



This is a repository copy of *Delaunay graph-based moving mesh method with damping functions*.

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/140807/>

Version: Published Version

---

**Article:**

Wang, Y., Qin, N. [orcid.org/0000-0002-6437-9027](https://orcid.org/0000-0002-6437-9027) and Zhao, N. (2018) Delaunay graph-based moving mesh method with damping functions. *Chinese Journal of Aeronautics*, 31 (11). pp. 2093-2103. ISSN 1000-9361

<https://doi.org/10.1016/j.cja.2018.08.008>

---

**Reuse**

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>



Chinese Society of Aeronautics and Astronautics  
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn  
www.sciencedirect.com



# Delaunay graph-based moving mesh method with damping functions



Yibin WANG<sup>a</sup>, Ning QIN<sup>b</sup>, Ning ZHAO<sup>a,\*</sup>

<sup>a</sup> College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>b</sup> Department of Mechanical Engineering, The University of Sheffield, Sheffield S1 3JD, United Kingdom

Received 6 October 2017; revised 20 November 2017; accepted 25 December 2017

Available online 22 August 2018

## KEYWORDS

Delaunay graph mapping;  
Dynamic mesh;  
Grid deformation;  
Mesh;  
Moving mesh

**Abstract** The fluid–structure interaction and aerodynamic shape optimization usually involve the moving or deforming boundaries, thus the dynamic mesh techniques are the key techniques to cope with such deformation. A novel dynamic mesh method was developed based on the Delaunay graph in this paper. According to the Delaunay graph, the mesh points were divided into groups. In each group, a factor ranging from 0 to 1 was calculated based on the area/volume ratio. By introducing a proper function for this factor, this method can control the mesh quality with high efficiency. Several test cases were compared with other dynamic mesh methods regarding mesh quality and CPU time, such as radial basis function method and Delaunay graph mapping method.

© 2018 Chinese Society of Aeronautics and Astronautics. Production and hosting by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

The dynamic mesh techniques are widely used in the fluid–structure interaction and aerodynamic shape optimization which involve the moving or deforming boundaries. For these numerical simulations, two aspects of the dynamic mesh method are very critical, one is the mesh quality after deforming, and the other is the efficiency. From the viewpoint of the mesh quality, physical analogy method,<sup>1–4</sup> such as the spring analogy approach<sup>5</sup> and elastic analogy,<sup>6,7</sup> can well maintain the mesh quality after the deformation. However, these meth-

ods need to solve certain equations, which normally require large computational cost. Particularly, for large shape changing, some of these methods may cause invalid cells, therefore, Farhat et al.<sup>8</sup> introduced the torsional springs to prevent the mesh from becoming invalid. Recently, some interpolation methods which can largely preserve the mesh quality near the boundaries were developed, e.g. Radial Basis Function (RBF) method<sup>9–11</sup> and Inverse Distance Weighting (IDW) interpolation method.<sup>12,13</sup> The RBF method, by using the displacement of the boundary nodes, constructs a matrix to interpolate the interior nodes from the original position to the new position. This method needs to solve two (for 2D) or three (for 3D)  $n_b$  by  $n_b$  matrixes, where  $n_b$  is the total number of boundary nodes. As the mesh size increases, the computational cost can be significant. To reduce the computational cost, Rendall and Allen<sup>14</sup> proposed an approximate RBF method with the data reduction algorithms. Though it improves the efficiency, some surface errors of mismatch may be introduced to the boundary. Later on, they developed a surface correction step

\* Corresponding author.

E-mail address: [zhaoam@nuaa.edu.cn](mailto:zhaoam@nuaa.edu.cn) (N. ZHAO).

Peer review under responsibility of Editorial Committee of CJA.



to fix the problem.<sup>15</sup> The IDW interpolation method, developed by Witteveen and Bijl,<sup>12,13</sup> simply uses the inverse distance as a weight for the displacement of each boundary to calculate the new position of the interior nodes. This method is faster than the original RBF method; however, it still needs to loop all the boundary nodes for each interior node, resulting in a large computational cost for large 3D problems. Luke et al.<sup>16</sup> proposed an explicit interpolation method based on the IDW method, which shows similar mesh quality as the RBF method, but with a relatively faster speed. Recently, Zhou and Li<sup>17</sup> developed a 2D dynamic mesh method based on disk relaxation. Later on, they managed to further improve the method and successfully applied it to the 3D cases.<sup>18</sup>

For the fluid–structure interaction problem, the mesh needs to be updated at each time step; while for the aerodynamic shape optimization, each new configuration needs a renewed mesh. Hence, due to the frequent use of the dynamic mesh, the efficiency of the dynamic method becomes very important for these problems. From the viewpoint of efficiency, the interpolation method, such as transfinite interpolation<sup>19–21</sup> or Delaunay Graph Mapping (DGM) method,<sup>22</sup> can quickly deform the mesh. The transfinite interpolation, however, can only be used on simple structured mesh. The Delaunay Graph Mapping (DGM) method can handle most of the deformation with quality mesh,<sup>23–25</sup> but it is difficult to deal with the rotation motion. Considering its high efficiency, for large 3D problems with small deformation, it is a better choice than the RBF and IDW methods. In general, most of the current methods either generate high quality mesh with large computational cost, or generate lower quality mesh with high efficiency. Recently, a new method successfully combines the RBF and DGM methods with both advantages,<sup>26</sup> which is called the

DGRBF. It is based on the Delaunay graph to divide the mesh nodes into groups, then uses the RBF method to interpolate the mesh nodes to its new position. As a result, the large matrix becomes a series of 3 by 3 (for 2D) or 4 by 4 (for 3D) small matrixes; therefore the computational cost is substantially decreased. Later on, the IDW function was implemented into the DGM method which shows similar mesh quality as the DGRBF but slightly better efficiency.

