A Fast Grid Deformation Algorithm And It's Aplication

Jianzhong Chai, Xianli Han, Hailin Che
(The First Aircraft Institute of AVIC-I)
(710089 Yanliang, Xi'an,Shaanxi ,P. R. of China)

***Keyword**: grid deformation, Delaunay map, unsteady flows*

Abstract

A simple and efficient dynamic grid deformation technique is proposed for computing unsteady flow problems with geometrical deformation, relative body movement or shape changes due to aerodynamic optimisation and fluid-structure interaction. A Delaunay graph of the solution domain is first generated, which can be moved easily during the geometric dynamic deformation, even for very large distortion. A one to one mapping between the Delaunay graph and the computational grid is maintained during the movement. Therefore the new computational grid after the dynamic movement can be generated efficiently through the mapping while maintaining the primary qualities of the grid. While most dynamic grid deformation techniques are iterative based on the spring analogy, the present method is non-iterative and much more efficient. On the other hand, in comparison with dynamic grid techniques based on transfinite interpolation for structured grids, it offers both geometric and cell topology flexibility, which is crucial for many unsteady flow problems involving geometric deformation and relative motions. The unsteady aerodynamics for the flying wing was computed during deforming it's wing and body, which is demonstrated regarding their efficiency and grid quality.

1. Introduction

With the development for CFD in recent years, it has been used to investigate complex flowfields and aid to solve the difficult engineering problems. At the same time it is also being involving the multidisciplinary applications for design optimization, aeroelastics, control surface analysis and aeroservoelastics, etc. The combination of CFD with other disciplines frequently involves deforming geometries due to design modifications, surface movement, or structural loads. For CFD, with the deforming geometries, it will cause the mesh in the flow field will also be changed. These changes not only include surfaces mesh but also include the volume mesh in the flow field. All these change will cause the mesh deformation. The coupled mechanics for CFD and other mechanics such as computational structured mechanics(CSM) will need a robust and efficient mesh deformation tools. The mesh deformation becomes very important for CFD to deal with unsteady computation depending on time and multidisciplinary applications fordesign optimization, aeroelastics, control surface analysis and aeroservoelastics, and thermal analyses. For example, for unsteady flow computation of the pitching of aerofoil, the oscillation of the aerofoil will change the coordinates for aerofoil, which we can call it boundary perturbation. How to propagate this boundary perturbation into the field mesh is the aim of the mesh deformation. There are two methods to propagate the boundary perturbations into flowfield[1]: (1) mesh regeneration (2) mesh deformation. Around these two methods many ideas are developed in recent years.

Trans-Finite interpolation (TFI) is a general three-dimensional method that is widely used [2],[3],[4],[5]for cases involving multiple deforming boundary faces, especially for the structured grid regeneration and deformation. Most structured grid regeneration and deformation techniques are based on transfinite interpolation (TFI). Gaitonde and Fiddes have provided a mesh regenerating technique based on TFI with exponential blending functions[6]. The choice of blending functions has a considerable influence on the quality and robustness of the field mesh. Soni has proposed a set of blending functions based on arclength[7]; such a set is extremely effective and robust for mesh regeneration and deformation. Jones and Samareh have presented an algorithm for general multiblock mesh regeneration and deformation based on Soni's blending functions [8]. Hartwich and Agrawal have used a variation of the TFI method [9]. They have introduced two new techniques: the use of the ì slave-masterî concept to semiautomate the process, and the use of a Gaussian distribution function to preserve the integrity of meshes in the presence of multiple body surfaces. Wong et al. have used Algebraic and Iterative Mesh 3D (AIM3D), which is based on a combination of algebraic and iterative methods [10]. Leatham and Chappell have used a Laplacian technique more commonly used for unstructured mesh deformation [11]. Based on the algebraic method, the quaternion algebra[12],[13],[14] scheme was one of successful method to treat more complex mesh. TFI combines the speed and efficiency of an algebraic method with the ability to handle fully 3D perturbations. Modifications are being made in this method in recent years and is being used complex perturbation. But the efficiency and robustness of this method is based on the single mesh type, for hybrid grids or unstructured grids the special treatment must be added, which loss this method's efficiency and robustness.

The spring analogy scheme, first developed by Batina[15], which model the mesh as a network of lineal springs and solve the static equilibrium equations for this network to determine the new locations of the grid points. Farhat[16] proposed a modified spring analogy by adding additional nonlinear torsion springs to avoid the non-positive cell volume problem associated with the lineal spring network. Murayama, M., Nakahashi, K., and Matsushima K.[17] proposed a method to treat the spring stiffness with the angle between faces, which made this method can deal with the problems with large movement and large deformation of surfaces. D. G. Martineau and J M Georgala[18] proposed the dual mesh to treat the arbitrary shape of mesh, and then used the modified spring analogy to the dual mesh. The movement of the mesh first was operated on dual mesh using the modified spring analogy, then backed to the original mesh. This method was successfully used in different arbitrary type meshes and hybrid mesh. Chen[19] developed an "Exterior BEM Solver" that has a unified feature for the deforming flowfield grid generation of all grid systems. Assuming that the CFD mesh is to be embedded in an infinite linear elastic medium where the CFD surface grid is treated as a deformable hollow slit, a pseudo elastostatic problem with semi-infinite elastic domain can be formulated. This is a perfect boundary element problem since BEM only requires modeling the surface of the body and, therefore, is ideally suitable for dealing with the infinite elastic domain. Because of the spring analogy scheme for moving the mesh need solving the static equilibrium equations for this network to determine the new locations of the grid points, so the iteration steps can't avoid and this will spend much computation time, especially for three-dimensional computation. At the same time, for viscous mesh, the special treatment must be given to obtain the viscous stretched mesh. For three-dimensional viscous complex computation, the time for interpolating the perturbation of boundaries to volume grids using the spring analogy is unbeatable.

