Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

High-order mesh movement with Spring-Field and vector-adding

Tuo Liu[a] *, C. Bruce Hilbert[b]

[a]SimCenter, the University of Tennessee at Chattanooga, 701 E. M.L. King Blvd., Chattanooga, TN 37403, USA
[b]Branch Technology, 100 Cherokee Blvd, Chattanooga, TN 37405, USA

Abstract

A new method for moving boundary problems with viscous mesh layers is presented. This method is extended from a former mesh generation approach, which is based on the advancing front method and a spring model named Spring-Field. This improved technique can not only be used for high-order mesh movement but also for high-order mesh generation. Based on the curved mesh deformation strategy named vector-adding, it can handle the curved mesh movement with minimal extra computational time compared with the linear mesh movement. Several 2D boundary movement cases are tested: boundary translations, boundary rotations, and boundary morphing. The results demonstrate that this approach can handle large boundary movements while preserving a good mesh quality.
© 2016 The Authors. Published by Elsevier Ltd.
Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: Mesh movement; Mesh morphing; High-order; Viscous; Anisotropic; Mesh generation; Spring-Field; Vector-adding

1. Introduction

In recent decades, computational design and analysis has become a fundamental part of industry. One of the major challenges in this area is how to handle moving boundaries and interfaces. Many applications must face this challenge, which require that spatial domain moves or deforms with time. To handle this, some methods choose a meshless strategy, such as using radial basis interpolation functions [1], and the others discretize spatial domains through meshing. The latter methods can generally be divided into two classes: boundary-fitted (boundary-conforming) methods or non-boundary-fitted methods. Boundary-fitted methods could be divided into two sub-classes: mesh-topology-changing and mesh-topology-unchanging. Mesh-topology-unchanging methods can generally be divided into two sub-classes, by the strategies of how to deform the mesh: physical analogy and interpolation [2]. The physical analogy approach describes the mesh deformation by a physical process and then models this process to deform the

* Corresponding author. Tel.: +1-423-425-5497; fax: +1-423-425-5517.
E-mail address: tuo-liu@mocs.utc.edu

1877-7058 © 2016 The Authors. Published by Elsevier Ltd.
Peer-review under responsibility of the organizing committee of IMR 25.

mesh. The representative method in this sub-class is the spring method. The spring approach views the mesh as a network of springs. After each step of boundary motion, the mesh is deformed to a new shape through the balance of spring forces. The pioneer of this method is Batina [3]. A torsional spring method was introduced by Farhat et. al [4, 5] to prevent cell collapse. Similarly, several elasticity-based approaches have been developed [6-8]. However, this kind of method can only handle small motion. Sometimes local remeshing is still required.

Recently a novel mesh generation method has been developed for generating high-order meshes with viscous layers [9]. In this method, the 2D high-order mesh generation process is divided into three steps: the boundary grid, the domain linear grid, and the domain curved grid generation. The domain linear grid generation is based on the advancing-front method and a spring approach which is referred to as Spring-Field [9]. For convenience in this paper we call it the Advancing-Front-Spring-Field method with abbreviation AFSF. The "Spring-Field" is readily extensible to mesh movement or morphing such as the spring models mentioned above in physical analogy category.

In this research note, we first improve this method with an edge-based advancing-front approach instead of the point-based one to increase its robustness. Then we adjust the spring models in the previous mesh generation method for a better implementation of mesh movement. Finally, we test this method with several cases: mesh generation cases with complex geometries and mesh movement cases with large deformations.

2. Improvement of AFSF mesh generation method

AFSF linear mesh generation method [9] is used to generate the mesh points layer by layer as same as advancing-front methods [10, 11], while a spring model named Spring-Field [9] is provided to smooth the mesh. However, it is necessary to increase the robustness of the method before adapting it for mesh movement. The increase of robustness is accomplished mainly by two adjustments as follows.

