

Towards High-order Accurate Numerical Simulation of Unsteady Flow Physics over Domains with Large Deformation

Kan Liu¹, Lai Wang², Meilin Yu³ (✉)

Department of Mechanical Engineering

University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250

Abstract: This paper presents the development of a high-order flux reconstruction (FR) formulation for unsteady flow simulation with dynamic grid algorithms. Specifically, the high-order FR formulation for the Navier-Stokes equations in an arbitrary Lagrangian-Eulerian (ALE) format is developed for numerical simulation on moving domains. A hybrid moving grid algorithm consisting of algebraic grid smoothing and grid regeneration methods is developed to resolve domains with large deformation. The ‘dist-mesh’ technique is used for mesh regeneration, and local Lagrange interpolation within finite elements is used for flow field reconstruction. Several unsteady flow cases are studied to verify the effectiveness of the new method developed in this work.

1. Introduction

Many engineering problems features unsteady flows over moving geometries, such as flows over turbomachinery, and vehicles with revolving or flapping wings. However, moving geometries, especially those involving multiple flexible bodies with large relative motion, pose tremendous challenge on accurate and efficient numerical simulation. Generally, there are three ways to handle moving geometries in numerical simulation, namely, the mesh-free methods [1, 2], the immersed boundary methods [3, 4], and the ALE methods [5] with dynamic body-fitted meshes. Comparing with the other two methods, ALE with dynamic meshes can maintain the mesh quality near moving boundaries, especially for high Reynolds number flows; but the mesh moving and regeneration algorithms can be very complex, and therefore, hard to design to achieve high computational efficiency. In this study, we will develop a hybrid moving mesh and mesh regeneration technique to accelerate the dynamic mesh approach, and verify this technique with unsteady flows over moving domains.

There are basically two approaches to handle dynamic meshes. One is mesh deformation (including rigid-body motion), which can resolve small, local mesh deformation or simple mesh movement, and the other one is mesh regeneration, which can handle large deformation and/or large relative motion among multiple bodies. The algebraic mesh generation/deforming technique originates from trans-finite interpolation (TFI) [6]. It has been widely adopted for generating static meshes around complex geometries, and smoothing dynamic meshes by the

¹ Graduate Student, Department of Mechanical Engineering, AIAA student member, email: kan7@umbc.edu

² Graduate Student, Department of Mechanical Engineering, AIAA student member, email: bx58858@umbc.edu

³ Assistant Professor, Department of Mechanical Engineering, AIAA senior member, email: mlyu@umbc.edu

agency of its numerical efficiency. For each specific problem, the blending function used to control the mesh quality and algorithm robustness needs to be modified. Furthermore, the algebraic approach may generate crossed grid elements and negative volumes [7] when applied to cases which have multiple bodies with large deformation or relative movement. To conquer the weakness of the algebraic technique, researchers have developed many other meshing algorithms. Batina first developed the spring analogy scheme [8], which treats the mesh as a linear springs network and solve the equilibrium equation of this to determine the locations of the grid points. After that, Degand and Farhat [9], and Blom [10] proposed nonlinear approach for the spring analogy scheme. Liu *et al.* [7] presented a deforming mesh technique based on Delaunay graph mapping. Further, Persson [11] proposed a general mesh method named ‘dist-mesh’ by combining spring analogy with the Delaunay triangulation algorithm. This method is easy-to-use and can generate meshes of high quality with high computing efficiency. In this work, the algebraic mesh deforming technique developed in our previous work [12] will be used to efficiently handle grid movement with relatively small deformation. To handle large mesh deformation or relative motion, the mesh quality will be measured following the approach proposed by Field [13]. When it is below a certain threshold, the ‘dist-mesh’ method will be activated for mesh regeneration.