As the RBF and IDW function can improve the mesh quality of the DGM method, are there any other functions with simpler form which can further improve both the mesh quality and efficiency? To achieve this, the functions should meet two critical conditions, firstly the functions can maintain the mesh quality near the wall; secondly the form of the function should be as simple as possible to maintain the high efficiency. In this paper, a new dynamic mesh method based on the above idea is developed. This method is based on the Delaunay graph to divide the mesh nodes into groups, and then it introduces a damp function for each group to control the mesh deforma-

**Table 1** Damp functions.

Parameter	Definition
$\varphi_1$	$(1 - r)^2(r + 1)$
$\varphi_2$	$(1 - r)^4(4r + 1)$
$\varphi_3$	$(1 - r)^6(35r^2 + 18r + 1)/3$
$\varphi_4$	$(1 - r)^8(32r^3 + 25r^2 + 8r + 1)$
$\varphi_5$	$(1 - r)^3(r + 1)$
$\varphi_6$	$(1 - r)^2$
$\varphi_7$	$1 - r$

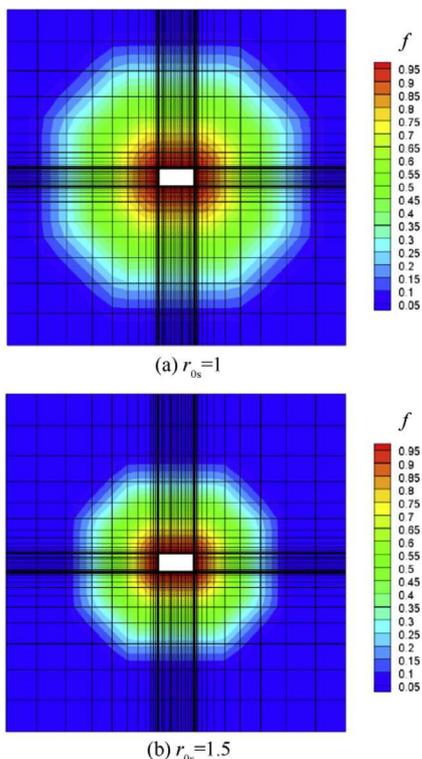


Fig. 1 Contours of  $f$  with different  $r_{0s}$

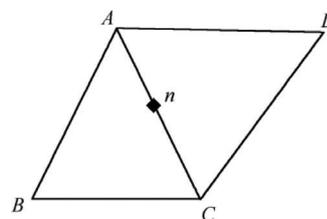


Fig. 2 Discontinuity occurs for nodes on the edge of Delaunay triangle.

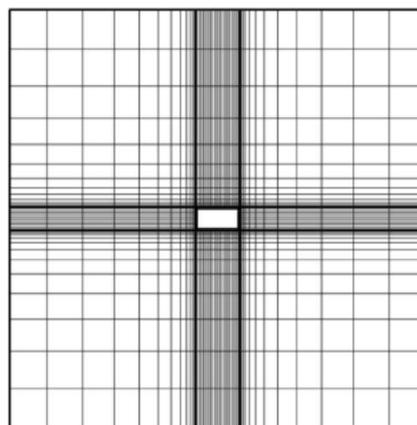


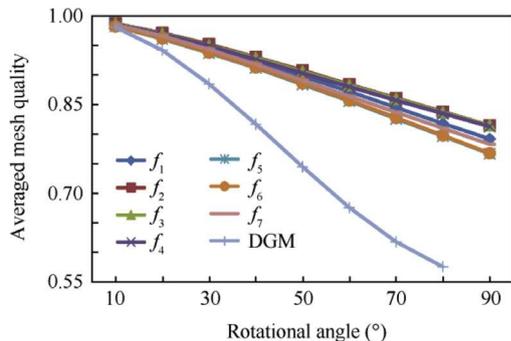
Fig. 3 Original mesh of rotating rectangle.

tion; therefore the overall mesh quality can be effectively controlled and improved. In addition, the damp function only uses local information; therefore the computational cost is low. Different forms of the functions are compared and discussed. Sev-

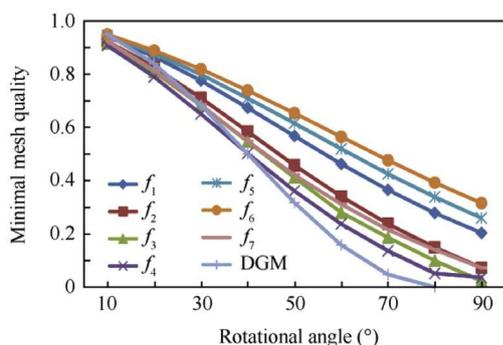
eral test cases are compared with the RBF, IDW, DGM and DGRBF methods, which show higher mesh quality with higher efficiency.

### 2. Delaunay graph mapping method with damp function

The general procedure of the Delaunay Graph Mapping with damp Function (DGMF) method is similar to the original DGM method. However, being different from the original one, it utilizes the area ratio to calculate an intermediate variable  $r$  rather than the coordinates. By using this variable  $r$ , a damp function which controls the displacement of the interior nodes is calculated. The procedures are set as follows:

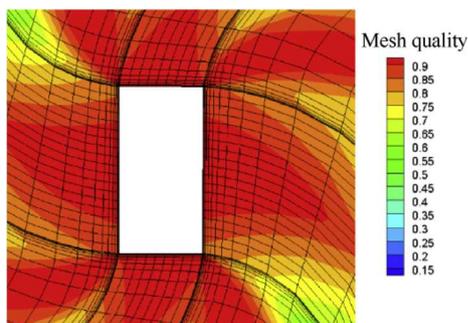
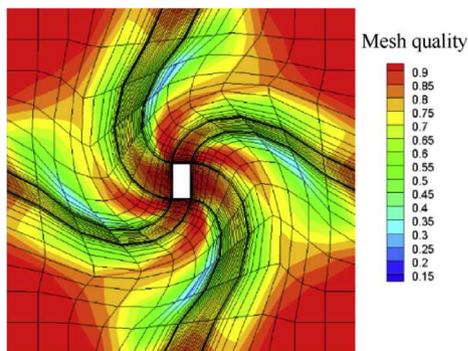


(a) Averaged mesh quality

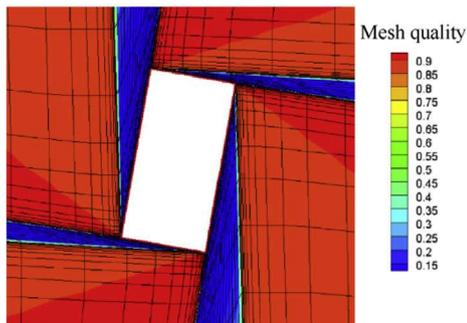
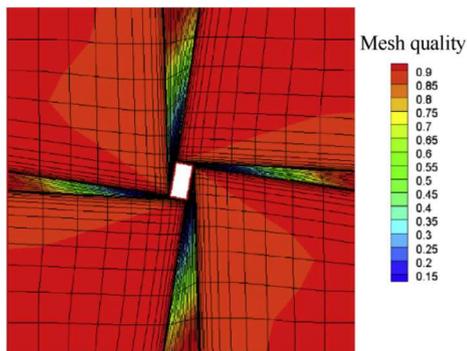


