Chinese Journal of Aeronautics, (2018), xxx(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University
Chinese Journal of Aeronautics
cja@buaa.edu.cn
www.sciencedirect.com

Mesh deformation on 3D complex configurations using multistep radial basis functions interpolation

Gang WANG *, **Xin CHEN**, **Zhikan LIU**

School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

Received 14 August 2017; revised 7 September 2017; accepted 16 November 2017

KEYWORDS

Data reduction;
Greedy method;
Mesh deformation;
Radial basis function;
Unstructured mesh

Abstract The Radial Basis Function (RBF) method with data reduction is an effective way to perform mesh deformation. However, for large deformations on meshes of complex aerodynamic configurations, the efficiency of the RBF mesh deformation method still needs to be further improved to fulfill the demand of practical application. To achieve this goal, a multistep RBF method based on a multilevel subspace RBF algorithm is presented to further improve the efficiency of the mesh deformation method in this research. A whole deformation is divided into a series of steps, and the supporting radius is adjusted in accordance with the maximal displacement error. Furthermore, parallel computing is applied to the interpolation to enhance the efficiency. Typical deformation problems of the NASA Common Research Model (CRM) configuration, the DLR-F6 wing-body-nacelle-pylon configuration, and the DLR-F11 high-lift configuration are tested to verify the feasibility of this method. Test results show that the presented multistep RBF mesh deformation method is efficient and robust in dealing with large deformation problems over complex geometries.

1. Introduction

Mesh deformation is commonly involved in CFD simulations that have geometric boundary variations. Typical cases of these problems are aeroelastic[1,2] and aerodynamic shape optimization.[3,4] In these cases, adjustments should be made to a computational mesh according to the deformation of the wall boundary at each physical time step or design loop. For complex 3D configurations whose mesh nodes could amount to hundreds of millions, it is considerably time-consuming to deform a computational mesh at each step or design loop. Furthermore, mesh quality is crucial to the stability and accuracy of CFD simulations. It is important to develop a qualified mesh deformation method with high efficiency, excellent robustness, and ability to preserve mesh quality. At present, a variety of mesh deformation schemes have been introduced depending on the mesh topology and the specific application in literature. Based on the deforming structure, strategies for mesh deformation can be categorized into two classes: physical analogy and interpolation.[5]

Physical analogy methods, usually provided with connectivity information of a computational mesh, use physical models

* Corresponding author.
E-mail address: wanggang@nwpu.edu.cn (G. WANG).
Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER | Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2018.01.028
1000-9361

to determine the positions of internal nodes conform to boundary motions. The spring analogy[6–8] is one of the most widely used methods in this category, which models the connections between grid nodes as springs. Hence, the new positions of the nodes are determined by solving static equilibrium equations. However, this method lacks efficiency, and the mesh quality is difficult to be preserved when undergoing large deformation. The Finite Element Analogy (FEA)[9,10], viewed as an extension of the spring analogy, assumes that each mesh node is connected to its neighbors by elastic solids. This kind of method is robust and accurate but time-consuming owing to the costs of solving large-scale control equations.

Differing from the aforementioned class, interpolation methods have wider applicability and computational efficiency because connectivity information is not required.[11] The Trans-Finite Interpolation (TFI) method[12,13] generates inner grids by implementing transfinite interpolation along mesh lines with the motion of the surface boundary. This method is efficient but limited to be merely applied to structured mesh topologies, and will result in crossover of the mesh. Huang et al.[14] proposed a "layering blend deformation" technique based on the basis quaternion technique, which combines the layering arithmetic with TFI technique. This method is able to deal with problems with large deformations. Delaunay graph mapping method[15] generates a Delaunay graph of the solution domain whose grid motion can be obtained according to the boundary motion based on the one to one mapping between the Delaunay graph and the computational grid. This method is of high efficiency except that the mesh quality is tough to preserve in occurrence of large boundary deformation.

