ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2019, Vol. 40, No. 11, pp. 1848–1852. © Pleiades Publishing, Ltd., 2019.

Approximate Methods of the Surface Mesh Deformation in Two-dimensional Case

A. A. Rybakov[1]* and S. S. Shumilin[1]**

(Submitted by A. M. Elizarov)

[1] *Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS)—Branch of Scientific Research Institute of System Analysis of the Russian Academy of Sciences, Moscow, 119334 Russia*

Received July 5, 2019; revised July 10, 2019; accepted July 15, 2019

Abstract—Numerical simulation of the surface ice accretion includes the work of various solvers that are performed iteratively and exchange data with each other. The calculation execution chain consists of the work of the gas-dynamic solver, the calculation of the liquid phase, the calculation of the thickness of the accreted ice on the surface grid and the rebuilding of the surface. After rebuilding is done, the modelling process goes to the next iteration in the gas-dynamic solver. Thus, the performance of a qualitative rebuilding of the surface computational grid taking into account the accumulated ice affects all further calculations. The article discusses approximate methods of rebuilding the surface mesh according to the ice accretion in each cell for the two-dimensional case and estimates their accuracy.

DOI: 10.1134/S1995080219110258

Keywords and phrases: *mesh deformation, displaced areas, gradient descent.*

1. INTRODUCTION

The task of calculating the ice accretion of aerodynamic surfaces is extremely relevant today. Of the most well-known software products that implement this functionality, we can emphasize such packages as FENSAP-ICE [1], LEWICE [2], CANICE [3], CHT2D [4] and many others. As a rule, these packages are multi-step and provide for iterative rebuilding of the surface mesh as the ice increases. Such rebuilding is required to be performed, as with the growth of ice, it is necessary to re-launch the aerodynamic solver to obtain actual data on the motion of the medium around the new deformed body [5, 6]. To obtain a more accurate profile of accreted ice, it is required to rebuild the surface mesh as often as possible, therefore, the speed of the mesh rebuilding algorithms is also significant.

Of particular interest is the development of algorithms for rebuilding surface grids in three-dimensional space. Among them there is a rebuilding algorithm with local saving of the cell volume [7], which found its implementation in the iceSurf tool [8]. However, work on the development of algorithms for rebuilding surfaces in two-dimensional space is also under way, and the results of these studies are more-less transferred to work in three-dimensional space, as demonstrated in [9].

In this paper, we have analyzed approximate methods for rebuilding a surface mesh only for the two-dimensional case; however, they can be generalized to three-dimensional objects. Further, the task of rebuilding the surface mesh is considered in an abstract form without reference to the problems of ice accretion.

*E-mail: rybakov.aax@gmail.com

**E-mail: shumilin@jscc.ru

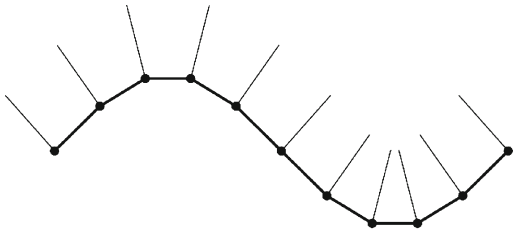

Fig. 1. Surface mesh with designated direction of movement of the nodes.

2. PROBLEM OF REBUILDING THE SURFACE IN TWO-DIMENSIONAL SPACE

Consider the geometric problem of rebuilding a surface in two-dimensional space in general form. Let n cells of the surface grid be given, each of which is represented by a segment of length l_i (that is, the total number of nodes is $n+1$). The direction of change of the surface of each cell is known (the direction of the normal to the segment), as well as the direction of movement of each node $\overline{g_i}$, $|\overline{g_i}| = 1$. It coincides with the direction of the sum of the unit normals drawn to the incident cells. Moreover, for a two-dimensional case, this direction lies on the bisector of the angle formed by two incident cells [10] (Fig. 1).