(b) Minimal mesh quality

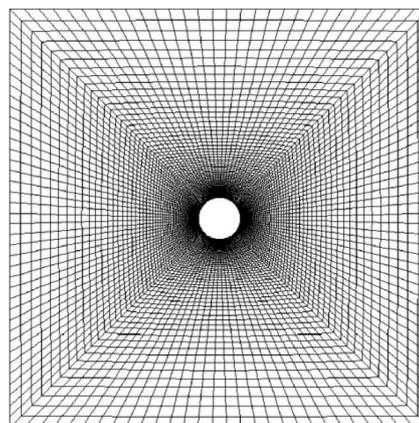
**Fig. 4** Comparison of mesh quality with different damp functions and DGM method.



**Fig. 5** Mesh quality contour (DGMDF by  $f_1$ ).



**Fig. 6** Mesh quality contour (DGM).



**Fig. 7** Original mesh of rotating circle.

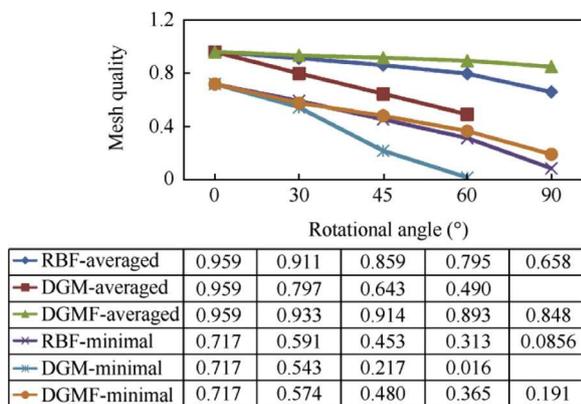


Fig. 8 Comparison of mesh quality for rotating circle.

- (1) Generate the Delaunay graph using all the boundary nodes of the original mesh.
- (2) Find the parent Delaunay element for each mesh node point so that all the mesh points are grouped by the Delaunay Graph elements.
- (3) Calculate  $r$  for each node using the area or volume ratio of Delaunay elements (triangle for 2D and tetrahedron for 3D).
- (4) Calculate the damp function  $f(r)$  for each node.
- (5) Calculate the new node position for all the mesh nodes using the damp function  $f(r)$ .

At step (3), an intermediate variable  $r$  is calculated by the area ratio as follows:

$$r = \sum_{i=1}^n e_i r_0 \tag{1}$$

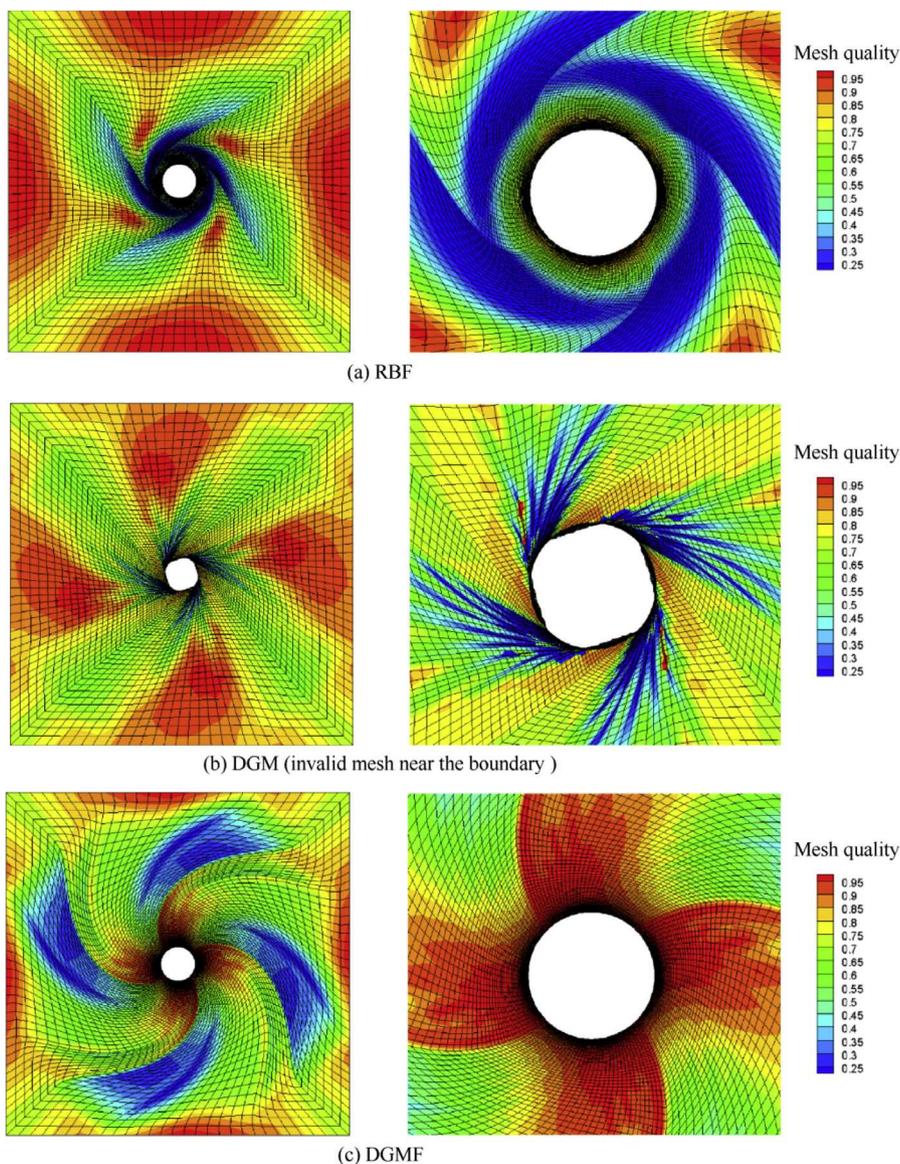


Fig. 9 Mesh quality contour (90°rotation).

