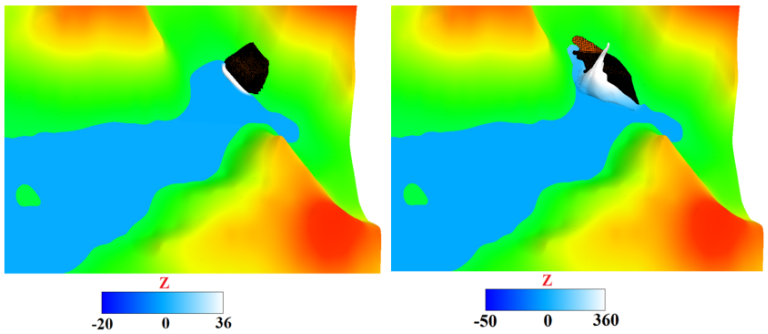

FIGURE 23. Wave formation by the landslide

Figure 23 shows the propagation of the tsunami wave at different time points across entire Lituya Bay. The figures demonstrate a huge tsunami wave that formed as a result of the landslide in Gilbert Inlet. The wave ran up onto the opposite shore of the bay and reached a height of 520 m (see Figure 24), which corresponds to real data, according to which the wave reached a height of 524 m. Then, the wave spread along entire Lituya Bay. The velocity of the tsunami in the bay was 160 km/h. The calculated velocity was about 140 km/h.

As can be seen from the figures, the wave rolled over the island of Cenotaph (Figure 19), the elevation of which is 50 m above sea level, which corresponds to field observations. According to [24, 25, 27] water in the bay practically calmed down as early as half

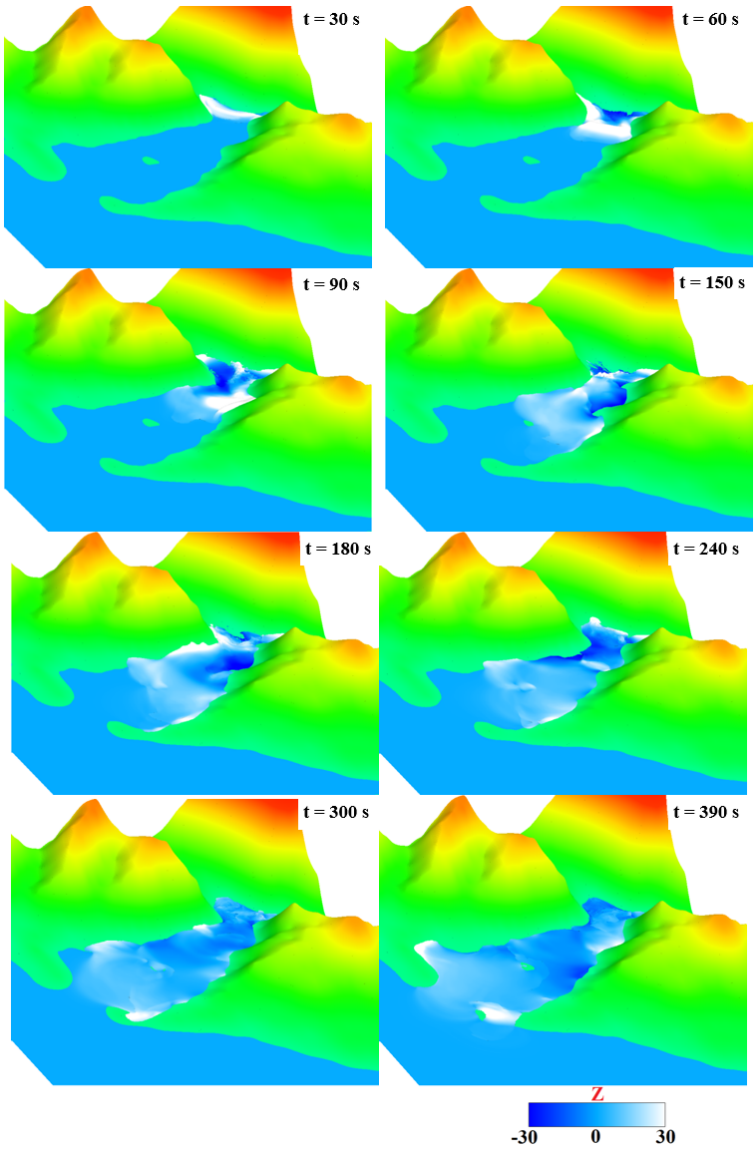

FIGURE 24. Tsunami propagation along Lituya Bay

an hour after the landslide occurred. Figure 25 shows a picture of the water surface at the time of 30 minutes, and there are practically no waves in the bay

Figure 26 shows the run-up line that shows how far the wave has moved inland. The numbers along the line indicate how high the wave was as it moved inland across the mountainous coast of the bay (the observed data are marked in brackets). The calculated heights are in good agreement with the observations.

