

Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid bodies movement

S.A. Soukov¹

Keldysh Institute of Applied Mathematics Russian Academy of Sciences

¹ ORCID: 0000-0002-0667-6955, ssoukov@gmail.com

Abstract

The article deals with the problem of initializing the field of the signed distance function to the surface of a moving solid body of arbitrary shape. A combined algorithm is proposed for fast calculation of approximate values of a function with a controlled loss of accuracy. The idea is to interpolate the function over the cells of an adaptive grid with local switching to find the distance to surface triangulation. This algorithm can be used both for visualizing the motion of surfaces and for solving various geometric problems arising in the process of numerical modeling of physical processes. The error in determining the function does not depend on the shape of the body and the features of the movement trajectory. The paper contains a description of an algorithm for generating an interpolation grid taking into account a given computational error and an algorithm for calculating the signed distance to triangulation using a binary search tree. Using the examples of processing a spherical surface and a cruise missile model, the possibility of using a combined approach for visualizing the motion of solid bodies and in numerical calculations of gas-dynamic flows is demonstrated.

Keywords: signed distance field, data visualization, level-set method, surface triangulation.

1. Introduction

In solving a wide range of scientific problems related to geometric modeling, computer animation, image processing, simulation of physical processes and visualization of the results of numerical experiments, the signed distance function is used to describe the shape of bodies [1]. The body geometry Ω with a boundary $\partial\Omega$ (fig. 1) is specified by a function $\phi_{\partial\Omega}(\mathbf{x})$, that takes positive values in the external subdomain Ω^+ and negative values in the subdomain Ω^- inside the body. The zero isosurface $\phi_{\partial\Omega}$ corresponds to the body boundary.

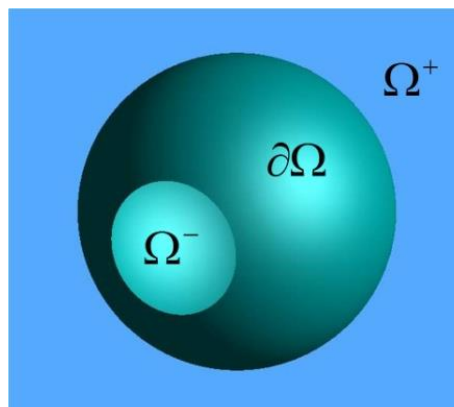


Fig. 1. Determination of the signed distance function for the sphere surface.

The absolute value of the function is equal to the shortest distance to the body boundary:

$$\phi_{\partial\Omega}(\mathbf{x}) = \begin{cases} \inf_{\mathbf{x}_B \in \partial\Omega} \|\mathbf{x} - \mathbf{x}_B\|, & \mathbf{x} \in \Omega^+ \\ 0, & \mathbf{x} \in \partial\Omega \\ -\inf_{\mathbf{x}_B \in \partial\Omega} \|\mathbf{x} - \mathbf{x}_B\|, & \mathbf{x} \in \Omega^- \end{cases}.$$

In the problems of numerical modeling of liquid and gas flows $\phi_{\partial\Omega}$, it is usually used to reconstruct the boundaries of obstacles moving in the flow on the elements of the computational grid [2] and adapt the grid to the current position $\partial\Omega$. In the latter case, the unit gradient vector becomes an additional adaptation criterion $\nabla\phi_{\partial\Omega}$. The values $\phi_{\partial\Omega}$ can also be the coefficients of the equations of the mathematical model [f.e., 3]. The visualization of the results of computational experiments here includes the display of the body surface moving along the calculated trajectory.

In terms of requirements for the accuracy of the determination $\phi_{\partial\Omega}$, the area of numerical modeling is conventionally divided into three zones (fig. 2). The minimum error should be ensured near the body surface, where it affects the accuracy of reconstruction $\partial\Omega$ (zone A) and is equivalent to the error in determining the coefficients of the equations of the mathematical model (zone B). In zone C, the values $\phi_{\partial\Omega}$ and components of the vector $\nabla\phi_{\partial\Omega}$ are used only as criteria for grid adaptation, which makes it possible to increase the level of acceptable computational error.

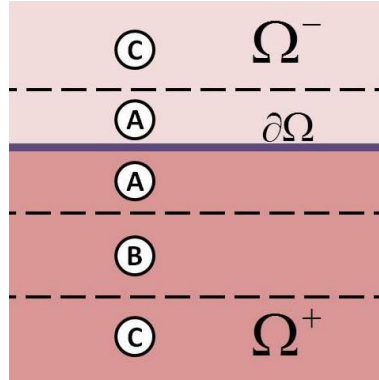


Fig. 2. Allocation of zones in the area of solving the problem.

Methods [4-6] based on the numerical solution of the transfer equation are usually used to simulate field changes $\phi_{\partial\Omega}$ during displacement Ω

$$\frac{\partial\phi_{\partial\Omega}}{\partial t} + \mathbf{u}\nabla\phi_{\partial\Omega} = 0. \quad (1)$$

The body's trajectory is set by the instantaneous velocity. $\mathbf{u}(\mathbf{x}, t)$. The approach is relatively easy to generalize to the case of deformation $\partial\Omega$, but has a number of significant drawbacks from the point of view of modeling the motion of solid bodies. So, even with numerical integration (1) using high-precision methods, the accumulation of errors leads to a gradual distortion of the boundary. An additional obstacle for the practical application of software implementations of the corresponding algorithms can be their high resource intensity. In addition, the values of the required function are determined at the nodes or centers of the cells of the computational meshes. Therefore, the search $\phi_{\partial\Omega}(\mathbf{x})$ at an arbitrary point is associated with interpolation, which leads to a loss of precision.

Theoretically, analytical algorithms should provide the maximum calculation speed $\phi_{\partial\Omega}$ with a minimum error. However, for bodies whose surface consists of many curved faces, the derivation of an explicit functional relationship between distance and coordinate is a serious problem. In this case, a discrete body model is considered. The surface $\partial\Omega$ is approximated by a set of polygons. The time it takes to calculate the distance to a discrete model depends on its dimension (the total number of polygons) and the efficiency of the data ordering method used.

In computer graphics, the signed distance function is considered as an implicit representation of solid and deformable surfaces in algorithms for constructing voxel and polygonal object models [7], as well as in algorithms for generating images using the ray tracing method [8]. In this case, the function field is set on regular or adaptive grids, where exact values $\phi_{\partial\Omega}$ are stored at the vertices, and approximate values at arbitrary points are calculated by interpolation over cells [9-11]. This algorithm is characterized by a time-fixed error and high performance, but in its pure form it is not applicable in the calculations of physical processes. The maximum error in determining the signed distance in zones A and B is equal to several percent of the step of the computational mesh. Therefore, the number of cells and the amount of data describing the structure of the interpolation grid can go beyond the limitations of the size of the geometry representation.