It is required to find such values of local shifts of grid nodes h_i that the covering area between the old surface and the new surface for each grid cell (S_i) differ as small as possible from the required value a $T_i = l_i H_i$.

To solve this problem, first, it's needed to calculate the covering area for each individual cell.

2.1. The task of Calculating the Covering Area when Moving Nodes of a Single Cell

Consider a cell represented on a plane by a segment AB of length l. When moving A and B to the new points A_1 and B_1 respectively quadrangle AA_1B_1B is formed. It is required to find its area expressed explicitly through parameters $a = |\overline{AA_1}|$ and $b = |\overline{BB_1}|$ (Fig. 2).

To solve the problem, we drop perpendiculars from the points A_1 and B_1 to a straight AB. Their projections will be points A_2 and B_2 respectively. The required area can be represented as follows:

$$S_{AA_1B_1B} = S_{A_2A_1B_1B_2} - S_{AA_1A_2} - S_{BB_1B_2}.$$

Denote the angle between the vectors $\overline{AA_1}$ and $\overline{AB}$ as α, and the angle between the vectors $\overline{BB_1}$ and $\overline{BA}$ as β. Then the required area is calculated explicitly in the following form:

$$S_{AA_1B_1B} = \frac{1}{2}(l - a\cos\alpha - b\cos\beta)(a\sin\alpha + b\sin\beta) + \frac{1}{2}a^2\sin\alpha\cos\alpha + \frac{1}{2}b^2\sin\beta\cos\beta,$$

$$S_{AA_1B_1B} = \frac{1}{2}\big(l(a\sin\alpha + b\sin\beta) - ab\sin(\alpha+\beta)\big).$$

2.2. Gradient Descent Solution

The gradient descent method is one of the simplest optimization method for finding the local minimum of a function. Provided that at any point of the function it is possible to calculate its gradient, then an iteration sequence is constructed starting from some initial approximation x_0 [11]: $x^{k+1} = x^k - \gamma_k \nabla f(x_k)$, where $\gamma_k \geq 0$ sets the step length and, respectively, the rate of gradient descend.

The gradient method finds its main application in the task of finding the minimum or maximum of a function. The direction of the anti-gradient is the direction of the fastest decreasing of the function. The main problem of the method is to choose the step γ. For large values of the step, there is a chance to "jump over" the minimum of the function. In addition, the method does not guarantee finding a global minimum.

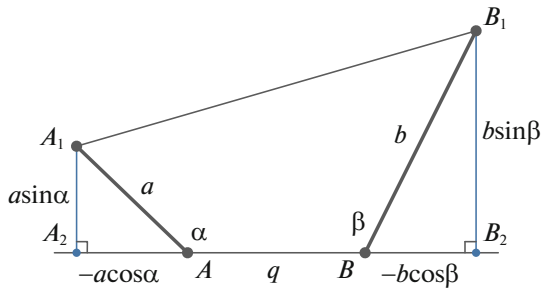

Fig. 2. Calculation of the covering area when moving the nodes of the cell.

Consider the solution of the problem by the method of gradient descent. The unknown parameters are the magnitudes of the shifts of the nodes of the grid h_i. Based on the solution of the local problem of determining the covering area, we can define the covering area when a separate cell moves as:

$$S_i = \frac{1}{2}\big(l_i(h_i \sin\alpha_i + h_{i+1}\sin\beta_i) - h_i h_{i+1}\sin(\alpha_i + \beta_i)\big).$$

The deviation of the covering area in the cell from the true value will be called the value $\delta_i = S_i - T_i$, and its error is its square $d_i = \delta_i^2$. The total error when rebuilding the surface is defined as the sum of errors for all cells: $D = \sum_{i=0}^{n-1} d_i$. When finding the optimal solution, it is required to minimize the total error. To find the gradient, it is required to calculate the partial derivatives of the function D over all unknowns h_i. These derivatives can be written explicitly: $\frac{\partial D}{\partial h_i} = \frac{\partial d_{i-1}}{\partial h_i} + \frac{\partial d_i}{\partial h_i}$, where

$$\begin{cases} \frac{\partial d_{i-1}}{\partial h_i} = \delta_{i-1}(l_{i-1}\sin\beta_{i-1} - h_{i-1}\sin(\alpha_{i-1} + \beta_{i-1})), \\ \frac{\partial d_i}{\partial h_i} = \delta_i(l_i \sin\alpha_i - h_{i+1}\sin(\alpha_i + \beta_i)). \end{cases}$$

