Scientific Visualization, 2021, volume 13, number 5, pages 52 - 64, DOI: 10.26583/sv.13.5.05

Visualization of the CFD Calculations Results on Adaptive Mixed Meshes

S.A. Soukov[1]

Keldysh Institute of Applied Mathematics RAS

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

Abstract

The article discusses the problem of visualizing the results of computational fluid dynamics simulations on cell-centered adaptive mixed meshes. An algorithm for converting hierarchical adaptive mixed meshes consisting of tetrahedra, triangular prisms, quadrangular pyramids, and hexahedra to a conformal form is described. The main idea of the proposed algorithm is to decompose a subset of mesh cells with hanging nodes on the edges into tetrahedra and pyramids with the addition of new vertices at the polyhedra mass centers. The mesh functions interpolation algorithm is based on the discrete analogue of the formula for the integral representation of the gradient. Application of the algorithm allows to show the fields of mesh functions using software for data visualization on irregular grids. On the example of a stationary supersonic flow visualization at the sphere surface using the Tecplot tools, the advantages and disadvantages of various data presentation options are demonstrated. These include displaying results on a hanging node mesh without changing the topology, transferring data to a starting conformal mesh, polyhedra data visualization, and converting an adaptive mesh to conformal form with cell decomposition and data interpolation.

Keywords: computational fluid dynamics, data visualization, mixed meshes, adaptive mesh refinement.

1. Introduction

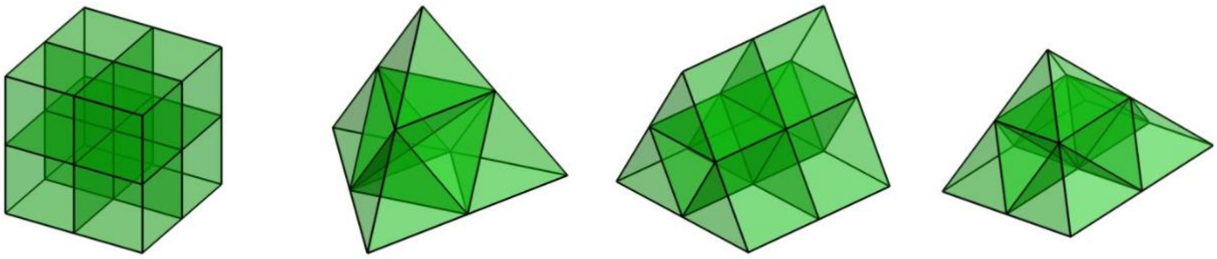
In modern computational fluid dynamics, methods of static and dynamic adaptation of meshes are used to improve the solution accuracy and reduce the computation time. The adaptive mesh is refined during the simulation, taking into account the features of the flow structure. For example, the adaptive mesh is refined near the shock wave. The first papers describing the technology and the results of calculations on adaptive meshes were presented more than a quarter of a century ago [1, 2]. Nevertheless, the problems of developing new methods, optimization and application of known refinement techniques are of interest at the present time [3, 4].

In terms of frequency of references in the literature, the main approaches are based on hierarchical decomposition and coarsening of elements. The most common example of meshes of this type is octree grids, which are used both directly in calculations [5] and for solving other applied problems [6]. The practical advantage of using hierarchical meshes in CFD is the logical simplicity of the implementation of dynamic adaptation procedures for simulating unsteady flows. For a fixed total number of cells, the fine mesh zone moves arbitrarily within the computational domain. The hierarchical topology does not formally have restrictions on the difference between the sizes of neighboring leaf elements. However, in mathematical physics calculations, the difference in the refinement levels of neighboring calculation cells, as a rule, is limited to one. For example, an octree leaf cell always has either one or four neighbors on each face.

In the case of modeling flows near surfaces of complex shapes, mixed meshes are used for spatial discretization of the system of gas-dynamic equations. They consist of so-called basic polyhedra (tetrahedron, triangular prism, quadrangular pyramid, hexahedron) and polyprisms with polyhedra. Polyprisms and polyhedra are generated by transforming mesh blocks of triangular prisms and tetrahedra [7].

Mixed meshes data visualization and analysis are supported by most postprocessing and scientific visualization software tools. The topology of basic polyhedra is described either explicitly with an indication of the cell type (VTK file format [8]), or the hexahedron format with degenerate edges and faces is used (Tecplot package [9]). For example, a triangular prism is defined as a hexahedron at six mesh nodes with two opposite edges contracted to points. The description of the topology of a polyhedron mesh contains ordered lists of flat faces nodes and the pairs of indices of cells adjacent along the faces. The visualization algorithms work with the mesh functions values specified at the nodes. But the data storage formats allow the determination of the flow variable values both at the nodes and at the mesh element centers. In the last case, the interpolation of values is performed by visualization tools at the stage of reading the input files.