Petri Fast and William D. Henshaw[20] proposed overset grid mesh to simulation the movement of bodies. The key idea of this method is to use the overset grid method with a thin, body-fitted grid near the deforming boundary, while using fixed Cartesian grids to cover most of the computational domain. But for the unsteady calculation it had to interpolate the flow properties between meshes at different time steps, which cost large mount of time, especially for unstructured grids. The work of Lohner, Yang and Baum[21] was the separation of a flexible store using unstructured grids. Their grid generator was optimized to limit regridding to the neighbourhood of the deforming object. However, they were forced to regrid the whole computational domain periodically to obtain a high-quality finite element mesh. The same disadvantage of this method existed as the overset grids, also needed the interpolation between meshed at different time steps for unsteady calculation.

From the above discussion, it is concluded that neither the TFI technique nor the spring analogy scheme can be generalized to deal with any given mesh systems with less time and less special treatment. In this paper, we are developing an innovative method to deform the mesh with arbitrary shapes, called "map" method. So called this name because defining a place in a map usually using some coefficients such as longitude and latitude, whatever we move the map, we still can using the coefficients belongs to the place to find the place, and will never crossover each other. When we put the "map" idea into the mesh movement we can get a very new method to move the mesh. The authors developed this method as the following steps: 1) First generating the map, this map should cover the whole flowfield ; 2) secondly defining the coefficients for the points of the flowfield; 3) then moving the map according to the demand by design; 4) Finally relocating the flowfield points in the moved "map", such we can get the moved mesh and can guarantee the mesh quality after moving and at the same time the computing time is very small comparing the whole computing time because of no needing the smoothing iteration by other method such as spring analogy scheme. The following are the details for this new method. Different cases, involving in-viscous mesh for arbitrary grid elements and viscous mesh such as hybrid grid for two and three-dimensional cases, are used to demonstrate this new method. At the same time we also demonstrate this new method can move the mesh for large displacement.

2. Method

As describing the above, this new method is separated into four steps: a) The generation of the 'map'; b)locating the mesh points in the map; c)moving the map; d) relocating the mesh points in the map. In order to show the procedures of this method directly, the hybrid 2-d grid for aerofoil is adopted to give the details of this method, as shown in figure 1.

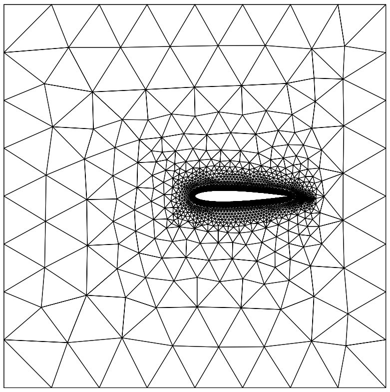


Figure 1 The hybrid grid of the aerofoil

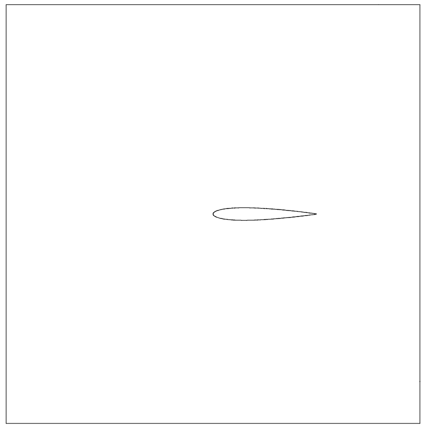
2.1 The generation of map

The map should cover the whole domain, and should have a clear structure. In our approach, we use the triangles(for two dimensional cases) or tetrahedrals(for three dimensional cases) as the map because triangles or tetrahedrals can fill arbitrary region so as to guarantee to extend this method to any complex configures. These "map" elements can be constructed using the body points and the boundary points or selected points which can represent the solid face. The Delaunay method[22] is employed to connect these points to form triangles or tetrahedrals. The connected triangles or tetrahedrals can be assured that the maximum of the minimum angle of a triangle or tetrahedral using this method, and any of arbitrary points can be constructed to triangles or tetrahedrals without special treating for special points. This is the base for that we can use "mapping" method for any arbitrary geometry's mesh movement.

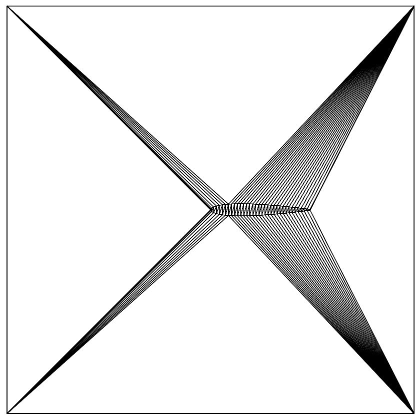
The points which are selected to form the "mapping" are very important to succeed the "mapping" method to move or deform mesh.

Commonly, we first must know which solid face will move and which solid surface will not move. The points lying on these solid surfaces are often cheesed to form the “mapping” elements. But, because the time for forming the “mapping” elements and searching the location of a point of the flowfield will a little big if we deal the very complex geometry which the solid surfaces have a large number of points, so we often selecting points which can represent the solid surface. The farfield boundary points are often selected to form the “mapping” elements, and if the boundary are very uniform, such as the quadrilateral, we only select the very representing points, the four points which form the quadrilateral.