In this study, a high-order accurate FR method [14, 15, 16, 17] is further developed to resolve flow simulation on domains with large deformation. Our previous work [18, 19] implemented the ALE formulation for both the compressible and incompressible Navier-Stokes equations which can be directly solved in mesh deforming conditions. To handle large mesh deformation, the ‘dist-mesh’ technique [11] is used to regenerate the mesh. Then a Lagrange interpolation approach, which has been used by many researchers [20, 21, 22] in immersed boundary methods and moving grid methods, is adopted to reconstruct flow fields

The remainder of this paper is organized as follows. In Sect.2, numerical method, dynamic mesh algorithm and simulation setup are introduced. In Sect.3, results from vortex propagation presented. Sect.4 briefly concludes the study and discusses the future work.

2. Numerical methods

2.1 Governing equations

Unsteady compressible Navier-Stokes(N-S) equations in conservation form in the physical domain(t, x, y) can be written as:

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \quad (1)$$

where Q is the vector of conservative variables, and F and G are the total fluxes including both the inviscid and viscous flux vectors.

After introducing a time-dependent coordinate transformation from the physical domain (t, x, y) to the computational domain (τ, ξ, η), one can rewrite Eq. (1) as:

$$\frac{\partial \tilde{Q}}{\partial \tau} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0, \quad (2)$$

where

$$\begin{cases} \tilde{Q} = |J|Q \\ \tilde{F} = |J|(Q\xi_t + F\xi_x + G\xi_y) \\ \tilde{G} = |J|(Q\eta_t + F\eta_x + G\eta_y) \end{cases} \quad (3)$$

Herein, $\tau = t$, and ξ and η , which vary from -1 to 1, are the local coordinates in the computational domain. Then, as the transformation shown above, the Jacobian matrix J will be the following:

$$J = \frac{\partial(t, x, y)}{\partial(\tau, \xi, \eta)} = \begin{pmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & 1 \end{pmatrix}. \quad (4)$$

Since J is a non-singular matrix, its inverse transformation must also exist. The inverse of J is

$$J^{-1} = \frac{\partial(\tau, \xi, \eta)}{\partial(t, x, y)} = \begin{pmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & 1 \end{pmatrix}. \quad (5)$$

The Geometric Conservation Law (GCL) for the time-dependent coordinate transformation can be written as:

$$\begin{cases} \frac{\partial(|J|\xi_x)}{\partial \xi} + \frac{\partial(|J|\eta_x)}{\partial \eta} = 0 \\ \frac{\partial(|J|\xi_y)}{\partial \xi} + \frac{\partial(|J|\eta_y)}{\partial \eta} = 0 \\ \frac{\partial|J|}{\partial t} + \frac{\partial(|J|\xi_t)}{\partial \xi} + \frac{\partial(|J|\eta_t)}{\partial \eta} = 0 \end{cases} \quad (6)$$

Eq. (2) can be reformulated in the physical domain by using the relationship between the grid velocity $\vec{V}_g = (x_t, y_t)$ and (ξ_t, η_t) as given below,

$$\begin{cases} \xi_t = -\vec{V}_g \cdot \nabla \xi \\ \eta_t = -\vec{V}_g \cdot \nabla \eta \end{cases} \quad (7)$$

On applying the GCL identities, Eq. (2) can then be express as:

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} - \vec{V}_g \cdot \nabla Q = 0 \quad (8)$$

We note that GCL is automatically satisfied when a chain-rule approach is used to implement Eq. (8) [23, 19].

2.2 FR formulation

In FR, the flux terms in Eq. (2), i.e., \tilde{F} and \tilde{G} , are treated as a combination of local fluxes \tilde{F}^{Loc} and \tilde{G}^{Loc} , and correction fluxes \tilde{F}^{Cor} and \tilde{G}^{Cor} , which are expressed as:

$$\begin{cases} \tilde{F}(\xi, \eta) = \tilde{F}^{Loc}(\xi, \eta) + \tilde{F}^{Cor}(\xi, \eta) \\ \tilde{G}(\xi, \eta) = \tilde{G}^{Loc}(\xi, \eta) + \tilde{G}^{Cor}(\xi, \eta) \end{cases} \quad (9)$$