One adjustment is changing the advancing-front framework from the node-based ($node_{parent}\ to\ node_{child}$) one to the edge-based ($edge_{parent}\ to\ edge_{child}$) one. The advantage of this adjustment is that it can significantly simplify implementation to benefit programming and to speed up the program. The simplification of maps (nodes, edges, ets.) benefits the linear mesh generation. And the parent-child-edge structures procedurally matches the "Pipe" structures as shown in Fig. 1 [9], which benefits the high-order mesh generation with the vector-adding method [9].

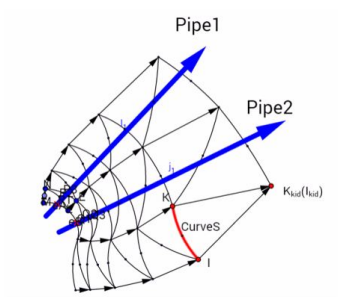


Fig. 1. The pipe-like mesh structure

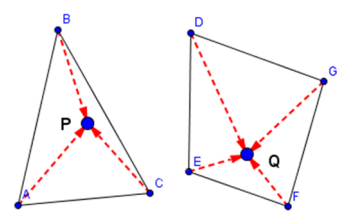


Fig.2 An advancing front closing diagram

The other adjustment is adding one more node layer framework as a "super node" layer. In the previous AFSF method, one node is only allowed to have four neighbors in four directions (right, up, left, down). This structure leads to a robustness issue especially when the mesh extrudes inwards such as shown in Fig. 2. Now with the super nodes as a map mainly used to merge nodes, the front-nodes-merging case in Fig. 2 could be solved. The super node layer has the following characters: a) Each basic node has a super node attached to it; b) One basic node can only map to one super node. One super node can map to multiple basic nodes; c) If several basic nodes overlap, only one super node is active in that group. This super node will absorb all basic nodes at this overlapped location and the other super nodes are deactivated.

The robustness obtains improvements not only by a better overlapped-nodes strategy, but also by a clarifying code structure because the operations on mesh nodes now can be divided into two layers.

After the above adjustments, several mesh movement cases are tested. Some results show the AFSF method needs to be further adjusted to benefit mesh movement. The main adjustments are described as follows.

First, the change is made to the mass of nodes in spring models. To solve the oscillation issue which a typical force model generally faces, the mass of an oscillating node increases to annihilate the oscillation in [9]. For mesh generation this strategy works. But for mesh movement, the balancing position of a node is sometimes out of the range of oscillation. So increasing the mass of a node needs a new strategy: In first several iterations of force balance process, the mass will be constant as 1. Then if there is still an oscillation, the mass increases by multiplying 1.5 in each cycle of oscillation.

Second, the change is made to the relationship between the angle spring model and the edge spring models. The angle spring may disturb the node movement if the node is too far from the target position such as in a large movement case. So for mesh movement, the angle spring is blocked instead of always active, until the node moves close enough to its target position. We also separate the extruding edge springs from the neighbor edge springs and the angle springs. Though the two groups share the same time step, now they have their own characteristic moving distances and max-moving-distance limitations, by which the spring models cooperation becomes more manageable.

Third, the change is made to extruding edge spring model to accelerate the convergence process, and to accommodate the compressed mesh movement cases. The extruding spring force, which is the main force to drag the node to its target location in large boundary motion cases, is adjusted as shown in the force equation Eq. 1

$$Force_{edge_extruding} = Coe_{aniso} \times CoeK_{edge} \times \frac{Len_{ideal} - Len_{real}}{Len_{ideal}} \tag{1}$$

Where $\mathrm{Len_{ideal}}$, as the ideal length of edge, is used instead of $\mathrm{Len_{ideal}} + \mathrm{Len_{real}}$ [9] to amplify the difference of the force value generated by different $\mathrm{Len_{real}}$. An anisotropic coefficient $\mathrm{Coe_{aniso}}$ is introduced, which has a different value for the two nodes attached to one edge spring to provide a proportional compression of edges. They are $\mathrm{Coe_{aniso}} = \frac{1}{Growth_ratio}$ for the node in the lower mesh layer, and $\mathrm{Coe_{aniso}} = 1$ for the other node. $\mathrm{CoeK_{edge}}$ is 1.