The adaptation methods for mixed meshes and regular grids are essentially the same. Isotropic decomposition of basic polyhedra is performed using fixed decomposition stencils (fig. 1). New nodes are placed at the midpoints of the edges, at the quadrangular face centers, and at the hexahedron mass centers. After splitting tetrahedra, hexahedrons and prisms, eight elements of a same type appear. The quadrangular pyramid is split into four tetrahedrons and six pyramids. In the polyhedral zones, at the first adaptation step, the cells are divided into basic polyhedra, to which the standard decomposition stencils are subsequently applied. The implementation of the latter approach is included in the Ansys Fluent software package (PUMA technology [10]).



a) hexahedron b) tetrahedron c) prism d) pyramid

Fig. 1. Mixed mesh cell decomposition types.

Locally cell refinement leads to the appearance of "hanging nodes". Technically, an adaptive hanging node mesh topology can be stored in the same format as an unstructured conformal mesh topology. But at the same time, the border of adjacent zones of leaf cells at different refine levels is displayed as an additional internal border. Fig. 2 shows an example of visualization of an adaptive mesh for simulating the flow around a ballistic model (hereinafter, the illustrations are obtained using the Tecplot tools). The appearance of the inner boundary leads to discontinuities of the level lines and mesh functions isosurfaces.

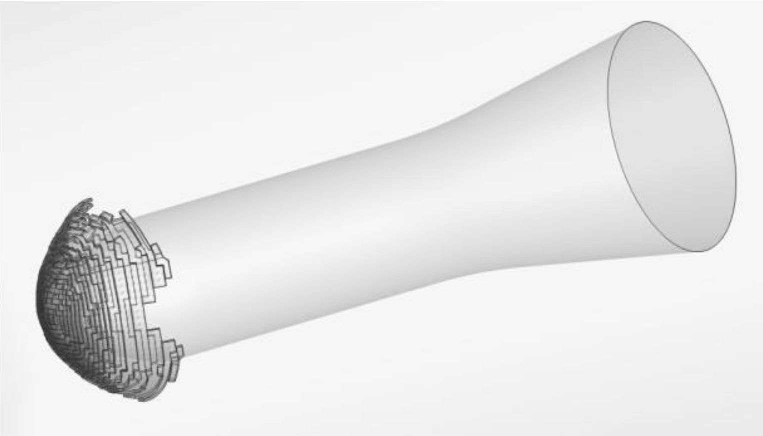


Fig. 2. Inner border when rendering a adaptive hanging node mesh.

There are two options for solving the problem: the development of special visualization algorithms, or the transformation of an adaptive mesh into a conformal mesh consisting of arbitrary or base polyhedra, followed by visualization of the results by means of viewing data on unstructured meshes. A generalized review of visualization methods and software for adaptive meshes is given in [11]. In terms of their functionality, narrowly focused applications are expected to lose out to more popular and widespread software packages for conformal mesh data visualization. In addition, the geometry representation format used in the computational kernel, as a rule, differs from the format supported by the viewer. Thus, to display the solution, the storage of special files is necessary. An important question is the duration of support and the support terms of specialized software products by the developer. Therefore, the variant with the transformation of the adaptive mesh looks more general from a practical point of view, although it has similar disadvantages. The implementation of the approach is associated with the development of an algorithm for transforming the topology and recording individual files to visualize the flow structure.

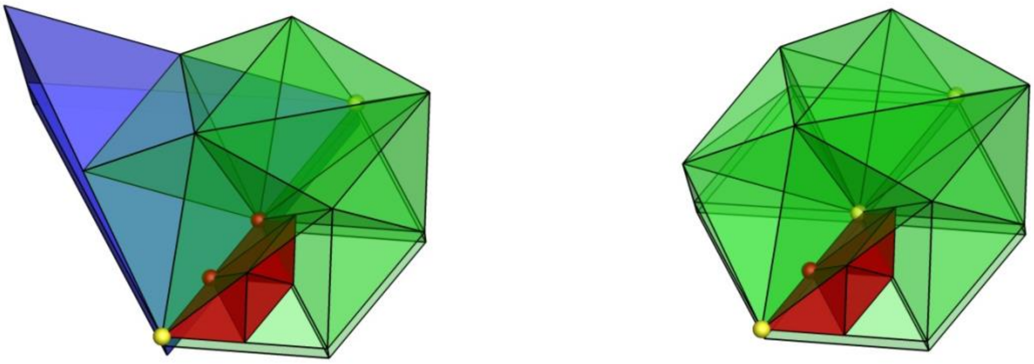
One of the well-known methods for transforming an adaptive mesh consisting of basic polyhedra is presented in [12]. The cell decomposition type is selected according to the number and distribution of hanging nodes on its edges and faces. The approach is focused primarily on generating a computational mesh containing elements of a regular geometric shape. Therefore, for some configurations of the node location, it is proposed to repeat the procedure of refinement of the mesh cells, which leads to the expansion of the adaptation zone and multiple values interpolation. The code and software implementation of the algorithm are distinguished by complex logic for transforming elements and a large amount of computation.

This paper describes a data visualization-oriented, logically simple algorithm for transforming hierarchical adaptive meshes with elements of the tetrahedron, prism, pyramid, and hexahedron type to conformal form. On the example of the problem of visualizing the structure of a stationary supersonic flow near the surface of a sphere using the Tecplot package, the advantages and disadvantages of various data presentation options for viewing the results of simulations on adaptive mixed meshes are shown.

