Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **6**, No. 1, pp. 120-134

DOI: 10.4208/aamm.2013.m199
February 2014

An Efficient Dynamic Mesh Generation Method for Complex Multi-Block Structured Grid

Li Ding, Zhiliang Lu* and Tongqing Guo

Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, 29 Yudao Street, Nanjing 210016, Jiangsu, China

Received 22 April 2013; Accepted (in revised version) 18 July 2013

Available online 13 December 2013

Abstract. Aiming at a complex multi-block structured grid, an efficient dynamic mesh generation method is presented in this paper, which is based on radial basis functions (RBFs) and transfinite interpolation (TFI). When the object is moving, the multi-block structured grid would be changed. The fast mesh deformation is critical for numerical simulation. In this work, the dynamic mesh deformation is completed in two steps. At first, we select all block vertexes with known deformation as center points, and apply RBFs interpolation to get the grid deformation on block edges. Then, an arc-length-based TFI is employed to efficiently calculate the grid deformation on block faces and inside each block. The present approach can be well applied to both two-dimensional (2D) and three-dimensional (3D) problems. Numerical results show that the dynamic meshes for all test cases can be generated in an accurate and efficient manner.

AMS subject classifications: 65Z05

Key words: Multi-block structured grid, mesh deformation, radial basis functions, transfinite interpolation.

1 Introduction

Currently, the use of unstructured grid is very popular in Computational Fluid Dynamics (CFD) due to its great flexibility and easy generation. However, the unstructured grid suffers the difficulty of capturing the thin boundary layer on the solid surface, especially at high Reynolds numbers. In contrast, the structured grid is preferred to capture the boundary layer. The structured grid is usually associated with coordinate transformation. When a complex geometry such as aircraft is considered, a single-block structured grid is very difficult to be generated. Instead, the multi-block structured grid is often used, in which, the whole computational domain is divided into a number of blocks, and

*Corresponding author.
Email: luzl@nuaa.edu.cn (Z. L. Lu)

the structured grid is generated in each block. Clearly, the grid in each block is structured and the block edges are unstructured. When the complex object is moving, the dynamic change of resultant multi-block structured grid is still a challenging issue in CFD. Basically, there are two major approaches for the dynamic change of mesh. They are mesh deformation approach and mesh reconstruction approach. As compared with mesh reconstruction, mesh deformation is more efficient and accurate, and is more widely employed.

For a single structured mesh deformation, Transfinite Interpolation [1–6] (TFI) method is most frequently used, which generates the dynamic mesh with an algebraic interpolation. The computational effort of TFI is proportional to the grid number. So its efficiency is still very high when the grid number is large. However, TFI is very difficult to be applied for a complex geometry when a multi-block structured grid is used. This is because the mesh points on block edges are irregularly distributed when the mesh is deformed.

On the other hand, it is found that Radial Basis Functions [7–9] (RBFs) interpolation is a desirable approach for scattered data interpolation, which could be used to compute the mesh deformation on block edges. As a new interpolation method, RBFs method has been successfully applied to the multivariate interpolation in a fluid-structure-interaction problem [10], the numerical simulation of a flapping foil [11] and the aerodynamic shape optimization [12, 13]. Based on RBFs, Rendall and Allen [14] proposed a unified algorithm for the fluid-structure interpolation and mesh motion. In RBF interpolation, the computational effort is usually proportional to the cube of the total grid number. Thus, when RBF interpolation is applied to the whole grid, the computational effort for mesh deformation will be extremely large for a complex problem with a large grid number. Some attempts [15, 16] have been made with a reduced size of equation system to improve the efficiency of RBF interpolation. Nevertheless, the computational effort is still very large.

In this paper, a hybrid mesh deformation method, which is especially suitable for a complex multi-block structured grid, is developed. In the approach, the RBF interpolation is only utilized to generate the dynamic mesh on block edges, and TFI is adopted to generate structured mesh on block faces and within each block. The present hybrid method is firstly applied to the dynamic mesh generation for one two-dimensional (2D) and one three-dimensional (3D) problems with different complexity. Then numerical simulation of a 2D unsteady flow is conducted. Based on the given multi-block structured grids, the numerical results demonstrate that the dynamic meshes for all test cases can be generated in an accurate and efficient manner by using present method.

2 Mesh deformation on block edges by RBFs interpolation

2.1 RBFs interpolation

RBFs [17] use the interpolation function f to describe the displacement in the whole physical space, and f can be approximated by a weighted sum of the basis functions

$$f(x)=\sum_{i=1}^{N_b}a_i\cdot\varphi\left(\|\mathbf{x}-\mathbf{x}_{bi}\|\right)+\psi(\mathbf{x}), \tag{2.1}$$

where $\mathbf{x}_{bi}=[x_{bi},y_{bi},z_{bi}]$ are the coordinates of the selected center points with known displacements, N_b is the number of center points, φ is a given basis function with respect to the Euclidean distance $\|\bullet\|$ and $\psi=b_0+b_1x+b_2y+b_3z$ is a linear polynomial. The coefficients a_i and b_0, b_1, b_2, b_3 are evaluated from the collocation condition and additional requirement, which are formulated as follows

$$f(\mathbf{x}_{bi})=\mathbf{d}_{bi},\quad \sum_{i=1}^{N_b}a_i=\sum_{i=1}^{N_b}a_ix_{bi}=\sum_{i=1}^{N_b}a_iy_{bi}=\sum_{i=1}^{N_b}a_iz_{bi}=0, \tag{2.2}$$

