Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0085

Vol. **21**, No. 5, pp. 1282-1309
May 2017

Delaunay Graph Based Inverse Distance Weighting for Fast Dynamic Meshing

Yibin Wang[1,*], Ning Qin[2] and Ning Zhao[1]

[1] *College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, 29 Yudao Road Nanjing 210016, China.*
[2] *Department of Mechanical Engineering, The University of Sheffield, Sheffield S1 3JD, UK.*

Received 12 June 2016; Accepted (in revised version) 9 October 2016

Abstract. A novel mesh deformation technique is developed based on the Delaunay graph mapping method and the inverse distance weighting (IDW) interpolation. The algorithm maintains the advantages of the efficiency of Delaunay graph mapping mesh deformation while it also possesses the ability of better controlling the near surface mesh quality. The Delaunay graph is used to divide the mesh domain into a number of sub-domains. On each sub-domain, the inverse distance weighting interpolation is applied, resulting in a similar efficiency as compared to the fast Delaunay graph mapping method. The paper will show how the near-wall mesh quality is controlled and improved by the new method.

AMS subject classifications: 65D18, 68U05, 68w99

Key words: Dynamic mesh, inverse distance weighting, Delaunay graph mapping.

1 Introduction

Aerodynamic shape optimization, flapping wing and dynamic aeroelastic fluid-structure interaction problems require the interface or boundary to deform during the solution process. In order to propagate the displacement of the boundary into the solution domain, accurate and efficient meshing techniques are needed. There are mainly three types of methods to handle such problems, namely re-mesh, overset mesh and dynamic mesh techniques. The re-mesh techniques regenerate the mesh to account for the deformation of the interface, hence the solution needs to be interpolated from the old mesh to the new mesh. For the aforementioned problems, the mesh needs to be frequently updated,

*Corresponding author. *Email addresses:* Yibin.wang@nuaa.edu.cn (Y. Wang), n.qin@sheffield.ac.uk (N. Qin), zhaoam@nuaa.edu.cn (N. Zhao)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

which makes the re-mesh approach very time-consuming; furthermore the interpolation between two different meshes also causes extra error for the numerical solution. The overset mesh method is now widely used in simulation of the rigid body motion when large body movement occurs. The dynamic mesh methods use certain techniques to move the mesh nodes according to the deformation. It is generally more efficient than the re-mesh method and the solution interpolation is avoided. The two key issues of this method are the efficiency and mesh quality of the resultant mesh. During the shape deformation process, the mesh needs to be renewed frequently, especially in the dynamic fluid-structure interaction simulation the mesh needs to be updated for each time step, and therefore a fast dynamic mesh technique can efficiently decrease the computational time. The accuracy of the CFD simulation largely relies on the mesh quality, particularly the mesh quality near the wall boundary, since in this region a poor mesh can result in large truncation errors. As a result, high quality and high efficiency are the most important aspects of the dynamic mesh techniques.

Dynamic mesh techniques can be generally classified into two main categories, based on either physical analogy or interpolation [1]. The physical analogy approach, such as the spring analogy approach [2], normally requiring the connectivity information of the mesh, uses certain physics processes to propagate the mesh deformation from the boundary to the solution domain. Though this method has been successfully applied to many unsteady and optimization problems, it is a relatively expensive approach due to the necessary iterations for solving the associated spring systems. For large deformation, it may lead to invalid mesh cells. Farhat *et al.* [3] introduced the torsional springs to prevent the mesh from becoming invalid which greatly improves the robustness of the method. By solving a set of partial differential equations, Loehner and Yang [4,5] developed a method based on elastic analogy. Particularly, by solving a bi-harmonic set of equations, Helenbrook [6] found that both mesh quality and orthogonality are superior to the Laplace equation based method. In general, this type of method is computationally expensive and not suitable for the large deformation problem.

An interpolation method, by applying some interpolation schemes, directly obtains the displacement or the new coordinates of each node. The radial basis function (RBF) method, proposed by de Boer [7], is able to maintain the mesh quality near the boundary. However it needs to solve three large matrices depending on the node number on the surfaces. For practical 3D problems, it becomes very inefficient. Rendall and Allen [8] proposed an approximate RBF method with a data reduction algorithm. It effectively improves the efficiency of the RBF method. On the other hand, this method also introduces surface point mismatching errors. In order to solve this problem, they added a surface correction step [9]. Witteveen and Bijl [10] proposed a direct interpolation strategy by using inverse distance weighting (IDW) interpolation, which is faster than the original RBF method. Luke *et al.* [1] proposed a fast explicit interpolation method based on IDW method, which has a similar mesh quality as the RBF method, but with a relatively faster speed. For large practical 3D problems, these methods still involve substantial computational cost. Liu et al. [11] developed a dynamic mesh method based on Delaunay graph

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

mapping. Van der Burg *et al.* [12] applied it to the aeroelastic problem in a high lift study, in which the mesh deformation could be done "in a couple of minutes" for a large 3D problem with 7 million tetrahedral cells using the Delaunay graph mapping method. Though it is a fast interpolation scheme, this method cannot preserve well the mesh quality near the boundary for large deformation. Recently, Wang *et al.* [13] developed a new method, which combines the Delaunay graph mapping method and the RBF method. It manages to maintain the high efficiency of the Delaunay graph mapping method, while significantly improves the mesh quality for large deformation; especially the mesh quality near the wall is well preserved.

Recently some hybrid methods were developed, which combine the physical analogy and interpolation methods [14–17] to reduce the computational cost and maintain the mesh quality. Gopalakrishnan [18] developed a hybrid method which combines spring analogy and transfinite interpolation. Zhou and Li [19] developed a 2D dynamic mesh method based on disk relaxation. Recently they further improved the method, and successfully applied to 3D cases [20]. These methods show similar high quality mesh as the RBF method and the IDW method. The efficiency is not as good as the fast interpolation methods such as the Delaunay graph mapping method.

A novel method of dynamic mesh based on Delaunay graph and inverse distance weighting (IDW) is presented in this paper. This method is mainly based on the basic framework of the Delaunay graph mapping method (DGM) [11], by introducing the inverse distance weighting interpolation into the last mapping step of DGM, which acquires the ability to control the quality of the meshes near the boundary. Several test cases were compared with the RBF method, the IDW method and the Delaunay graph mapping method, regarding both mesh quality and CPU time. This new method maintains the efficiency of the original Delaunay graph mapping method, while it also shows high near wall mesh quality comparable to the RBF and IDW methods.

2 Delaunay graph based inverse distance weighting interpolation method (DG-IDW)

2.1 Delaunay graph mapping method

In many CFD applications such as fluid structure interaction, aeroelastic problems, flapping wing simulations and aerodynamic optimization, the Delaunay graph mapping method [11] has been widely used, for its high efficiency and easy implementation. In bio-fluid simulation [21], flapping wing simulation [22] and shock control aerodynamic shape optimization [23], it efficiently deformed the mesh with quality cells for moderate deformation. However, for large deformation, especially when the Delaunay graph becomes invalid after deformation, the mesh quality becomes invalid. In addition, due to the lack of mesh quality control, the mesh quality near the wall may quickly degrade as the mesh deforms significantly.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

2.2 Delaunay graph mapping method

The inverse distance weighting interpolation method was first developed by Witteveen and Bijl [10]. It is an explicit method for multivariate interpolation of scattered points. By using the known displacement of the boundary nodes, the displacement of the interior nodes can then be calculated by interpolation. The boundary nodes are generally classified into two categories, one is static boundary normally referring the boundary without any deformation; the other is deformed boundary referring the boundary involving deformation. The displacement of the interior nodes can be calculated by

$$\mathbf{s(x)} = \frac{\sum_{i=1}^{n_d} s_i {r_i}^{-c_d}}{\sum_{i=1}^{n_d} {r_i}^{-c_d} + \sum_{i=1}^{n_s} {r_i}^{-c_s}}, \tag{2.1}$$

where $\mathbf{s(x)} = [dx,dy,dz]^T$ the displacement of an interior node, x is the vector of coordinate, n_d is the total number of deformed boundary nodes, n_s is the total number of static boundary nodes, r_i is the Euclidian distance between the interior node and boundary node i. c_d and c_s are the user defined shape parameters for deformed boundary and static boundary. $\mathbf{s_i} = [dx_i,dy_i,dz_i]^T$ is the displacement of a boundary node i. Thus the coordinates of the interior nodes can be calculated

$$\mathbf{x_{new}} = \mathbf{x_{old}} + \mathbf{s(x)}. \tag{2.2}$$

For different motions, c_d and c_s can be chosen differently. The resultant mesh quality depends highly on the shape parameters. The proper setting of these parameters varies from case to case and, therefore, it is difficult for beginners to properly choose the parameters. In addition, it requires summing up all the weights of surface nodes for each mesh point, resulting in a significant computational cost. The motivation of this paper is to develop a novel method to significantly improve the efficiency of the IDW method.

2.3 The procedures of Delaunay graph based IDW method (DG-IDW)

The procedure of the Delaunay graph based IDW method (DG-IDW) is similar to the original Delaunay graph mapping method (DGM). The main difference is in the last step, where the IDW function is used to map the nodes rather than using the area or volume ratios as in the DGM. The algorithm is set out in the following procedure:

(a) Generating the Delaunay graph by using all the boundary nodes of the original mesh (the original configuration and mesh was shown in Fig. 1 and Fig. 2 respectively, the resultant Delaunay graph was shown in Fig. 3).

(b) Locating the mesh points in the graph, namely finding out which Delaunay triangle contains which mesh node, in Fig. 4. It is noted that the first two steps (a) and (b) are only needed once at the beginning of the unsteady simulation, though the mesh needs to be renewed repeatedly. This is quite different from the original DGM

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

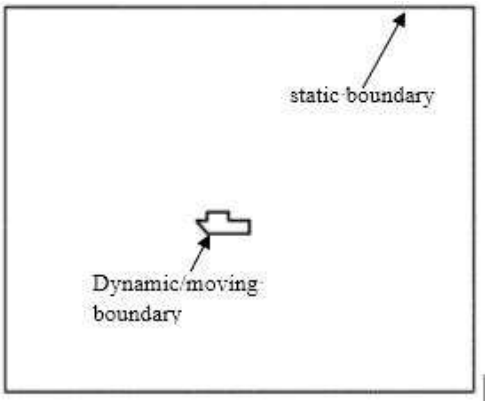


Figure 1: Original configuration.