Local fluxes are constructed using only flow information within a specific element. Therefore, local fluxes are element-wise continuous, and have jumps on element boundaries. To ensure conservation and numerical stability, common or numerical fluxes, i.e., \tilde{F}^{Com} and \tilde{G}^{Com} in current context, are reconstructed on element boundaries using local flow information via Riemann solvers [23] for the inviscid fluxes and/or via the first Bassi-Rebay (BR1) approach [24] for the viscous fluxes. The numerical fluxes are then used to correct the local fluxes, and forms the correction fluxes \tilde{F}^{Cor} and \tilde{G}^{Cor} .

On substituting Eq. (9) into Eq. (2), the governing equations then read

$$\begin{aligned} \frac{\partial \tilde{Q}}{\partial \tau} + \left(\frac{\partial \tilde{F}^{Loc}}{\partial \xi} + \frac{\partial \tilde{G}^{Loc}}{\partial \eta} \right) + \left(\frac{\partial \tilde{F}^{Cor}}{\partial \xi} + \frac{\partial \tilde{G}^{Cor}}{\partial \eta} \right) \\ = \frac{\partial \tilde{Q}}{\partial \tau} + \frac{\partial \tilde{F}^{Loc}}{\partial \xi} + \frac{\partial \tilde{G}^{Loc}}{\partial \eta} + \tilde{\delta}^{Cor} = 0. \end{aligned} \quad (10)$$

Herein, $\tilde{\delta}^{Cor} = \partial \tilde{F}^{Cor} / \partial \xi + \partial \tilde{G}^{Cor} / \partial \eta$ is named the correction field.

To approximate the solution \tilde{Q} within the computational domain, a multi-dimensional polynomial of degree p is defined by its value at a set of $N_p = \frac{(p+1)(p+2)}{2}$ solution points. The solution points for a third-order accurate scheme are shown in Figure 1.

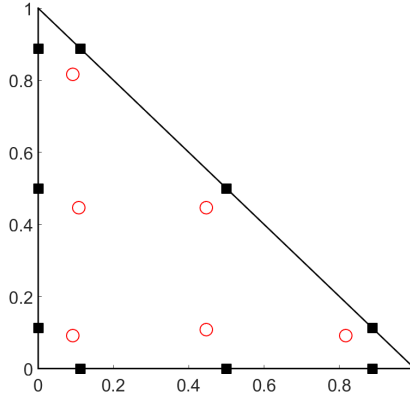


Figure 1. Solution points (circles) and flux points (squares) in the reference element for $p = 2$.

Eq. (10) can be expressed in the physical domain as:

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F^{loc}}{\partial x} + \frac{\partial G^{loc}}{\partial y} - \vec{V}_g \cdot \nabla Q + \delta^{Cor} = 0, \quad (11)$$

where the correction field in the physical domain is $\delta^{Cor} = \tilde{\delta}^{Cor} / |J|$. Readers are referred to Refs. [37, 41, 42, 43, 44] for more information on this method.

2.3 Dynamic grid strategies – the ‘Dist-mesh’ method

The basic idea of the ‘dist-mesh’ method is to solve the force equilibrium equation of each

element edge (bar) of triangle elements generated by the Delaunay algorithm [25]. First, the coordinates of a 2D mesh node are collected in a N-by-2 array p :

$$p = [x \ y]. \quad (12)$$

The force vector $F(p)$ has horizontal and vertical components at each mesh node as following:

$$F(p) = [F_{int,x}(p) \ F_{int,y}(p)] + [F_{ext,x}(p) \ F_{ext,y}(p)], \quad (13)$$

where F_{int} represents the internal forces from the bars, and F_{ext} are the external forces which are the reactions from the boundaries. $F(p)$ depends on the topology of the bars connecting the joints. Since the Delaunay algorithm generates the input points without overlapping each other, every edges is shared by at most two triangles. In the process, the force vector $F(p)$ is not a continuous function of p , as the topology (the connectivity of each mesh node) is changing by the Delaunay algorithm when the nodes move.