where $\mathbf{d}_{bi}$ is the discrete known displacement on the boundary. Then, the coefficients a_i and b_0, b_1, b_2, b_3 can be obtained by solving the following equation system

$$\begin{bmatrix}\mathbf{M} & \mathbf{P}\\ \mathbf{P}^T & 0\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}=\begin{bmatrix}\mathbf{d}_b\\ 0\end{bmatrix},\quad \mathbf{M}=\begin{bmatrix}\varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n}\\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \varphi_{n1} & \varphi_{n2} & \cdots & \varphi_{nn}\end{bmatrix}, \tag{2.3}$$

where

$$\mathbf{P}=\begin{bmatrix}1 & x_{b1} & y_{b1} & z_{b1}\\ 1 & x_{b2} & y_{b2} & z_{b2}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_{bn} & y_{bn} & z_{bn}\end{bmatrix},\quad \mathbf{a}=\begin{bmatrix}a_1\\ a_2\\ \vdots\\ a_n\end{bmatrix},\quad \mathbf{b}=\begin{bmatrix}b_0\\ b_1\\ b_2\\ b_3\end{bmatrix},\quad \mathbf{d}_b=\begin{bmatrix}d_{b1}\\ d_{b2}\\ \vdots\\ d_{bn}\end{bmatrix}. \tag{2.4}$$

$\varphi_{ij}=\varphi(\|\mathbf{x}_{bi}-\mathbf{x}_{bj}\|)$. There are several mature methods to solve Eq. (2.3), such as the fast iterative techniques [18] and methods based on partition of unity [19].

After the interpolation function f is determined, the displacement $\mathbf{d}_{inter}$ for any interior point $\mathbf{x}_{inter}=[x_{inter},y_{inter},z_{inter}]$ can be calculated by

$$\mathbf{d}_{inter}=f(\mathbf{x}_{inter}). \tag{2.5}$$

The available basis functions can be divided into two main categories: functions with compact and functions with global support [20]. Boer has discussed the effects of different basis functions on mesh quality in [21]. Table 1 lists several frequently used RBFs with global support [22,23]. RBFs with spline type function $f(x)=\|x\|^3$ is adopted in this work.

Table 1: RBFs with global support.

No.	Name	Abbrevi	$f(x)$
1	Spline type	R_n	$\|x\|^3$
2	Thin plate spline	TPS	$\|x\|^2\log\|x\|$
3	Multiquadric biharmonics	MQB	$\sqrt{a^2+\|x\|^2}$
4	Inverse multiquadric	IMQB	$\sqrt{1+(a^2+\|x\|^2)}$
5	Quadric biharmonics	QB	$1+\|x\|^2$
6	Inverse quadric	IQB	$1/(1+\|x\|^2)$
7	Gaussian	Gauss	$e^{-\|x\|^2}$

2.2 Mesh deformation on block edges

2.2.1 Choice of center points for RBFs interpolation

Conventionally, all the grid points with known displacements are selected as the center points. Because the number of center points directly controls the size of matrix **M** in Eq. (2.3), the computational efficiency is very low in the case of a large number of center points, such as the multi-block grid for a complex geometry. In this work, only the block vertexes with known displacements are selected as center points. This can greatly speed up the process of mesh deformation. An example is shown in Fig. 1. When all the grid points with known displacements (red points) are selected as the center points, there are 504 center points. On the other hand, when only the block vertexes with known displacements are selected as center points, there are only 16 center points. The corresponding sizes of matrix **M** are respectively 507×507 for the former and 19×19 for the latter. As a consequence, their calculation time differs about 19,000 times.

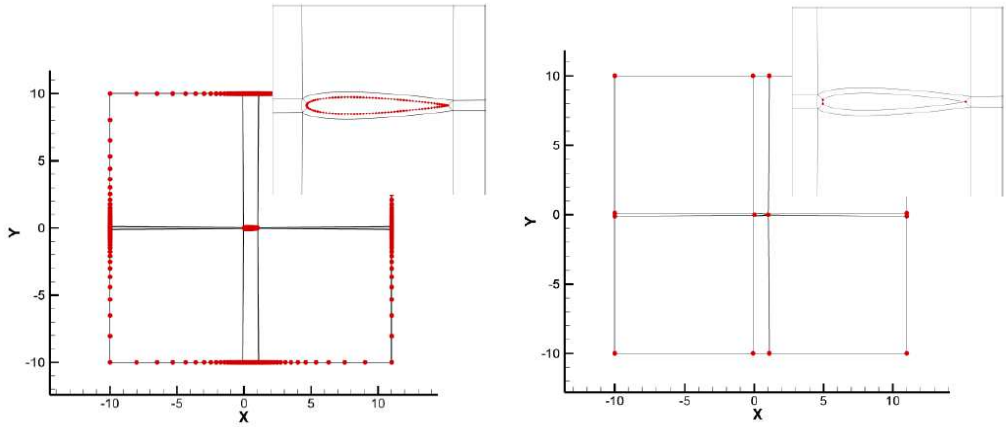
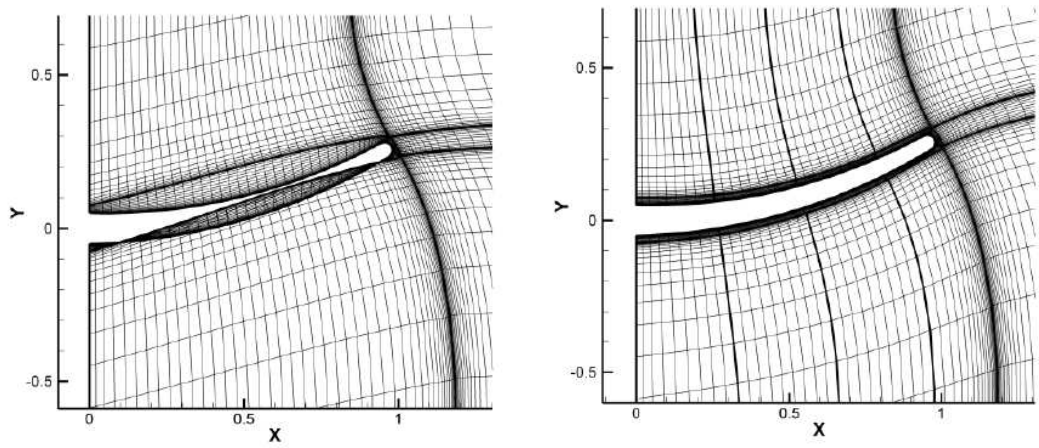