In this paper, a combined approach to computation of $\phi_{\partial\Omega}$ is presented, which combines interpolation and finding the distance to a discrete surface model. The fundamental possibility of its use in numerical calculations is shown in [12] by the example of modeling two-dimensional flows. Details of the implementation of the combined approach in three-dimensional case are discussed below. A description of an algorithm for generating adaptive interpolation grids and an algorithm for calculating the signed distance to triangulation is given. Examples and parameters of grids for initialization of the signed distance field in problems of visualization of surfaces and modeling of gas-dynamic flow around solids are presented.

2. Signed distance to triangulation

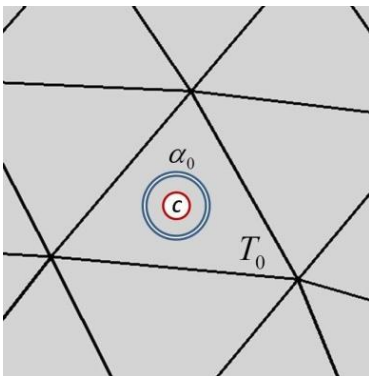
Spatial triangulation is one of the main approaches for body boundaries discretization. The surface $\partial\Omega$ is approximated by triangulation T , which consists of N_T oriented triangles:

$$\partial\Omega = T = \bigcup_{i=1}^{N_T} T_i.$$

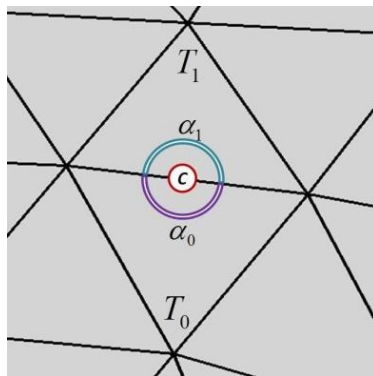
Triangle orientation condition means compliance with the order of listing its vertices (e.g., counterclockwise from the side Ω^+) that can uniquely determine the direction of the unit outward normal \mathbf{n}_{T_i} to the plane of the triangle. In turn, at any point $\mathbf{c} \in T$, the vector of the angle-weighted unit pseudonormal can be found

$$\mathbf{N}_{\mathbf{c}} = \frac{\sum_{\mathbf{c} \in T_i} \alpha_i \mathbf{n}_{T_i}}{\left\| \sum_{\mathbf{c} \in T_i} \alpha_i \mathbf{n}_{T_i} \right\|}.$$

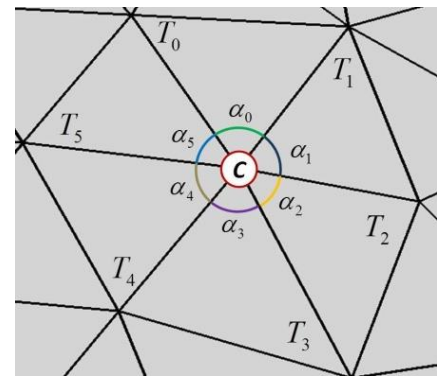
The direction of the pseudo-normal is calculated as the sum of the incident normals with respect to the triangles \mathbf{c} , taken with angle coefficients [13]. For \mathbf{c} , located strictly inside the triangle (fig. 4a), the coefficient takes on a value 2π , and the pseudonormal coincides with the normal to the plane of the triangle. The angle coefficient at the points located on the edges of the triangulation are equal π (fig. 4b). In this case, $\mathbf{N}_{\mathbf{c}}$ is codirectional with the vector of the sum of the normals of two triangles adjoining along the edge. If \mathbf{c} coincides with the triangulation vertex (fig. 4c), then the coefficients are equal to the angles at the vertex.



a) point inside the triangle



b) point on the edge



c) the vertex of the triangulation

Fig. 3. Determination of angle coefficients.

The signed distance from point \mathbf{x} to triangulation T is calculated by the formula

$$\phi_T(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}\| \text{sign}((\mathbf{x} - \mathbf{c}) \cdot \mathbf{N}_c),$$

where \mathbf{c} is the point closest to \mathbf{x} on the triangulation surface. The direction of the gradient vector is given as

$$\nabla \phi_T(\mathbf{x}) = \begin{cases} \mathbf{N}_c, & \phi_T(\mathbf{x}) = 0 \\ \frac{\mathbf{x} - \mathbf{c}}{\|\mathbf{x} - \mathbf{c}\|}, & \phi_T(\mathbf{x}) > 0 \\ \frac{\mathbf{c} - \mathbf{x}}{\|\mathbf{c} - \mathbf{x}\|}, & \phi_T(\mathbf{x}) < 0 \end{cases}.$$

The basic algorithm for determining $\phi_T(\mathbf{x})$ involves finding the minimum in absolute value in the process of sequentially calculating the distances $\phi_{T_i}(\mathbf{x})$ to each of the triangles. In optimized algorithms, exhaustive search of triangles is eliminated by preliminary sorting the data according to the geometric principle. Ordering allows you to constrain the traversal to the triangles in the nearest neighborhood \mathbf{x} .

Sorting objects based on k-d trees [14], which are a kind of binary search tree, is efficient for problems of finding the minimum distances in space. The principle of constructing a k-d tree consists of a recursive bisection of the solution space by planes perpendicular to the coordinate axes. The triangulation T is immersed in a box whose edges are parallel to the coordinate axes. The bounding boxes is taken as the construction area of the k-d tree and becomes its root. The orientation and position of the partition planes of the tree nodes are chosen so that the descendants appearing as a result of decomposition contain an equal number of triangulation vertices. Recursive bisection of nodes continues until the specified number of vertices is reached or is interrupted by the constraint on the maximum tree depth. At the end of the procedure, lists of triangles belonging to them are made for the leaves. A triangle refers to a tree node when it is partially or completely inside the box associated with that node. Triangulation elements intersecting with the faces of boxes of several nodes are included in the lists of each of them. Fig. 4 shows an example of uniform triangulation of the surface of ballistic model HB-2 (fig. 4a) and the partition of its triangles at the level of the leaf nodes of the k-d tree (fig. 4b). The geometric model of the body is taken from [15]. The triangles at the intersection with the borders of the leaves of the tree are filled in black in fig. 4b.