Last, the change is made to the neighbor edge spring model and the angle spring model. The neighbor edge spring strengthens if the edge aspect ratio (the length of neighbor edge divided by the length of extruding edge in the same attached mesh element) is too small. The aim of this adjustment is to obtain a smoother interior mesh in the mesh movement cases with large motions. For angle spring, the force function is changed to

$$Force_{angle_spring} = CoeK_{angle} \times f(Angle_{advanced}) \tag{2}$$

where $f(Angle_{advanced})$ is a polynomial function of $Angle_{advanced}$, with the same sign as α_{appro} [9]. $Angle_{advanced}$ is sum of $|\alpha_{\mathrm{appro}}|$ and Fac_{shape} [9]. $|\alpha_{\mathrm{appro}}|$ reflects the deviation of real extruding direction from ideal extruding direction. Fac_{shape} distinguishes the anisotropic mesh elements close to boundaries and isotropic mesh elements far from boundaries. $CoeK_{angle}$ is 1. The change is also made to the direction of the angle spring force. Now the angle spring model shares the same force direction formula with the neighbor edge spring model.

An extra benefit of adjusting AFSF method for mesh movements is that, after all the above adjustments, the method can still cover mesh generations. That indicates that by using this method, we could generate the mesh first. Then under the same mesh topology, we could move the mesh without adjusting any input coefficients of the method.

3. Results

The high-order meshes are obtained by deforming linear pipe-like-structure meshes with the method named vector-adding [9]. In this section, all the triangular meshes are obtained from mixed-element meshes by simply cutting the quads into triangles. All the triangular meshes have positive Jacobian with cubic curves (P3), which is enforced by a harsh Jacobian-checking process with 900 checking points in each triangle.

3.1. Improvement of the robustness of mesh generation

With the adjustments described in section 2, now AFSF mesh generation approach can handle more complex geometries without changing the input coefficients. Fig. 3 shows one test case mentioned as SimCenter-logo. The initial extruding length of the first mesh layer is 5×10^{-5}. The boundary mesh and the domain mesh growth ratios

are 1.3. The other two SimCener-logo cases with different growth ratio 1.1, 1.2 are also tested. By simply cutting quads into triangles, triangular meshes are obtained with 0 obtuse triangle which has an angle larger than 135 degrees in three cases.

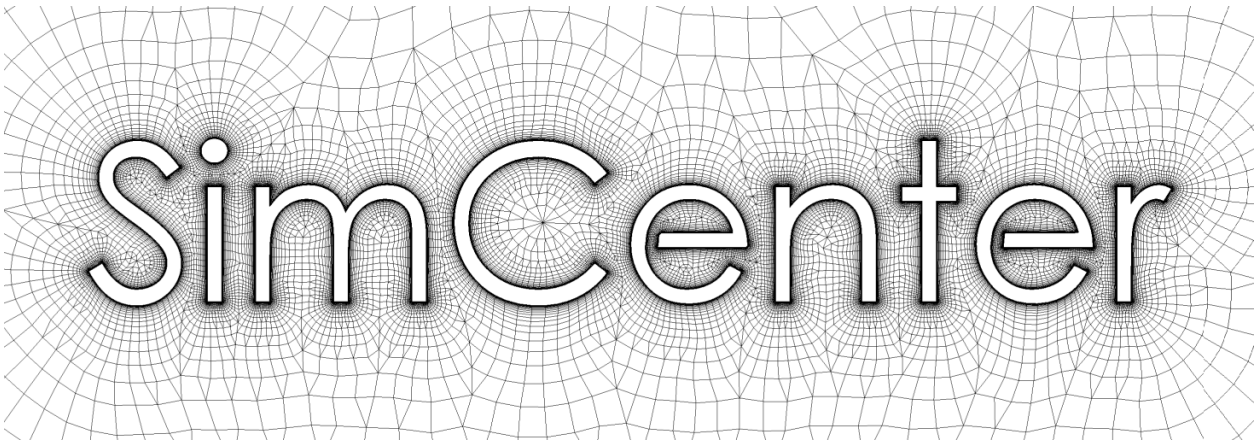
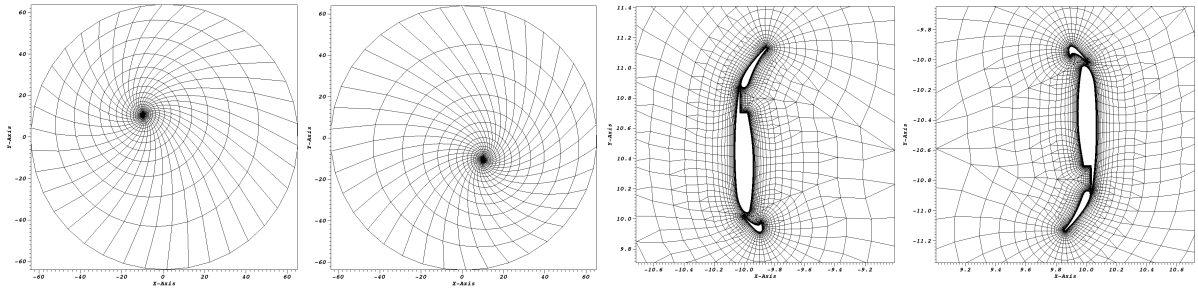