The system $F(p) = 0$ should be solved for a set of equilibrium positions of p . Due to the discontinuity in the force function and the external reaction forces at the boundaries, a trivial approach to solve this system is to adopt an artificial time-dependence. For some $p(0) = p_0$, a system of ODEs without any physic units is written as:

$$\frac{dp}{dt} = F(p), t \geq 0. \quad (14)$$

If any stationary solution is found, it will satisfy the system $F(p) = 0$. In Eq. (14), a forward Euler method is used to approximate the solution. At the discretized artificial time $t_n = n\Delta t$, the approximate solution $p_n \approx p(t_n)$ is updated by:

$$p_{n+1} = p_n + \Delta t F(p_n). \quad (15)$$

When evaluating the force function, both the coordinates of each node and the triangulation topology are known. The external reaction forces behave in the following way: all nodes that go outside the region during the update are moved back to the closest boundary node or just deleted to satisfy the requirement that forces act normally to the boundary. Thus, the points can move along the boundary, but not go outside.

Each bar has a force-displacement relationship $f(l, l_0)$ depending on its current length l and original length l_0 . In this work, a linear approach for $f(l, l_0)$ is used as:

$$f(l, l_0) = \begin{cases} k(l_0 - l) & \text{if } l < l_0 \\ 0 & \text{if } l \geq l_0 \end{cases}, \quad (16)$$

which is the bar's response to the repulsive forces but it will not respond to the attractive force. Although the nonlinear function may generate better meshes, the piecewise linear force function still generates acceptable results.

2.4 Solution interpolation method

When the mesh is regenerated, a local solution interpolation from the original element to the new element will be performed using the Lagrange interpolation. Specifically, the interpolated value $Q^{new}(x)$ on the new elements can be written as

$$Q^{new}(\mathbf{x}) = \sum_{j=1}^{N_s} L_j(\mathbf{x}) Q_j^{old}, \quad (17)$$

where $Q^{new}(\mathbf{x})$ is the value for the new element whose coordinate is \mathbf{x} , $L_j(\mathbf{x})$ is the multi-dimensional Lagrange polynomial associated with the solution point j of the old element in which the solution points of the new element are located. Thus, values at every solution point of the new element can be calculated from the multi-dimensional Lagrange polynomial with degree p .

3. Numerical results

3.1 Order of accuracy study

The convergence rate of the solver is tested by a scalar equation using the L_2 error with a 3rd order scheme ($p = 2$). In this case both quadrilateral and triangular elements are studied with stationary and moving grids. Sizes of meshes tessellated with regular elements, defined as number of elements in x direction and y direction, are 10×10 , 20×20 , 40×40 , 80×80 , respectively. The grid deformation strategy is presented as follows:

$$\begin{cases} dx(t) = A_x \cdot \sin(2\pi \cdot f_x \cdot x_{ref}) \cdot \sin(2\pi \cdot f_y \cdot y_{ref}) \cdot \sin(2\pi \cdot f_n \cdot t) \\ dy(t) = A_y \cdot \sin(2\pi \cdot f_x \cdot x_{ref}) \cdot \sin(2\pi \cdot f_y \cdot y_{ref}) \cdot \sin(2\pi \cdot f_n \cdot t) \end{cases} \quad (18)$$

Herein, A_x and A_y are the amplitudes of the grid deformation in x and y directions. f_x , f_y and f_n are frequencies in space and time, respectively. x_{ref} and y_{ref} are the original coordinates of the mesh nodes. And t is the physical time. In this test, these parameters are set as follows: $A_x = A_y = 1.0$, $f_x = f_y = 0.1$, and $f_n = 1.0$. For the time scheme, an explicit three-stage strong stability preserving Runge-Kutta method is adopted. The time step is set to 0.001 s. Therefore, the maximum CFL number within all the mesh sizes is 0.016.