2. Adaptive mesh transformation algorithm

The algorithm for transforming a hierarchical adaptive mesh to a conformal form is applicable for mixed meshes consisting of polyhedra C_i of four types: tetrahedron, triangular prism, quadrangular pyramid, and hexahedron. The values of the mesh functions f_i are determined at the centers of mass of the elements $\mathbf{x_i} = (x_i, y_i, z_i)$. The mesh topology must comply with the limit for a difference of no more than one nesting level between the leaves adjacent to the edge. There can be no more than one hanging node on the edge of a leaf cell, and the number of its neighbors on each face is equal to one or four cells. Otherwise, the mesh is preliminarily reduced to the specified form.

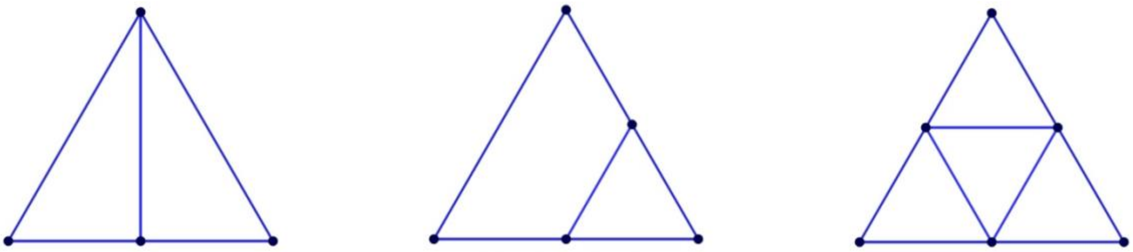
Fig. 3a shows an example of an adaptive mesh topology, where two hanging nodes (red spheres) are located on a common edge of two cells (cell faces are painted in blue, vertices of a common edge are marked with yellow spheres). Here, cells of three levels of the hierarchy border along the edge. This is against the restrictions. By analogy with [12], the problem is solved by uniformly refining the cells of the upper level. As a result of decomposition, the edge is divided into two parts, only one of which has a hanging node (fig. 3b).



a) edge with two hanging nodes b) edge with one hanging node

Fig. 3. Matching mesh topology to constraints.

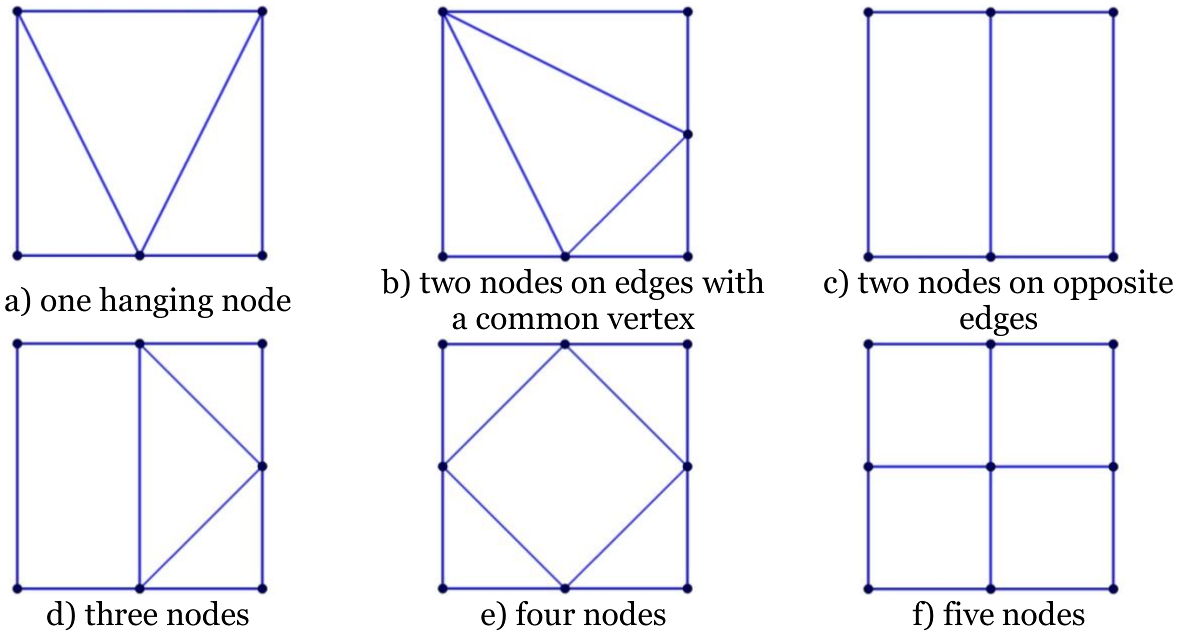
The essence of the proposed mesh transformation algorithm is a sequential partitioning of cells with hanging nodes. Cell decomposition is performed in two steps. The first step is to split the hanging node faces. Fixed subdivision types of triangles and quadrangles are shown in fig. 4 and fig. 5. The subdivision type is determined by the number of hanging nodes. For quadrangles with two hanging nodes, their relative position is additionally taken into account: on adjacent (fig. 5b) or opposite (fig. 5c) edges. At the end of the first step, the mesh cell becomes a convex polyhedron with the total number of flat faces from 5 to 30.



a) one hanging node b) two hanging nodes c) three hanging nodes

Fig. 4. Subdivision types for triangular faces.