Fig. 3. SimCenter logo mesh. Mixed-elements with growth ratio 1.3

3.2. High-order mesh movement

Several types of boundary movement and morphing are tested. Benefiting from this mesh movement approach, there is no need for intermediate moving steps. All the mesh movement and morphing cases are moved in one step, and only the final shapes or the destinations of the boundaries are needed. The first group of test cases are combined motions of boundary translations and rotations, as shown in Fig. 4. The test geometry is NACA 30P30N airfoil, with an initial extruding length 5×10^{-5} and a domain mesh growth ratio 1.3. The initial airfoil is located near the origin. First test case is shown in Fig. 4. (a) left and (b) left, where the boundaries rotate 90 degrees with a rotation center at (0,0), then move with a vector $\mathrm{v} = \langle -10,10 \rangle$. Second test case is shown in Fig. 4. (a) right and (b) right, where the boundaries rotate negative 90 degrees with a rotation center at (0,0), then move with a vector $\mathrm{v} = \langle 10, -10 \rangle$. For the two triangular meshes obtained from mix-element meshes, there is no obtuse triangle with an angle larger than 135 degrees.



(a) a whole view of the linear mixed-elements meshes (b) a close view of the linear mixed-elements meshes

Fig. 4. 30P30N combined motion cases. (a) a whole view; (b) a close view;

The second group of test cases is about boundary morphing. The test geometry is NACA0012 with the same initial extruding length and growth ratio as 30P30N cases. The mesh boundary morphs with the Hicks-Henne bump functions[12]. The airfoil first changes to a larger size (phase 1). Then it moves to the left with a five-time-body-length distance as it becomes larger (phase 2), as shown in Fig. 5. All meshes in Fig. 5 are high-order meshes with cubic curves (P3). For the two triangular meshes obtained from mixed-element meshes, there is no obtuse triangle with an angle larger than 150 degrees.

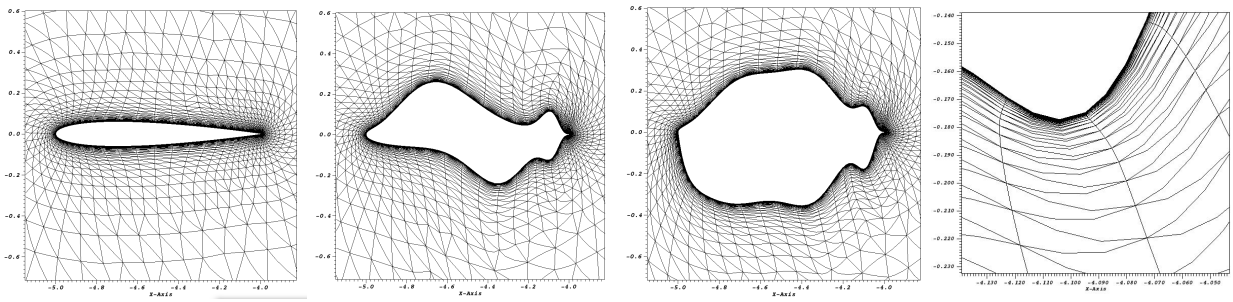


Fig. 5. NACA0012 morphing cases. From left to right, original, phase 1, phase 2, a close view of a convex area in phase 2.