where  $e_i$  is the area or volume ratio (detailed definition can be found in Ref.<sup>22</sup>), and  $n = 3$  for 2D,  $n = 4$  for 3D.  $r_0$  is the user defined constant for the boundary nodes, for all static boundaries  $r_{0s} \geq 1$ , while for all dynamic boundaries  $r_{0d} = 0$ . By changing  $r_{0s}$ , the impact region of  $f(r)$  (the detailed form of the damp function  $f(r)$  will be discussed in the next paragraph) can be adjusted (Fig. 1). In the figures, the outer boundary is the static boundary, while the inner boundary is the dynamic boundary. The mesh nodes in the red region which are adjacent to the dynamic boundary will move similarly as the dynamic boundary since  $f$  is close to 1, thus the mesh quality can be maintained in this region. The mesh nodes in the blue region which is close to the static boundary will almost keep unchanged, because  $f$  is close to 0. From the figures, it is clear that by changing the value of  $r_{0s}$ , the size of the blue and red region can be adjusted. By increasing  $r_{0s}$ , one can compress the red region and enlarge the blue region.

The damp function used in this paper is in the form of

$$f_i(r) = \begin{cases} 0 & r > 1 \\ \varphi_i & r \leq 1 \end{cases} \quad (2)$$

where  $\varphi$  is a function ranging from 0 to 1. In order to maintain high efficiency, computational expensive operations are not considered, such as exponent calculation used in the IDW function. Therefore, the simple polynomial is used in this paper. In Table 1, the functions tested and compared in this paper are listed. These functions are generally the radial basis function or modified radial basis function whose value ranges from 0 to 1; different forms and orders can affect the change of the deformation of the mesh. The comparison of these functions will be discussed in Section 3.1.

The change of the boundary can be mainly classified into three categories, i.e., rigid body translation, rigid body rotation and shape deformation. Let the displacement of translation be  $t$ , the Eulerian angle be  $\theta$ , and the displacement of shape deformation be  $d$ , then the displacement of the interior nodes can be calculated as follows:

$$s_x = ft_x + fd_x + (x - x_c)\cos(\theta f) + (y - y_c)\sin(\theta f) + x_c \quad (3)$$

$$s_y = ft_y + fd_y + (y - y_c)\cos(\theta f) - (x - x_c)\sin(\theta f) + y_c \quad (4)$$

where  $x$  and  $y$  are the original coordinates, and  $x_c$  and  $y_c$  are coordinates of the center.

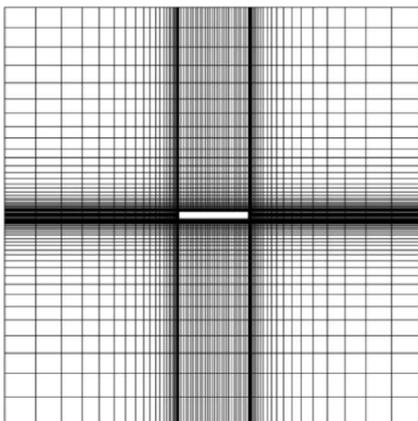


Fig. 10 Original mesh of twisted bar.

For the 3D problem, let  $\theta = (\alpha, \beta, \gamma)$  be the Eulerian angle, the total displacement of the interior nodes is in the form of

$$s = f \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + f \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} + \begin{bmatrix} \cos(f\gamma) & \sin(f\gamma) & 0 \\ -\sin(f\gamma) & \cos(f\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(f\beta) & \sin(f\beta) \\ 0 & -\sin(f\beta) & \cos(f\beta) \end{bmatrix} \begin{bmatrix} \cos(f\alpha) & \sin(f\alpha) & 0 \\ -\sin(f\alpha) & \cos(f\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} x_0 - x_0 \quad (5)$$

where  $x_0$  is the original coordinates. Let  $q(f\theta)$  be the quaternion, and Eq. (5) can be rewritten as

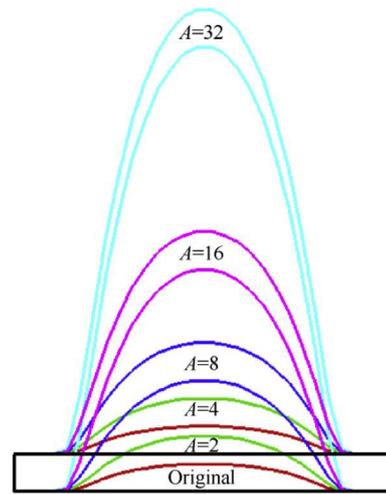
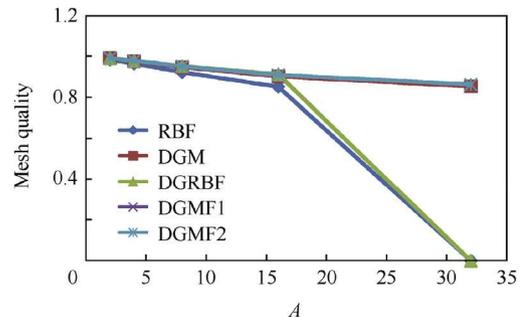
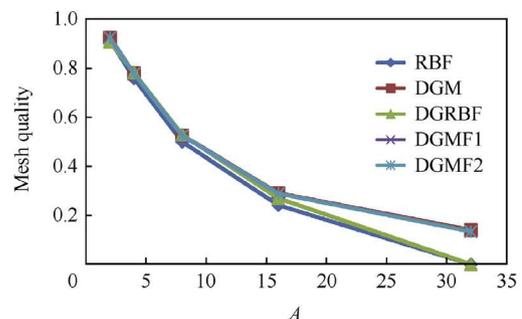


Fig. 11 Deformation of the bar as  $A$  increases.