The Delauney method which is adopted to connect these points to form triangles or tetrahedrals is followed by three steps[22]: (a) First forming the background mesh which covers the whole domain; (b)Using Delauney criterion(the circum-circle or circum-sphere of a triangle or tetrahedral will not include other points except the points which construct the triangle or tetrahedral by itself) to insert the flowfield and boundary points; (c) Delete the unwanted grids and we can get the mesh using given points by Delauney method. The details of this method can be got from the reference [22]. Here we only use the points which can represent the solid faces or boundaries to form the triangles and tetrahedrals with Delauney method, so the number of triangles and tetrahedrals are not large. The time of forming these elements will very small. Because we only use these elements as “map”, and the “map” should cover the whole mesh which will be moved or deformed, so we don’t need to delete the unwanted grids which lie in the solid faces, which is the most complex and important techniques of generating mesh using Delauney method. Figure 2 (b) is the mesh which using Delauney method, the whole grid can be used as a map to cover the whole mesh which need moving or deforming.



(a) The boundaries and solid surface

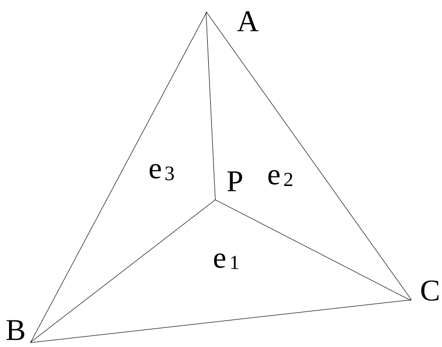


(b) The constructed “mapping” elements by Delauney method

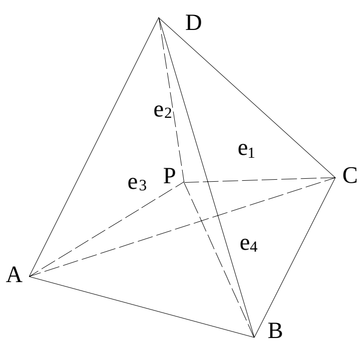
Fig.2 The boundaries and solid surface of plate hybrid grid and the constructed mapping elements using them by Delauney method

2.2 Locating the mesh points in the map

We can choose any rules to locate the mesh points in the “mapping”. But these rules must satisfy that the positions of the mesh points in the “mapping” is unified and accordingly can accurately find these points using these rules. Such as, in a map, usually using longitude and latitude to define the position of a place and accordingly we can use the longitude and latitude belongs to the place to find this place. Another example is that we use Cartesian coordinates (x,y,z) to define a point in Cartesian system, and accordingly we can use these coordinates (x,y,z) to find out the exact point accurately. In this paper, the triangles or tetrahedrals form the "mapping", so we must find a criterion to define the points of flowfield in the "mapping" using some coefficients, and can find any points using their coefficients. In this paper we use the area coefficients to define points for 2D and the volume coefficients to define points for 3D. These coefficients have three parts for 2-dimensional case and have four parts for 3-dimensional case. Figure 3(a) demonstrates the construction of the coefficients of a point in a "mapping" element for 2-dimensional case, if a mesh point P is found in the "mapping" element ABC, then using the area method, we can get three coefficients: e_1, e_2, e_3, which relative to the points A,B,C, respectively. and figure 3(b) for 3-dimensional case. The formulation (1) is the expression of these coefficients for 2-dimensional case and (2) for 3 dimensional case



(a)



(b)

Figure 3 The display of the coefficients corresponding points

$$e_1 = \frac{\begin{vmatrix} x_P & y_P & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}} \quad e_2 = \frac{\begin{vmatrix} x_A & y_A & 1 \\ x_P & y_P & 1 \\ x_C & y_C & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}} \quad e_3 = \frac{\begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_P & y_P & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}} \quad (1)$$

$$e_1 = \frac{\begin{vmatrix} x_P & y_P & z_P & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}} \quad e_2 = \frac{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_P & y_P & z_P & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}}$$

$$e_3 = \frac{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_P & y_P & z_P & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}} \quad e_4 = \frac{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_P & y_P & z_P & 1 \end{vmatrix}}{\begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}} \quad (2)$$

In this paper, we define the above coefficients for mesh points which lie in the same "mapping" element. So the first step we must find out the mesh points belong to which mapping elements. This step can be done by distinguishing the sign of the above coefficients: for a mesh point, if the sign of any of these coefficients for a certain "mapping" element is negative, we can decide that this mesh point lies outside of the certain "mapping" element. On the contrast, if the signs of all the coefficients for a certain "mapping" element are positive or zero, we can say this mesh point lies in the certain "mapping" element. All the mesh points can be divided into different "mapping" elements, and every "mapping" element will contain a certain number of mesh points. These mesh points' coefficients are obtained using the above expressions in the "mapping" element which cover these mesh points. Every coefficient is positive. So we can get the position in "mapping" elements with corresponding coefficients. The following steps are used to get the corresponding coefficients of the mesh points in "mapping" elements:

1) Divide the mesh points into groups, and a group of mesh points should be in a mapping element;
2) Using the expression (1) or (2) to compute the coefficients of mesh points. Note that the coefficients of mesh points should be equal or greater than zero;
3) Endow the coefficients to mesh points, and these coefficients will keep constant during the movement of the mesh.

These coefficients can also be used to get the mesh points' characteristic using the "mapping" element points' characteristic by linear interpolation. For example, for a point P, if U denotes the characteristic, and P1, P2, P3 are points of the "mapping" element which contain the point P, so the characteristic of P is can be obtain using the above coefficients:

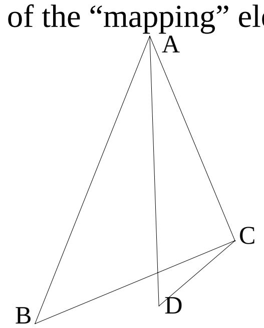
$$U_P = \frac{e1 \cdot U_1 + e2 \cdot U_2 + e3 \cdot U_3}{e1 + e2 + e3} \tag{3}$$

Here U can be any of the characteristics, such as coordinates.