Mesh deformation examples are presented in Figure 2. The results of convergence rate are shown in Table 1 and Table 2. It can be observed that the order of accuracy of both quadrilateral and triangular elements can reach its optimal value in stationary or moving grid simulations. And the tables also show that comparing with that from the stationary grid, the absolute L_2 error from the dynamic one increases. However, the order of accuracy is still well maintained.

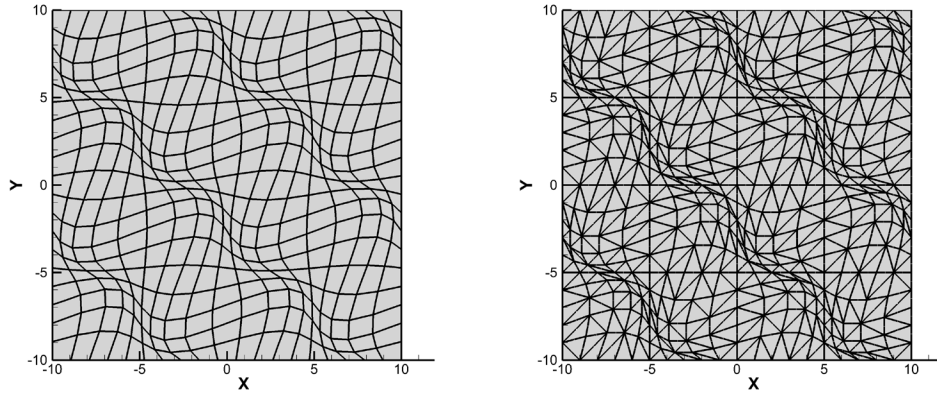


Figure 2. Deformation of quadrilateral elements and triangular elements.

Table 1. Convergence rate for Quadrilateral elements

Quadrilateral element				
	Stationary grid		Moving grid	
Mesh size	Max L_2 Error	Order of accuracy	Max L_2 Error	Order of accuracy
10 by 10	0.007783135305		0.015426034622	
20 by 20	0.000770146110	3.3	0.001964748249	3.0
40 by 40	0.000088315054	3.1	0.000186254477	3.4
80 by 80	0.000010610625	3.1	0.000017387800	3.4

Table 2. Convergence rate for Triangular elements

Triangular element				
	Stationary grid		Moving grid	
Mesh size	Max L_2 Error	Order of accuracy	Max L_2 Error	Order of accuracy
10 by 10	0.008217738632		0.012440746642	
20 by 20	0.000763009590	3.4	0.001408945705	3.1
40 by 40	0.000098648429	3.0	0.000157141687	3.2
80 by 80	0.000012524947	3.0	0.000017596073	3.2

Researchers [26, 27] indicates that if the Gauss-Legendre points are used in numerical

quadrature, a $(2p + 1)$ th order super-accuracy for a p th degree DG scheme can be achieved on quadrilateral elements. In this paper, similar tests have been done for meshes tessellated with quadrilateral and triangular elements using the scalar wave propagation equation. Results of convergence rate of the 3rd order scheme (refinement in grid spacing: 20×20 and 40×40) are presented in Figure 3.