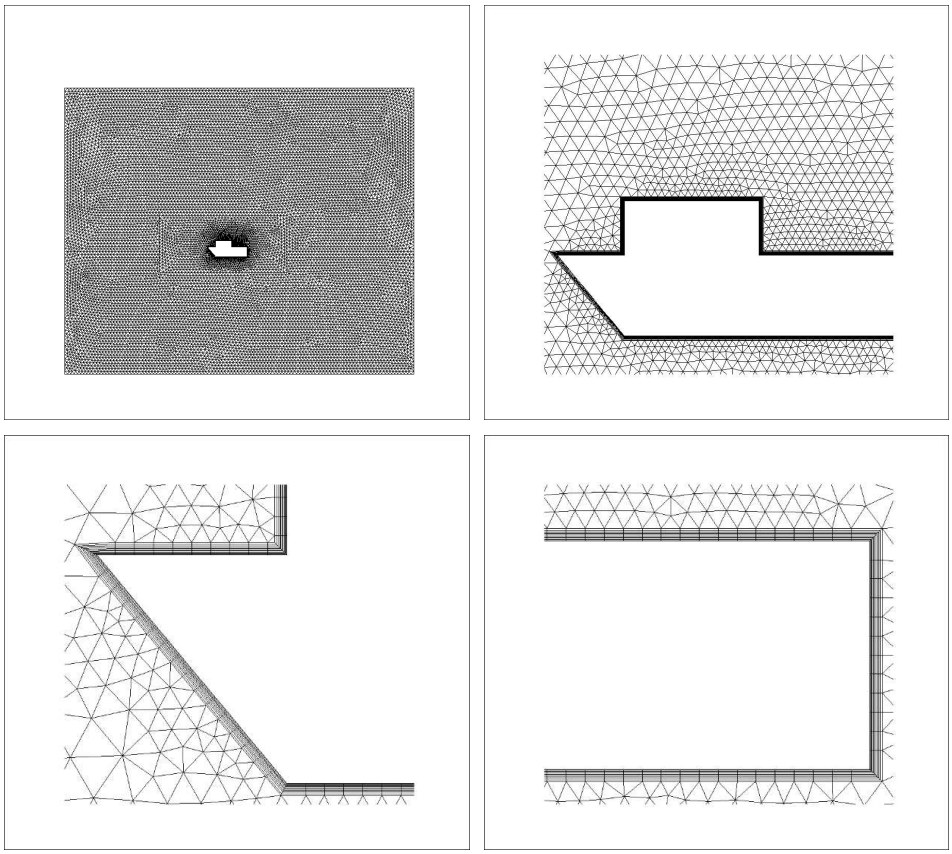


Figure 2: Original mesh.

method which requires generating the Delaunay graph repeatedly during the unsteady simulation process. This will be further discussed in Section 3.1.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

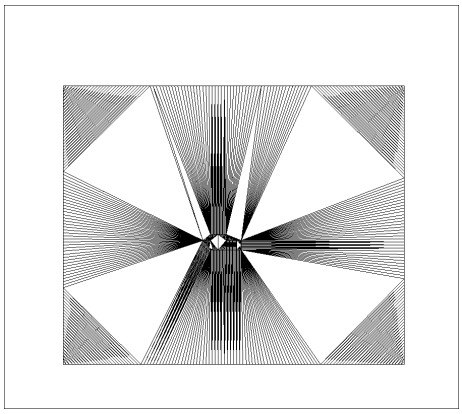


Figure 3: Original Delaunay graph.

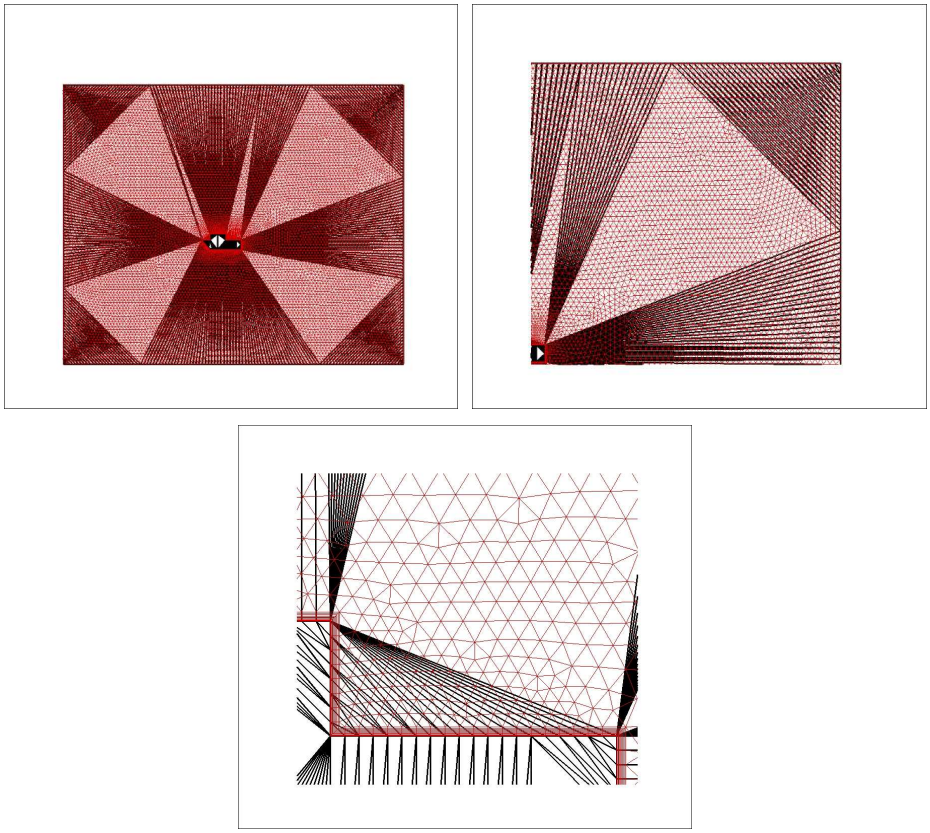


Figure 4: Identification of the nodes in the Delaunay graph.

(c) Moving the Delaunay graph according to the specified geometric motion/deformation, in Fig. 5.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

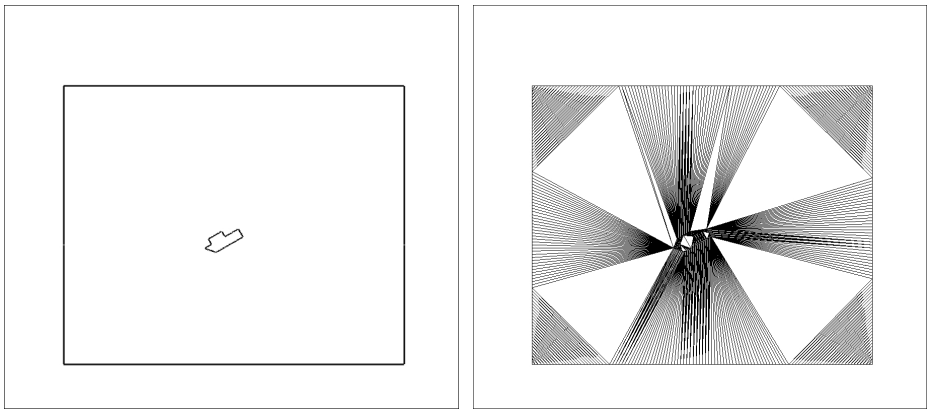


Figure 5: Deformed configuration and Delaunay map.

(d) Mapping the mesh points in each Delaunay triangle according to the IDW interpolation. The displacement and Euler angle of each node in a Delaunay triangle is

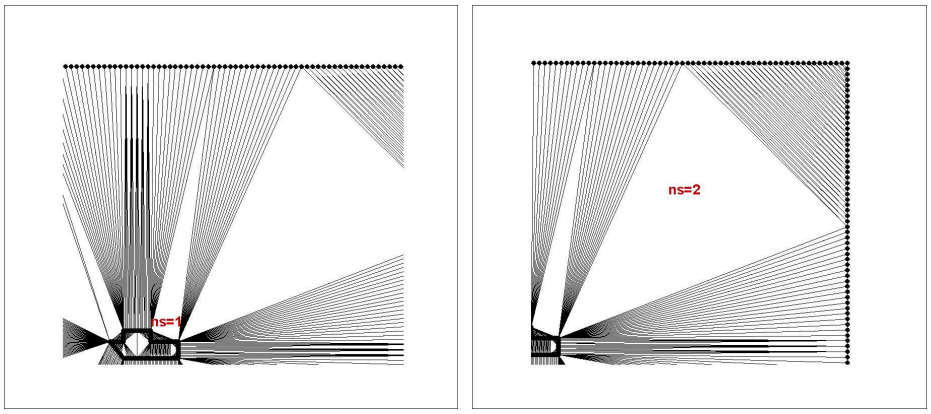
$$\mathbf{s}(\mathbf{x})=\begin{bmatrix} dx \\ dy \\ \gamma \end{bmatrix}=\begin{cases} \dfrac{a\sum_{i=1}^{n_d} s_i {r_i}^{-2}}{a\sum_{i=1}^{n_d} {r_i}^{-2}+2{r_i}^{-1}}, & n_s=1,\ n_d=2, \\ \dfrac{2as_d {r_d}^{-2}}{2a{r_d}^{-2}+\sum_{i=1}^{n_s} {r_i}^{-1}}, & n_s=2,\ n_d=1, \\ \dfrac{a\sum_{i=1}^{n_d} s_i {r_i}^{-2}}{a\sum_{i=1}^{n_d} {r_i}^{-2}+\sum_{i=1}^{n_s} {r_i}^{-1}}, & \text{else.} \end{cases} \tag{2.3}$$

For 3D it is

$$\mathbf{s}(\mathbf{x})=\begin{bmatrix} dx \\ dy \\ dz \\ \alpha \\ \beta \\ \gamma \end{bmatrix}=\begin{cases} \dfrac{a\sum_{i=1}^{n_d} s_i {r_i}^{-2}}{a\sum_{i=1}^{n_d} {r_i}^{-2}+3{r_i}^{-1}}, & n_s=1,\ n_d=3, \\ \dfrac{3as_d {r_d}^{-2}}{3a{r_d}^{-2}+\sum_{i=1}^{n_s} {r_i}^{-1}}, & n_s=3,\ n_d=1, \\ \dfrac{a\sum_{i=1}^{n_d} s_i {r_i}^{-2}}{a\sum_{i=1}^{n_d} {r_i}^{-2}+\sum_{i=1}^{n_s} {r_i}^{-1}}, & \text{else,} \end{cases} \tag{2.4}$$

where dx, dy, dz are the displacement due to the translation and shape deformation, α, β, γ are Euler angles; where $a=[7,7,1.4]^T$ is for 2D, and $a=[7,7,7,1.4,1.4,1.4]^T$ is for 3D. According to the numerical tests, the above setting can generally preserve the mesh quality for all the test cases in this paper. Users can adjust these parameters to control the size of the region around the moving body, and in which the high quality mesh is well preserved. Larger value of a means larger region of mesh around the moving body has a high quality mesh, but with a possible penalty of lower minimal mesh quality; while smaller value of a means smaller region of mesh around the moving body is well preserved but with a higher minimal mesh quality. User can base on the flow feature to adjust this parameter to control the size of the region. **s(x)** is the Euler angle and the displacement of the interior nodes due to both translation and deformation, s_i is the displacement and Euler angle of a moving boundary node