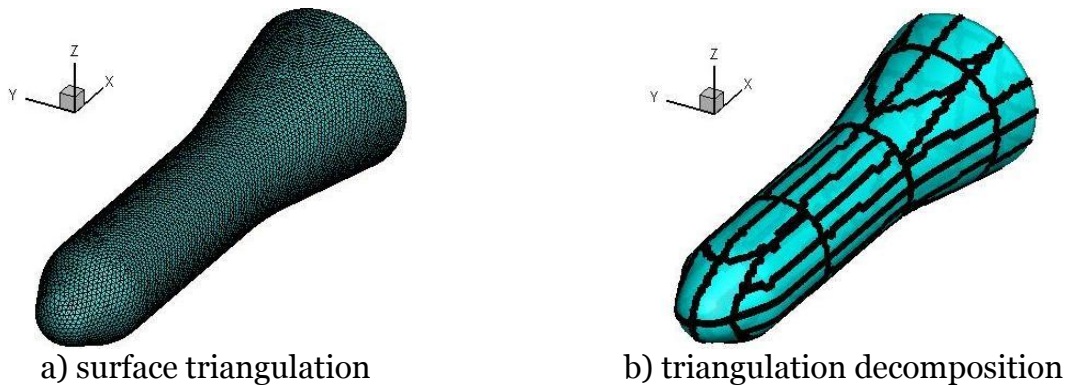
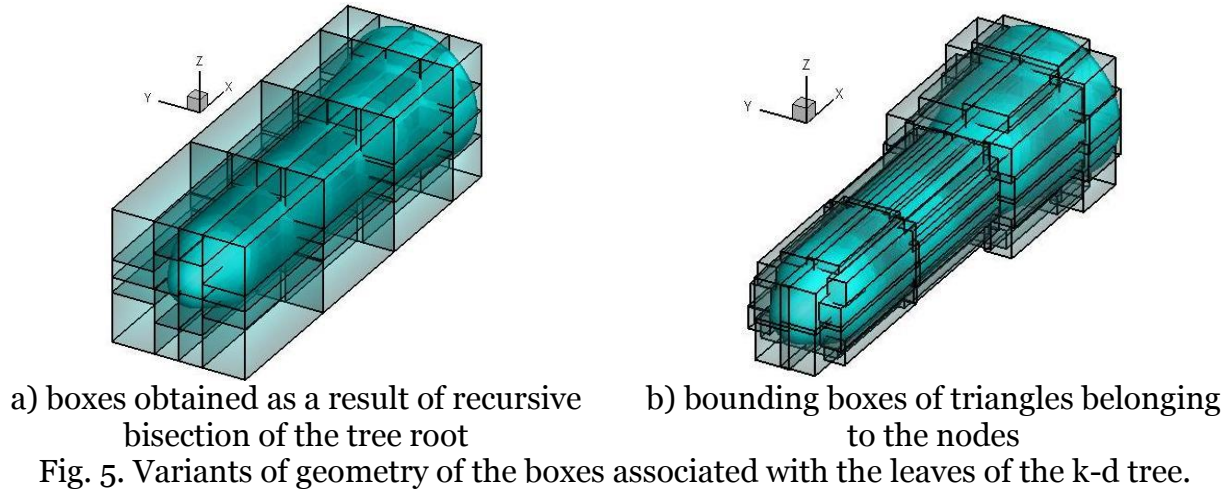


Fig. 4. Ballistic model HB-2.

The search procedure $\phi_T(\mathbf{x})$ implements descent along a k-d tree by recursively calling the node processing function [16]. The order and necessity of traversing the parent's child nodes are set depending on the current status of the problem solution. Leaf node processing consists of sequential calculation of the distances $\phi_{T_i}(\mathbf{x})$ to the triangles associated with it. The starting goal of the search is to determine the tree closest to the leaf \mathbf{x} . The minimum in absolute value of the signed distances to its triangles becomes the radius of the local search area centered on the point \mathbf{x} . Further, when descending the tree, only those branches and leaves that intersect with the local search area are considered. If the distance between the point and

the next processed triangle is less than the radius of the area, then its size is correspondingly reduced. It should be noted that, in the general case, the box associated with the node k-d of the tree coincides with the minimum bounding box of the set of triangles belonging to the node only at the root of the tree. The bounding boxes of triangles intersect due to the presence of common elements, but, as a rule, they have a smaller volume (fig. 5). Therefore, from the standpoint of reducing calculations more efficient is the intersection checking unit and the local region search using triangles bounding boxes.



If we assume that the resource intensity of the determination procedure $\phi_T(x)$ is proportional to the number of processed triangles N_{check} , then the acceleration from using a binary search tree can be estimated by the value of the coefficient $S_{kd} = N_T/N_{check}$. The number of triangles N_{check} depends on several factors: the shape of the triangulated surface, the method of decomposition of the nodes of the k-d tree, and the specific location of the point x in relation to T . The minimum amount of computation in the search for $\phi_T(x)$ corresponds to the variant of descending the tree with testing triangles only containing the point x leaf node. The probability of the location of a point and the nearest triangle inside one leaf increases for x located near T . Therefore, it can be assumed that in the vicinity of triangulation, that is, in areas where high accuracy of calculation $\phi_{\partial\Omega}$ is important, the acceleration coefficient S_{kd} will also increase.

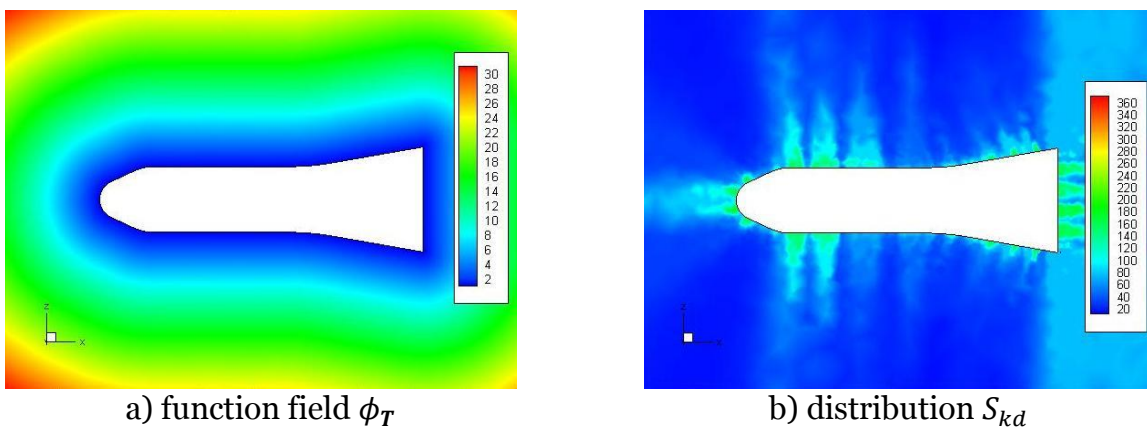


Fig. 6 shows the distribution of values of the function ϕ_T and the acceleration coefficient S_{kd} on the outer side of the surface of the ballistic model HB-2. In this case, there is no unambiguous relationship between the value of the coefficient and the distance to the surface. However, in the illustrations, you can see that the value S_{kd} decreases as you move from the boundary along the rays, the origin at points $c \in T$ and the direction coincides with the vector N_c .

3. Combined signed distance algorithm

The combined algorithm implements an accelerated procedure for determining the approximate values of the signed distance function with a controlled loss of accuracy. The solution area is filled with an adaptive mesh G consisting of cubic cells. The grid nodes store the values of the signed distance function to the triangulation ϕ_T and the components of the gradient vector $\nabla\phi_T$. The grid cells are divided into two types: interpolation (subdomain G_I) and model (subdomain G_T).