The presented comparison of the results calculated within the framework of the Navier-Stokes equations with the observations showed that there is a fairly close

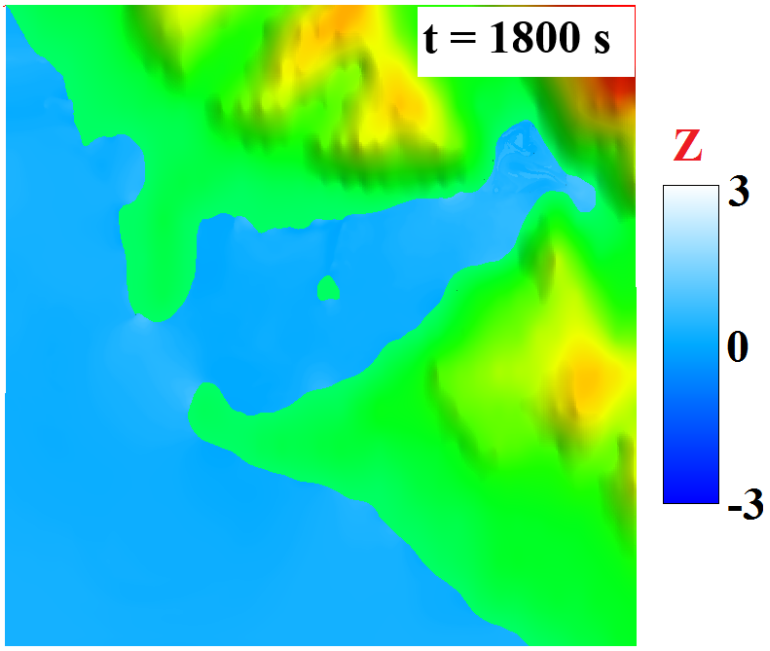

FIGURE 25. Water surface in the bay 30 minutes after the landslide

agreement between them in the maximum run-up both, in the tsunami velocity in the bay, and in the run-up heights along the entire bay.

The comparison showed that the numerical results obtained are in good agreement with the historical data.

To test the implemented mesh generation method in tsunami simulations, the cases of tsunami propagation on the Volga River near Nizhny Novgorod and underwater landslides in Kamchatka Bay of the Pacific Ocean near the coast of the Kamchatka Peninsula were also simulated [8].

6. Conclusion

A mesh generator was developed and implemented for constructing mesh models of water areas for tsunami simulations, a methodology for constructing the mesh models was developed, and the mesh generator was verified on test models of varying complexity and detail.

The implemented generator of computational meshes for tsunami simulations makes it possible to resolve water areas with a required level of detail depending on the features of the ongoing physical processes with coarsened or refined description of the model geometry in the regions where it is necessary. The mesh generator operates in the automatic mode, which significantly reduces labor and time costs for preparing/adjusting the model for calculations and allows preparing computational meshes for modeling tsunami waves more efficiently at a qualitatively new level.

The generator was used to construct mesh models, which were employed for numerical simulations by the Logos software package of tsunami propagation from

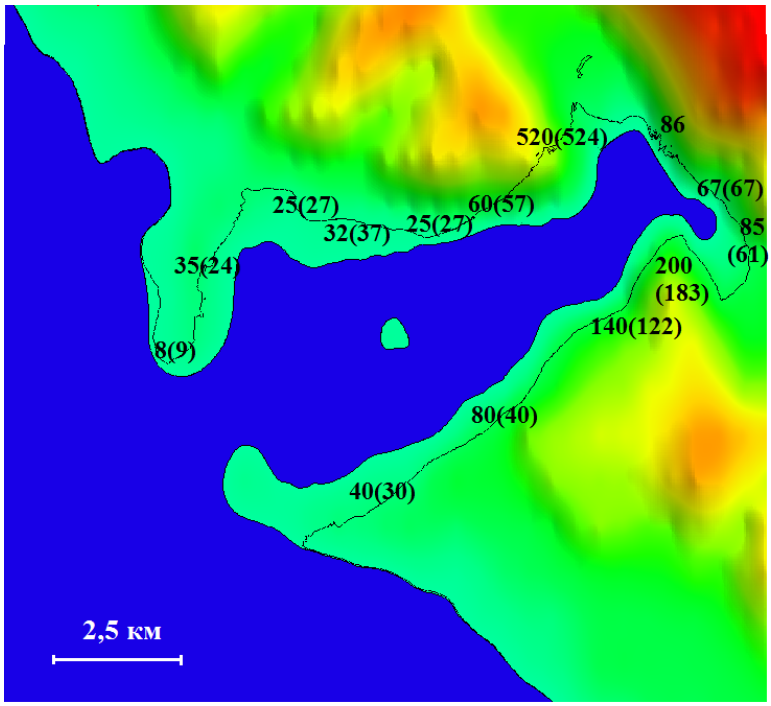

FIGURE 26. Heights of the wave run-up on the shore

an underwater landslide in the Kamchatka Bay of the Pacific Ocean near the coast of the Kamchatka Peninsula.