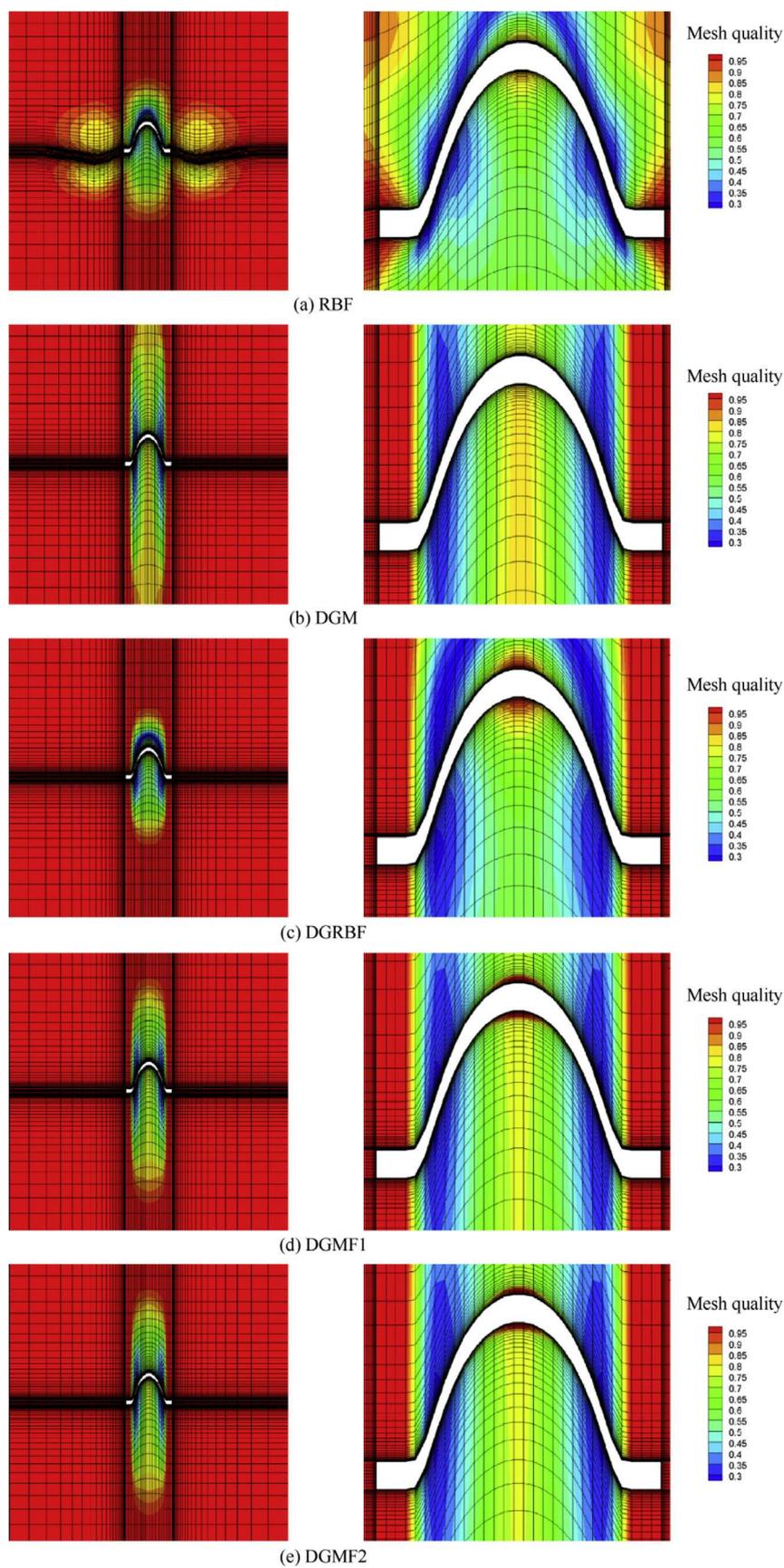


(a) Averaged mesh quality as  $A$  increases



(b) Minimal mesh quality as  $A$  increases

Fig. 12 Comparison of mesh quality for twisted bar.



**Fig. 13** Mesh quality contour ( $A = 16$ ).

$$s = f \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + f \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} - \mathbf{q}\mathbf{x}_0\mathbf{q}^{-1} - \mathbf{x}_0 \tag{6}$$

Normally for the numerical problems involving rigid body motion, the translation and rotation of the moving boundary are known. For deformation, if the moving boundary can be expressed as a known function,  $d$  can be directly computed; if not,  $d$  can be calculated in the following way.

$$d = \frac{\sum_{i=1}^{n_d} e_i d_i}{\sum_{i=1}^{n_d} e_i} \tag{7}$$

where  $d_i$  the deformation displacement of the dynamic boundary nodes in a Delaunay triangle/tetrahedron, and  $n_d$  the number of the dynamic boundary nodes in the Delaunay triangle/tetrahedron.

Due to the interpolation based on different triangles/tetrahedrons, discontinuity may occur for the DGRBF method. As shown in Fig. 2, Node  $n$  is on the sharing edge of the Delaunay triangle  $\triangle ABC$  and  $\triangle ACD$ . Obviously its displacement can be computed by using either triangle. However, for shape deformation, the results can be inconsistent when different triangles are used. For the unstructured meshes, the nodes appearing on an edge of the Delaunay triangle maybe rare; however, for the structured meshes it happens from time to time. For the DGMF method, the discontinuity problem aforementioned is resolved. Taking Node  $n$  in Fig. 2 as an example and presuming that Nodes  $B, C$  and  $D$  are dynamic boundary nodes, since Node  $n$  is on Edge  $AC$ ,  $e_b$  and  $e_d$  are equal to 0; thus, for either Delaunay triangle, the displacement for the shape deformation computed from Eqs. (3), (4) and (7) are the same  $d_n = fd_c$ .

### 3. Results and discussion

#### 3.1. Rotating rectangle

One of the major disadvantages of the original Delaunay mapping method is its difficulty in handling large rotational problems. Hence, as the first test case, the capability for dealing with the large rotational problem was tested. The original mesh is shown in Fig. 3.

The rectangle in the middle rotated around its center from  $10^\circ$  to  $90^\circ$ . Therefore, the interior nodes of the mesh can be computed as

$$s_x = (x - x_c)\cos(\theta f) + (y - y_c)\sin(\theta f) + x_c$$

$$s_y = (y - y_c)\cos(\theta f) - (x - x_c)\sin(\theta f) + y_c$$

It should be noted that all the meshes were deformed directly from the original mesh without any intermediate steps in this paper. Therefore, only one Delaunay graph was needed. The  $r_{0s} = 1$  was used for all the test cases in this paper; one can restrain the impact region by increasing this value. The effect of different damp functions is compared in Fig. 4, regarding the averaged mesh quality and minimal mesh quality. In this figure, the result of the DGM method is also included. The size-skew metric  $Ft$  was used in this paper which assesses the skewness and the size of the mesh.<sup>27</sup> The essential properties of the size-skew metric are

$Ft = 1$  element has equal angles and the same size as the initial element.