Figure 1: Selection of center points: left is all points at the boundary and right is block vertexes at the boundary.

2.2.2 Block division

As computational efficiency greatly depends on the number of center points, it is unpractical to select all points on the boundary as center points. Thus, only block vertexes are chosen as RBF center points in this work. However, if the block division is not properly made, the displacement information of boundary may not be reflected in RBF interpolation very well, and this may lead to errors in interpolation. For cases with relative movement of some bodies or boundary with soft deformation, blocks should be divided to assure that the displacement information of boundary can be considered commendably. As a rule of thumb, the block division principle is to keep the dimension of block in deformation direction $0.2\sim0.3$ times greater than the dimension of block in other directions. For the left picture in Fig. 2, before block division, the shape of block around boundary is prolate, since the deformation direction of block is Y direction, grid intersection appears after interpolation. As shown in the right picture of Fig. 2, a very good mesh quality is obtained after all the grid blocks surrounding the airfoil are subdivided into several blocks along the chord wise.



(a) (b)

Figure 2: Deformed meshes before and after grid block subdivision.

2.2.3 Mesh generation on block edges

The blue point in Fig. 3 is a point on one of red curves whose displacements are unknown. Now set all block vertex points (black points) as center points and blue point as interpolative dot, the displacement of blue point can be calculated by all black points easily with RBF interpolation method. Similarly, other grid points on this edge and other edges can be worked out one by one. After that, deformation of all block edges (green curves and red curves) is generated as shown in Fig. 4.

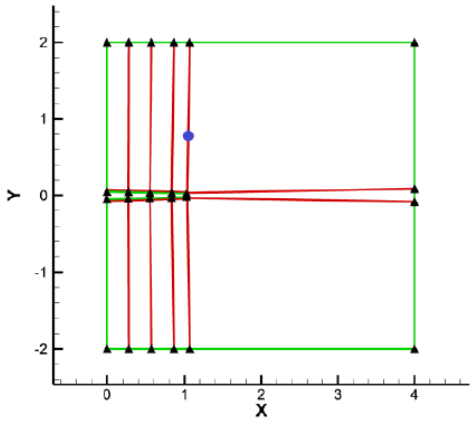
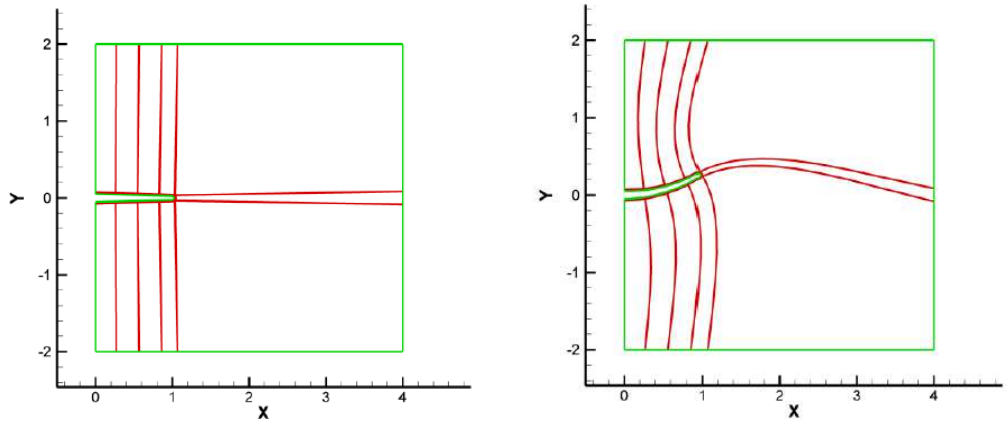


Figure 3: Generation of mesh deformation on block edge.



(a) (b)

Figure 4: Deformed block edges before and after RBFs interpolation.

3 Mesh deformation on block faces and within each block by TFI interpolation

3.1 Transfinite interpolation

Transfinite interpolation [24] means the construction of a function over a planar domain or a three-dimensional space that matches a given function on the boundary, and has various applications, notably in geometric modeling and finite element methods.