The presented method for constructing volume mesh models taking into account bathymetric data will make it possible to construct a model of any water area in the World Ocean in the most time-saving manner. Additionally, it offers the capability for interactive identification of tsunami source zones and wave run-up areas, enabling the assessment of wave characteristics directly in the wave formation region, as well as in the area of breaking and inland propagation.

References

[1] Goto C., Ogawa Y., Shuto N., Imamura N. *Numerical method of tsunami simulation with the leap-frog scheme (IUGG/IOC Time Project)*, IOC Manual, UNESCO, 1997, №35, 96p.

[2] Yalciner A.C., Ozer C., Karakus H., Ozyurt G., Pelinovsky E., Zaitsev A., Kurkin A. *Modeling and visualization of tsunamis: Mediterranean examples*, In book: Tsunami and Nonlinear Waves, 2007, p.273-283.

[3] Zahibo N., Pelinovsky E., Yalciner A., Kurkin A., Kozelkov A.S., Zaitsev A. *Modelling the 1867 Virgin Island Tsunami*, Natural Hazards and Earth System Sciences, 2003, v.3, № 5, p.367–376.

[4] Wei G., Kirby J.T., Grilli S.T., Subramanya R. *A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves*, J. Fluid Mech, 1995, v.294, p.71–92.

[5] Kirby J., Wei G., Chen Q., Kennedy A., and Dalrymple R. *Fully Nonlinear Boussinesq Wave Model Documentation and Users Manual*, Center for Applied Coastal Research Department of Civil Engineering University of Delaware, Newark DE 19716, Research Report №CACR-98-06, September 1998.

[6] Landau L.D., Lifshitz V.M. *Hydrodynamics*, M.: Nauka, 1988. (In Russian)

[7] Ferziger J.H., Peric M. *Computational Method for Fluid Dynamics*, Springer-Verlag, New York, 2002, 423p.

[8] Sarazov A.V., Kozelkov A.S., *Modification of the IDW method for numerical simulation of aerodynamics problems on large grids*, Siberian Electronic Mathematical Reports, 2024, v.21, №2, p. 591-620, https://doi.org/10.33048/semi.2024.21.042/

[9] Kozelkov A.S., Tyatyushkina E.S., Kurulin V.V. and Kurkin A.A. *Influence of Turbulence Effects on the Runup of Tsunami Waves on the Shore within the Framework of the Navier–Stokes Equations*, Fluids, 2022, 7, 117, https://doi.org/10.3390/fluids7030117.

[10] Kozelkov A.S. *The Numerical Technique for the Landslide Tsunami Simulations Based on Navier-Stokes Equations*, Journal of Applied Mechanics and Technical Physics, 2017, v.58, №7, pp.1192-1210

[11] Kozelkov A., Tyatyushkina E., Kurkin A., Kurulin V., Kurkina O., Kochetkova O., *Fluid flow simulation in a T-connection of square pipes using modern approaches to turbulence modeling*, Siberian Electronic Mathematical Reports, 2023, v.20, №1, p. 25-46, http://semr.math.nsc.ru

[12] Kozelkov A.S., Kurkin A.A., Kurulin V.V., Pelinovsky E.N., Plygunova K.S., Tyatyushkina E.S., Utkin D.A. *Verification of the LOGOS Software Package for Tsunami Simulations*, Geosciences 2020, 10, 385; doi:10.3390/geosciences10100385

[13] Kozelkov A.S., Kurkin A.A., Pelinovsky E.N., Tyatyushkina E.S., Kurulin V.V., Tarasova N.V. *Landslide-type tsunami modelling based on the Navier-Stokes Equations*, Science of tsunami Hazards, Journal of Tsunami Society International, 2016, vol. 35, №3, p.106-144

[14] Kozelkov A., Kurkin A., Utkin D., Tyatyushkina E., Kurulin V., Strelets D. *Application of Non-Reflective Boundary Conditions in Three-Dimensional Numerical Simulations of Free-Surface Flow Problems*, Geosciences 2022, 12, 427, https://doi.org/10.3390/geosciences12110427

[15] Horrillo J., Wood A., Kim G.B., Parambath A. *A simplified 3-D Navier-Stokes numerical model for landslide-tsunami: Application to the Gulf of Mexico*, Journal of Geophysical Research: Oceans, 2013, v.118, p.6934–6950.