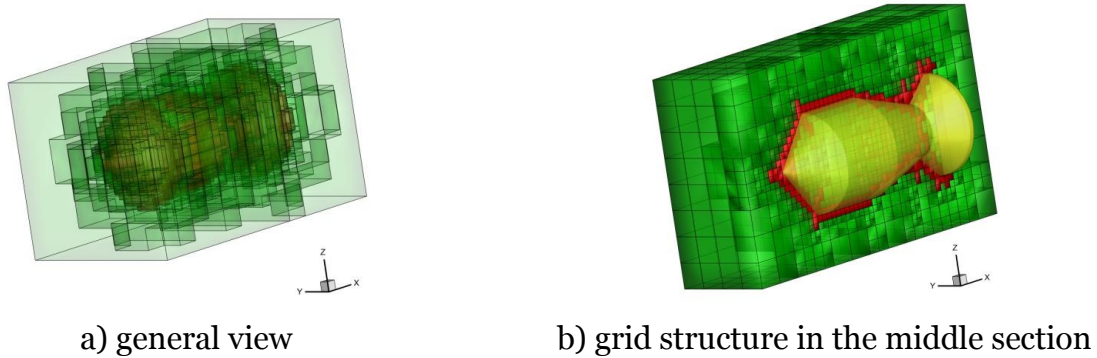


Fig. 7. An example of constructing a grid for a bullet.

Fig. 7 shows an example of an adaptive grid for computation with respect to the axisymmetric surface of the bullet. The body boundary consists of cylindrical and conical surfaces with a diameter of 8 to 15.5 mm. Bullet length - 34 mm. Red color in fig. 7b shows a subdomain of model cells.

The way to calculate the values of the required function at a point \mathbf{x} depends on the type of a cell:

$$\phi_G(\mathbf{x}) = \begin{cases} P(\mathbf{x}), & \mathbf{x} \in G_I \\ \phi_T(\mathbf{x}), & \mathbf{x} \in G_T \end{cases}.$$

Within the interpolation cell, the signed distance function is replaced by a polynomial

$$P(\mathbf{x}) = C_0xyz + C_1xy + C_2xz + C_3yz + C_4x + C_5y + C_6z + C_7.$$

Coefficients $C_{k=0,1, \dots, 7}$ are determined from the values $\phi_T(\mathbf{x})$ at the cell nodes (trilinear interpolation). A similar interpolation approach is used to reconstruct the gradient vector components. Inside the model cells, the signed distance to the surface triangulation is explicitly calculated.

The generation mechanism G consists in hierarchical isotropic refinement of cubic grid cells (step h_0). That is, the grid has an octree topology with two types of leaf cells. The structure G is limited to the maximum grinding depth at the level l_d (cell edge $h_{l_d} = h_0/2^{l_d}$). Decomposition of cells in the process of constructing the grid extends to compliance with predetermined interpolation error conditions

$$|\phi_G(\mathbf{x}) - \phi_T(\mathbf{x})| \leq R_\phi(\phi_T(\mathbf{x})). \quad (2)$$

The function $R_\phi(\phi_T(\mathbf{x})) > 0$ describes the relationship between the acceptable computational error and the signed distance to triangulation. A level cell l_d that does not satisfy condition (2) belongs to a subdomain G_T . In addition, tree branches become model cells if, after their recursive splitting to the level l_d , no descendant of the interpolation type appears.

The estimation of the interpolation accuracy is realized by comparing the values ϕ_G and ϕ_T at the nodes of the control grid (step $h_t = h_0/2^{l_t}$, $l_t > l_d$). Thus, checking the fulfillment of condition (2) in the level l_m of cell includes comparing the exact and approximate values of the function at $(2^{l_t-l_m} + 1)^3 - 8$ points. The set of points of the control grid can be supplemented with a list of coordinates of the vertices located on the surface and in the immediate vicinity of the triangulation.

An interpolation grid is generated in a body-related coordinate system. The choice of its origin and directions of the axes takes into account the shape of the body. For example, for an aircraft, the origin of the associated coordinate system is placed at the center of mass, and the longitudinal, vertical, and lateral axes are determined by the aircraft design. The current position of the body is set by the coordinates of the center and the directions of the axes of the associated coordinate system in the coordinate system of the computational domain (fig. 8).

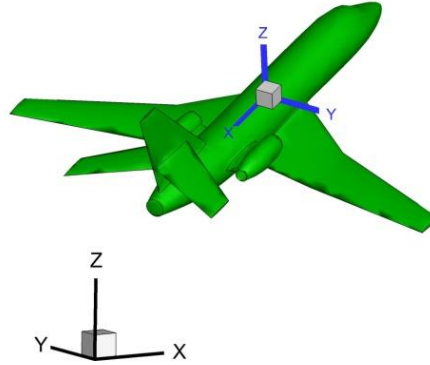


Fig. 8. Positioning the body in the computational domain.

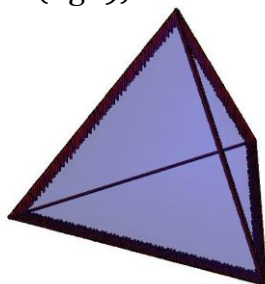
The axes of the linked coordinate system are displayed in blue, and the axes of the computational domain coordinate system are displayed in black.

Finding a signed distance to the surface of a moving object involves transforming coordinates, calculating a distance, and reverse rotation of the gradient vector. Geometrically, the situation can be interpreted so that Ω moves along the computational domain together with G .

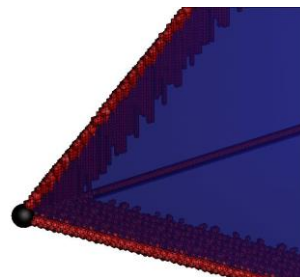
From the point of view of performance, the bottleneck of the combined algorithm is the local transition from interpolation to explicit search ϕ_T within the subdomain of model cells. An indirect criterion for assessing the size G_T and efficiency of using the combined grid can be the ratio of the volume of the subdomain to the triangulation area

$$h_{G_T} = V_{G_T}/S_T.$$

Assuming that the model type cells are grouped near the surface of the body, the coefficient h_{G_T} should correspond to the thickness of the transition zone to work with triangulation. The actual volume V_{G_T} and distribution pattern of the model cells depend on the shape of the surface under consideration, the triangulation step, orientation $\partial\Omega$ relative to the edges G , the basic step h_0 , the refinement depth l_d , and the type of function R_ϕ . With an increase in the decomposition depth, the volumes of prisms obtained by stretching the surface triangles along the perpendiculars to their planes are gradually excluded from G_T . Model cells are grouped mainly at the vertices and edges of T , as well as near the bisector planes of acute dihedral angles (fig. 9).



a) general view



b) the pattern of the distribution of cells near the vertex

Figure: 9. An example of the distribution of model grid cells for calculation ϕ_T relative to the box faces.