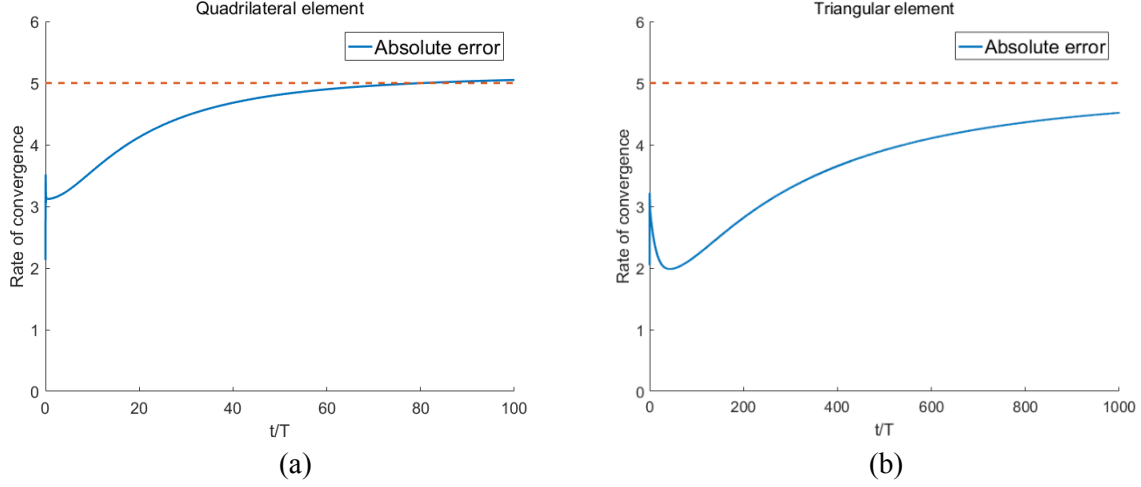


Figure 3. Rate of convergence of polynomial order $p = 2$ with wave speed $c = 4$.

From Figure 3, it can be observed that the quadrilateral elements can reach super accuracy $(2p + 1)$ within 100 periods. This has good agreement with the results from the work by K. Asthana *et al.* [26]. However, the rate of convergence of triangular elements is significantly slower than that of the quadrilateral elements: even after 1000 periods, the rate of convergence cannot reach the same level of the quadrilateral elements.

3.2 Vortex propagation with dynamic mesh

In order to demonstrate the numerical performance of the hybrid moving mesh and mesh regeneration method, a vortex propagation case is simulated on a dynamic mesh which has large deformation. Both algebraic mesh smoothing and mesh regeneration methods are used in this simulation. A ring-like mesh is generated in the computational domain shown in Figure 4.

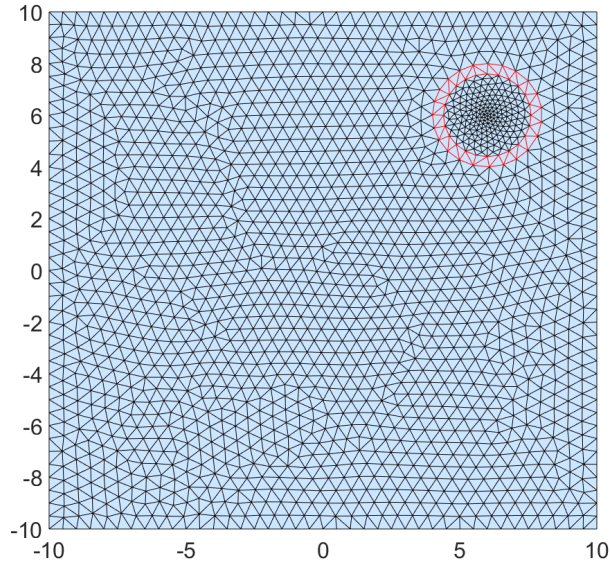


Figure 4. Mesh setup for the vortex propagation problem.

In every physical time step, the mesh inside the ring will rotate, meanwhile the ring-like mesh will deform. When the deformation is large (shown in Figure 5), the mesh will be regenerated.