In the second step, the volume of the polyhedron is divided into pyramids. The faces of the cells become the bases of the pyramids, and a new mesh node located in the center of mass of the polyhedron becomes the common vertex. As a result, the cell of the adaptive mesh splits into tetrahedrons and quadrangular pyramids.



a) one hanging node

b) two nodes on edges with a common vertex

c) two nodes on opposite edges

d) three nodes

e) four nodes

f) five nodes

Fig. 5. Subdivision types for quadrangular faces.

Fig. 6 shows illustrations of examples of transformation of elements. A tetrahedron with five nodes on its edges is divided into ten tetrahedra and two quadrangular pyramids (fig. 6a). A hexahedron with two hanging nodes splits into ten tetrahedra and three pyramids (fig. 6b).

a) decomposition of tetrahedron

b) decomposition of hexahedron

Fig. 6. Examples of transformation of mesh cells.

The face decomposition types are unambiguous. Regardless of the processing order of cells of the same hierarchy level, their common face will be divided into polygons in the same way. Special cases of face decomposition coincide with isotropic refinement algorithms during mesh adaptation. Decomposition types of faces with the maximum number of hanging nodes (fig. 4c and fig. 5f) correspond to the variants of splitting the faces from the side of four cells of the next hierarchy level. Thus, upon completion of the decomposition of cells with hanging nodes, the hierarchical adaptive mesh will be converted to a conformal form.

To calculate the values of the variables at the mass centers of the nested cells C_{ch}, the polynomial defined on the parent cell C_p is used

$$f_{ch} = k_p^x(x_{ch} - x_p) + k_p^y(y_{ch} - y_p) + k_p^z(z_{ch} - z_p) + f_p.$$

Linear reconstruction of variables corresponds to the conservative interpolation condition

$$f_p|C_p| = \sum f_{ch}|C_{ch}|,$$

where $|C_i|$ is the volume of the i-th mesh cell.

The polynomial reconstruction coefficients are calculated on the basis of a discrete analogue of the formula for the integral representation of the gradient

$$\left(k_i^x, k_i^y, k_i^z\right) = \frac{1}{|C_i|}\sum_{faces} f_{face}\mathbf{S}_{\mathbf{face}}.$$

Here f_{face} denotes the values of the function at the mass centers of flat faces with square vectors $\mathbf{S_{face}}$.

If the cell C_i borders on the cell C_j of the same or the up level of the hierarchy, the value of the function in the center of their common face is equal to the arithmetic mean of the values of the function in the cell centers

$$f_{face} = \frac{1}{2}(f_i + f_j).$$

If on the opposite side of the face of cell C_i there are four cells C_{j_0}, C_{j_1}, C_{j_2} C_{j_3} of the next level of subdivision, the formula is used for averaging

$$f_{face} = \frac{\sum_{m=0,1,2,3} f_i + f_{j_m}}{8}.$$

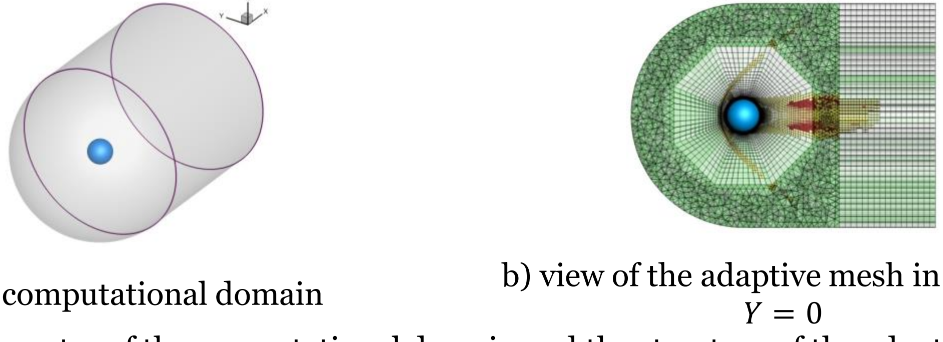
The value of the function in the center of the face located on the boundary of the computational domain is equal to the value in the cell center.

The algorithms for cell decomposition and transfer of the mesh function values are in good agreement with the format for representing the topology of adaptive meshes. Thus, the software implementation of the approach does not require the initialization of additional geometric parameters or cell connectivity relations. The chosen interpolation method provides sufficient accuracy of displaying the flow structure and determining its characteristic parameters (for example, the distance to the detached shock wave) by the visualization software tools. However, to minimize errors to the level of the results of internal algorithms for analyzing the solution of a CFD kernel, it is more preferable to use second order methods for calculating gradients [13] with the subsequent determination of the values of the variables at the mesh nodes.