Minimization V_{G_T} due to cell refinement leads to an undesirable increase in dimension G , that is, the total number of cells in the grid. This problem is partially solved by using multiple grids to calculate approximate values ϕ_G . Generation of a separate grid G_i in the vicinity of each of the surface features makes it possible to locally maximize the step h_{l_d} . For a set of nested grids, the order of their initialization and traversal is specified. Subdomains of intersection with previously constructed grids are excluded from the generation region of each subsequent grid. The cells in them are assigned a model type without testing the interpolation accuracy. A similar order of walking is observed when searching for a distance to the surface. Fig. 10 illustrates an example of the layout of a quad of nested grids for determining distance relative to the surface of an aircraft. Grids G_1 and G_2 with high-density cells surround the engines. The grid G_3 contains the object and its immediate vicinity. The last grid G_4 with the maximum cell size is used to interpolate the distance in the far-field region.

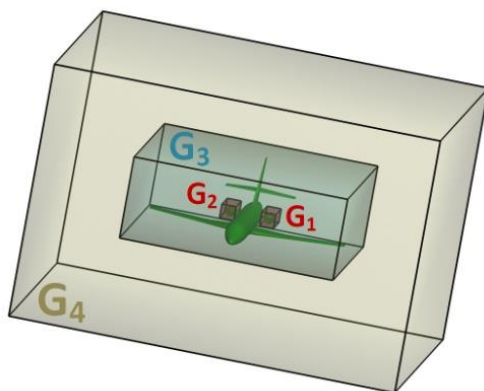


Fig. 10. An example of the layout of a set of nested grids.

4. Computational experiment

The combined approach to calculating the signed distance field is implemented as a library of subroutines for systems with multicore processors (CPUs). The library includes modules for constructing a k-d tree, calculating the signed distance to triangulation, a module for generating and visualizing interpolation grids, as well as directly implementing the combined algorithm. The procedure for determining the approximate values of the signed distance function works with its own data structures, the initialization of which is carried out in a preliminary stage. Thus, the software implementation of the combined algorithm without any modifications is added to the codes of both sequential and parallel programs for the numerical simulation of physical processes and data visualization. In MPI applications, all processes of the group read the topology of the interpolation grid and the description of the triangulation, after which the problem is solved in a sequential mode. To parallelize a loop on a multicore architecture with a search for distances to the surface at a given set of points, it is enough to add the corresponding OpenMP directive to the program code.

Below is a description of two examples of generating interpolation grids. The first grid is built around a simplified cruise missile model to determine the signed distance function field in numerical calculations. The second grid contains a sphere of unit diameter and is used to render the surface on the elements of the mixed mesh.

The appearance of the cruise missile model is shown in fig. 11.

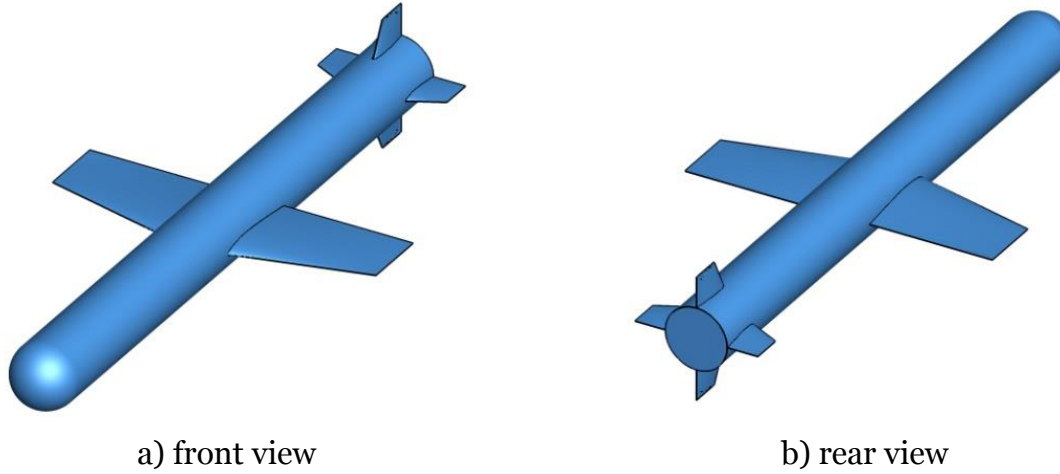


Fig. 11. Surface model.

The geometric parameters of the object and surface triangulation are given in tab. 1.

Table 1

Parameter	Value
Number of vertices	160322
Number of triangles	322154
Bounding box	10 x 5 x 2
Triangulation step	0.00553
Surface area	44.3162

The area of construction of the interpolation grid is an enclosing box of the model with the edges increased by the diameter of the rocket body ($D \approx 1$). The piecewise function of the admissible interpolation error is:

$$R_\phi(\phi_T(\mathbf{x})) = \begin{cases} 0.075 \cdot |\phi_T(\mathbf{x})|, & \phi_T(\mathbf{x}) < -2.5\delta \\ 10^{-12} + 0.005 \cdot |\phi_T(\mathbf{x})|, & -2.5\delta \leq \phi_T(\mathbf{x}) \leq 2.5\delta \\ 0.05 \cdot \phi_T(\mathbf{x}), & 2.5\delta < \phi_T(\mathbf{x}) \leq 5\delta \\ 0.1 \cdot \phi_T(\mathbf{x}), & 5\delta < \phi_T(\mathbf{x}) \end{cases}$$

The coefficient $\delta = 8.2207116E - 02$ corresponds to the thickness of the turbulent boundary layer on a flat plate at an incident flow velocity $2.5M$ and environment parameters at an altitude of 10 km. Other parameters of the grid generation are given in tab. 2.

Table 2

Parameter	Value
Grid generation area	11 x 6 x 3.2
Basic step h_0	0.2
Minimum cell edge length h_{l_d}	0.0125 ($l_d = 4$)
Interpolation accuracy testing step h_t	0.00625 ($l_t = 5$)

A general view of the grid constructed to determine the values of the signed distance function is shown in fig. 12. Like the surface model, the grid has a symmetric structure with respect to planes $Y = 0$ and $Z = 0$. Therefore, for clarity, the illustrations show a quarter of it against the background of the rocket surface. The outer boundaries and structural features of the subdomain of interpolation cells are highlighted in green, and the subdomain of model type cells is highlighted in red.