By using the Boolean sum of interpolation functions, a 3-D TFI can be written as

$$\Delta \mathbf{x}_{i,j,k} = U+V+W-UV-VW-UW+UVW, \tag{3.1}$$

where the grid deformation is defined as

$$\Delta\mathbf{x}_{i,j,k}=\left[\Delta x_{i,j,k},\Delta y_{i,j,k},\Delta z_{i,j,k}\right]. \tag{3.2}$$

The univariate products are

$$U=(1-\alpha_{i,j,k})\Delta\mathbf{x}_{1,j,k}+\alpha_{i,j,k}\Delta\mathbf{x}_{i\max,j,k}, \tag{3.3a}$$
$$V=(1-\beta_{i,j,k})\Delta\mathbf{x}_{i,1,k}+\beta_{i,j,k}\Delta\mathbf{x}_{i,j\max,k}, \tag{3.3b}$$
$$W=(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i,j,1}+\gamma_{i,j,k}\Delta\mathbf{x}_{i,j,k\max}, \tag{3.3c}$$

and the tensor products are evaluated as follows

$$\begin{aligned}UV=&(1-\alpha_{i,j,k})(1-\beta_{i,j,k})\Delta\mathbf{x}_{1,1,k}+\alpha_{i,j,k}(1-\beta_{i,j,k})\Delta\mathbf{x}_{i\max,1,k}+(1-\alpha_{i,j,k})\beta_{i,j,k}\Delta\mathbf{x}_{1,j\max,k}\\&+\alpha_{i,j,k}\beta_{i,j,k}\Delta\mathbf{x}_{i\max,j\max,k},\end{aligned} \tag{3.4a}$$
$$\begin{aligned}VW=&(1-\beta_{i,j,k})(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i,1,1}+\beta_{i,j,k}(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i,j\max,1}+(1-\beta_{i,j,k})\gamma_{i,j,k}\Delta\mathbf{x}_{i,1,k\max}\\&+\beta_{i,j,k}\gamma_{i,j,k}\Delta\mathbf{x}_{i,j\max,k\max},\end{aligned} \tag{3.4b}$$
$$\begin{aligned}UW=&(1-\alpha_{i,j,k})(1-\gamma_{i,j,k})\Delta\mathbf{x}_{1,j,1}+\alpha_{i,j,k}(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i\max,j,1}+(1-\alpha_{i,j,k})\gamma_{i,j,k}\Delta\mathbf{x}_{1,j,k\max}\\&+\alpha_{i,j,k}\gamma_{i,j,k}\Delta\mathbf{x}_{i\max,j,k\max},\end{aligned} \tag{3.4c}$$
$$\begin{aligned}UVW=&(1-\alpha_{i,j,k})(1-\beta_{i,j,k})(1-\gamma_{i,j,k})\Delta\mathbf{x}_{1,1,1}+\alpha_{i,j,k}(1-\beta_{i,j,k})(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i\max,1,1}\\&+(1-\alpha_{i,j,k})\beta_{i,j,k}(1-\gamma_{i,j,k})\Delta\mathbf{x}_{1,j\max,1}+(1-\alpha_{i,j,k})(1-\beta_{i,j,k})\gamma_{i,j,k}\Delta\mathbf{x}_{1,1,k\max}\\&+\alpha_{i,j,k}\beta_{i,j,k}(1-\gamma_{i,j,k})\Delta\mathbf{x}_{i\max,j\max,1}+\alpha_{i,j,k}(1-\beta_{i,j,k})\gamma_{i,j,k}\Delta\mathbf{x}_{i\max,1,k\max}\\&+(1-\alpha_{i,j,k})\beta_{i,j,k}\gamma_{i,j,k}\Delta\mathbf{x}_{1,j\max,k\max}+\alpha_{i,j,k}\beta_{i,j,k}\gamma_{i,j,k}\Delta\mathbf{x}_{i\max,j\max,k\max}.\end{aligned} \tag{3.4d}$$

In order to maintain the grid distribution laws of the initial grid, an arc-length-based TFI method [25] is applied in this work. The arc-length-based control functions along i, j, k directions are defined as

$$\alpha_{1,j,k}=0, \qquad \beta_{i,1,k}=0, \tag{3.5a}$$
$$\gamma_{i,j,1}=0, \qquad \alpha_{i,j,k}=\frac{\sum_{m=2}^{i}\left\|\mathbf{x}_{m,j,k}-\mathbf{x}_{m-1,j,k}\right\|}{\sum_{m=2}^{i\max}\left\|\mathbf{x}_{m,j,k}-\mathbf{x}_{m-1,j,k}\right\|}, \tag{3.5b}$$
$$\beta_{i,j,k}=\frac{\sum_{m=2}^{j}\left\|\mathbf{x}_{i,m,k}-\mathbf{x}_{i,m-1,k}\right\|}{\sum_{m=2}^{j\max}\left\|\mathbf{x}_{i,m,k}-\mathbf{x}_{i,m-1,k}\right\|}, \qquad \gamma_{i,j,k}=\frac{\sum_{m=2}^{k}\left\|\mathbf{x}_{i,j,m}-\mathbf{x}_{i,j,m-1}\right\|}{\sum_{m=2}^{k\max}\left\|\mathbf{x}_{i,j,m}-\mathbf{x}_{i,j,m-1}\right\|}. \tag{3.5c}$$

3.2 Structured mesh on block faces

After computing the displacement of mesh points on all block edges by RBF interpolation method, the displacement of grid points in the interior of faces is computed by a 2-D TFI method as described below. Since the interpolation is a two-dimensional one, all terms related to **W** in Eq. (3.1) are removed in this sub-section. Take Fig. 5 as an example. For any structured face, displacements of point $P(i,j)$ are computed by 8 points.

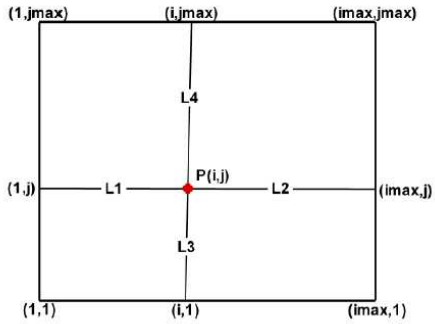


Figure 5: 2-D TFI sketch.