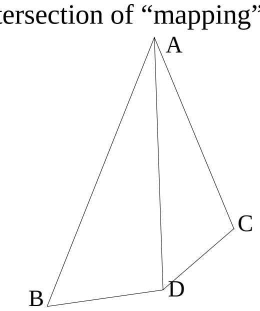
2.3 Moving the "mapping" elements

As discussed before, the "mapping" elements are constructed using the outer points and solid points which are representing the solid. The mapping elements should cover the whole mesh points. Generally the outer points stay still, and the solid points move according the movement of the solid faces and reflect the exactly movement of the solid faces. The movement of solid surfaces include the pitching of aerofoil, the relative movement between solid surfaces, the distortion of solid, and so on. The points which are chosen to construct the "mapping" elements must represent the real geometry of solid, and still can reflect the exactly real geometry during the movement of the solid surface. The most simple is to select all the solid surface mesh points to be the "mapping" element points. The outer points which are chosen to construct the "mapping" elements generally are easier to be selected, generally the points representing the boundaries which can cover the whole mesh.

During the movement of the solid surface points, there maybe encounter an error which makes one of the mapping elements' area become negative. This error will happen when the "mapping" element points crossover, as showed by the following figure 4(a). This error can be overcome by swapping the edges of the "mapping" elements. After swapping we must track back the last two steps. At this time we have to repeat the step 2.2: we have to divide the mesh points in groups again, calculate their coefficients using the expression (1) or (2), and store the coefficients for each mesh point. After doing the above steps then we can move the mesh elements again. If there still is error encountered, the divided smaller movement is used till no error encountered. When we finish the above checking and no error is encountered, then we can continue the movement of mapping elements. Figure 5 shows the result of the movement of the "mapping" elements.

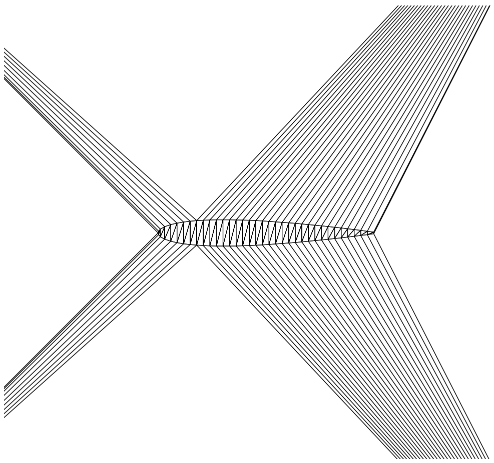


(a) The intersection of "mapping"elements



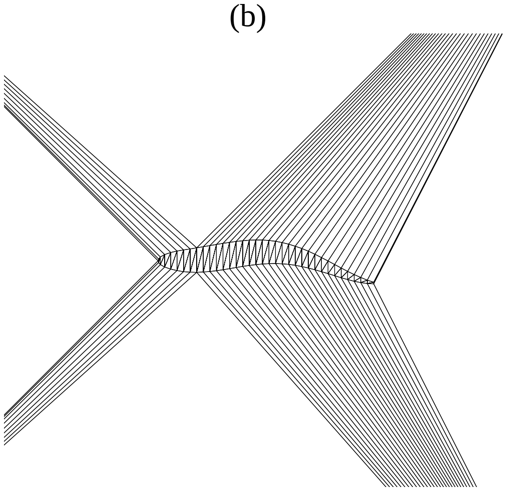
(b) The corrected "mapping" elements

Figure 4 The intersection of "mapping" elements and correction



(a) Before movement

(b)



(b) After movement

Figure 5 The result of the movement of mapping elements

This is the most important step for moving the mesh points using this new method. It is the base that no mesh grids crossover and can get high quality of mesh elements after movement.

2.4 Relocating the mesh points in the "mapping" elements

This step is to get the new position of the mesh points. When the movement is achieved, the final step using this method is to relocate the mesh points in the map. Using the above calculated coefficients and the moved "mapping" elements we can get the new position of the mesh points. For each group mesh points which lie in the same "mapping" we can use the following expressions to relocate their coordinates by linear interpolation.

For 2-dimensional case, the expression is:

$$x'_P = \frac{e_1 \cdot x'_A + e_2 \cdot x'_B + e_3 \cdot x'_C}{e_1 + e_2 + e_3} \tag{4}$$

$$y'_P = \frac{e_1 \cdot y'_A + e_2 \cdot y'_B + e_3 \cdot y'_C}{e_1 + e_2 + e_3} \tag{5}$$

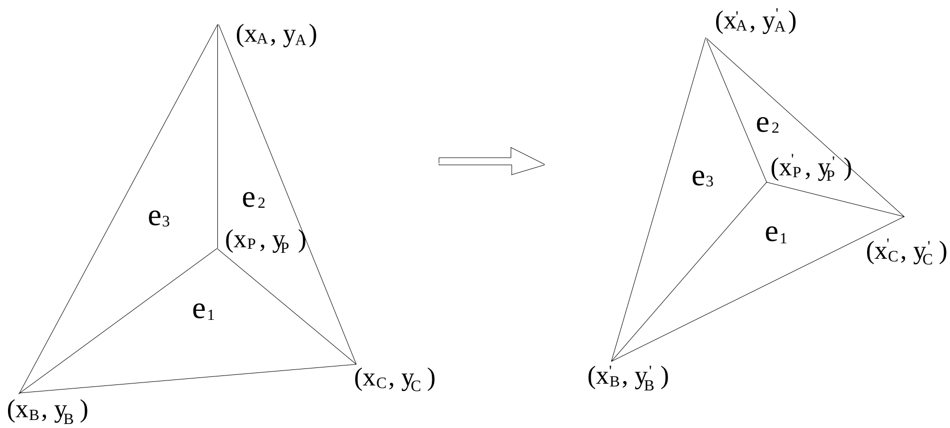
For 3-dimensional cases, the expressions are:

$$x'_P = \frac{e_1 \cdot x'_A + e_2 \cdot x'_B + e_3 \cdot x'_C + e_4 \cdot x'_D}{e_1 + e_2 + e_3 + e_4} \tag{6}$$

$$y'_P = \frac{e_1 \cdot y'_A + e_2 \cdot y'_B + e_3 \cdot y'_C + e_4 \cdot y'_D}{e_1 + e_2 + e_3 + e_4} \tag{7}$$

$$z'_P = \frac{e_1 \cdot z'_A + e_2 \cdot z'_B + e_3 \cdot z'_C + e_4 \cdot z'_D}{e_1 + e_2 + e_3 + e_4} \tag{8}$$