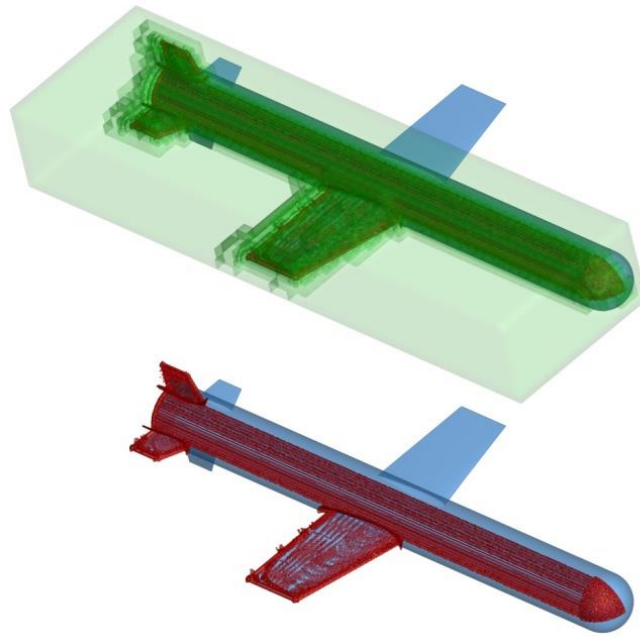
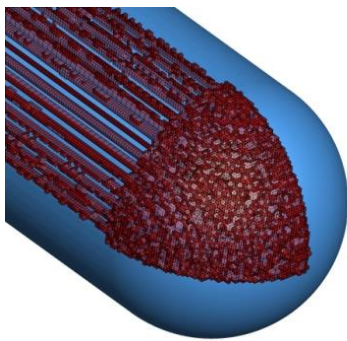


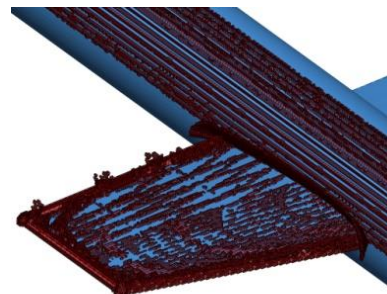
Fig. 12. General view of the grid.

In total, the grid contains 5541528 leaf cells, of which 18.5% belong to the subarea of explicit calculation of the distance to triangulation. A visualization of the distribution of model cells near various parts of the structure is shown in fig. 13.

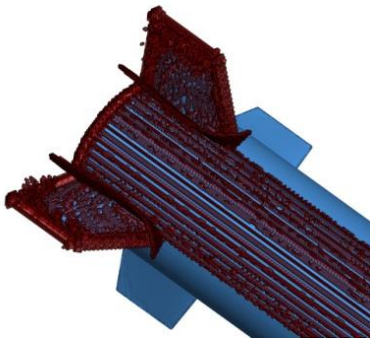
The thickness of the zone of the model cells is $h_{G_T} = 0.028819 \approx 0.35\delta$. In this case, the maximum distance from the surface to the cell of the model type is 0.206421. The illustrations show that the outermost cells are located along the bisector planes of the dihedral angles between the surfaces of the body and wings. The averaged coefficient of acceleration of the search for the distance to triangulation in the centers of mass of the model cells takes on a value $S_{kd} \approx 414$ that is approximately 3 times greater than the average value of the coefficient over the entire region of grid generation.



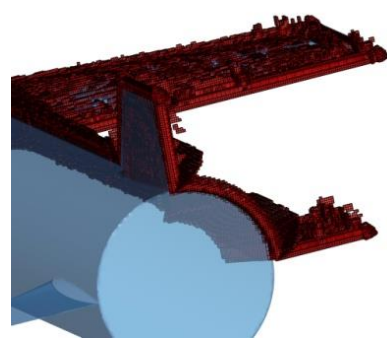
a) the nose



b) the wing



c) the tail, side view



d) the tail, rear view

Fig. 13. Visualization of the zone of model cells.

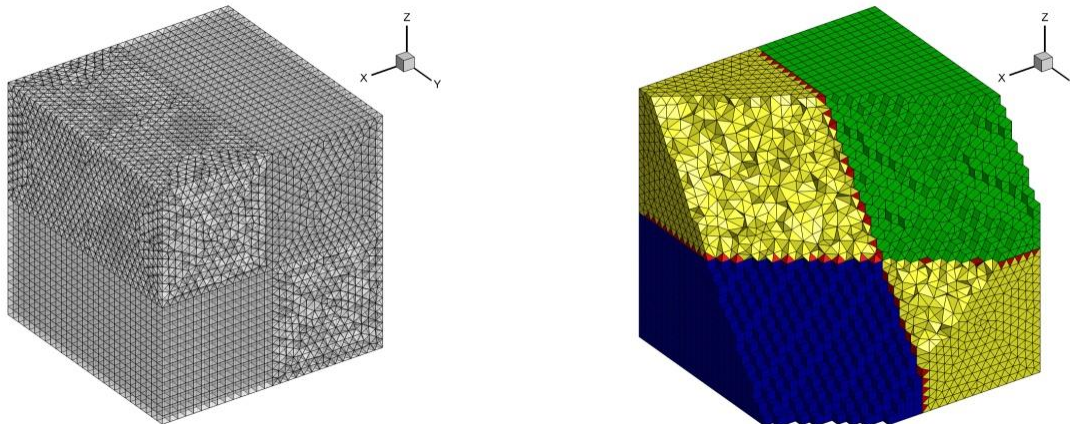
The performance of the software modules and the actual accuracy of interpolation were estimated by the results of determining ϕ_G at the nodes of a uniform orthogonal grid. The mesh is built inside the grid generation region and contains more than $20 \cdot 10^6$ vertices that do not match the points of the interpolation accuracy check at the grid generation stage. Average ratio between the actual and the specified calculation error

$$d_R(\mathbf{x}) = \frac{|\phi_G(\mathbf{x}) - \phi_T(\mathbf{x})|}{R_\phi(\phi_T(\mathbf{x}))}$$

by the mesh vertices that are inside the interpolation cells is $d_R \approx 0.0578$. That is, the interpolation error does not, on average, exceed 6% of the threshold value. However, grid generation with discrete accuracy testing in the general case does not guarantee strict observance of the given computational error. In 197 vertices belonging to the $|\phi_T(\mathbf{x})| \leq 0.09$ subdomain, the actual interpolation accuracy is lower than the specified one. The maximum value of $d_R(\mathbf{x}) = 2.184$ is fixed at $\phi_T(\mathbf{x}) \approx 0.00042$. The interpolation error here is 1.1% of the absolute value of ϕ_T instead of the specified 0.5%. In turn, the maximum error absolute value $|\phi_G(\mathbf{x}) - \phi_T(\mathbf{x})| \approx 5.9 \cdot 10^{-4}$ corresponds to $d_R(\mathbf{x}) = 1.354$ or a deviation of 0.68% from the absolute value of ϕ_T . These results are comparable in accuracy with another approaches (for example, [6]) to calculating the signed distance function to the boundary of a moving body of complex shape. It is also worth noting that we are talking about finding the values of the function at arbitrary points in space, and not about modeling the change in the field on some mesh.

The use of the combined algorithm reduces the computation time by more than 300 times compared to the time to find the exact distance to surface triangulation. The body geometry description (k-d search tree structure, surface triangulation, and interpolation grid topology) takes up 250 MB of disk space.