The specific interpolation functions are

$$\Delta\mathbf{x}_{i,j} = U + V - UV, \tag{3.6}$$

where U, V and UV are defined as follows

$$U = L_2\Delta\mathbf{x}_{1,j} + L_1\Delta\mathbf{x}_{i\max,j}, \quad V = L_4\Delta\mathbf{x}_{i,1} + L_3\Delta\mathbf{x}_{i,j\max}, \tag{3.7a}$$

$$UV = L_2L_4\Delta\mathbf{x}_{1,1} + L_1L_4\Delta\mathbf{x}_{i\max,1} + L_2L_3\Delta\mathbf{x}_{1,j\max} + L_1L_3\Delta x_{i\max,j\max}, \tag{3.7b}$$

and L is the arc length.

3.3 Structured mesh within each block

After computing the displacement of mesh points on all the faces, the TFI interpolation method is also used to compute the displacement of mesh points in the interior of each block. For each single structured block, displacements of point $P(i,j,k)$ are calculated by 26 points in the block as shown in Fig. 6.

The coordinates of mesh points can be computed by Eq. (3.1), but relevant functions are specifically defined as

$$U = L_2\Delta\mathbf{x}_{1,j,k} + L_1\Delta\mathbf{x}_{i\max,j,k}, \tag{3.8a}$$

$$V = L_4\Delta\mathbf{x}_{i,1,k} + L_3\Delta\mathbf{x}_{i,j\max,k}, \tag{3.8b}$$

$$W = L_6\Delta\mathbf{x}_{i,j,1} + L_5\Delta\mathbf{x}_{i,j,k\max}, \tag{3.8c}$$

$$UV = L_2L_4\Delta\mathbf{x}_{1,1,k} + L_1L_4\Delta\mathbf{x}_{i\max,1,k} + L_2L_3\Delta\mathbf{x}_{1,j\max,k} + L_1L_3\Delta\mathbf{x}_{i\max,j\max,k}, \tag{3.8d}$$

$$VW = L_4L_6\Delta\mathbf{x}_{i,1,1} + L_3L_6\Delta\mathbf{x}_{i,j\max,1} + L_4L_5\Delta\mathbf{x}_{i,1,k\max} + L_3L_5\Delta\mathbf{x}_{i,j\max,k\max}, \tag{3.8e}$$

$$UW = L_2L_6\Delta\mathbf{x}_{1,j,1} + L_1L_6\Delta\mathbf{x}_{i\max,j,1} + L_2L_5\Delta\mathbf{x}_{1,j,k\max} + L_1L_5\Delta\mathbf{x}_{i\max,j,k\max}, \tag{3.8f}$$

$$\begin{aligned} UVW = {} & L_2L_4L_6\Delta\mathbf{x}_{1,1,1} + L_1L_4L_6\Delta\mathbf{x}_{i\max,1,1} + L_2L_3L_6\Delta\mathbf{x}_{1,j\max,1} \\ & + L_2L_4L_5\Delta\mathbf{x}_{1,1,k\max} + L_1L_3L_6\Delta\mathbf{x}_{i\max,j\max,1} + L_1L_4L_5\Delta\mathbf{x}_{i\max,1,k\max} \\ & + L_2L_3L_5\Delta\mathbf{x}_{1,j\max,k\max} + L_1L_3L_5\Delta\mathbf{x}_{i\max,j\max,k\max}, \end{aligned} \tag{3.8g}$$

and L is the arc length.

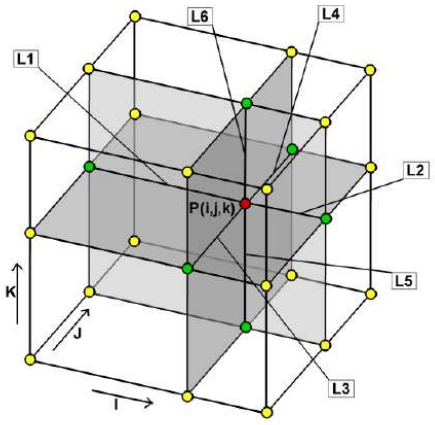


Figure 6: 3-D TFI sketch.

4 Numerical tests

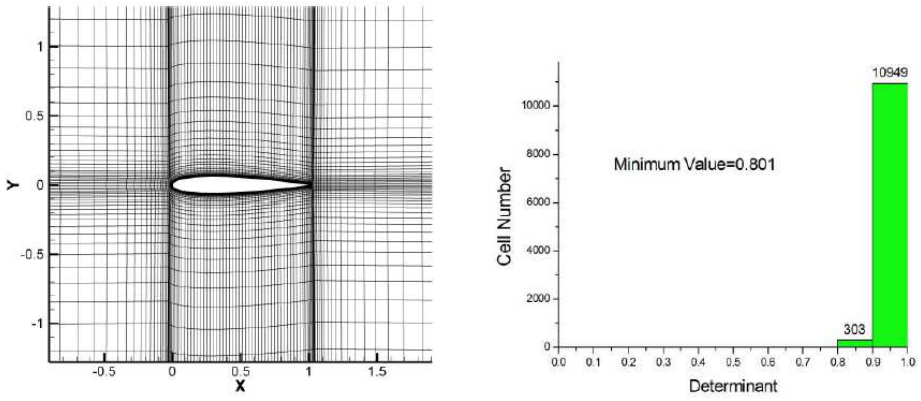
The present RBFs-TFI hybrid method has been applied to one 2D and one 3D mesh deformation problems with different complexity: a rigid airfoil and a wing-body combination. After that, simulation of unsteady inviscid flows around a moving airfoil is performed to show the accuracy of this moving mesh method. Based on the given multi-block grids, the dynamic meshes for the above test cases have been generated in an accurate and efficient manner. All the CPU times are taken from a single 2.93GHz Intel processor.