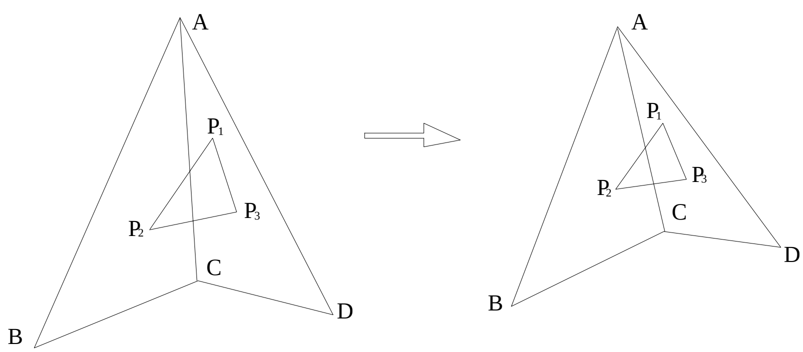
here (x', y') or (x', y', z') are the new coordinates of the mesh points, i.e., the deformed mesh's coordinates. $(x'_{A,B,C}, y'_{A,B,C})$ or $(x'_{A,B,C,D}, y'_{A,B,C,D}, z'_{A,B,C,D})$ are the moved "mapping" element's point coordinates. For 2-dimensional cases, the "mapping" element is triangle which has 3 points(A,B and C); For 3-dimensional cases, the "mapping" element is tetrahedral which has 4 points(A,B,C and D). e_1, e_2, e_3, e_4 are the coefficients of a mesh point which lies in a certain mapping element, and are computed using the formula (1) or (2). If we use the unmoved mapping elements' coordinates, we can get the original coordinates of the mesh points. During the movement, these coefficients retain constant since they are computed using (1) or (2), and can directly be used in the relocating mesh point procedure using the above expressions (4)-(8). The figures 6 showed this procedure for 2-dimensional case, and also similar to 3-dimensional case.



(a) Before movement (b) After movement

Figure 6. The relocation of mesh points during the mesh movement

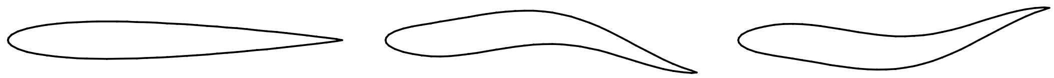
This step is the final step for this method. The deformed mesh can be got from this step. In this step we will encounter the cases: the points of a grid element maybe not lie in the same mapping element (showed in the figure 7, triangles ABC and ACD are “mapping” elements, while $P_1P_2P_3$ is one of the grid element), it seems that the moved mesh maybe crossover each other which will lead the failure of this method. Fortunately, if we use above coefficients computed by the formula (1) or (2), this error will not happen if the moved “mapping” elements have no intersection. This coefficients guarantee the success of the mesh movement without intersecting each other.



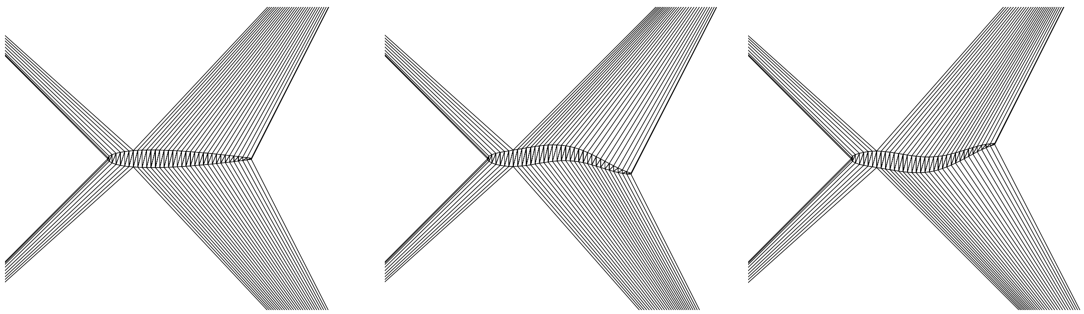
(a) Before movement (b) After movement

Figure 7 The demonstration of the possible intersection for mesh elements

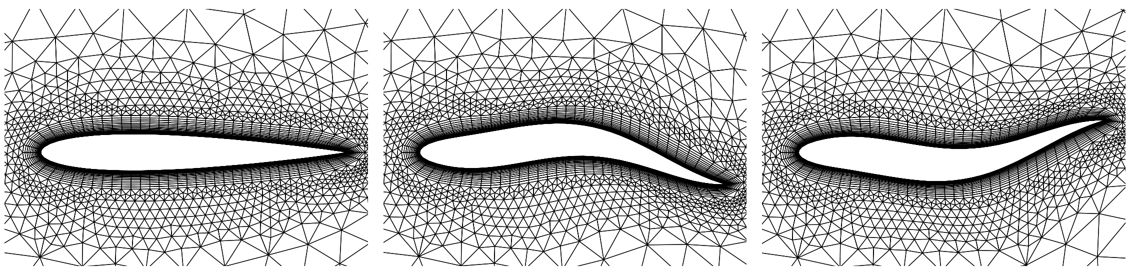
Due to the mapping elements’ coordinates had moved, the mesh points also were moved smoothly using the formula (4)-(8). The following figure 8 showed the results of the mesh for hybrid grid. Good quality of the moved mesh can be obtained and the moved mesh can be obtained by one step without any iterations.