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



(a) one node on the static boundary (b) two nodes on the static boundary

(c) three nodes on the static boundary (d) no node on the static boundary

Figure 6: Four types of Delaunay triangles.

i in Delaunay triangle or tetrahedron. s_d is the displacement and Euler angle of the single moving boundary node in Delaunay triangle or tetrahedron. n_d is the total number of dynamic boundary nodes of a Delaunay triangle or tetrahedron, n_s is the total number of static boundary nodes of a Delaunay triangle or tetrahedron (for 2D, there are four different types of Delaunay triangles as shown in Fig. 6), r_i is the Euclidian distance between the interior node and boundary node i. r_s is the distance from the current position x to the single static boundary node of a Delaunay triangle/tetrahedron; while r_d is the distance from the current position to the single moving boundary node of a Delaunay triangle/tetrahedron. Based on the displacement and Euler angle the new position of the interior mesh nodes can be directly calculated

$$\mathbf{x}=\begin{bmatrix}x_0\\y_0\\z_0\end{bmatrix}+\begin{bmatrix}dx\\dy\\dz\end{bmatrix}+\begin{bmatrix}\cos\gamma & \sin\gamma & 0\\-\sin\gamma & \cos\gamma & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\0 & \cos\beta & \sin\beta\\0 & -\sin\beta & \cos\beta\end{bmatrix}\begin{bmatrix}\cos\alpha & \sin\alpha & 0\\-\sin\alpha & \cos\alpha & 0\\0 & 0 & 1\end{bmatrix}, \quad (2.5)$$

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

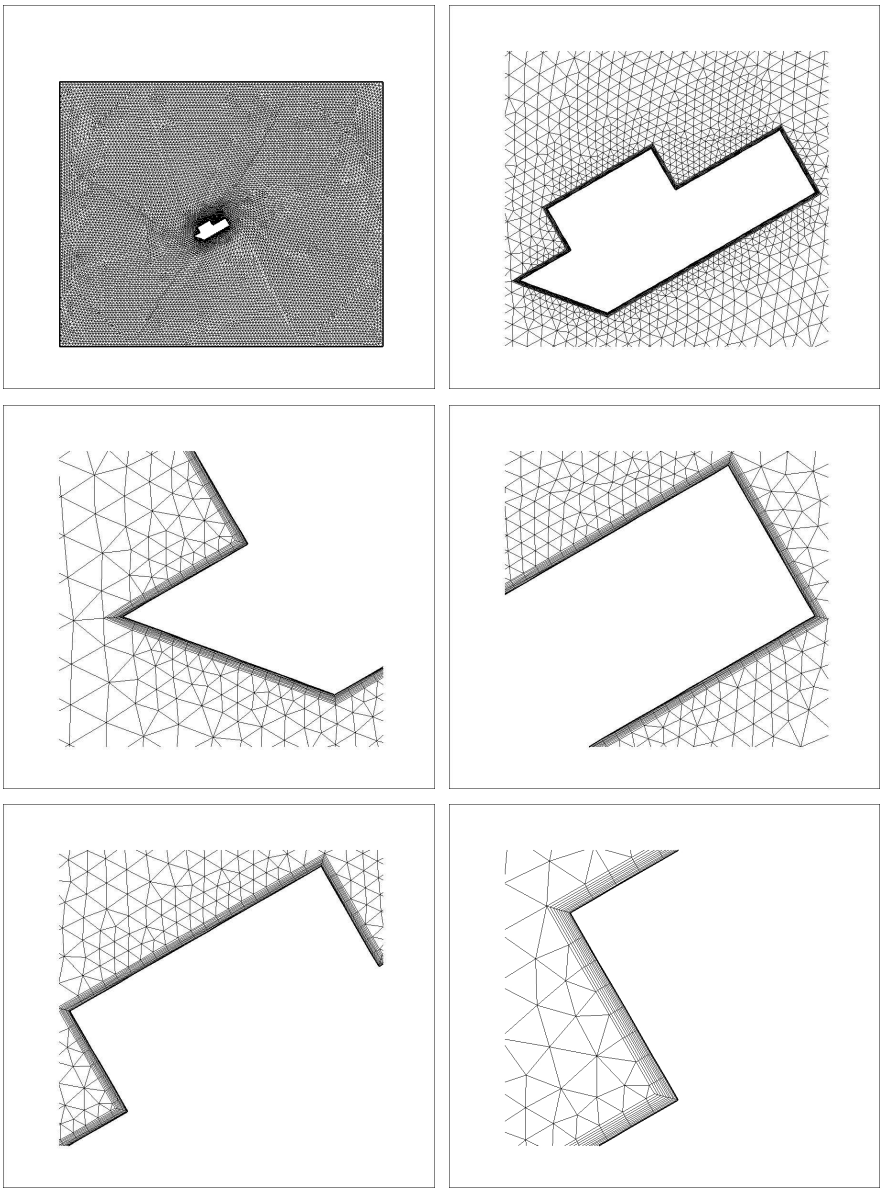


Figure 7: Deformed mesh.

where $\begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ is the original coordinates. In Fig. 7 the deformed mesh was shown where the meshes around the moving boundary were well preserved.

Steps 1-3 are exactly the same as the original Delaunay graph mapping method; hence the details of these steps are skipped in this paper. More details can be found in [8]. In Step 4, the IDW interpolation is used to calculate the displacement (or Euler angle) of the

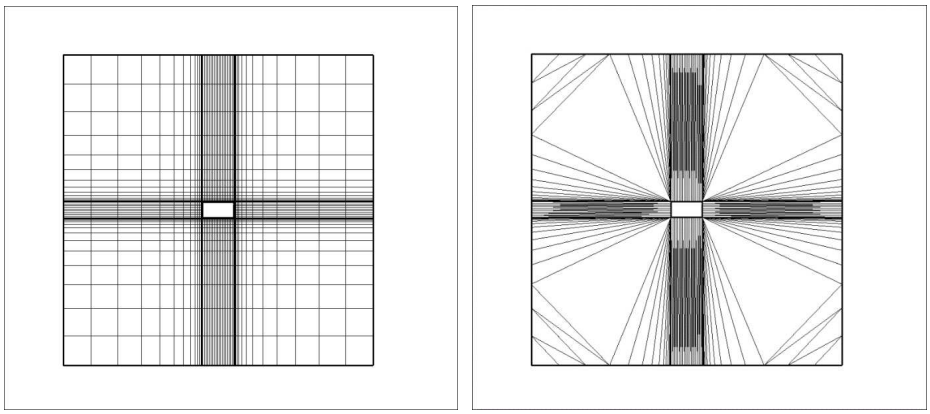
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

internal mesh nodes from the given displacement (or Euler angle) of the Delaunay triangle nodes on the boundary, while the original Delaunay mapping method uses surface or volume ratios to calculate the displacement of inner nodes.

3 Results and discussions

3.1 Test case 1: rotation and translation

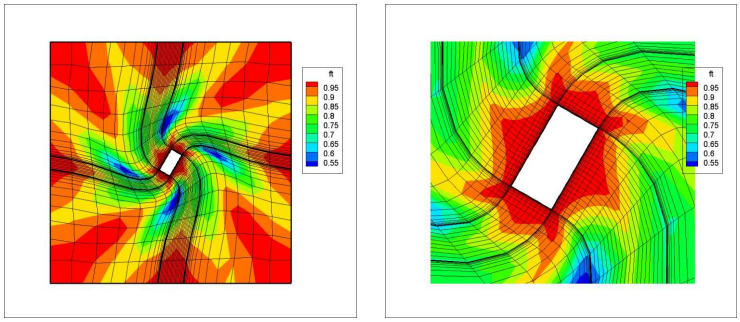
The first test case is a structured mesh containing a rectangle object with 2340 nodes, shown in Fig. 8(a). the original Delaunay graph was shown in Fig. 8(b). In this paper, all the performances are measured on a single core 2.7GHz Intel i7 processor. The inner rectangle was rotated around its center by 30°, 45°, 60°, 90°, respectively. For all the test cases in this paper, the meshes were generated by only one mapping for fair comparison. The mesh quality is compared with the Delaunay graph mapping method (DGM), the inversed distance weighting method (IDW) and the radial basis function method (RBF) in Tables 1 and 2. The detailed definition of mesh quality metric can be found in [24]. ft$=1$ represents the best mesh quality and ft$=0$ the worst. As can been seen from the tables, the IDW method gave the best mesh quality for both averaged and minimal mesh quality; while the Delaunay graph based inversed distance weighting method (DG-IDW) shows second best averaged and minimal mesh quality for most of the cases. In Table 3, the CPU time of the four methods were compared. The DGM and the DG-IDW show the best efficiency, the CPU time of the two methods was 0.001s, while the CPU time of the IDW is one order larger than the DGM and the DG-IDW methods; and the RBF method is two orders larger than the DGM and the DG-IDW methods. In Fig. 9, the resultant meshes generated by these four methods are shown, with 45° rotation. As shown in the figures, only the DGM method failed to keep the mesh quality near the wall, while the



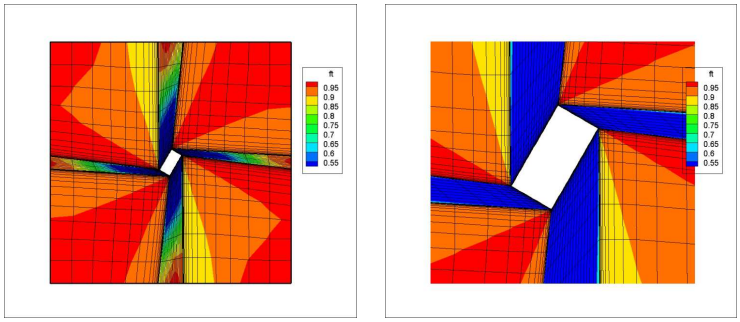
(a) (b)

Figure 8: (a) Original mesh; (b) original Delaunay graph.