The second example demonstrates the possibility of using the combined algorithm in computer graphics. The problem of reconstruction and visualization of the surface of a unit sphere on the elements of a mixed mesh is considered (fig. 14). A quasi-uniform mesh with a step $h_{mesh} \approx 0.0625$ fills the volume of a cube with an edge length of 1.8 (fig. 14a). It consists of 24881 vertices and 77442 elements: 5684 hexahedrons (highlighted in blue in fig. 14b), 57316 tetrahedrons (highlighted in yellow), 12818 triangular prisms (highlighted in green), and 1625 quadrangular pyramids (highlighted in red).



a) mesh area and boundary discretization

b) the structure of the mixed mesh in the section

Fig. 14. Quasi-uniform mixed mesh.

The parameters of the interpolation grid for finding the signed distance relative to the spherical surface are given in tab. 3.

Table 3

Parameter	Value
Grid generation area	2 x 2 x 2
Basic step h_0	0.1
Minimum cell edge length h_{l_d}	0.00625 ($l_d = 4$)
Interpolation accuracy testing step h_t	0.0015625 ($l_t = 6$)

The function definition error is limited to 5% of its absolute value:

$$R_\phi(\phi_{\partial\Omega}(\mathbf{x})) = 10^{-6} + 0.05 \cdot |\phi_{\partial\Omega}(\mathbf{x})|.$$

The boundary of the object in this case is described analytically

$$\phi_{\partial\Omega}(\mathbf{x}) = \sqrt{x^2 + y^2 + z^2} - 0.5.$$

That is, the interpolation error is tested against exact values $\phi_{\partial\Omega}$.

Since the experiment assumes the use of a grid solely for the purpose of visualizing the surface, then all leaf cells of the l_d level, regardless of the final calculation error, are assigned an interpolation type ($h_{G_T} = 0$). The grid structure in the center slice is shown in fig. 15.

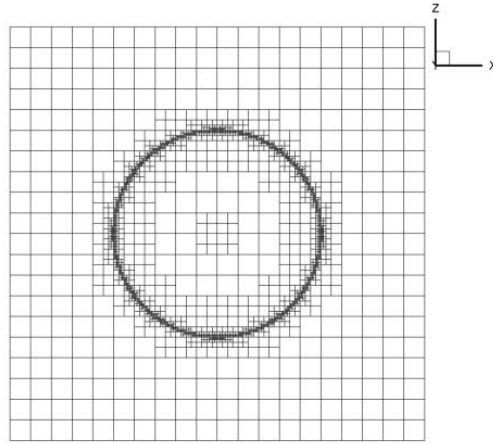


Fig. 15. The structure of the interpolation grid in the center slice.

The visualization of the sphere boundaries on the elements of the mixed mesh is performed in two ways. The first method is analogous to voxelization. The surface is approximated by the outer boundary of the set of mesh elements that are entirely inside the sphere. The second method is based on the use of the level set method. The boundary is reconstructed with a polygonal mesh corresponding to the isosurface $\phi_{\partial\Omega} = 0$.

In order to increase the degree of image detail, the original conformal mesh ($MESH_B$) is twice adapted (meshes $MESH_{A1}$ and $MESH_{A2}$) to the spherical surface (fig. 16). The adaptation mechanism consists in hierarchical isotropic refinement of elements with the addition of new vertices at the midpoints of the edges, at the centers of quadrangular faces and volumes of hexahedra. The adaptation area is formed from cells, the vertices of which are located on opposite sides of the sphere surface.

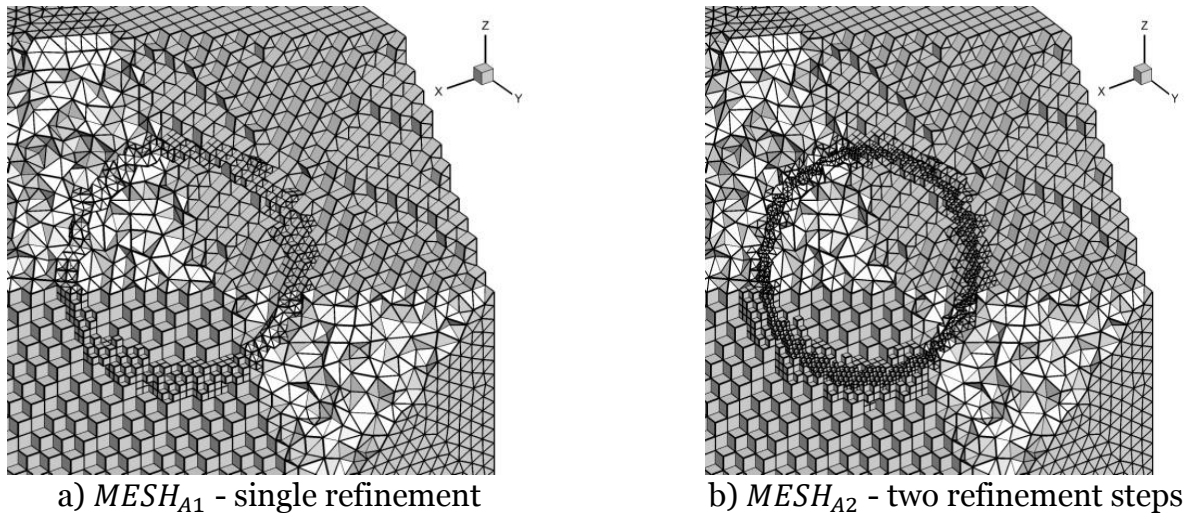


Fig. 16. Mesh structure after adaptation.

The signed distance field specified at the mesh vertices is initialized by interpolation over the cells of the generated mesh. To build a polygonal mesh and visualize the surface, the TecPlot toolkit [17] was used.

The contours of the sphere, obtained as a mapping of the outer boundary of the set of mesh cells inside it, are shown in fig. 17. Mesh adaptation improves the accuracy of the surface reconstruction, but even in the general plan, all images are noticeably different from the original object.

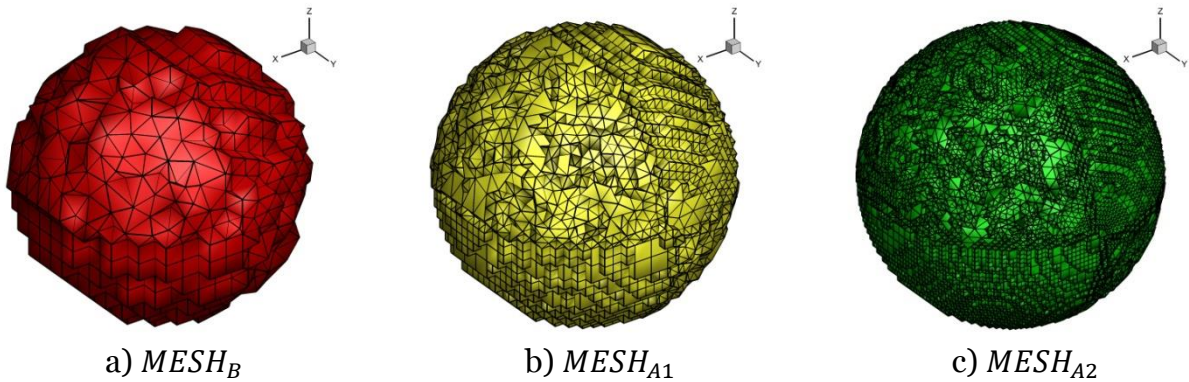


Fig. 17. Visualization of a spherical surface by the boundary of a set of internal mesh cells.