(a) Solid surfaces deformation



(b) The movement of "mapping" elements



(c) The final moved mesh(Relocating the mesh points in "mapping" elements)

Figure 8 The procedure of the deformation of mesh with different deformation for solid surface

For a conclusion of this method, the procedure of this method can be summarized as following:

[1] Using the boundary points and selected solid surface points(it seems that no much more additional time will be spend if we choose all the solid surface points) and Delauney method to construct the mapping elements;

[2] Grouping the mesh points to make sure that the mesh points in a group lie in a "mapping" elements, then computing the coefficients of mesh points using the formula (1) or (2), and storing these coefficients;

[3] According the deformation of the solid surfaces or the relative movements between different bodies, moving the mapping elements. In fact, this movement is very simple, and we only change the coordinates of the "mapping" elements' points into the coordinates of the moved solid surfaces or new coordinates of the bodies after the relative movements;

[4] Checking the intersection between the mapping elements, and if the intersection happens we split the movement into small movements and go to step [1], and if no intersection, continue the following the step. Generally, this intersection seldom is encountered;

[5] Relocating the mesh points using the expression (4)-(8) using the coefficients computed in step [2] and the coordinates of new moved mapping elements, then the deformed mesh can be obtained;

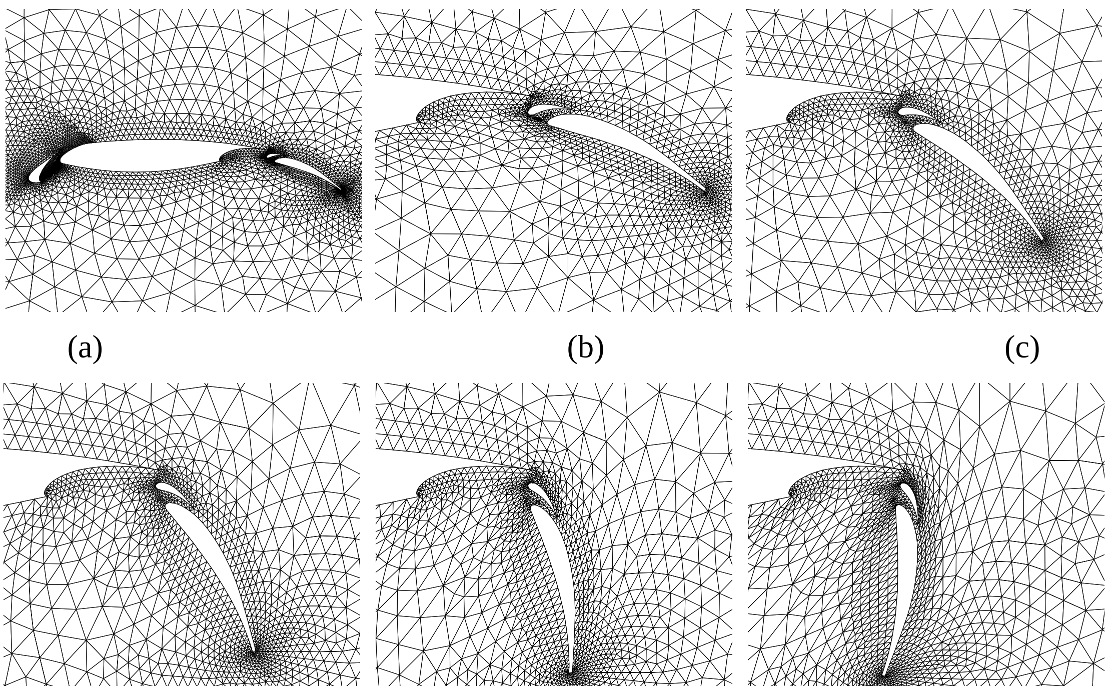
[6] Repeating [1]—[5] till we finish the movement.

From the above steps we can see we need no iterations, which made this method cost much less time for mesh movement. And, during the movement we have no special treatment for different types of mesh, i.e. viscous mesh, non-viscous mesh or hybrid mesh, so this method can be widely used in different types of mesh. For validating this method, different types of mesh are used in the following section.

3 Test Cases

(1) The relative movement between the elements of multi-element aerofoil for in-viscous and viscous meshes

In this case, the four-element aerofoil is used as the relative movement test case. In this case, the flaps had relative movement for the main aerofoil. Figure 9(a) demonstrates the view of the total mesh. Figure9(b),(c),(d),(e),(f) demonstrated the mesh after relative movement at different positions. The high quality of mesh still was obtained after the final movement(figure9(f))



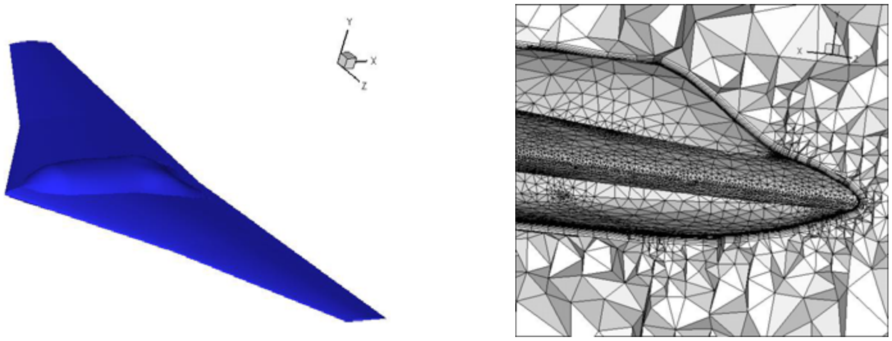
(a) (b) (c)

(d) (e) (f)

Figure 9. The relative movement between main foil and flaps (non-viscous mesh)

(2) Three-dimensional flying-wing deformation with hybrid grids

In this case, the model is used as the flying-wing with hybrid grids. The computational parameters chosen as follows : the Mach number is 0.85, the Reynolds number is 2×10^{6} ,and the attack angle α=3°. The flying-wing deformation is the relative movement at the different wings positions with variable sweep and the folding wing. The time is used the 30s when the wings folded from -30^{0} to 30^{0}. When the variable sweep wings is from 0^{0} to 90^{0},the time sames.