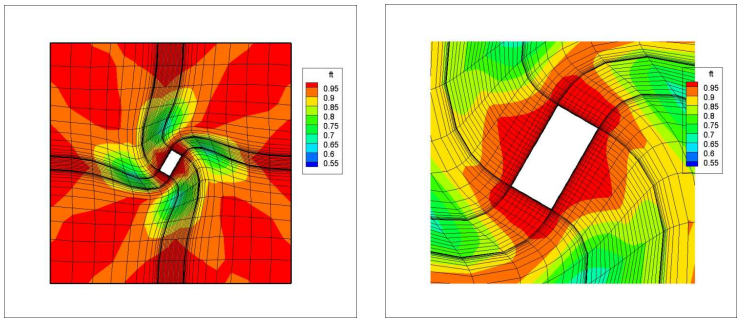
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



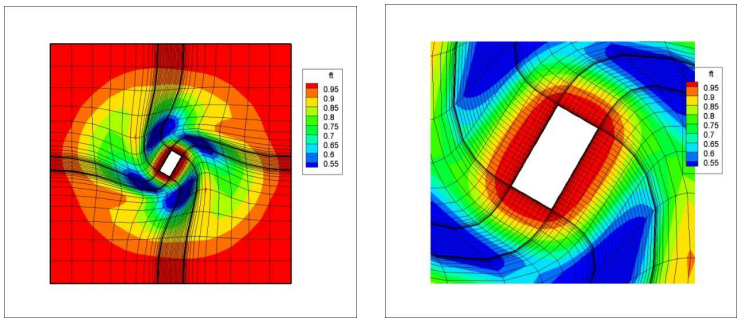
(a) DG-IDW



(b) DGM



(c) IDW



(d) RBF

Figure 9: The deformed mesh by different methods (rotation by 45°).

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

Table 1: Averaged mesh quality.

	DGM	IDW	RBF	DG-IDW
30°	0.885	0.970	0.954	0.952
45°	0.781	0.946	0.907	0.920
60°	0.676	0.919	0.851	0.884
90°	-	0.863	0.739	0.814

Table 2: Minimal mesh quality.

	DGM	IDW	RBF	DG-IDW
30°	0.684	0.882	0.784	0.784
45°	0.408	0.768	0.580	0.627
60°	0.159	0.641	0.363	0.469
90°	-	0.396	0.356	0.208

Table 3: CPU time.

	DGM	IDW	RBF	DG-IDW
30°	0.001	0.015	0.065	0.001
45°	0.001	0.016	0.062	0.001
60°	0.001	0.016	0.062	0.001
90°	-	0.016	0.063	0.001

rest of the methods all maintained the mesh quality in this region. In Tables 1 and 2, it is shown that the DGM method fails to give the valid mesh for 90-degree rotation, this is because the Delaunay graph has become invalid after moving (in Fig. 10), therefore for the DGM method the Delaunay graph needs to be regularly checked and regenerated for the large deformation. However, for the DG-IDW method which used the same Delaunay graph still generated valid meshes as the IDW and the RBF methods. It is because the DGM method uses the Delaunay triangle to move the volume mesh nodes, and these nodes must stay inside the triangle after each mapping. When the Delaunay triangles overlap with each other, the mesh becomes invalid. The DG-IDW method uses the Delaunay graph to divide the volume mesh nodes into different groups, and then applies the IDW interpolation in each group. This can greatly decrease the computational cost of the original IDW method. In addition, the volume mesh nodes are not confined inside the Delaunay triangle any more, thus the validity of the Delaunay graph after moving becomes irrelevant. Furthermore, different from the DGM method, the DG-IDW method only needs to generate the Delaunay graph once at the beginning of the simulation to divide the original mesh into groups.

For the same configuration and initial mesh, the inner rectangle is moved 2 units upwards and to the right. The mesh qualities are compared in Table 4. For the translation,

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

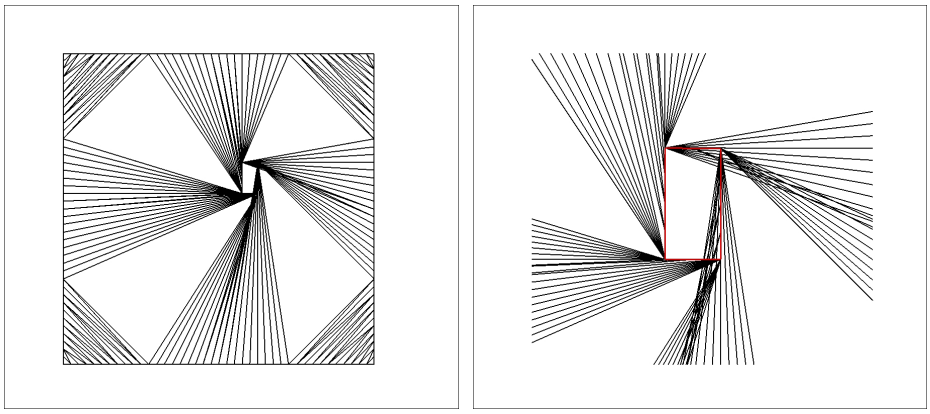


Figure 10: Invalid Delaunay graph after 90-degree rotation.

the DGM method showed the best minimal mesh quality, while its averaged mesh quality is second worst; the RBF method gave high averaged mesh quality but with worst minimal mesh quality; both average and minimal mesh qualities for the IDW method are ranked at the third place. The DG-IDW method shows the best averaged mesh quality and the second best minimal mesh quality. Therefore the DG-IDW method shows better properties for the translation motion. In Fig. 11, the four resultant meshes are shown with the mesh quality contours. Both the RBF and the DG-IDW methods maintain the mesh quality near the moving boundary, while the DGM and IDW methods failed to maintain the mesh quality in the same region. The CPU time are also compared in Table 4, in which the DG-IDW and the DGM show again the highest efficiency.

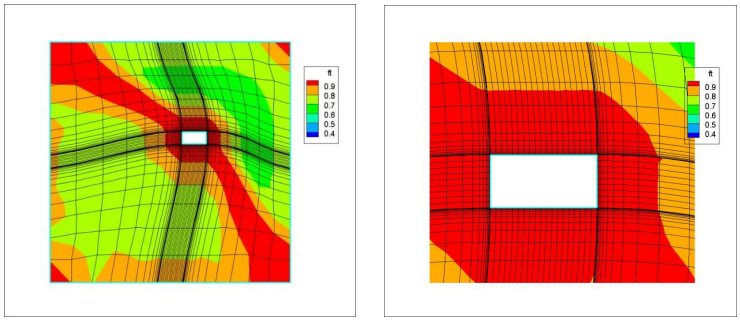
Table 4: Mesh quality and CPU time comparison.

	DGM	IDW	RBF	DG-IDW
averaged	0.792	0.777	0.860	0.881
minimal	0.649	0.422	0.223	0.591
CPU time	0.001	0.033	0.107	0.001

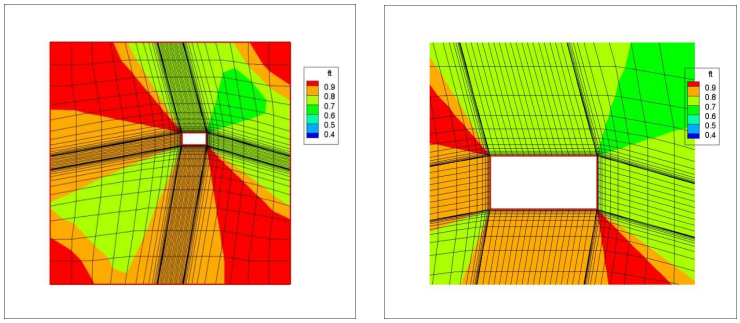
3.2 Test case 2: unstructured mesh for NACA0012 airfoil

An unstructured mesh with 1647 nodes and 3054 cells for NACA0012 airfoil was used as the second test case. The original mesh and its Delaunay graph are shown in Figs. 12 and 13. In order to test the capability of the four methods, a sliver cell is put right behind the trailing edge, as shown in Fig. 13. Any slight degradation of the mesh quality may cause the cell to become invalid. The airfoil is moved 2 units upwards and to the right. The deformed meshes are shown in Figs. 14 and 15. As can been seen form the figures, the RBF, DG-IDW and IDW methods maintain the mesh quality near the airfoil, while

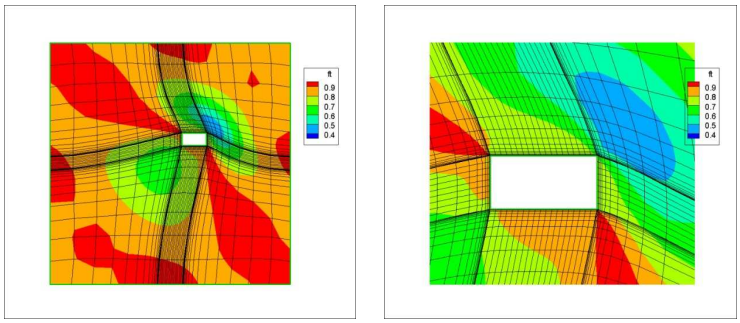
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



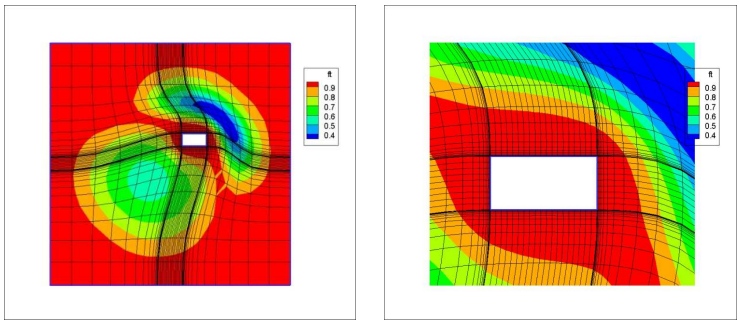
(a) DG-IDW



(b) DGM



(c) IDW



(d) RBF

Figure 11: The resultant mesh by different method (translation).

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

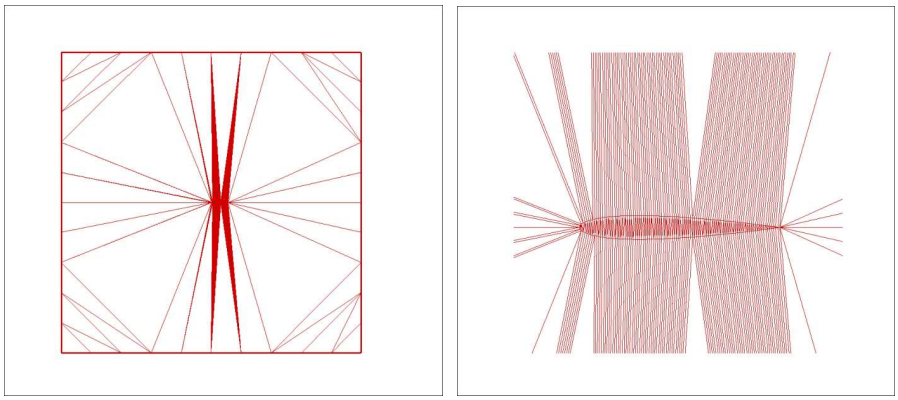


Figure 12: Original Delaunay graph.