Since it was firstly applied by Boer et al.[16] to mesh deformation, the Radial Basis Functions (RBF) interpolation algorithm has raised considerable concern due to its adaptation (suitable for both structured and unstructured mesh topologies), capabilities of operating on multivariate data set (no connectivity relationship required), and maintenance of mesh quality (mesh topology will be preserved). The major factor which influences the efficiency of RBF interpolation is the number of control points. The greedy method was applied by Rendall and Allen[17], which reduces the dimensions of the basis matrix by intelligently selecting a subset of surface points as control points based on the displacement error. The problem here is that only the selected points are moved to the exact place while the others are moved by the global interpolation. As a result, a correction process needs to be carried out to ensure the accuracy of the geometry. Kedward et al.[18] developed a multiscale RBF mesh deformation method which captures global and local motions using all the surface points. This method does not need a second correction progress, and the sparsity introduced can be exploited. Niu et al.[19] proposed a novel Dynamic-Control-Point RBF (DCP-RBF) mesh deformation algorithm, which employs a dynamic set of control points to perform large mesh deformations with a quite small increase in the computational expense. RBF interpolation has been combined with some other mesh deformation algorithms. For example, Qin et al. combined the Delaunay graph scheme with RBF interpolation.[11] RBF interpolation is applied to each Delaunay graph sub-domain using vertices as control points, thus reducing the size of the interpolation matrix. This method combines merits from both the efficiency of the Delaunay graph mapping mesh deformation method and the better control of quality of grids close to the surface from the RBF method. Liu et al.[20] presented an RBFs-MSA method, which combines the benefits of the moving-submesh approach with the RBF interpolation method.

In authors' previous work, a 'double-edge' greedy supporting point selection algorithm using a multi-level subspace method[21] was adopted to reduce computational consumption. This method is efficient and robust in dealing with large deformations. However, it has been proven that the efficiency of this method tends to decline over time when dealing with complex mesh deformation problems.

In this work, a multistep RBF interpolation algorithm is presented to further ease the computational cost based on the multi-level subspace method. Differing from the previous method, the entire deformation process is viewed as consisting of hierarchical deformations, which is automatically partitioned. Since the coordinates of all the mesh nodes can be shared by all the processors, the back substitution procedure of surface and volume nodes displacements can be carried out in parallel to save the computational cost. Three test cases as representatives of dynamic mesh applications are presented to demonstrate the superiority of this method.

2. Multistep mesh deformation method

The term RBF refers to a series of functions with the general form written as:

$$\phi = \phi(\|\boldsymbol{d}\|) \tag{1}$$

where $\|\boldsymbol{d}\|$ denotes the Euclidean distance, which means that the basic variable of RBF interpolation is the spatial distance between the nodes. The interpolation function $f(\boldsymbol{r})$, representing the displacements of the mesh nodes, can be approximated by a weighted sum of basis functions:

$$f(\boldsymbol{r}) = \sum_{i=1}^{N} w_i \phi(\|\boldsymbol{r} - \boldsymbol{r}_i\|) \tag{2}$$

where the index i identifies the supporting center of RBFs, which is located on the moving surface. $\boldsymbol{r}$ is the position vector of the unknown mesh node. w_i is the weight coefficient corresponding to the ith radial basis function. $\phi(\|\boldsymbol{r} - \boldsymbol{r}_i\|)$ is the general form of the adopted radial basis function. N is the number of volume nodes involved in the interpolation.

Basis functions can be categorized as global, local, and compact.[17] Boer et al.[16] has made a comparison between different functions. In this work, Wendland C2 function[22,23] is selected as the basis function because of its satisfactory performance, which is:

$$\phi(\eta) = \begin{cases} (1-\eta)^4(4\eta+1) & 0 \leqslant \eta < 1 \\ 0 & \eta \geqslant 1 \end{cases} \tag{3}$$

where $\eta = \frac{\|\boldsymbol{r}-\boldsymbol{r}_i\|}{R}$ with R denoting the supporting radius of RBF series. Practice indicates that an appropriate supporting radius is of significant importance for this interpolation method. On one hand, a larger value of the supporting radius R will lead to a better scatter of the deformation away from the boundary but make the matrix dense. On the other hand, a lower value may generate unsmooth deformation but save the computational costs. The interpolation problem of the surface deformation is described in the following matrix expressions:

Please cite this article in press as: WANG G et al. Mesh deformation on 3D complex configurations using multistep radial basis functions interpolation, *Chin J Aeronaut* (2018), https://doi.org/10.1016/j.cja.2018.01.028