（a）model （b）hybrid grids

Figure 10 Three-dimensional flying-wing deformation with hybrid grids

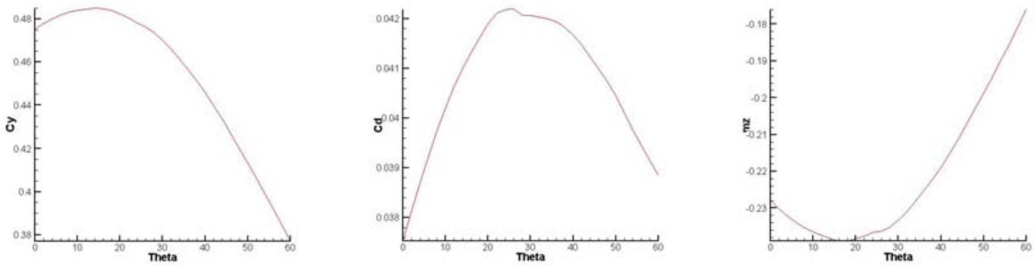


Figure 11 The display of Aerodynamicas coefficient during the wings- folding

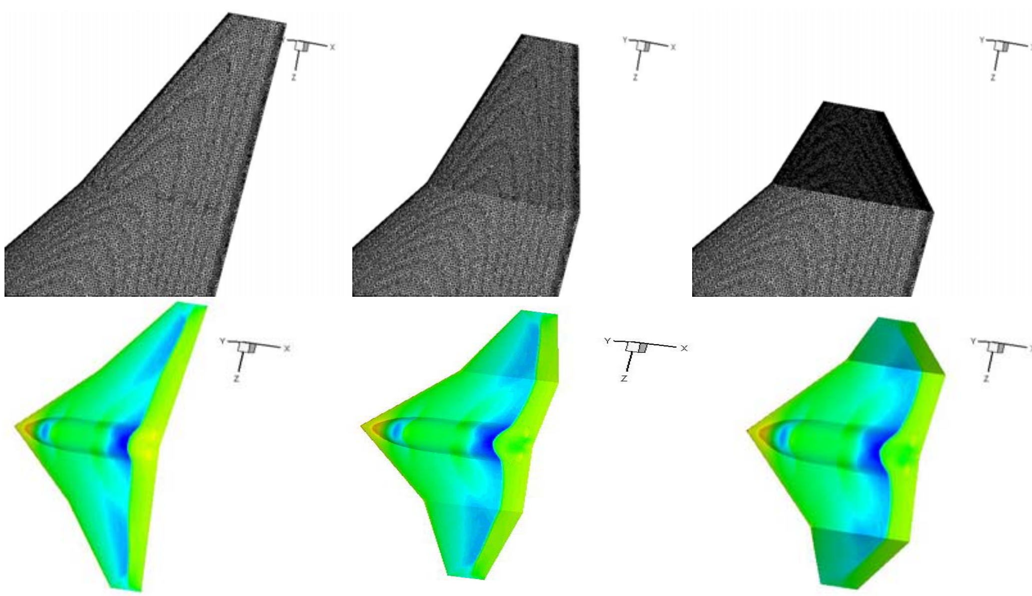


Figure 12 The display of Pressure coefficient and Grids during the wings- folding

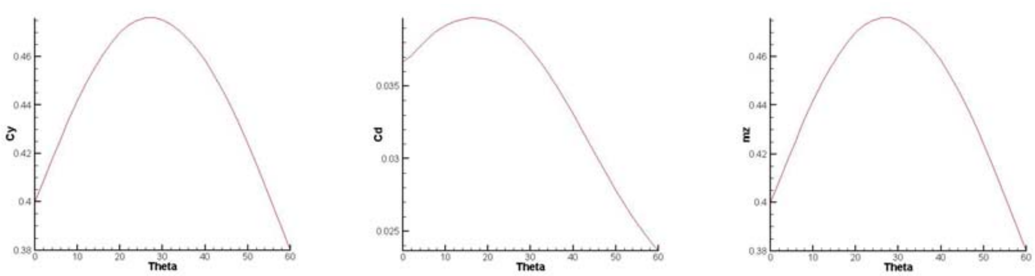


Figure 13. The display of aerodynamic coefficients during wing -variable sweep

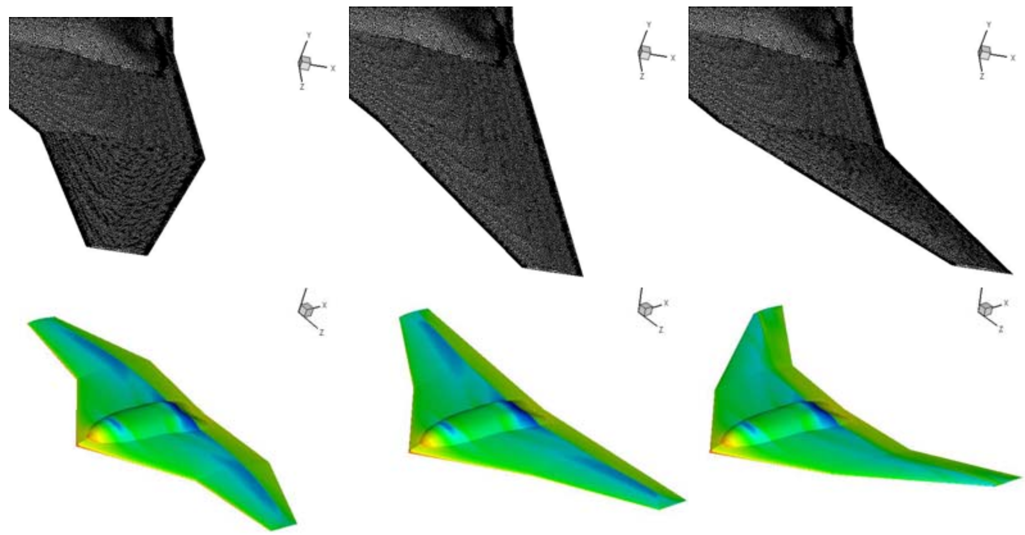


Figure 14. The display of Pressure coefficient and Grids during wing -variable sweep