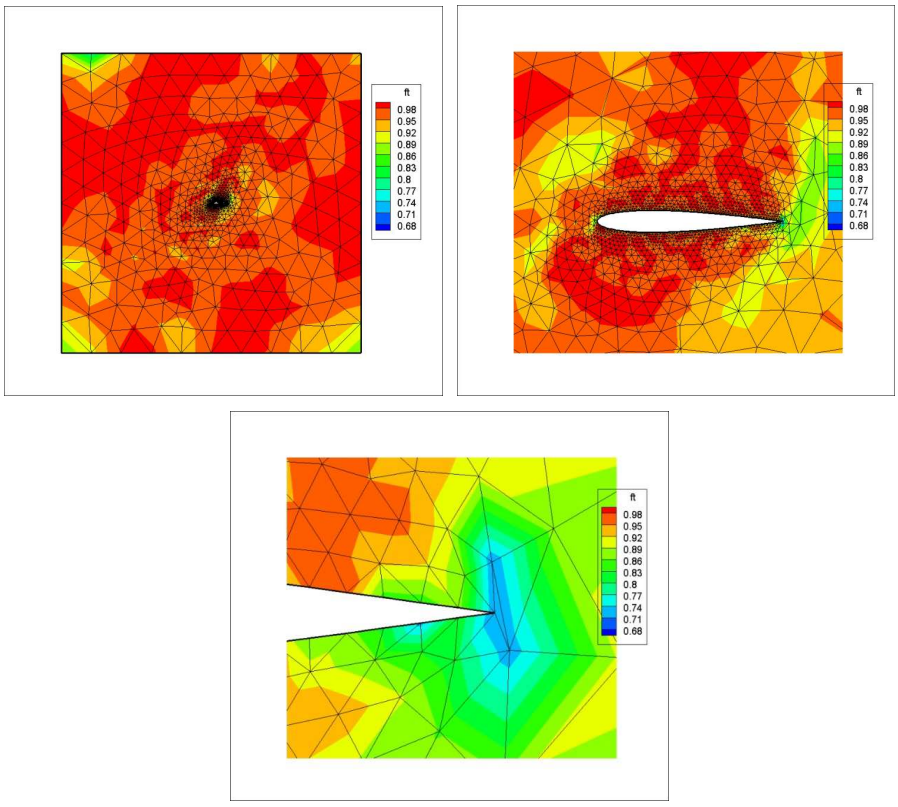
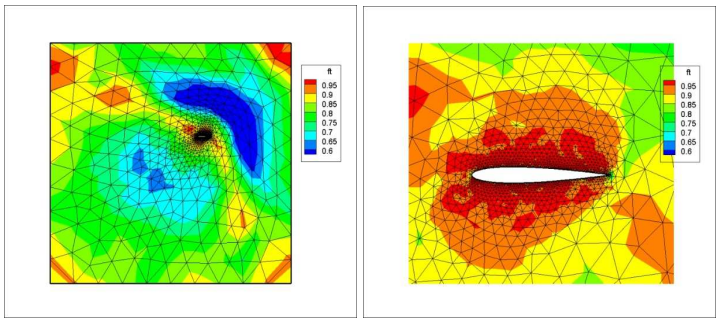


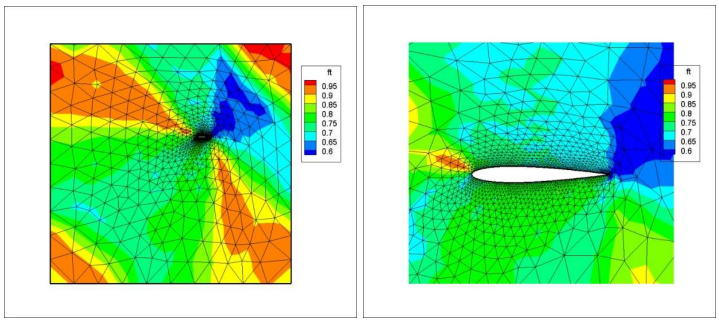
Figure 13: Original mesh.

the DGM method largely degrades the mesh quality in the same region. Particularly, in Fig. 15 the mesh near the trailing edge is shown. The mesh generated by the DGM method becomes invalid due to mesh overlapping. In Table 5, the mesh qualities are

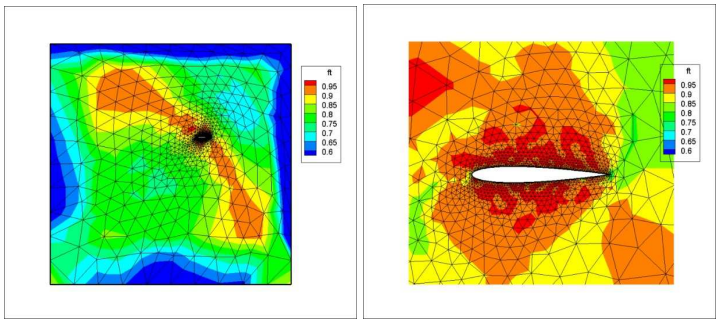
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



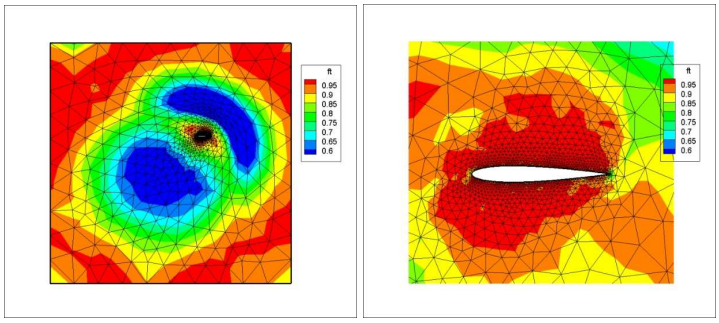
(a) DG-IDW



(b) DGM



(c) IDW



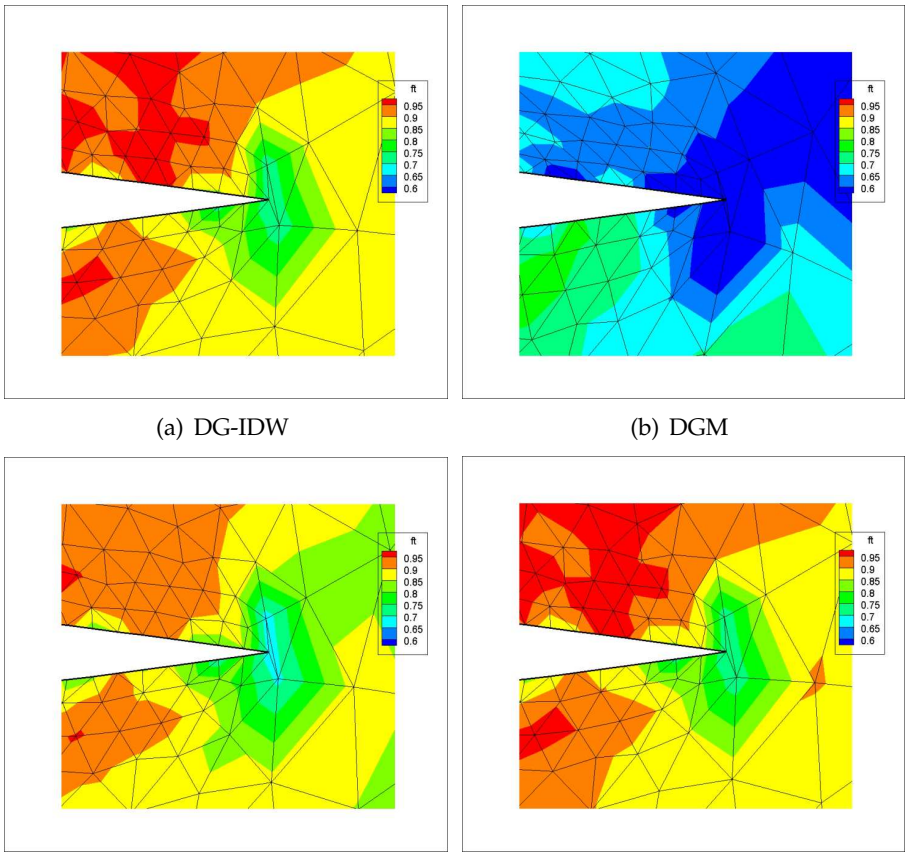
(d) RBF

Figure 14: The resultant mesh by different methods (translation).

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

Table 5: Mesh quality comparison.

	DGM	IDW	RBF	DG-IDW	original
averaged	0.736	0.895	0.895	0.908	0.965
minimal	-0.185	0.116	0.176	0.176	0.177



(a) DG-IDW (b) DGM

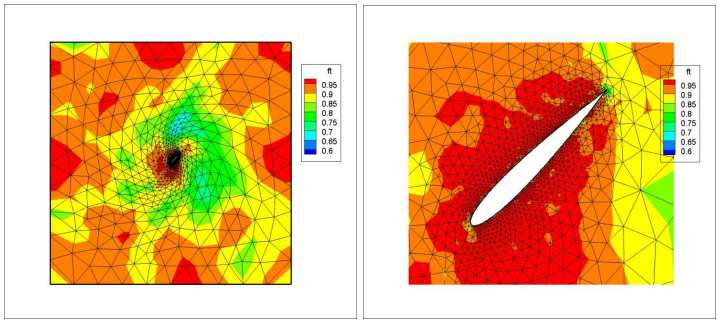
(c) IDW (d) RBF

Figure 15: The mesh quality contour near the trailing edge (translation).

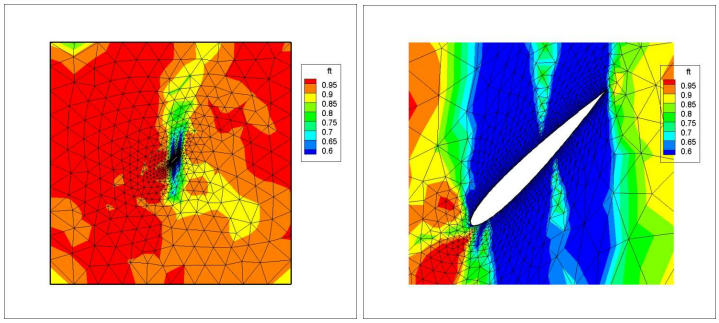
compared. The DG-IDW method shows the best mesh quality for both averaged and minimal mesh qualities.