4.1 Determinant

In order to compare the dynamic mesh quality with initial mesh, a parametric evaluation criterion should be introduced. In this paper, mesh quality inspection process is accomplished by software "ANSYS ICEM CFD". ICEM CFD is one of the most popular mesh generation softwares in the world. There are many mesh checking methods in ICEM CFD [26], such as Quality, Aspect Ratio and Determinant. The Determinant [27], more properly defined as the relative determinant, is the ratio of the smallest determinant of the Jacobian matrix divided by the largest determinant of the Jacobian matrix, where each determinant is computed at each node of the element. The Determinant can be found for all linear hexahedral, quadrahedral, and pyramidal elements. A Determinant value of 1 would indicate a perfectly regular mesh element, 0 would indicate an element degenerating in one or more edges, and negative values would indicate inverted elements.

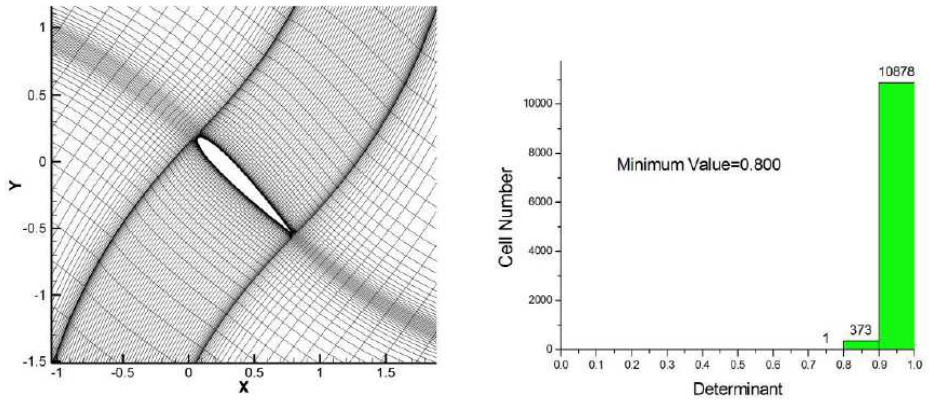
4.2 Test case 1: rigid airfoil

Test case 1 is a NACA0012 airfoil rotating 45 degrees about its quarter-chord point. For this case, the number of grid blocks is 12, and the number of grid cells is 11,252.



(a) (b)

Figure 7: Initial mesh and mesh quality of NACA0012.



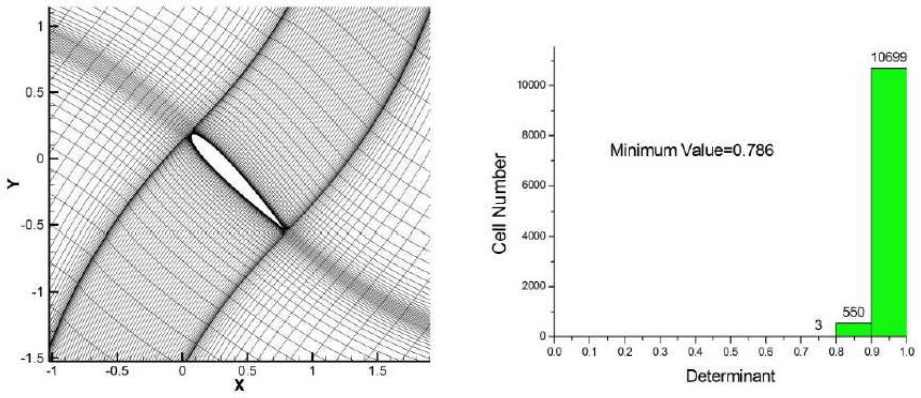
(a) (b)

Figure 8: Deformed mesh and mesh quality of NACA0012 using RBFs method.

Fig. 7 shows the initial mesh and its mesh quality histogram. Both RBFs and present RBFs-TFI hybrid method are applied for the dynamic mesh generation. As shown in Fig. 8 and Fig. 9, the mesh quality of the RBFs-TFI hybrid method is comparable with RBFs. On the other hand, the CPU-time required by RBFs is 1,217ms, and it is only 4ms needed by the RBFs-TFI hybrid method which is about 0.32% of the former. This test case indicates that the RBFs-TFI hybrid method is as accurate as RBFs, but it is much more efficient.

4.3 Test case 2: wing-body

Test case 2 is a wing-body combination of a DLR-F4 airplane. For this case, the number of grid blocks is 318, and the number of grid cells is 2,565,588. For this case, the wing is



(a) (b)

Figure 9: Deformed mesh and mesh quality of NACA0012 using RBFs-TFI hybrid method.