4 Conclusion

This paper provided a new method for fast deformation or movement of the mesh with arbitrary shapes. Using this method can save much time during the deformation or movement of the mesh compared with the spring analysis or other methods. This method also makes the computation of unsteady flow with hybrid grids easy because it needs no additional treatment for different types of mesh. The only difficult of this method is that the mesh generator for constructing the “mapping” elements has to be coupled into the procedure of the deformation or movement of the mesh. Anyway, it provides a new method for fast deformation or movement of mesh, and can save much time for deformation or movement of mesh. Additionally, it can stand large displace of movement or deformation without special treatment, which make it more easy to compute complex unsteady flow calculation with complex configures.

Reference

[1] Jamshid A. Samareh, Application of Quaternions for Mesh Deformation, *NASA/TM*-2002-211646, 2002

[2] Byun, C., and Guruswamy, G. P., “A Parallel, Multi-block, Moving Grid Method for Aeroelastic Applications on Full Aircraft,” *AIAA Paper* 98-4782, September 1998.

[3]Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., “Aerodynamics Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation,” *AIAA Paper* 96-0094,January 1996.

[4] Wang, Z. J. and Przekwas, A. J. “Unsteady Flow Computation Using Moving Grid with Mesh Enrichment,” AIAA Paper 94-0285, January 1994.

[5]Jones, W. T. and Samareh-Abolhassani, J., “A Grid Generation System for Multi-Disciplinary Design Optimization,” *AIAA Paper* 95-1689, June 1995.

[6] Gaitonde, A. L., and Fiddes, S. P., Three-Dimensional Moving Mesh Method for the Calculation of Unsteady Transonic Flows, *Aeronautical Journal*, Vol. 99, No. 984, 1995, pp. 150-160.

[7] Soni, B. K., Two- and Three-Dimensional Grid Generation Internal Flow Applications *AIAA Paper* 85-1526, January 1985.

[8] Jones, W. T., and Samareh, J. A., A Grid Generation System for Multidisciplinary Design Optimization AIAA Paper 95-1689, June 1995.

[9] Hartwich, P. M., and Agrawal, S., Method for Perturbing Multiblock Patched Grids in Aeroelastic and Design OptimizationApplications, *AIAA Paper 97-2038*, 1997.

[10] Wong, A. S. F., Tsai, H. M., Cai, J., Zhu, Y., Liu, F., Unsteady Flow Calculations with a Multi-Block Moving Mesh Algorithm, *AIAA*-2000-1002, January 2000.

[11] Leatham, M., and Chappell, J. A., On the Rapid Regeneration of Hybrid Grids Due to Design Driven Geometry Perturbation, Sixth International Conference on Numerical Grid Generation in Computational Field Simulation, Mississippi State University, 1998, pp. 533-542.

[12] Shoemake, K., ìAnimating Rotation with Quaternion Curves,î SIGGRAPH 1985 Conference Proceedings, July, 1985, pp. 245-254.

[13] Philips, W. F., Hailey, C. E., Gerbert, G. A., Review of Attitude Representation Used for Air Kinematics, *Journal*

Aircraft, Vol. 38, No. 4, July-August 2001, pp.718-737.

[14] Jamshid A. Samareh, Application of Quaternions for Mesh Deformation, NASA/TM-2002-211646,2002.

[15] J.T. Batina. Unsteady euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA J.*, 29(3):pp. 327-333, 1991.

[16] Farhat, C, Degand, C, Koobus, B. and Lesoinne, M, "An Improved Method of Spring Analogy for Dynamic Unstructured Fluid Meshes," *AIAA 1998, Structural Dynamics and Materials Conference, AIAA* 98-2070, Apr. 20-23, 1998.

[17] Murayama, M., Nakahashi, K., and Matsushima, K., "Unstructured Dynamic Mesh for Large Movement and Deformation," *AIAA* Paper 2002-0122, 2002.

[18] D. G. Martineau, J M Georgala, A mesh movement algorithm for high quality generalized meshes, *AIAA* 2004-0614, 2004

[19] Chen, P. C. and Hill, L. R., "A Three-Dimensional Boundary Element Method for CFD/CSD Grid Interfacing," *AIAA Paper* 99-1213, April 1999.

[20] Petri Fast and William D. Henshaw, "Time accurate simulation of viscous flow around deformation bodies using overset grids" 15th AIAA Computational Fluid Dynamics Conference 11-14 June 2001 Anaheim, CA, *AIAA 2001-2604*

[21] Lohner, R., Yang, C., and Baum, J. D., "Rigid and Flexible Store separation simulations using dynamic unstructured grid technologies," *First AFOSR Conference on Dynamic Motion CFD, June 1996,* edited by L. Sakell and D. Knight, 1996, pp. 3-30.

[22] Pirzadeh, Shahyar, Recent progress in unstructured grid generation, Aerospace Sciences Meeting and Exhibit, 30th, Reno, NV, Jan. 6-9, 1992, *AIAA-1992-0445*

Copyright Statement