[16] Kozelkov A.S., Kurkin A.A., Pelinovsky E.N., Kurulin V.V., Tyatyushkina E.S. *Numerical modeling of the 2013 meteorite entry in Lake Chebarkul, Russia*, Nat. Hazards Earth Syst. Sci., 2017, v. 17, p. 671–683

[17] Schneider, R. *A grid-based algorithm for the generation of hexahedral element meshes*, Engineering with computer. 1996. V.12, No.3-4, pp 168-177.

[18] Borisenko O.N., Cherenkova M.V., Chukhmanov N.V., Giniyatullina A.G., Kuzmenko M.V., Popova N.V., Potekhina E.V., Turusov M.R., Smolkina D.N. *An automatic generator of unstructured polyhedron meshes in the preprocessor of the LOGOS software package*, Voprosy Atomnoy Nauki i Tekhniki. Ser. Mathematical Modeling of Physical Processes. 2018. Sarov. Issue 2. P.25-39. (In Russian)

[19] Tomac M., Eller D. *Towards automated hybrid-prismatic mesh generation*, Procedia Engineering. 2014. V.82. pp 377-389.

[20] Garimella R. V. *Polyhedral mesh generation and optimization for non-manifold domains*, Proc. 22nd Int. Meshing Roundtable. Springer International Publishing. 2014. pp.313-330.
[21] J.F. Thompson, B.K. Soni, N.P. Weatherill (Eds.) *Handbook of Grid Generation*, CRC Press, Boca Raton, FL, 1999.
[22] Jiao X., Heath M.T. *Feature detection for surface meshes*, Proc. 8th Int.Conf. on Numerical Grid Generation in Computational Field Simulations. 2002. pp. 705–714.
[23] Blazhnova K.A., Borisenko O.N., Cherenkova M.V., Chukhmanov N.V., Giniyatullina A.G., Smolkina D.N., Timaeva T.E. *Improving the quality of volume cell faces when generating unstructured meshes in the Logos software package*, Voprosy Atomnoy Nauki i Tekhniki. Ser. Mathematical Modeling of Physical Processes. 2022. Sarov. Issue 3. P.73-85. (In Russian)
[24] Fritz, H.M., Mohammed, F., Yoo, J. *Lituya Bay landslide impact generated mega-tsunami 50th Anniversary*, Pure Appl. Geophys., 166, 153–175, https://doi:10.1007/s00024-008-0435-4, 2009.
[25] Aufleger M., Franco A., Gems B., Moernaut J., Schneider-Muntau B., Strasser M. *Lituya Bay 1958 Tsunami – detailed pre-event bathymetry reconstruction and 3D-numerical modelling utilizing the CFD software Flow-3D*, Natural Hazards and Earth System Science. https://doi.org/10.5194/nhess-2019-285, 2019.
[26] Efremov V.R., Kozelkov A.S., Kurkin A.A., Kurulin V.V., Tyatyushkina E.S., Utkin D.A. *Evaluation of numerical diffusion of the finite volume method in surface wave modeling*, Computational Technologies. 2019. V. 24. № 1. P. 106-119. (In Russian)
[27] Arcas D., Asunci?n M., Castro M.J., Gonz?lez-Vida J.M., Mac?as J., Ortega-Acosta S., S?nchez-Linares C. *The Lituya Bay landslide-generated mega-tsunami – numerical simulation and sensitivity analysis*, Nat. Hazards Earth Syst. Sci., 2019, v.19, p. 369-388, https://doi.org/10.5194/nhess-19-369-2019.

Nikolay Vasil'evich Chukhmanov
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: NVChukhmanov@vniief.ru

Ol'ga Nikolaevna Borisenko
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: ONBorisenko@vniief.ru

Andrey Sergeevich Kozelkov
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia,
Federal State-Funded Higher Education Institution "Nizhny Novgorod State Technical University n.a. R.E. Alexeyev",
Nizhny Novgorod, Russia,
Moscow Aviation Institute (National Research Univ.),
Moscow, Russia
Email address: ASKozelkov@vniief.ru

Dina Nikolaevna Smolkina
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: DNSmolkina@vniief.ru

Marina Vital'evna Cherenkova
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: MVCherenkova@vniief.ru

Milana Vladimirovna Kuz'menko
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: MVKuzmenko@vniief.ru

Anastasiya Gennad'evna Giniyatullina
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: AGGiniatullina@vniief.ru

Tat'yana Evgen'evna Timaeva
Federal State Unitary Enterprise "Russian Federal Nuclear Center - All-Russian Research Institute of Experimental Physics" (FSUE "RFNC-VNIIEF"),
Sarov, Russia
Email address: TETimaeva@vniief.ru