3. Computational experiment

The features of using various approaches to the presentation of data for visualizing the calculation results on locally adaptive mixed meshes using the Tecplot are considered using the example of a supersonic flow ($M = 2$, $Re = 300$) near the sphere surface. The problem statement, calculation method, flow structure and parameters are discussed in detail in [13].

A doubly connected computational domain is a union of a cylinder and a sphere, inside which there is a streamlined body (fig. 7a). At the stage of generating the starting grid, the computational domain is divided into several tens of domains with common flat and curved faces. The boundary layer zone is filled with tetrahedrons and prisms. At the outer boundaries of the computational domain, the mesh consists of tetrahedra and hexahedra. In the rest of the space, hexahedral, tetrahedral and prismatic mesh blocks alternate. The quadrangular pyramids are used to join the triangular faces of the tetrahedra with the quadrangular faces of the hexahedra and prisms. The total dimension of the conformal mesh is 1.3·10^5 nodes and 2.7·10^5 elements.



a) computational domain

b) view of the adaptive mesh in the slice $Y = 0$

Fig. 7. The geometry of the computational domain and the structure of the adaptive mesh for simulating the supersonic flow around a sphere.

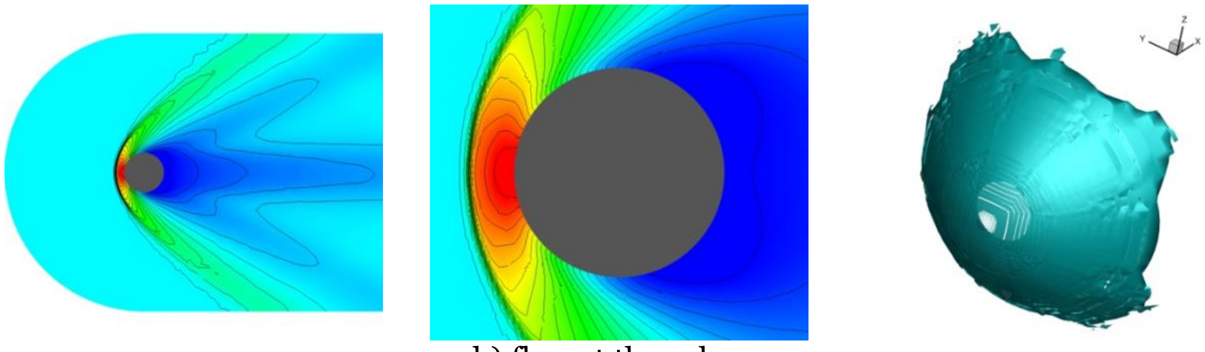
During the simulation, the mesh is adapted twice. The experimentally chosen combination of adaptation criteria allows to refine the mesh without reference to geometric coordinates. Mesh refinement is performed near the isosurface of the local Mach number $M = 1$ (transition from supersonic to subsonic flow velocity), along the boundaries of the recirculation zone, as well as in the regions of maximum pressure gradients and velocity modulus. The structure of the final adaptive mesh is shown in fig. 7b. The cells of the first refinement level are colored yellow, the cells of the second level are marked in red. The refinement zone goes through the boundary layer zone and crosses the boundaries of the mesh blocks. As a result of local decomposition, the total number of computational cells increases by 2.37 times. The hanging node mesh dimension is $4.5\cdot10^5$ nodes and $6.5\cdot10^5$ elements.

Comparative visualization of the flow is performed for four data presentations:

- visualization of the values specified in adaptive mesh cell centers;
- transfer of values to the mass centers of the starting grid cells;
- converting a hanging nodes mesh to a polyhedral mesh;
- transform adaptive mesh with elements decomposition.

The choice of these approaches is due to the possibility of using software packages for visualization of solutions specified on conformal meshes.

Visualizing the flow on an adaptive mesh without any transformations is the simplest approach to viewing the results. But as noted above, the main problem of the approach is associated with the appearance of artificial internal border, which lead to discontinuities of isosurfaces. Fig. 8a shows the density distribution in the slice Y = 0. A continuous color palette and an additional twenty-one level lines are used. In the general view, the contour lines appear to be continuous. When scaling the image (fig. 8b), you may notice minor gaps that have little impact on the viewing of the results. However, the three-dimensional isosurface $\rho = 1.35$ constructed on a hanging node mesh (fig. 8c) consists of many disconnected parts located at a noticeable distance from each other. In this case, the level $\rho = 1.35$ does not refer to the characteristic parameters of the flow and is used exclusively as the most illustrative example to demonstrate the problems of visualizing the results. The incorrect viewing of the isosurface is explained by the fact that it passes through the boundaries of the different refinement level zones. Thus, the approach is applicable for monitoring intermediate calculation results, but cannot be used at the stage of visualizing the final results and detailed analysis of the flow structure.