Also, when implementing the gradient descent method, it is required to monitor the compliance of additional conditions that are imposed on the unknown h_i. For example, an obvious condition is that $h_i \geq 0$ is satisfied, which prevents the grid from moving in a negative direction. In this work, more stringent conditions $\min\{H_{i-1}, H_i\} \leq h_i \leq \max\{H_{i-1}, H_i\}$ were used, which do not allow the values of the displacement of the grid nodes to go beyond the limits of the displacements of the grid cells incident to them.

3. APPROXIMATE SOLUTION SCHEMES

Solving the problem of rebuilding the grid using the gradient descent method is too resource-demanding as the grid size increases. In addition, the quality of the solution is often unsatisfactory, especially when hit in local minima. Therefore, to solve the problem, methods of approximate solution were proposed, based on approximation of the solution in each cell using primitive geometric shapes.

3.1. Solution by Rectangles Method

As a first method, we consider an approximation in which each grid node is shifted by a vector $\frac{1}{2}(H_{i-1} + H_i)\overline{g_i}$. This method corresponds to the approximation of the solution in each cell with a rectangle on the sides of l_i and H_i, and then averaging, as shown in Fig. 3.

It is worth noting the possible development of this approach through the use of multilayer approximation, as described in [12], however, this opportunity was not considered in this paper.

3.2. Solution by Trapezoid Method

In the trapezoid method, the solution in each cell is approached by a trapezoid with an area of T_i, the sides of which lie on the growth directions of two nodes belonging to the considered cell. After constructing the trapezoid for all grid cells, each internal node has two new potential positions for the shift (formed by the cell on the left and the cell on the right). Their average value is chosen as the final position (Fig. 4).

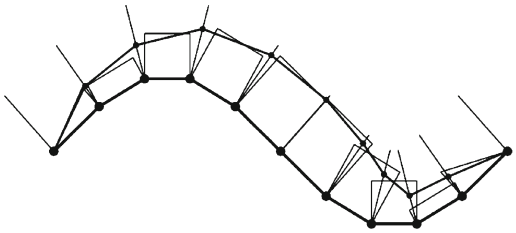

Fig. 3. Surface rebuilding using the rectangle method.

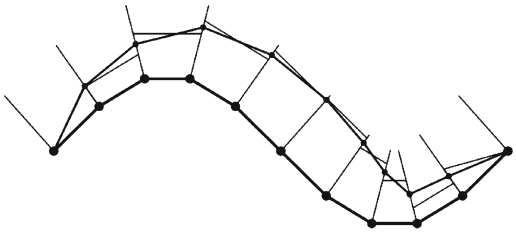

Fig. 4. Surface rebuilding by trapezoid method.

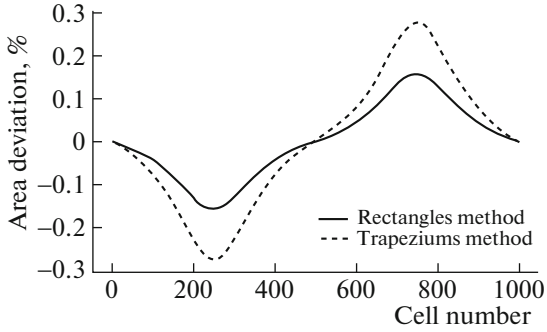

Fig. 5. Comparison of the accuracy of solutions using the rectangle method and the trapezoid method.