4. Summary and Future Works

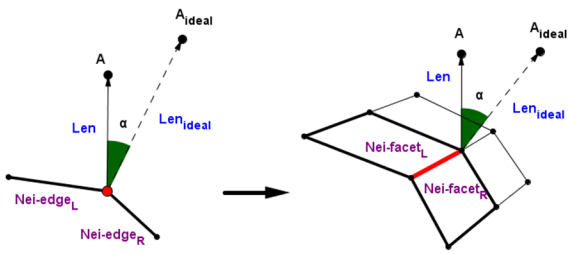


Fig. 6. An extension of spring models from 2D to 3D.

We made some adjustments to the previous AFSF linear mesh generation method to adapt it for the mesh movement purpose. With the improvements, the approach can not only handle linear mixed-elements mesh generation with complex geometries but also has the capability to manage mesh movements or mesh morphing. Furthermore, with the vector-adding method for these pipe-like-structure linear meshes, we can handle both high-order viscous mesh generations and movements. In future research, a more efficient design of the spring model could be provided to further improve the mesh quality in mesh movement cases.

Fig. 6. shows the concept of spring models in 3D compared with the ones in 2D. They are similar. The advancing-front method for 3D and 2D are also similar. Vector-adding method for 3D cases can easily be adjusted from the 2D one by using 3D vectors instead of 2D vectors. Considering these, an extension for 3D cases could be accomplished in future.

References

[1] A.K. Michler, Aircraft control surface deflection using RBF-based mesh deformation, International Journal for Numerical Methods in Engineering, 88 (2011) 986-1007.
[2] E. Luke, E. Collins, E. Blades, A fast mesh deformation method using explicit interpolation, Journal of Computational Physics, 231 (2012) 586-601.
[3] J.T. Batina, Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis, AIAA Journal, 29 (1991) 327-333.
[4] C. Farhat, C. Degand, B. Koobus, M. Lesoinne, Torsional springs for two-dimensional dynamic unstructured fluid meshes, Computer Methods in Applied Mechanics and Engineering, 163 (1998) 231-245.
[5] C. Degand, C. Farhat, A three-dimensional torsional spring analogy method for unstructured dynamic meshes, Computers & Structures, 80 (2002) 305-316.
[6] T.E. Tezduyar, M. Behr, S. Mittal, A. Johnson, Computation of unsteady incompressible flows with the stabilized finite element methods: Space-time formulations, iterative strategies and massively parallel implementations, in: ASME Pressure Vessels and Piping Division PVP, New York, 1992, pp. 7-24.
[7] R. Lohner, C. Yang, Improved ALE mesh velocities for moving bodies, Communications in Numerical Methods in Engineering, 12 (1996) 599-608.
[8] S.M. Shontz, S.A. Vavasis, A robust solution procedure for hyperelastic solids with large boundary deformation, Engineering with Computers, 28 (2012) 135-147.
[9] T. Liu, L. Wang, S.L. Karman, B. Hilbert, Automatic 2D high-order viscous mesh generation by Spring-Field and vector-adding, in: 54th AIAA Aerospace Sciences Meeting, AIAA SciTech, San Diego, CA, 2016.
[10] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, Journal of computational physics, 72 (1987) 449-466.
[11] R. Löhner, P. Parikh, Generation of three‐dimensional unstructured grids by the advancing‐front method, International Journal for Numerical Methods in Fluids, 8 (1988) 1135-1149.
[12] R.M. Hicks, P.A. Henne, Wing design by numerical optimization, Journal of Aircraft, 15 (1978) 407-412.