Development of information technologies and algorithms for generation of block-structured adaptive grids in complex domains

Yuriy Dimitrienko¹ and Andrey Zakharov¹

¹ Computational Mathematics and Mathematical Physics Department, Bauman Moscow State Technical University, 2-nd Baumanskaya, 5, Moscow, Russian Federation
E-mail: dimit@bmstu.ru, azaharov@bmstu.ru

Abstract. The paper covers the area of block-structured adaptive grid generation for finite-difference, finite-element and finite-volume methods. It deals with a problem of development of information technologies and algorithms for generating of such grids in complex geometries. Its key idea is to bring together the different existing techniques for computer-aided geometric design of domains, grid generation, and subdivision. The methods used for a construction of domain geometries and working with block-structured quasi-continuous grids are discussed. The results in the generation of three-dimensional adaptive grids for the domain of external flow near a hemisphere, flow around a surface of the high-speed aircraft of the Falcon HTV-2 type and for the domain of the gap between a wheel and a body of an aircraft are analyzed. The presented methods and algorithms for grid generation have the following advantages: 1. The grids are structured and mesh cells are hexahedra in the entire domain. 2. The grids can be applied for finite-difference methods. 3. The grid lines are adjusting with the flow in the whole domain.

1. Introduction

Development of unified computer-aided software packages for generating adaptive grids currently represents a large-scale problem of applied mathematics, computational physics, mechanics, biology and medicine, both in Russia and abroad [1, 2, 3]. Existing open-source and commercial software packages mainly use mesh generation methods that are universally applicable for complex geometry and focus on the generation of unstructured finite-element or finite-volume grids. Structuring and adaptation of a mesh, as well as the hexahedral shape of its cells, usually can be held only near the some boundaries of the domain. At the same time, for some problems of mathematical physics, it is important to hold the structuring and hexahedral shape of the mesh cells in the entire domain [4, 5, 6], the applicability of the mesh for finite-difference methods [5, 6, 7], geometric and dynamic adaptation of the mesh in the entire domain [8].

2. Construction of Domain Geometries

For the generation of block structured adaptive grids, the equations of boundary curves and surfaces of computational domains need to be written in parametric form. These equations are built on the spline interpolation method, which is based on a created grid of control points. The control points of the domain are located on the boundary surfaces and form the structured surface mesh.
The computational domain is constructed from a set of initial hexahedral blocks (primitives) by their combining and subsequent deformation. The deformation is performed by changing of coordinates of control points of the primitives. These coordinates can be read from a file.

It is possible to load the incoming information about geometric shapes of bodies from the solid simulation software. In this case the STL geometry definition file format supported by the most solid simulation programs is used. The functions for generation of points in given sections and along lines between the two specified points on a triangulated surface are implemented for the construction of the structured mesh of control points on the imported surfaces (Fig. 1). Then the predetermined thickness curvilinear blocks are generated based on data about the normals in cells. Fig. 1 illustrates the process of importing the geometry from STL file, view of the surface cubic splines and the construction of the curvilinear blocks.

### 3. Adaptive Grid Generation

A three-dimensional non-orthogonal coordinate system $X_i$, $i = 1, 2, 3$ is introduced to generate the adaptive mesh. In this coordinate system, the boundaries of each curvilinear block are the coordinate surfaces. To go to these coordinates, we use the explicit form of algebraic transformation, which refers to the Lagrangian coordinate transformation of transfinite interpolation methods [1, 2, 3, 9].

It is impossible to build an acceptable global transformation from a single block without singularities. Therefore, an approach for the grid generation is that the domains are generally six-sided and one-to-one matching at the boundaries. One current requirement for this approach is that the assignments of the parameterizations and coordinate directions at the boundary surfaces should be matched.

It is impossible to ensure the continuity of adaptive coordinate lines for some computational domains; therefore, the introducing of additional internal discontinuities is necessary. The mesh nodes on these internal discontinuities are generated twice. The parameters of each of these nodes correspond to the parameters of the adaptive coordinate system in which it was generated. The physical coordinates of these nodes are the same. Each node has a link to its twin. There are different classes of geometry topologies with internal discontinuities in the preprocessor (Fig. 2).

Let the coordinate lines $X^1$ and $X^2$ of different adaptive coordinate systems (green and red on the Fig. 2) are perpendicular to an internal discontinuity. If a derivative of a function at an
ordinary node \( j \) is approximated by the central differences

\[
\frac{\partial f}{\partial X^1} \approx \frac{f_{jR} - f_{jL}}{X_{jR}^1 - X_{jL}^1},
\]

the derivative at a node \( j \) on the internal discontinuity can be approximated for example as follows

\[
\frac{\partial f}{\partial X^1} \approx \frac{f_{jR} - f_j}{X_{jR}^1 - X_j^1} + \frac{f_j - f_{jD}}{X_{jT}^2 - X_{jD}^2},
\]

where \( j_R \) and \( j_L \) are the two neighbors of the node \( j \) in the direction \( X^1 \), \( j_D \) is the neighbor of the twin \( j_T \) of the node \( j \) in the direction \( X^2 \).

Results of a solution correspond to only one of the two twin nodes. The second node is a subsidiary and its parameters are used only for the calculations of the difference approximations and searching for the appropriate neighbors.

4. Examples of Grid Generation

Fig. 3 shows the different types of grids for the domain of external flow near a hemisphere. Fig. 3 (b)-(c) shows the possibility of converting the generated block structured hexahedral adaptive mesh into the finite-element or finite-volume grids. An essential feature of the resulting mesh in comparison with those obtained by means of the most open-source and commercial grid generators [10] is that the grid lines are adjusted with the flow in the whole computational domain.

Fig. 4 shows the block-structured adaptive mesh for the real complex curvilinear computational domains: the outer surface of the high-speed aircraft of the Falcon HTV 2 type and for the domain of the gap between a wheel and body of an aircraft.

5. Conclusions

The computer technologies to generate block-structured adaptive grids have been developed. They include the methods of computer-aided geometric design of domains, grid generation techniques, methods of working with block structured quasi-continuous grids for the certain types of curved domains. The generated grids can be applied for the finite-difference, finite-element or finite-volume methods.

The main advantages of the proposed technologies of adaptive grid generation in comparison with other approaches are the structuring and hexahedral shape of the mesh cells in the complex
curvilinear computational domains, the applicability of the mesh for finite-difference methods, the adjusting with the flow grid lines in the whole domain.

[1] Liseikin V D 1999 Grid Generation Methods (Heidelberg: Springer-Verlag)
[2] Shneiders R 1996 Proc. of 5th Int. Conf. on Numerical Field Simulations pp 699–708
[3] Farrashkhalvat M and Miles J P 2003 Basic Structured Grid Generation (Oxford: Butterworth-Heinemann)
[4] Kulik V V, Parkin A N and Navasardyan E S 2016 Chem. Petrol. Eng. 52 531–8
[5] Marchevskii I K and Puzikova V V 2017 J. Mach. Manuf. Reliab. 46 114–9
[6] Shumaev V V and Kuzenov V V 2017 J. Phys.: Conf. Series 891 012311
[7] Kuzenov V V, Dobrynina A O and Shumaev V V 2018 J. Phys.: Conf. Series 980 012023
[8] Kuzenov V V and Ryzhkov S V 2014 Appl. Phys. 3 26–30
[9] Eiseman P R 1992 Computers Math. Appl. 24 57–67
[10] Reznik S V, Mikhailovskiy K V and Prosuntsov P V MATEC Web Conf. (Tomsk) vol 92 (EDP Sciences) 01075