$Ft = 0$  element is degenerated.

The damp function  $f_2, f_3$  and  $f_4$  show better averaged mesh quality but worse minimal mesh quality, while  $f_5$  and  $f_6$  show worse averaged mesh quality with better minimal mesh quality. Function  $f_1$  is in the middle place among the all. The high-order functions ( $f_2, f_3$  and  $f_4$ ) slowly decay in the near boundary region, whereby the mesh quality in this region is well preserved; out of this region, they quickly decrease, resulting in a sudden transition whereby low-quality cells may be generated in this region, which cause a low minimal mesh quality. In contrast, the lower order functions ( $f_4$  and  $f_5$ ) decrease in a relatively even way, hence the transition is quite smooth, resulting in a high minimal mesh quality. However, the mesh quality in the near boundary region is not as good as the high order functions. It is found in this figure that function  $f_1$  shows moderate averaged mesh quality and minimal mesh quality, and the mesh quality near the boundary can be largely maintained. Therefore in this paper, this function was used as the damp function for the rest of the test cases. The deformed meshes are shown in Figs. 5 and 6. The new method well preserves the mesh quality near the moving boundary; while the DGM method does not.

#### 3.2. Rotating circle

To further test the capability of the new method in handling rotation motion, a rotating circle mesh (in Fig. 7) was tested. The radius of the circle is 1 unit and the outer boundary is a square with a 10-unit-side. The mesh quality is compared in Fig. 8. In this case, the DGMF method is compared with both the DGM and RBF methods. The DGMF method shows the best mesh quality among the three. As shown in Fig. 9, the DGMF method well reserves the mesh quality for  $90^\circ$  rotation, whereby the mesh quality contour further verifies that the DGMF method can well handle the rotation motion. The DGM method, as mentioned earlier, was difficult to handle the rotation problem, therefore for  $90^\circ$  rotation it failed to generate a valid mesh. Based on the displacement of the boundary nodes, the RBF method interpolates the displacement of interior nodes. However, for this case, it does not well preserve the mesh quality near the boundary. From the two test cases, it is found that this new method can well deform the mesh for rigid body rotation problems.

#### 3.3. Twisted bar with rotation

Normally, the DGM method is able to generate the high quality mesh for surface deformation problems such as twisted surface. In this case, the capability in shape deformation problem was tested along with the rotation motion. A bar with 10 unit

**Table 2** Comparison of mesh quality for twisted bar with rotation.

Mesh quality	Averaged	Minimal
RBF	0.747	0.014
DGM	0.793	0.019
DGRBF	0.862	0.296
DGMF1	0.873	0.496
DGMF2	0.880	0.530

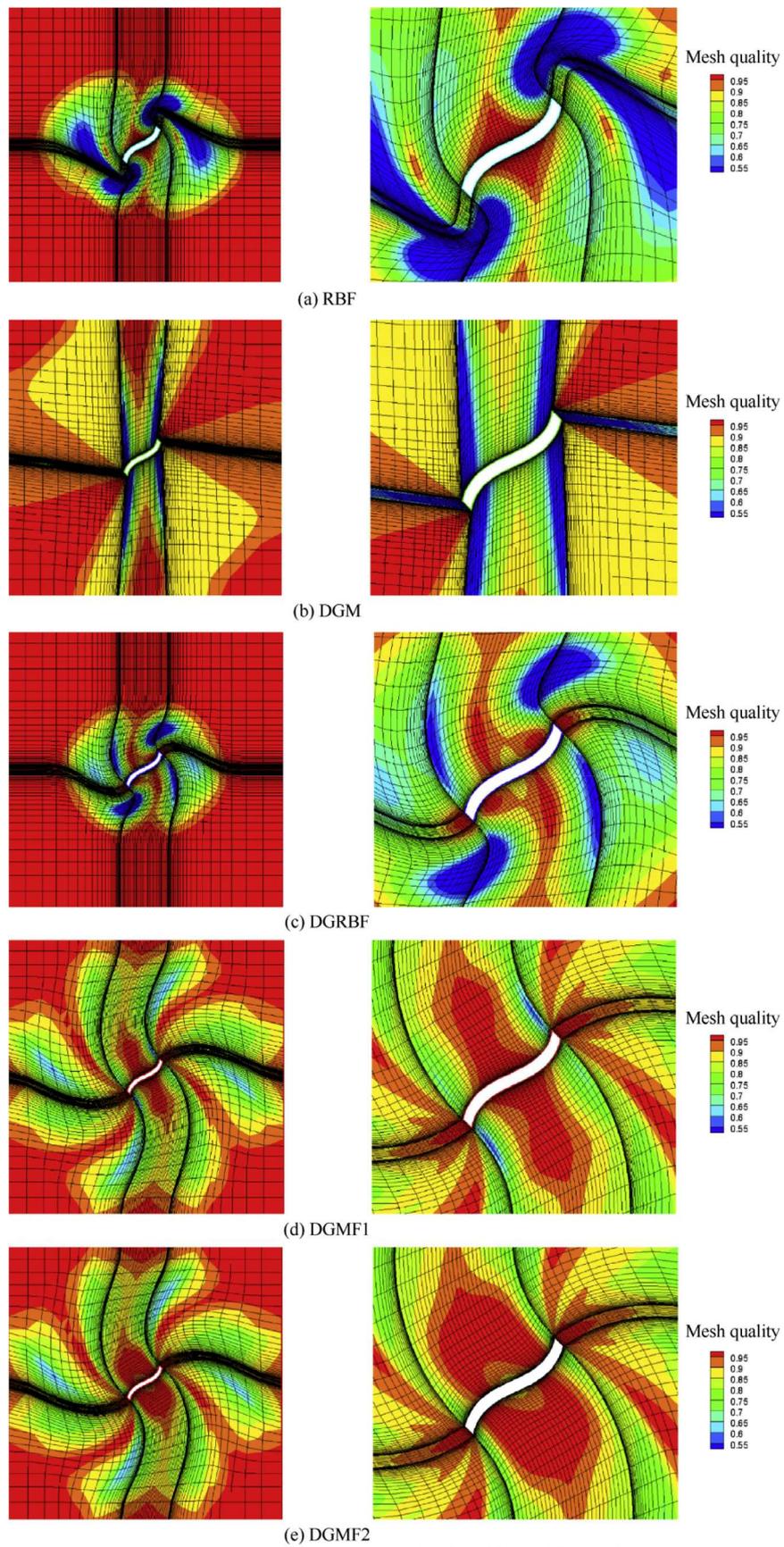


Fig. 14 Mesh quality contour of twisted bar with rotation.