The airfoil is then rotated by 45° with the deformed mesh shown in Figs. 16 and 17. For the rotation, the DGM method failed to retain the mesh quality near the airfoil. The low quality cells are all around the airfoil, which may cause significant numerical errors in the viscous flow simulation. On the other hand, the IDW, RBF and DG-IDW methods retain the mesh quality well near the airfoil. For rotation, the sliver cell behind the trailing edge does not become invalid for all the methods after mesh deformation. In Table 6, the

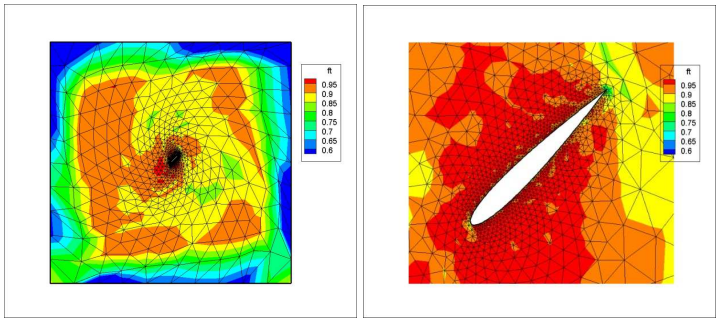
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



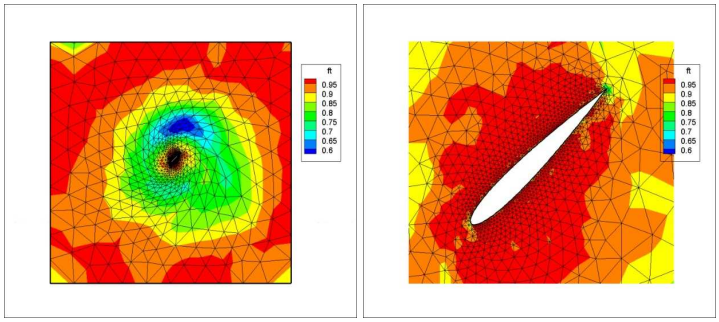
(a) DG-IDW



(b) DGM



(c) IDW



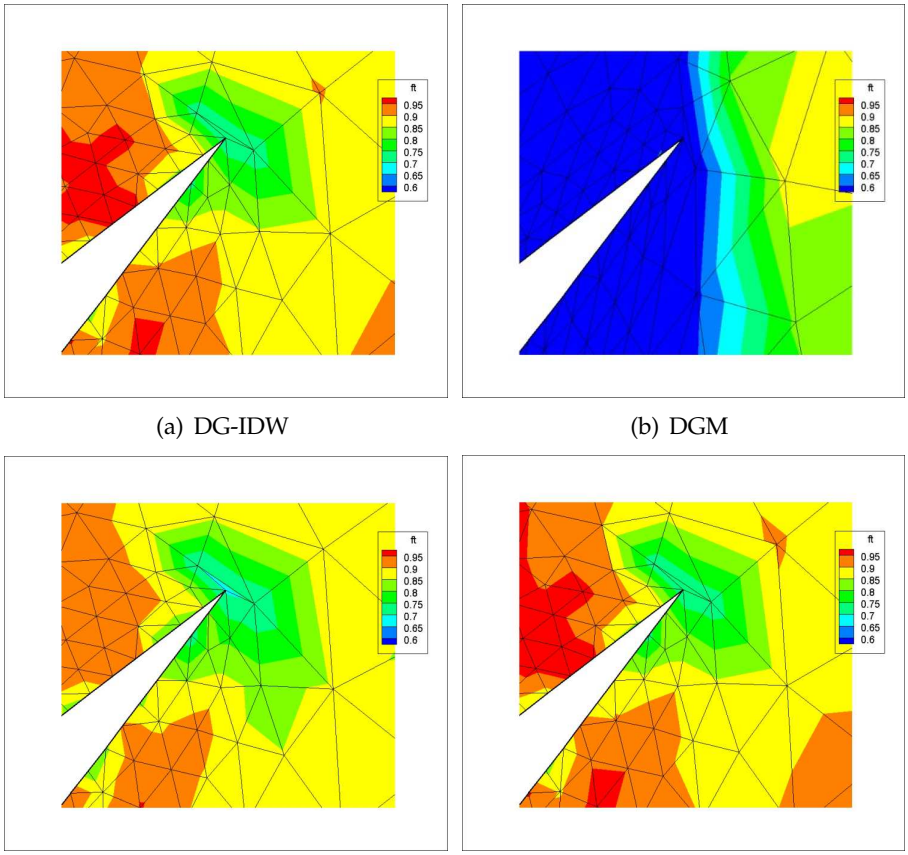
(d) RBF

Figure 16: The deformed mesh by different method (rotation).

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

Table 6: Mesh quality comparison.

	DGM	IDW	RBF	DG-IDW	original
averaged	0.942	0.927	0.923	0.933	0.965
minimal	0.151	0.162	0.176	0.177	0.177



(a) DG-IDW (b) DGM

(c) IDW (d) RBF

Figure 17: The mesh quality contour near the trailing edge (rotation).

mesh qualities are compared, in which the new method, i.e. the DG-IDW method, shows the best results for both averaged and minimal mesh qualities.

4 Test case 3 twisted bar

To further test the capability of the method for cases with twist (not simple translation or rotation), a structured mesh with 7520 nodes and 7308 cells is used, as shown in Fig. 18, for a twist deformation. The deformed meshes by using different methods are shown in

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

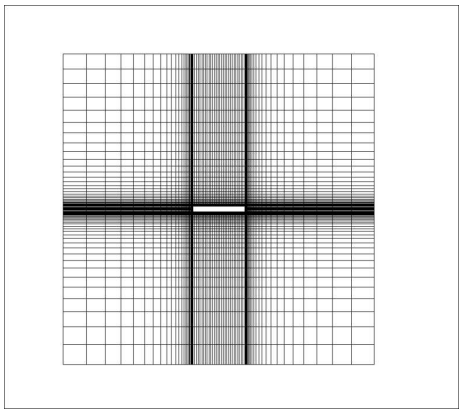


Figure 18: Original mesh.

Fig. 19. The DGM method and DG-IDW method show similar mesh quality contours, in which the mesh quality near the boundary is well preserved. The IDW method fails to maintain the mesh quality near the boundary. The RBF method largely maintained the mesh quality near the boundary but at the two ends, the mesh quality quickly degenerates. This means that for such deformation the DGM and DG-IDW methods can well preserve the mesh quality. In Table 7, the mesh quality and CPU time are compared. It is found that both the DGM and DG-IDW methods exhibit the best mesh quality and efficiency.

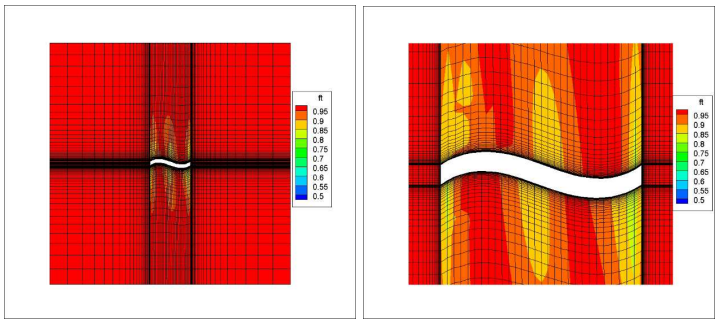
Table 7: Mesh quality and CPU time comparison.

	DGM	IDW	RBF	DG-IDW
averaged	0.981	0.943	0.914	0.980
minimal	0.792	0.414	0.666	0.799
CPU time	0.001	0.202	0.405	0.001

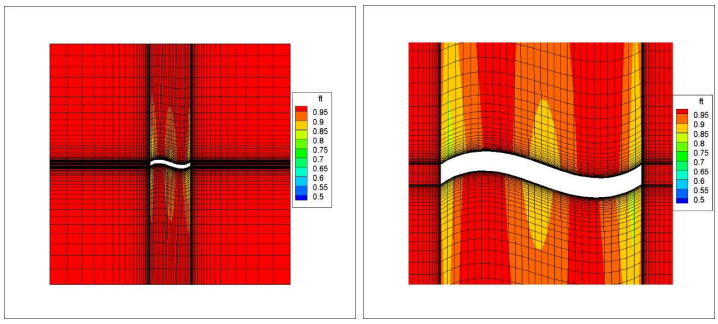
In Fig. 20, four deformed meshes are compared in which the bar in the middle is first twisted and then rotated by 45°. It should be noted here again, these meshes are all deformed by only one mapping without any middle steps. For this case, only the DG-IDW method preserved well the mesh quality near the boundary, while the RBF method partially preserved the mesh, similar as the previous case. The DGM method and IDW methods do not preserve well the mesh near the boundary; especially the DGM method leads to some low quality cells at the two ends of the bar. In Table 8, DG-IDW shows the best averaged mesh quality and second best minimal mesh quality. For this case, the RBF method showed the worst mesh quality among the four. From the efficiency point of view, the DG-IDW and DGM methods are much faster than the other two methods.