a) slice $Y = 0$

b) flow at the sphere surface

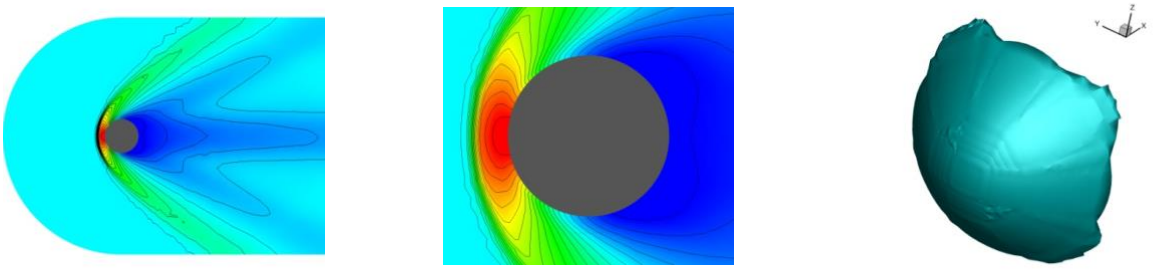
c) isosurface $\rho = 1.35$

Fig. 8. Visualization of the density distribution on an hanging node adaptive mesh.

The first way to remove discontinuities is to transfer the solution to the initial conformal mesh. In accordance with the finite volume method, the values of the mesh functions in the parent cell are calculated as the volume average of the values specified in the child element centers

$$f_p = \frac{\Sigma f_{ch}|C_{ch}|}{\Sigma|C_{ch}|}.$$

The flow pattern created from the values specified on the conformal mesh, as expected, does not contain isolines and isosurfaces discontinuities (fig. 9). At the same time, the transition to a coarse mesh leads to a worsening in image detail. For example, it can be seen that the isolines in the region of the detached shock wave are at an increased distance from each other in comparison with the images in fig. 8a and fig. 8b. That is, this approach limits the possibility of demonstrating small-scale details of the solution, for the purpose of resolving which the mesh refinement methods are used.



a) slice $Y = 0$ b) flow at the sphere surface c) isosurface $\rho = 1.35$

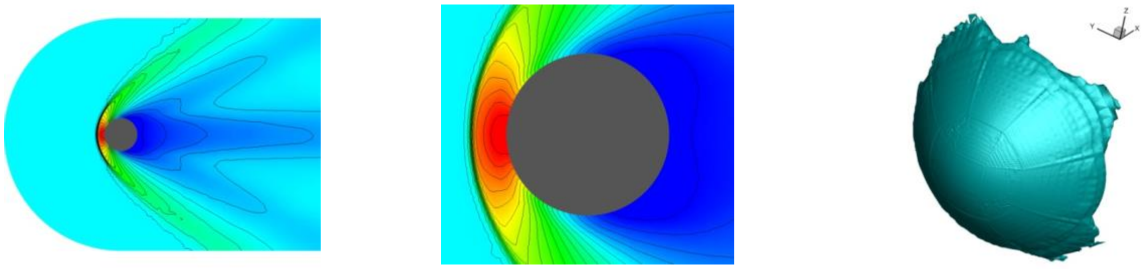
Fig. 9. Visualization of the density distribution on the original grid.

The second way to exclude interior border is to transform polyhedra with hanging nodes on the edges to polyhedra with flat polygonal faces. Here a triangle with a hanging node on one edge becomes a quadrilateral, a quadrilateral with hanging nodes on two edges becomes a hexagon, and so on. When a mesh element has four neighbors of the next refinement level along a face, then such a face is appropriately divided into parts. The number of mesh cells and the cell center coordinates do not change. Therefore, interpolation of the values of the mesh functions is not required. Fig. 10 shows an illustration of the transformation of a prism to a polyhedron with eleven planar faces. Two of the five faces of the cell are split into four parts. The remaining quadrangles become a pentagon and a hexagon. The hanging node triangular face transforms into a quadrilateral.

a) prism and its neighboring cells b) polyhedral cell

Fig. 10. Transformation of the prism.

The flow pattern created from the values of the mesh function at the polyhedra centers is shown in fig. 11. Level lines and 3D density isosurface are displayed without breaks. The detailing of the solution features is equivalent to the picture in fig. 8. And, therefore, this visualization method is the most preferable when using software packages that work with polyhedral meshes.



a) slice $Y = 0$ b) flow at the sphere surface c) isosurface $\rho = 1.35$

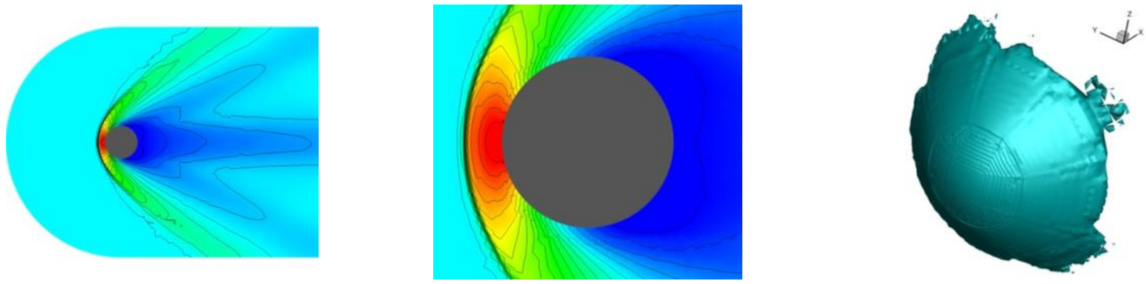
Fig. 11. Visualization of density distribution on a polyhedral mesh.