long and 1 unit thick was first twisted and then rotated. The original mesh is shown in Fig. 10. The shape deformation is based on the bump function, which is

$$y_1 = \begin{cases} A \exp\left(\frac{-1}{1 - (\frac{x}{4})^2}\right) + y & -4 \leq x \leq 4 \\ y & \text{else} \end{cases}$$

$$x_1 = x$$

The bar was bended to form a horseshoe shape, and as the parameter  $A$  increases the deformation becomes more severe in Fig. 11. The displacement of the interior node for shape deformation can be computed as

$$s_x = fA \exp\left(\frac{-1}{1 - (\frac{x}{4})^2}\right)$$

or by Eq. (7). Based on the differences between the calculations of the shape deformation displacement, the former one is called DGMF1, and the latter one which uses Eq. (7) is called DGMF2. The overall mesh qualities of the resultant meshes by different methods are illustrated in Figs. 12 and 13, the mesh quality contours are compared for  $A = 16$  case. As the  $A$  increases, both the averaged quality and minimal mesh quality

decrease for all the methods. Especially the minimal mesh quality quickly descends to 0 for the RBF and DGRBF methods, i.e., the meshes become invalid for  $A = 32$  case. The two DGMF methods and the DGM method show similar high averaged mesh quality and minimal mesh quality, for large deformation ( $A = 32$ ), all three methods can still generate valid mesh with proper averaged mesh quality.

The bar was twisted and then rotated by  $45^\circ$ . The deformed mesh was still based on the original mesh, particularly for the DGM, DGRBF, DGMF1 and DGMF2 methods, and the Delaunay graph was the same as the previous twisted case. The overall mesh quality and mesh quality contours are demonstrated in Table 2 and Fig. 14 respectively. Both the RBF and DGM methods fail to preserve the mesh quality near the two ends of the bar. Otherwise, the DGRBF and DGMF1&2 methods well reserve the mesh quality, and show the best averaged and minimal mesh quality among the all. From all the 2D cases, it can be concluded that the DGMF method can well handle the rotation problems, and for the shape deformation, it shows similar quality mesh as the DGM and DGRBF methods do.

### 3.4. 3D test cases

An unstructured mesh with 257909 nodes and 1507462 tetrahedrons based on a wing-body configuration was tested by the proposed DGMF1 method. The initial and deformed configurations are shown in Fig. 15, in which the wing was folded by  $30^\circ$ . The deformation is

$$z_1 = \begin{cases} z_0 + 300\sin 30^\circ & y < -500 \\ z_0 - (y_0 + 200)\sin 30^\circ & -500 \leq y \leq -200 \\ z_0 & \text{else} \end{cases}$$

Therefore the displacement of the interior nodes is

$$s_z = \begin{cases} 300\sin 30^\circ f & y < -500 \\ -(y_0 + 200)\sin 30^\circ f & -500 \leq y \leq -200 \\ 0 & \text{else} \end{cases}$$

The total CPU time for the DGM method is 0.356 s, while for the DGMF1 method, it is 0.356 s, too. The DGMF1 method shows the same efficiency as the DGM method. Further efficiency comparison will be discussed later. Fig. 16 shows the mesh quality contour, where the mesh near the deformed wing is well preserved, and most of the meshes show very good mesh quality.

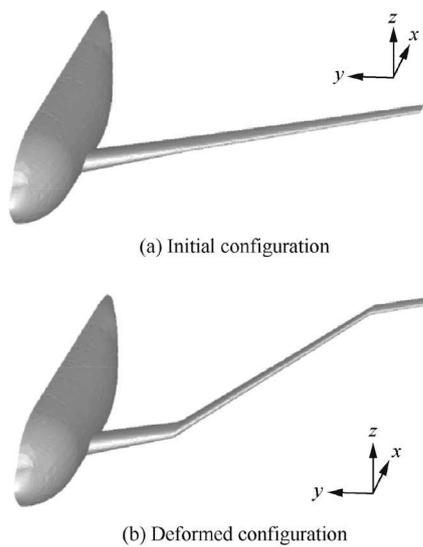


Fig. 15 Wing-body configurations.

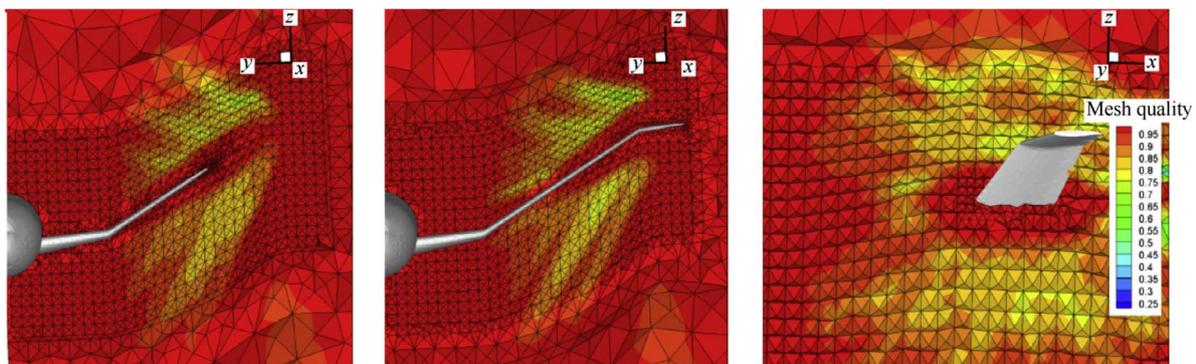


Fig. 16 Mesh quality contour for deformed wing.