3.3. Comparison of Solution Accuracy

To compare the accuracy of the solutions obtained using the methods described, we used a model two-dimensional surface grid represented by a single sinusoid period ($x \in [0, 2\pi]$). As a set of cell shifts (H_i), the same shifts equal to half of cell size were used. With an increase in the number of nodes, both approximate methods demonstrated values $\frac{D}{\sum_i T_i}$ tending to zero with minor deviations from each other and from the gradient descent method that was used for verification. A comparison of the values of δ_i for all cells for the proposed approximate methods was also carried out. The results of the comparison on the model grid with the number of nodes $n = 1000$ are shown Fig. 5.

It can be seen from this graph that the simpler method of rectangles is at the same time more accurate, since it provides smaller deviations from the exact solution on strongly convex and strongly concave sections of the grid.

4. CONCLUSION

Two simple approximate methods for rebuilding a surface mesh in two-dimensional space were considered. Comparison of the results of their work with a locally optimal solution obtained using the iterative gradient descent method showed the inexpediency of using the latter for industrial calculations. Of the two approximate rebuilding methods, the rectangle method demonstrated smaller deviations of the solution for individual grid cells in places of strong grid curvature. The considered approximate methods can be extended to the three-dimensional case, but this is beyond the scope of this article.

FUNDING

The work was done at the JSCC RAS as part of the state assignment for the topic 0065-2019-0016 (reg. no. AAAA-A19-119011590098-8). The supercomputer MVS-10P, located at the JSCC RAS, was used for calculations during the research.

REFERENCES

1. Y. Bourgault, H. Beaugendre, and W. G. Habashi, "Development of a shallow-water icing model in FENSAP-ICE," J. Aircraft **37**, 640–646 (2000).
2. C. Bidwell, D. Pinella, and P. Garrison, "Ice accretion calculations for a commercial transport using the LEWICE3D, ICEGRID3D and CMARC programs," AIAA Paper 99-0250 (AIAA, Reno, 1999).
3. A. Pueyo, D. Chocron, and F. Kafyeke, "Improvements to the ice accretion code CANICE," in *Proceedings of the 8th CASI Aerodynamics Symposium, Toronto, 2001*.
4. A. Pueyo, D. Chocron, F. Mokhtarian, and F. Kafyeke, "CHT2D: A 2D hot air anti-icing analysis tool," in *Proceedings of the 50th Annual General Meeting (AGM) and Conference, Canadian Aeronautics and Space Institute, 2003*.
5. S. Özgen and M. Canibek, "Ice accretion simulation on multi-element airfoils using extended Messinger model," Heat Mass Transfer **45**, 305–322 (2009).
6. H. Beaugendre, "A PDE-based 3D approach to in-flight ice accretion," PHD Thesis (McGill Univ., Montreal, QC, 2003).
7. J. Jiao, "Volume and feature preservation in surface mesh optimization," in *Proceedings of the 15th International Meshing Roundtable, Birmingham, AL* (Springer, 2006), pp. 359–374.
8. D. Thompson, X. Tong, Q. Arnoldus, E. Collins, D. McLaurin, and E. Luke, "Discrete surface evolution and mesh deformation for aircraft icing applications," in *Proceedings of the 5th AIAA Atmospheric and Space Environments Conference,* AIAA Paper 2013-2544 (2013).
9. A. Pueyo, "Efficient 3D artificial ice shapes simulations with 2D ice accretion codes using a 3-level correction," in *Proceedings of the SAE 2013 AeroTech Congress and Exhibition,* SAE Technical Paper 2013-01-2136 (2013), Vol. 7.
10. G. Fortin, A. Ilinca, J.-L. Laforte, and V. Brandi, "New roughness computation method and geometric accretion model for airfoil icing," J. Aircraft **41**, 119–127 (2004).
11. G. P. Akilov and L. V. Kantorovich, *Functional Analysis,* 2nd ed. (Pergamon, Oxford, 1982).
12. S. Bourgault-Côté, K. Hasanzadeh, P. Lavoie, and E. Laurendeau, "Multi-layer icing methodologies for conservative ice growth," in *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences EUCASS, 2017*.