In the absence of polyhedron rendering support, an adaptive mesh can be converted to a mixed conformal mesh of base polyhedra using the algorithm described in the previous section of the article. The dimensions of the adaptive mesh and the conformal mesh generated on its basis are given in tab. 1.

Table 1

	Adaptive mesh	Conformal mesh
Number of nodes	453413	500238
Number of tetrahedra	228634	658580
Number of pyramids	9781	146963
Number of prisms	99839	94197
Number of hexahedrons	307168	288326
Total number of cells	645422	1188066

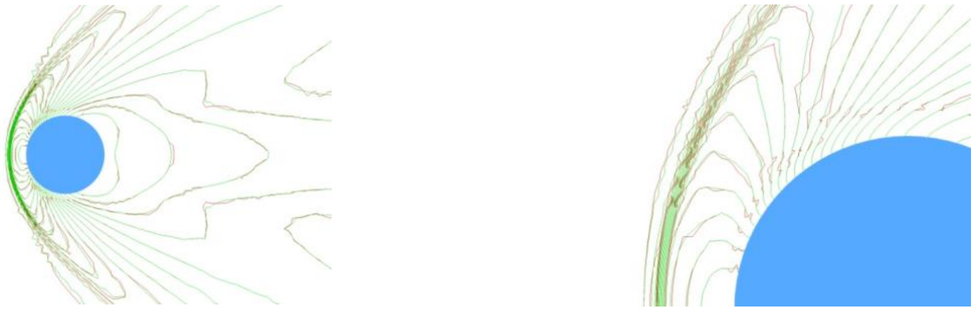
The use of the decomposition algorithm leads to a noticeable increase in the number of elements. In the case under consideration, after splitting 46825 cells (7.25% of the total number of elements of the adaptive mesh) with hanging nodes on the edges, their total number increases 1.84 times. At the same time, the distribution of mesh elements by type is changing. The relative number of hexahedra and prisms decreases, and the proportion of tetrahedra and pyramids increases. According to theoretical estimates for unstructured finite-volume algorithms, the amount of computation within one time step increases by about 1.57 times. But despite the increase in the number of elements, the visualization file with the grid topology and fields of values of five variables takes up 65 MB of disk space, which is 18% less than the size of the data file with the topology and results in the centers of the polyhedra. The increased size of the polyhedra topology description is explained by the use of a format with an explicit enumeration of lists of nodes of faces and an indication of the belonging of faces to elements.



a) slice $Y = 0$ b) flow at the sphere surface c) isosurface $\rho = 1.35$

Fig. 12. Visualization of the density distribution on the transformed mesh.

The flow pattern on the mesh obtained by the transformation (fig. 12) is extremely close to the data visualization on the polyhedral mesh (fig. 11). In a detailed analysis with overlapping images, local distortions of the curves appear. Fig. 13 shows the density isolines in the slice of the polyhedral mesh in green, the isolines built on the mesh with four types of elements are marked in red. Kinks in the red curves can be seen near the surface of the sphere in the boundary layer zone.



a) slice $Y = 0$

b) flow at the sphere surface

Fig. 13. Density contours for two mesh structures.

The indicated problem is not the result of incorrect operation of the mesh transformation algorithm or a consequence of interpolation errors in the values of mesh functions. Local distortions of the flow pattern are more or less present in all the illustrations above. Their appearance is due to an error in the internal algorithm of the visualization program, which is responsible for transferring values from the centers of the cells to the mesh nodes.

Fig. 14 shows the isolines of the potential flow pressure at the surface of the sphere. The green curves correspond to the exact solution specified at the nodes of the conformal mesh. The red curves are plotted according to the exact solution determined at the centers of mass of its elements. Contour lines are displayed in blue for the case of determining the pressure P_i^{node} at the nodes by averaging the values of the variable at the centers of the elements to which the nodes belong

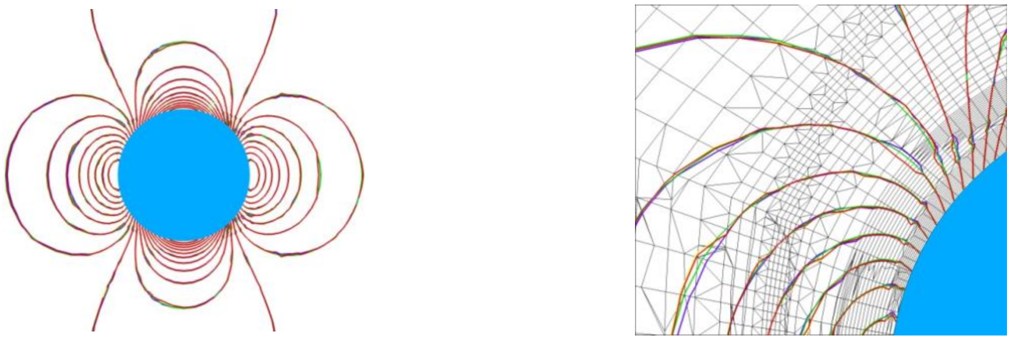
$$P_i^{node} = \frac{\sum P_k^{cell} w_k}{\sum w_k},$$