The second visualization method with the approximation of the sphere boundaries by a polygonal mesh gives a better result (fig. 18).

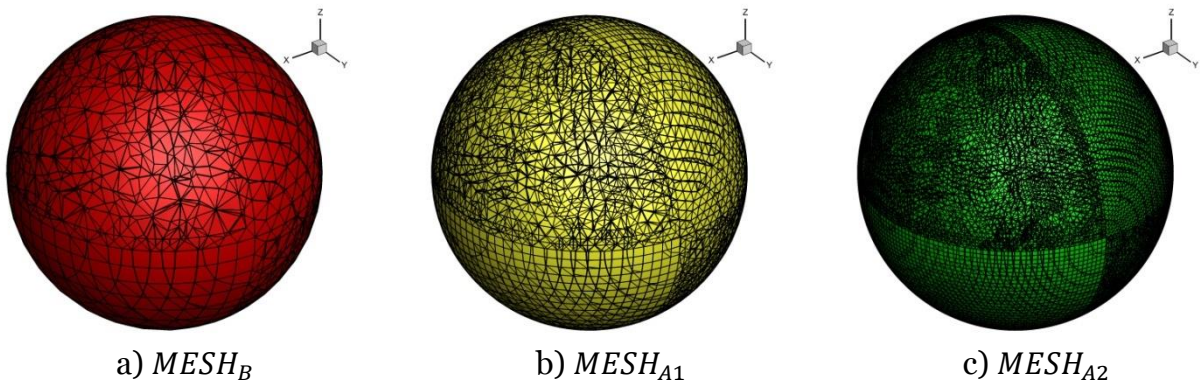


Fig. 18. Visualization of a surface based on a polygonal mesh.

Moreover, the difference between the surface reconstruction on different grids is visually practically indistinguishable. Numerical estimation (tab. 4) shows that the maximum distance from the vertices of the reconstructed surface to the actual surface of the sphere decreases after each step of the volume mesh adaptation. At the same time, the deviation of the polygonal mesh area from the surface area of the displayed object is also reduced.

Table 4

	$MESH_B$	$MESH_{A1}$	$MESH_{A2}$
Maximum distance	0.003267	0.000978	0.000385
Square deviation	0.518 %	0.136 %	0.039 %

This example confirms the possibility of using the software implementation of the interpolation approach in problems of visualization of solid surfaces.

5. Conclusion

This paper describes a combined algorithm for determining the signed distance to the boundaries of moving solid bodies. The accuracy of the presented algorithm does not depend on the shape of the body and the parameters of the trajectory of its movement. The combined approach can be used both in numerical calculations of physical processes and for visualization of objects, the geometry of which is specified by the signed distance field. The software implementation of the algorithm is characterized by increased performance and without additional modifications can be added to the codes of sequential and parallel applications. A further direction of work is to implement an interpolation procedure using high-order polynomials.

References

1. Jones M., Bærentzen A., Sramek M. 3D distance fields: A survey of techniques and applications // IEEE transactions on visualization and computer graphics., 2006, 12, 581-99. DOI: 10.1109/TVCG.2006.56.
2. Mittal R., Iaccarino G. Immersed boundary methods // Annu. Rev. Fluid Mech., 2005, v.37, p. 239–261.
3. Spalart P.R., Allmaras S.R. A one-equation turbulence model for aerodynamic flows // AIAA Paper 92-0439, 30th Aerospace Science Meeting, Reno, Nevada, 1992.
4. Osher S., Fedkiw R. Level set methods and dynamic implicit surfaces // New York: Springer-Verlag, 2003, 273p.
5. Shervani-Tabar N., Vasilyev O.V., Stabilized conservative level set method // Journal of Computational Physics, 2018, 375, pp. 1033-1044.
6. Morgan N., Waltz J. 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement // Journal of Computational Physics. 2017, 336, 492-512. DOI: 10.1016/j.jcp.2017.02.030.
7. Jones M. The production of volume data from triangular meshes using voxelization // Computer Graphics Forum, 1996, 15:311-318.
8. Hart J. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. // The Visual Computer, 1996, 12, 527–545. DOI: <http://doi.org/10.1007/s003710050084>.
9. Museth K. VDB: High-resolution sparse volumes with dynamic topology // ACM Trans. Graph. 32, 3, Article 27 (June 2013) 22 pages. DOI: <http://dx.doi.org/10.1145/2487228.2487235>.
10. Koschier D., Deul C., Bender J. Hierarchical hp-adaptive signed distance fields // In: Proceedings of ACM SIGGRAPH EUROGRAPHICS Symposium on Computer Animation (SCA), 2016.

11. Frisken S.F., Perry R.N., Rockwood A.P., Jones T.R. Adaptively sampled distance fields: a general representation of shape for computer graphics // *ACM Transactions on Graphics*, 2000, 249–254.
12. Abalakin I.V., Kozubskaya T.K., Soukov S.A., Zhdanova N.S. Numerical Simulation of Flows over Moving Bodies of Complex Shapes Using Immersed Boundary Method on Unstructured Meshes // In: Garanzha V., Kamenski L., Si H. (eds) *Numerical Geometry, Grid Generation and Scientific Computing. Lecture Notes in Computational Science and Engineering*, vol 131. Springer, Cham. 2019. DOI: https://doi.org/10.1007/978-3-030-23436-2_13
13. Baerentzen J.A., Aanaes H. Signed distance computation using the angle weighted pseudonormal // *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 3, pp. 243-253, May-June 2005. DOI: 10.1109/TVCG.2005.49.
14. Bentley J.L. Multidimensional binary search trees used for associative searching // *Communications of the ACM*. 1975, 18 (9): 509–517. DOI:10.1145/361002.361007.
15. Kryuchkova A.S. Numerical simulation of a hypersonic flow over HB-2 model using UST3D programming code // 2019 *J. Phys.: Conf. Ser.* 1250 012010. DOI: 10.1088/1742-6596/1250/1/012010.
16. Yianilos P. Data structures and algorithms for nearest neighbor search in general metric spaces // *Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. 1993. DOI: 10.1145/313559.313789.
17. TecPlot – CFD Visualization & Analysis Software URL: <https://www.tecplot.com/>