**Table 3** Comparison of CPU time.

Mesh size		CPU time (s)			
Total nodes	Total boundary nodes	DGM	DGMF	DGRBF	IDW
201920	10096	0.099	0.099	0.160	485.105
403840	10096	0.172	0.188	0.297	955.096
826080	41304	0.359	0.359	0.578	8295.252
8260800	41304	3.419	3.254	3.482	

The efficiency of the method was also evaluated by different sizes of the meshes. In Table 3 the CPU time calculated by different methods are compared. All the performances are measured on a single core 2.7 GHz Intel i7 processor. In the comparison four different methods are used, namely the DGM, DGMF, DGRBF and IDW. The RBF method is too time-consuming, so it is not included in this comparison, and the detailed information can be found in Ref.<sup>26</sup>. The DGM, DGMF and DGRBF methods show quite similar efficiency, which are significantly better than the IDW method. The DGM and DGMF methods are slightly better than the DGRBF method. The CPU time of the three Delaunay graph-based methods increases almost linearly as the total nodes of the mesh increase; while for the IDW method, it increases as the product of increase in the total nodes and total boundary nodes. The required CPU time for the Delaunay graph-based methods can be roughly 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  $N_{bp}$  refers to the number of boundary points. The second part is the identification process which is  $N_s N_{vp}$  where  $N_s$  is the number of searches (according to the numerical test, the  $N_s$  is normally less than 10 for 3D)<sup>26</sup> and  $vp$  stands for volume points. The computational costs of these two parts are exactly the same for all the Delaunay graph-based methods. The third part is the mapping process in which the cost is  $N_{vp}$  for all the Delaunay graph-based methods. The cost of the IDW method, however, is  $N_{bp} N_{vp}$ . From the perspective of computational complexity, it is clear that the costs of the Delaunay graph-based methods are similar but much lower than that of the IDW method. Thus it can be concluded that the Delaunay graph-based methods are much more efficient than the IDW method.

#### 4. Conclusions

A novel dynamic method based on the Delaunay graph was developed. This method can deform the mesh with both high quality and high efficiency, especially for large rotation problem. A range of large deformation and rotation test cases shows that this method can preserve the mesh quality near the boundary for both structured and unstructured mesh. This method inherits the advantages of the efficiency of the Delaunay graph method while addressing its robustness problem for large rotation. From the large 3D test cases, the required CPU time using this method increases almost linearly as the total node number increases, indicating its suitability for extremely large mesh computations.

#### Acknowledgements

This work is partially funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions

of China, National Natural Science Foundation of China (No. 11432007) and Natural Science Foundation of Jiangsu Province of China (No. BK20140805).

#### References

- Masud A, Bhanabagwanwala M, Khurram RA. An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction. *Comput Fluids* 2007;**36**(1):77–91.
- Anderson WK, Karman SL, Burdyslaw C. Geometry parameterization method for multidisciplinary applications. *AIAA J* 2009;**47**(6):1568–78.
- Burg C. Analytic study of 2D and 3D grid motion using modified Laplacian. *Int J Numer Meth Fluids* 2006;**52**(2):163–97.
- Helenbrook BT. Mesh deformation using the biharmonic operator. *J Numer Meth Eng* 2003;**56**(7):1007–21.
- Batina JT. Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA J* 1991;**29**(3):327–33.
- Loehner R, Yang C. Improved ALE mesh velocities for moving bodies. *Commun Numer Meth En* 1996;**12**(10):599–608.
- Bau JD, Luo H, Loehner R, Goldberg E, Feldhun A. Application of unstructured moving body methodology to the simulation of fuel tank separation from an F-16 fighter. Reston: AIAA; 1997. Report No.: AIAA-1997-0166.
- Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Comput Method Appl Math* 1998;**163**:231–45.
- de Boer A, van der Schoot M, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct* 2007;**85**(11):784–95.
- Wang G, Chen X, Liu Z. Mesh deformation on 3D complex configurations using multistep radial basis functions interpolation. *Chin J Aeronaut* 2018;**31**(4):660–71.
- Liu Y, Guo Z, Liu J. RBFs-MSA hybrid method for mesh deformation. *Chin J Aeronaut* 2012;**25**(4):500–7.
- Witteveen JAS, Bijl H. Explicit mesh deformation using inverse distance weighting interpolation. Reston: AIAA; 2009. Report No.: AIAA-2009-3996.
- Witteveen JAS. Explicit and robust inverse distance weighting mesh deformation for CFD. Reston: AIAA; 2010. Report No.: AIAA-2010-0165.
- Rendall T, Allen C. Efficient mesh motion using radial basis functions with data reduction algorithms. *J Comput Phys* 2009;**228**(17):6231–49.
- Rendall T, Allen C. Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors. *Int J Numer Meth Eng* 2010;**81**(1):89–105.
- Luke E, Collins E, Blades E. A fast mesh deformation method using explicit interpolation. *J Comput Phys* 2012;**231**(2):586–601.
- Zhou X, Li S. A new mesh deformation method based on disk relaxation algorithm with pre-displacement and post-smoothing. *J Comput Phys* 2013;**235**(2):199–215.
- Zhou X, Li S. A novel three-dimensional mesh deformation method based on sphere relaxation. *J Comput Phys* 2015;**298**:320–36.

19. Nakamichi J. Calculation of unsteady Navier-Stokes equations around an oscillating 3D wing using moving grid system. Reston: AIAA; 1987. Report No.: AIAA-1987-1158.
20. Rumsey CL, Anderson WK. Some numerical and physical aspects of unsteady Navier-stokes computations over airfoils using dynamic meshes. Reston: AIAA; 1988. Report No.: AIAA-1988-0329.
21. Morton SA, Melville RB, Visbal MR. Accuracy and coupling issues of aeroelastic Navier-stoke solution on deforming meshes. *J Aircraft* 1998;**35**(5):798–805.
22. Liu X, Qin N, Xia H. Fast dynamic grid deformation based on Delaunay graph mapping. *J Comput Phys* 2006;**211**(2):405–23.
23. van Der Burg JW, Freiherr Von Geyr H, Heinrich R, Eliasson P, Delille T, Krier J. Geometrical installation and deformation effects in high-lift flows. *AIAA J* 2009;**47**(1):60–70.
24. Zhang L, Chang X, Duan X, He Z. Applications of dynamic hybrid grid method for three-dimensional moving/deforming boundary problems. *Comput Fluids* 2012;**62**(15):45–63.
25. Wang H, Leskinen J, Lee D. Active flow control of airfoil using mesh/meshless methods coupled to hierarchical genetic algorithms for drag reduction design. *Eng Comput* 2013;**30**(4):562–80.
26. Wang Y, Qin N, Zhao N. Delaunay graph and radial basis function for fast quality mesh deformation. *J Comput Phys* 2015;**294**:149–72.
27. Knupp PM. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des* 2003;**39**(3):217–41.