From these two cases, it is found that for pure deformation, the DGM method can maintain the mesh quality well; however it does not perform well for the large rotation

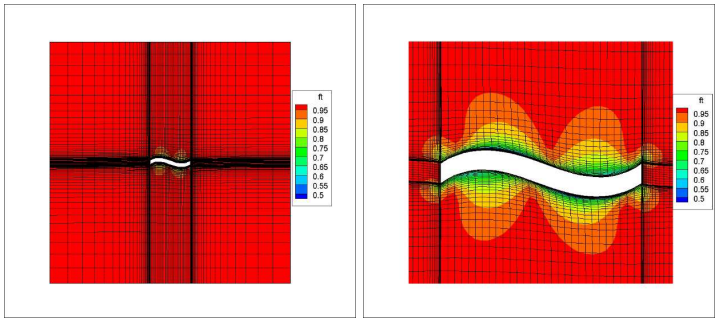
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



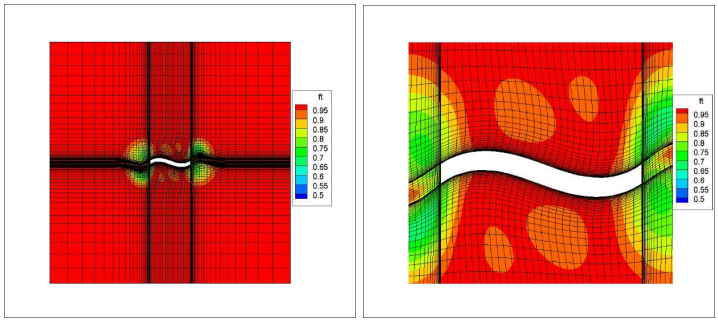
(a) DG-IDW



(b) DGM



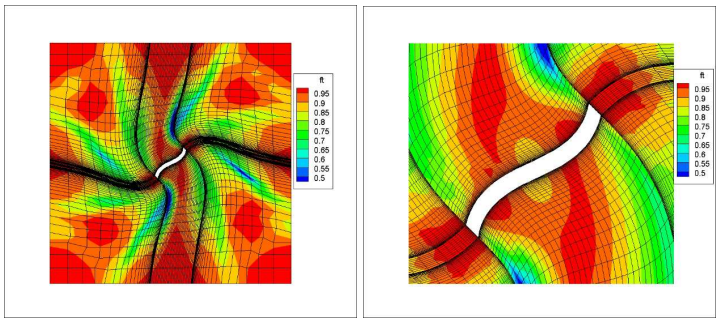
(c) IDW



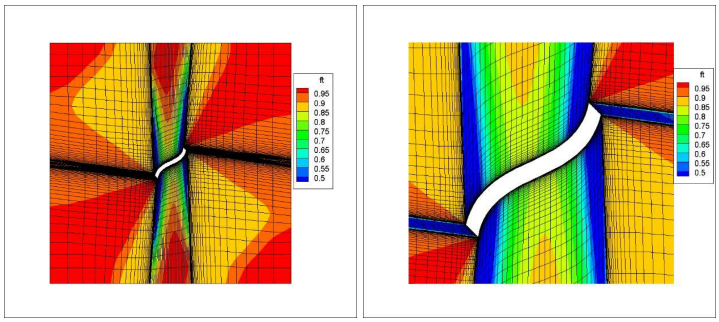
(d) RBF

Figure 19: Deformed mesh with mesh quality contour.

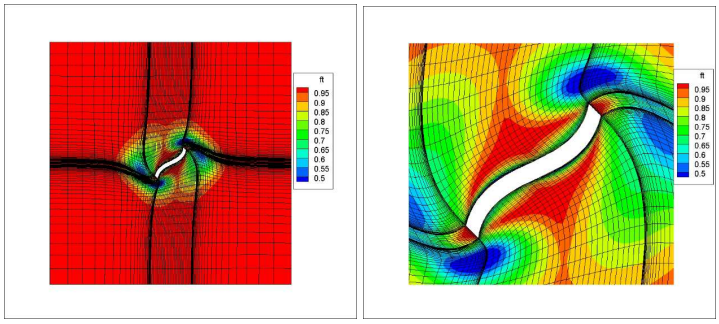
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



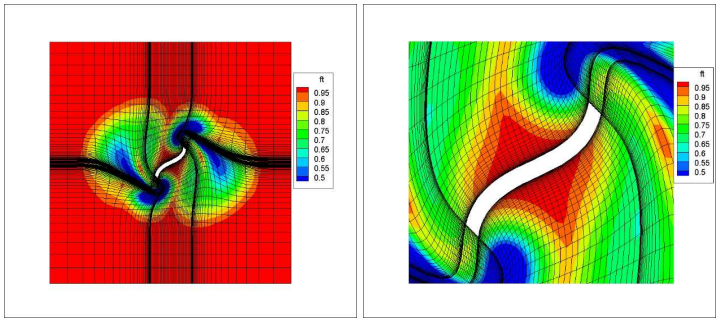
(a) DG-IDW



(b) DGM



(c) IDW



(d) RBF

Figure 20: Deformed mesh with mesh quality contour.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

Table 8: Mesh quality and CPU time comparison.

	DGM	IDW	RBF	DG-IDW
averaged	0.793	0.828	0.747	0.879
minimal	0.185	0.356	0.141	0.262
CPU time	0.001	0.202	0.402	0.001

regarding mesh quality near the boundary. However, the DG-IDW method can preserve the mesh quality well for both cases with a high efficiency. This is because, the DGM method uniformly distributes the skewness caused by the rotation, but the RBF and IDW methods are able to keep the skewness away from the moving boundary. As the combination of the DGM and IDW methods, the DG-IDW method inherits the characteristics of the IDW method, therefore it can largely preserve the mesh quality as the RBF and IDW methods do.

5 Moving mesh around a squeezed 3D sphere

As the first 3D case, a squeezed sphere is used, as shown in Fig. 21. To test the efficiency of the different methods, three different meshes from coarse to fine are shown in this case. In Table 9, the CPU time is compared. The DGM method is shown to be the most efficient, while the DG-IDW method is only slightly worse. Both the IDW method and RBF method are very time consuming, especially for the RBF method. As the mesh size increases, the required CPU time and memory for these two methods quickly become unaffordable. The required CPU time for the DG-IDW and the DGM methods in comparison can be divided into three parts. The first part is the generation of the Delaunay graph, of which the cost is normally $N_{bp}\log(N_{bp})$, where bp refers to number of boundary points. The second part is the identification process which is $N_s N_{vp}$ where N_s is the

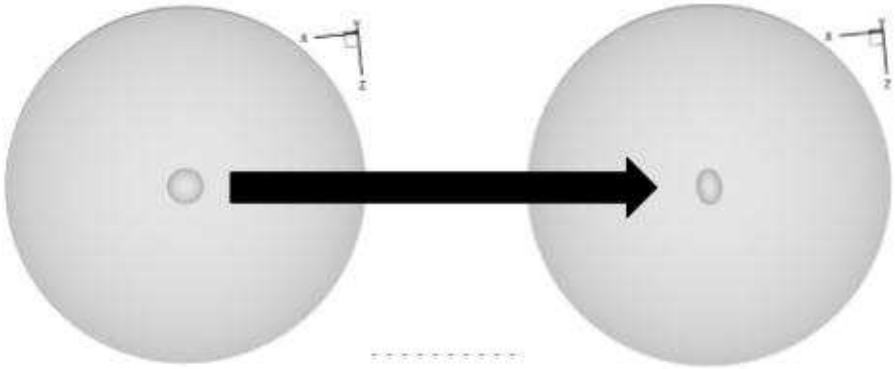


Figure 21: Original mesh.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

Table 9: Mesh quality and CPU time comparison.

	Total nodes	Total boundary nodes	DGM	IDW	RBF	DG-IDW
1	201,920	10,096	0.099s	485.105s	11692.154s	0.143s
2	403,840	10,096	0.172s	955.096s	-	0.237s
3	826,080	41,304	0.359s	8295.252s	-	0.451s

number of searches (according to the numerical test the N_s is normally less than 10 for 3D) [13,25], and vp stands for volume points. The computational costs of these two parts are exactly the same for the DGM and the DG-IDW methods. But luckily for the DG-IDW method, these two parts only need once for the whole unsteady simulation or optimization, thus the required information for the DG-IDW can be saved and used as an input file to generate different meshes. The third part is the mapping process in which the cost is $N_v p$ for both the DGM and the DG-IDW methods. The cost of the IDW method, however, is $N_{bp}N_{vp}$. Form the computational complexity point of view, it is clear that the cost of the DG-IDW method is similar as the DGM method, but more inexpensive than the IDW method. In practical term, however, due to the repeated generation and validation check of the Delaunay graph, the DGM method is less efficient than the DG-IDW method.

The mesh quality contours for the small size mesh are shown in Fig. 22 and the mesh quality is compared in Table 10. From the figures, it is noted that the DG-IDW method can largely preserve the mesh quality near the squeezed sphere, while the rest of the methods slightly degrade the mesh quality. Especially, the IDW method causes some low quality cells near the sphere. As can be seen from the table, the DG-IDW method shows better overall mesh quality among the four.

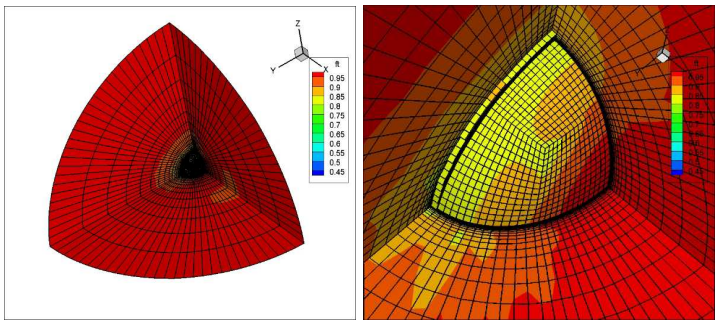
Table 10: Mesh quality and CPU time comparison.

	DGM	IDW	RBF	DG-IDW
averaged	0.845	0.880	0.833	0.836
minimal	0.552	0.414	0.705	0.613

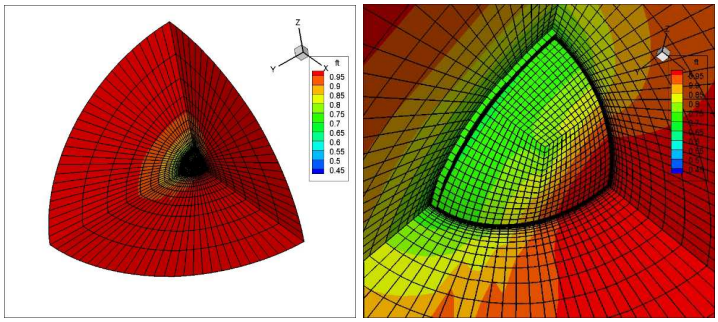
6 Deformed mesh around a wing-body configuration

An unstructured mesh with 257,909 nodes and 1,507,462 tetrahedrons based on a wing-body configuration was tested by the proposed DG-IDW method. The initial and deformed configurations are shown in Fig. 23, in which the wing is folded by 20°. The total CPU time for the DGM method is 0.503s, while for the DG-IDW method, it is 0.510s. The mesh quality ratio of the deformed mesh is shown in Fig. 24. For a fair comparison, the ratio of the mesh quality (f_{t1}/f_{t0}) is applied, where f_{t0} is the mesh quality of the original mesh, and f_{t1} was the mesh quality of the deformed mesh. The ratio near the wing

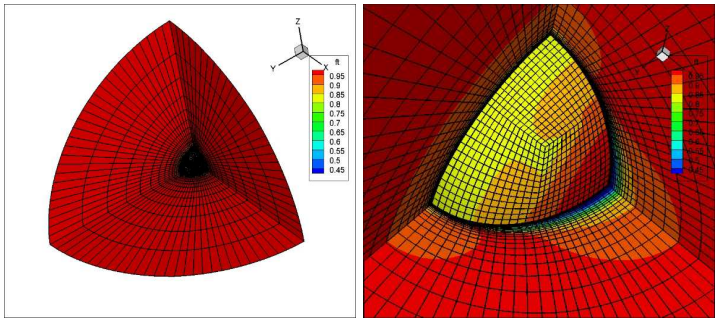
https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press