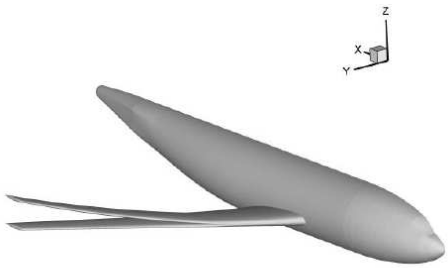


Figure 10: Initial and deformed configurations of DLR-F4.

undergone a vertical bending deformation and the maximum deformation at the wingtip is 10% of the semi-wingspan. Fig. 10 shows the initial and deformed configurations

For the RBFs method, a 2-D problem selects all block edge points with known displacements as the center points and the CPU-time cost is about 2∼3 orders of magnitude higher than present RBFs-TFI hybrid method, and a 3-D problem will select all block face points as the center points which is too large to lead to a unbearable CPU-time. Therefore, only the RBFs-TFI hybrid method is used for the 3-D problem.

Fig. 11 shows the final surface and symmetry plane mesh. Fig. 12 and Fig. 13 illustrate the mesh quality for the initial and dynamic mesh, respectively. The comparison indicates that the dynamic mesh quality almost maintains the same level as the initial mesh. On the other hand, the CPU-time cost by the RBFs-TFI hybrid method is only 2,169ms for this complex multi-block grid with millions of grid cells.

4.4 Test case 3: unsteady inviscid flow around airfoil

The third test case is numerical simulation of an unsteady flow associated with mesh deformation. For this case, a NACA0012 airfoil is forced to pitch around the quarter

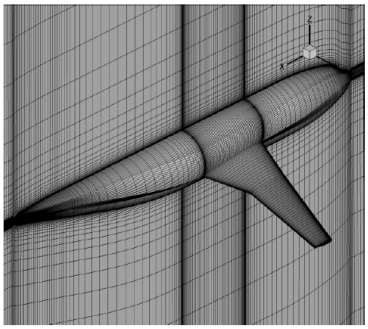


Figure 11: Deformed surface and symmetry plane mesh of DLR-F4.

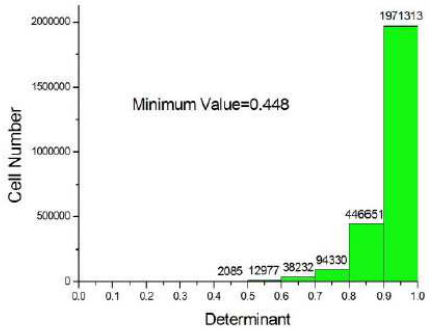


Figure 12: Initial mesh quality of DLR-F4.

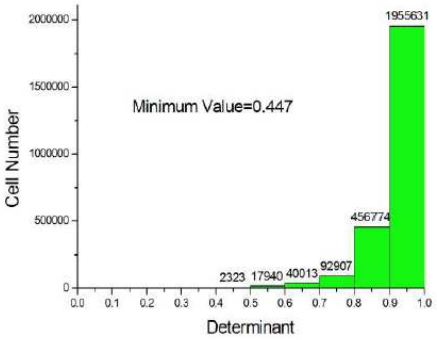


Figure 13: Deformed mesh quality of DLR-F4.

chord with a reduced frequency of 0.0814 [28]. The form of pitching motion is given as follows

$$\alpha = 0.016^\circ + 2.51^\circ \sin \omega t, \tag{4.1}$$

where the angle of attack is α, ω is the reduced frequency, t is the time, and the Mach number M_a is 0.755. In this case, only Euler equations are solved and the mesh has

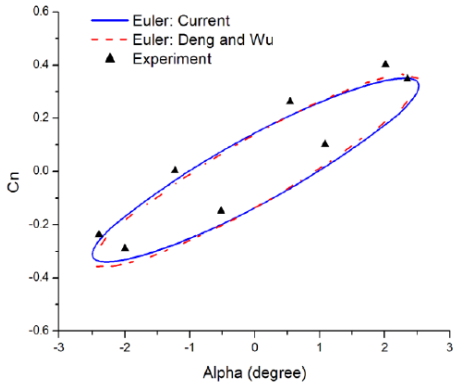


Figure 14: Calculated normal force coefficient by solving Euler equations.

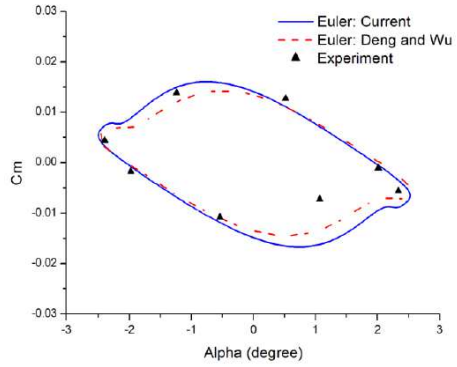


Figure 15: Calculated pitching moment coefficient by solving Euler equations.

been used in test case 1. The normal force coefficient and pitching moment coefficient calculated by Euler equations are compared with the reference data calculated by Deng et al. [29] and the experimental data [30] as depicted in Fig. 14 and Fig. 15.

Overall, the present numerical results in Fig. 14 and Fig. 15 have a good agreement with the date in literature. This implies that the present hybrid method can be effectively applied to solve the real complicated flow problems.