$$w_k = |C_k|/N_k.$$

The weight of the element w_k is equal to its volume divided by the number of vertices N_k. An alternative variant of averaging (yellow lines) is to define the weights as the inverse distances between the nodes and the centers of mass of the cells

$$w_k = \frac{1}{d(\mathbf{x_i},\mathbf{x_k})}.$$

All curves accurately reflect the solution pattern (fig. 14a). But the green isolines constructed from the values of the function at the mesh nodes have a smooth shape, and the curves obtained in one way or another from the solution at the centers of mass of the cells again show distortions (fig. 14b).



a) slice $Y = 0$

b) flow at the sphere surface

Fig. 14. Pressure isolines for a potential flow around the surface of a sphere.

The problem of errors in algorithms for transferring values of variables is present in problems of data visualization on any unstructured meshes. The most noticeable image defects are manifested on unstructured meshes with elements of irregular geometric shape, large differences in the sizes of neighboring polyhedra, numbers of edges, faces, etc. This class of discrete models includes both adaptive meshes with hanging nodes and various variants for converting them to a conformal form. The maximum visualization accuracy is achieved here only by using special high-precision interpolation algorithms, taking into account, among other things, the physical formulation of the problem.

4. Conclusions

The article discusses the issues of visualization of the results of CFD simulations on mixed locally adaptive meshes using unstructured data visualization tools. An algorithm is presented for transforming a hierarchical adaptive mesh with four types of cells to a conformal form by concerted decomposition into tetrahedrons and pyramids of a subset of cells with hanging nodes. On the example of visualization of supersonic flow using the Tecplot package, the advantages and disadvantages of various methods of data presentation are shown. The main approaches are the representation of the cells of the adaptive mesh in the form of polyhedra and the transformation of the topology using the proposed local decomposition algorithm.

References

1. Berger M.J., Colella P. Local adaptive mesh refinement for shock hydrodynamics // Journal of Computational Physics, vol. 82, Issue 1, 1989, pp. 64-84. DOI: https://doi.org/10.1016/0021-9991(89)90035-1.

2. Powell K., Roe P., Quirk J. Adaptive-Mesh Algorithms for Computational Fluid Dynamics. In: Hussaini M.Y., Kumar A., Salas M.D. (eds) Algorithmic Trends in Computational Fluid Dynamics // N.Y.: ICASE/NASA LaRC Series. Springer, 1993.

3. Afendikov A.L., Khankhasaeva Y.V., Lutsky A.E. et al. Numerical Simulation of the Recirculation Flow during the Supersonic Separation of Moving Bodies. Math Models Comput Simul 12, 282–292 (2020). DOI: https://doi.org/10.1134/S2070048220030035.

4. Tsvetkova V., Kozubskaya T., Kudryavtseva L., Zhdanova N. On mesh adaptation for supercomputer simulation of flows around solid bodies defined by immersed boundary method // Procedia Computer Science, 2020, 178, p. 404-413

5. Pavlukhin P., Menshov I. (2019) GPU-Aware AMR on Octree-Based Grids. In: Malyshkin V. (eds) Parallel Computing Technologies. PaCT 2019. Lecture Notes in Computer Science, vol 11657. Springer, Cham. https://doi.org/10.1007/978-3-030-25636-4_17.

6. Soukov S.A. Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid bodies movement // Scientific Visualization, 2020, vol. 12, num. 5, pp. 86-101. DOI: 10.26583/sv.12.5.08.

7. Garimella R.V., Kim J., Berndt M. (2014) Polyhedral Mesh Generation and Optimization for Non-manifold Domains. In: Sarrate J., Staten M. (eds) Proceedings of the 22nd International Meshing Roundtable. Springer, Cham. https://doi.org/10.1007/978-3-319-02335-9_18

8. ParaView – Multi-platform data analysis and visualization application URL: https://www.paraview.org/

9. Tecplot – CFD Visualization & Analysis Software URL: https://www.tecplot.com/

10. https://www.ansys.com/products/fluids/ansys-fluent

11. Weber G.H., Beckner V.E., Childs H., Ligocki T.J., Miller M.C., Van Straalen B., Bethel E.W. Visualization Tools for Adaptive Mesh Refinement Data // United States, 2007. https://www.osti.gov/servlets/purl/925532.

12. Mavriplis D.J. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes // 35th Aerospace Sciences Meeting and Exhibit. DOI: https://doi.org/10.2514/6.1997-857

13. Sozer E., Brehm C., and Kiris C.C. Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered CFD solvers // In AIAA Paper 2014-1440, 2014.

14. Soukov S.A. Parallelization for unstructured adaptive mesh CFD algorithm // Matem. Mod., 33:6 (2021), 31–44. DOI: https://doi.org/10.20948/mm-2021-06-03.