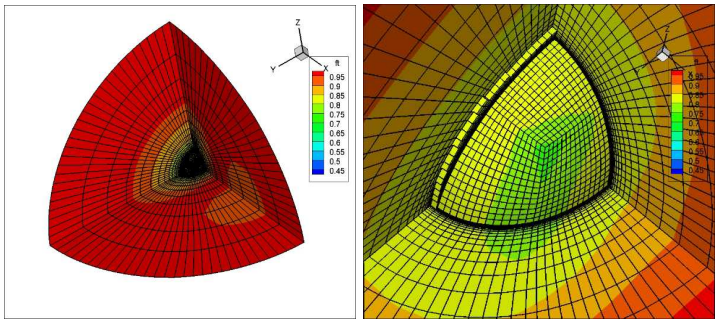
(a) DG-IDW



(b) DGM



(c) IDW



(d) RBF

Figure 22: Deformed mesh with mesh quality contour.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

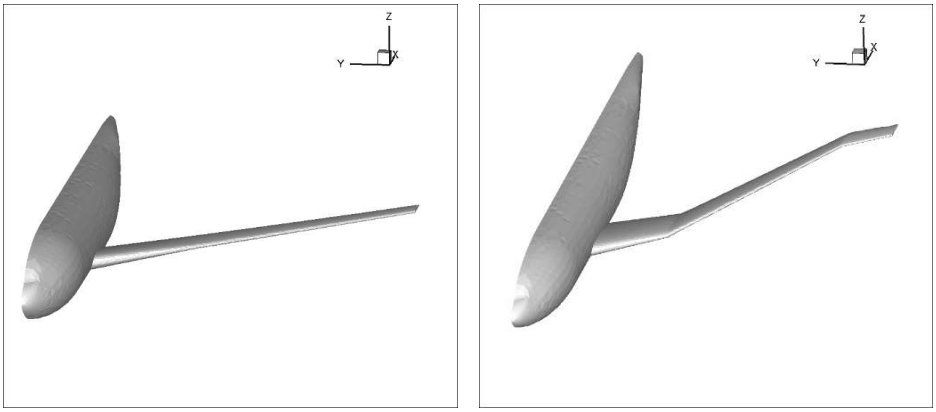


Figure 23: Wing-body configurations, (left) original, (right) deformed.

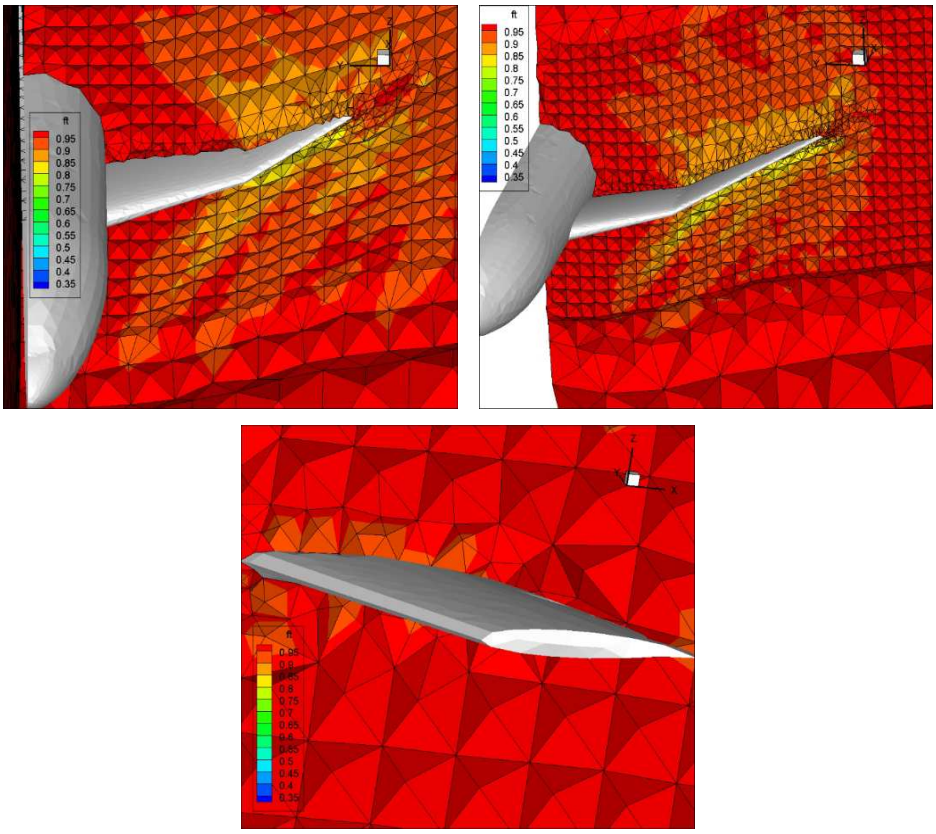


Figure 24: Mesh quality ratio contours (f_{t1}/f_{t0}) of the deformed mesh.

was all above 0.85, considering the relative large deformation, which suggests that the DG-IDW method maintains good mesh quality of the original mesh.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

7 Conclusions

A novel dynamic mesh deformation method is developed based on the combination of the Delaunay graph mapping method with inverse distance weighting interpolation. It maintains the high efficiency of the original Delaunay graph mapping method with improvement of the mesh quality. In particular, the mesh quality near the boundary is well preserved, which is very important for high Reynolds number viscous flow simulation. For large rotation deformation, the DGM method fails to preserve the mesh quality. The new method overcomes this problem without sacrificing the efficiency of the Delaunay graph mapping method, which is significantly faster than both the IDW and the RBF methods. The near wall mesh quality of this method is similar or even better in some cases than the IDW and the RBF methods.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities, NO. 3082014NS2014005.

References

[1] E. Luke, E. Collins, E. Blades, A fast mesh deformation method using explicit interpolation, J. Comput. Phys. 231(2012) 586-601.
[2] J.T. Batina, Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis, AIAA J. 29 (3) (1991) 327-333.
[3] C. Farhat, C. Degand, B. Koobus, M. Lesoinne, Torsional springs for two-dimensional dynamic unstructured fluid meshes, Comput. Method. Appl. M. 163 (1998) 231-245.
[4] R. Loehner, C. Yang, Improved ALE mesh velocities for moving bodies, Commun. Numer. Meth. En. 12 (1996) 599-608.
[5] J.D. Bau, H. Luo, R. Loehner, E. Goldberg, A. Feldhun, Application of unstructured moving body methodology to the simulation of fuel tank separation from an F-16 fighter, in: 35th Aerospace Sciences Meeting and Exhibit, AIAA Paper No. AIAA-1997-0166, Reno, NV, 1997.
[6] B. Helenbrook, Mesh deformation using the biharmonic operator, Int. J. Numer. Meth. Eng. 56 (2003) 1007-1021.
[7] A. de Boer, M. van der Schoot, H. Bijl, Mesh deformation based on radial basis function interpolation, Comput.Struct. 85 (2007) 784-795.
[8] T. Rendall, C. Allen, Efficient mesh motion using radial basis functions with data reduction algorithms, J. Comput. Phys. 228 (17) (2009) 6231-6249.
[9] T. Rendall, C. Allen, Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors, Int. J. Numer. Meth. Eng. 81 (1) (2010) 89-105.
[10] J.A.Witteveen, H. Bijl, Explicit mesh deformation using inverse distance weighting interpolation, in: 19th AIAA Computational Fluid Dynamics Conference, AIAA, San Antonio, Texas, AIAA Paper 2009-3996, 2009.
[11] X. Liu, N. Qin, H. Xia, Fast dynamic grid deformation based on Delaunay graph mapping, J. Comput. Phys. 211 (2006) 405-423.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press

[12] J.W. Van Der Burg, H. Freiherr Von Geyr, R. Heinrich, P. Eliasson, T. Delille, and J. Krier, Geometrical Installation and Deformation Effects in High-Lift Flows, AIAA J.,47(2009), 60-70
[13] Y. Wang, N, Qin, N. Zhao, Delaunay graph and radial basis function for fast quality mesh deformation. J. Comput. Phys., Volume 294 (2015), 149-172
[14] E. Lefrancois, A simple mesh deformation technique for fluid-structure interaction based on a submesh approach, Int. J. Numer. Meth. Eng. 75 (9) (2008) 10851101.
[15] J.H. Ko, S.H. Park, H.C. Park, Finite macro-element-based volume grid deformation for large moving boundary problems, International Journal for Numerical Methods in Biomedical Engineering 26 (12) (2010) 16561673.
[16] M. Miwa, K. Tani, T. Yamada, S. Wakao, A study of mesh deformation methods for magnetic field analysis, IEEJ Transactions on Electrical and Electronic Engineering 6 (5) (2011) 497502.
[17] D. Stadler, F. Kosel, D. Celic, A. Lipej, Mesh deformation based on artificial neural networks, International Journal of Computational Fluid Dynamics 25 (8) (2011) 439448.
[18] P. Gopalakrishnan, D.K. Tafti, A parallel boundary fitted dynamic mesh solver for applications to flapping flight, Comput. Fluids 38 (8) (2009) 15921607.
[19] X. Zhou, S. Li, A new mesh deformation method based on disk relaxation algorithm with pre-displacement and post-smoothing, J. Comput. Phys., 235(2013) 199-215.
[20] X. Zhou, S. Li, A novel three-dimensional mesh deformation method based on sphere relaxation, J. Comput. Phys., 298(2015), 320-336.
[21] L. Zhang,X. Chang,X. Duan,Z.He, Applications of dynamic hybrid grid method for three-dimensional moving/deforming boundary problems. Comput. Fluids, 62 (2012), 45-63
[22] K.B. Lee, J.H. Kim, C. Kim, Aerodynamic Effects of Structural Flexibility in Two-Dimensional Insect Flapping Flight, J of Aircraft, 48(3), (2011),894-909.
[23] H. Wang, J. Leskinen, D. Lee, Active flow control of airfoil using mesh/meshless methods coupled to hierarchical genetic algorithms for drag reduction design, Engineering Computations,30(4),(2013),562-580.
[24] P.M. Knupp, Algebraic mesh quality metrics for unstructured initial meshes, Finite Elem. Anal. Des. 39 (2003) 217-241.
[25] Y. Wang, N. Qin, G. Carnie, Zipper layer method for linking two dissimilar structured meshes, J. Comput. Phys. 255(2013), 130-148.

https://doi.org/10.4208/cicp.OA-2016-0085 Published online by Cambridge University Press