5 Conclusions

By combining the most favorable elements of RBFs and TFI, a novel hybrid mesh deformation method is proposed for a complex multi-block structured grid. By using the present RBFs-TFI hybrid method, the dynamic mesh generation with millions of grid cells can be efficiently achieved in several seconds. In the future work, the present hybrid mesh deformation method will be applied to aeroelastic problems and optimization design problems of airplane.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11372135), the National Basic Research Program of China ("973" Project) (Grant No. 2014CB046200) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] J. Reuther, A. Jameson, J. Farmer, L. Martinelli and D. Saunders, *Aerodynamics shape optimization of complex aircraft configurations via an adjoint formulation*, AIAA, 96-0094.
[2] C. Byun and G. P. Guruswamy, *A parallel multi-block moving grid method for aeroelastic applications on full aircraft*, AIAA, 98-4782.
[3] H. M. Tsai, A. S. F. Wong, J. Cai, Y. Zhu and F. Liu, *Unsteady flow calculations with a parallel multiblock moving mesh algorithm*, AIAA J., 39 (2001), pp. 1021–1029.
[4] A. L. Gaitonde, D. P. Jones and S. P. Fiddes, *A 2D Navier-Stokes method for unsteady compressible flow calculations on moving meshes*, Aeronautical J., 102 (1998), pp. 89–97.
[5] L. Dubuc, F. Gantariti, M. Woodgate, B. Gribben, K. J. Badcock and B. E. Richards, *A grid deformation technique for unsteady flow conputations*, Int. J. Numer. Methods Fluids, 32 (2000), pp. 285–311.
[6] S. A. Morton, R. B. Melville and M. R. Visbal, *Accuracy and coupling issues of aeroelastic Navier-Stokes solutions on deforming meshes*, J. Aircraft, 35 (1998), pp. 798–811.
[7] R. Frank, *Scattered data interpolation: Tests of some methods*, Math. Comput., 38 (1982), pp. 181–200.
[8] Z. M. Wu, *Multivariate compactly supported positive definite radial functions*, Adv. Comput. Math., 4 (1995), pp. 283–292.
[9] F. Bernal and G. Gutierrez, *Solving delay differential equations through BRF collocation*, Adv. Appl. Math. Mech., 1 (2009), pp. 257–272.
[10] A. Beckert and H. Wenland, *Multivariate interpolation for fluid-structure-interaction problems using radial basis functions*, Aerosp. Sci. Technol., 5 (2001), pp. 125–134.
[11] F. M. Bos, Numerical Simulation of Flapping Foil and Wind Aerodynamics: Mesh Deformation using Radial Basis Functions, Ph. D. Thesis, Dutch: Technical University Delft, 2009.
[12] S. Jakobsson and O. Amoignon, *Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization*, Comput. Fluids, 36 (2007), pp. 1119–1136.
[13] A. M. Morris, C. B. Allen and T. C. S. Rendall, *CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation*, Int. J. Numer. Methods Fluids, 58 (2008), pp. 827–860.
[14] T. C. S. Rendall and C. B. Allen, *Unified fluid-structure interpolation and mesh motion using radial basis functions*, Int. J. Numer. Methods Eng., 74 (2008), pp. 1519–1559.
[15] T. C. S Rendall and C. B. Allen, *Efficient mesh motion using radial basis functions with data reduction algorithms*, J. Comput. Phys., 228 (2009), pp. 6231–6249.
[16] A. H. van Zuijlen, A. de Boer and H. Bijl, *Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations*, J. Comput. Phys., 224 (2007), pp. 414–430.
[17] M. D. Buhmann, *Radial basis functions*, Acta Numer., 9 (2000), pp. 1–38.

[18] A. C. FAUL AND M. J. D POWELL, *Proof of convergence of an interative technique for thin plate spline interpolation in two dimensions*, Adv. Comput. Math., 11 (1999), pp. 183–192.
[19] H. WENDLAND, *Fast evalution of radial basis functions: methods based on partition of unity*, Approximation Theory X: Wavelets, Splines, and Applications, Vand erbilt University Press, 2002, pp. 473–483.
[20] H. WENDLAND, *On the smoothness of positive definite and radial functions*, Comput. Appl. Math., 101 (1999), pp. 177–188.
[21] A. DE BOER A, M. S. VAN DER SCHOOT AND H. BIJL, *Mesh deformation based on radial basis function interpolation*, Comput. Struct., 85 (2007), pp. 784–795.
[22] J. C. CARR, R. K. BEATSON, B. C. MCCALLUM, W. R. FRIGHT, T. J. MCLENNAN AND T. J. MITCHELL, *Smooth surface reconstruction from noisy range data*, First International Conference on Computer Graphics and Interaction Techniques, 2003.
[23] M. J. SMITH, C. E. S CESNIK AND D. H. HODGES, *Evaluation of some data transfer algorithms for noncontiguous meshes*, J. Aerosp. Eng., 13 (2000), pp. 52–58.
[24] GORDON WILLIAM, THIEL LINDA, *Transfinite mapping and their application to grid generation*, In Thomson, Joe, Numerical Grid Generation, pp. 117–134.
[25] W. T. JONES AND J. SAMAREH, *A grid generation system for multi-disciplinary design optimization*, AIAA, 95-1689.
[26] ANASYS Software Corporation, ANASYS ICEM CFD 14.0 User Manual, Printer in U.S.A, 2011.
[27] K. J. BATHE, Finite Element Procedures, Prentice Hall, 1996.
[28] NING GU, ZHILIANG LU AND TONGQING GUO, *Simulation of viscous flows around a moving airfoil by field velocity method with viscous flux correction*, Adv. Appl. Math. Mech., 4 (2012), pp. 294–310.
[29] F. DENG, Y. Z. WU AND X. Q. LIU, *Numerical simulation of two-dimensional unsteady viscous flow based on hybrid dynamic grids*, Journal of Nanjing University of Aeronautics & Astronautics, 39 (2009), pp. 444–448.
[30] N. C. LAMBOURNE, R. H. LANDON AND R. J. ZWAAN, *Compendium of unsteady aerodynamic measurements*, AGARD R-702, 1982.