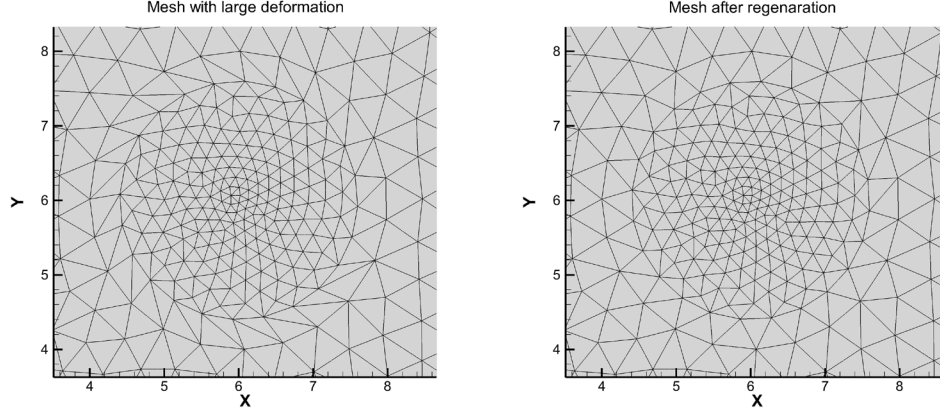
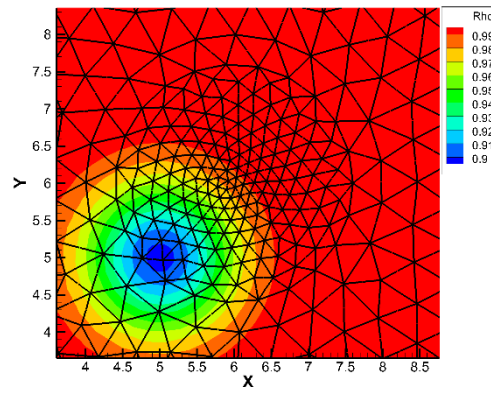


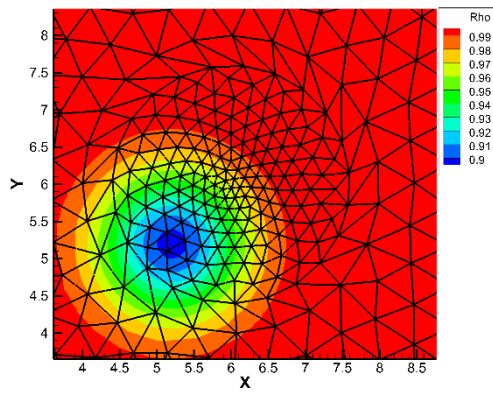
Figure 5. Mesh before and after regeneration.

The process of vortex passing over this dynamic mesh region is shown in Figure 6. When the mesh is deforming, the moving grid solver will be adopted. And when the mesh is regenerated, the variables on the old mesh will be interpolated to the new mesh with the Lagrange interpolation method. In this simulation, the grid velocity is given by its analytical solution. When the rotation angle reaches 10 degrees, the mesh is regenerated. It is clear from Figure 6 that mesh regeneration does not distort the shape of the vortex.

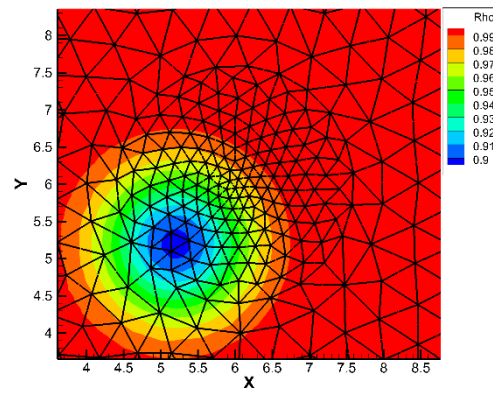
Initial vortex and grid position



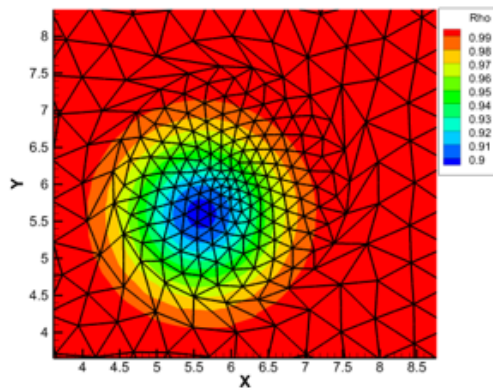
Vortex and grid position before the 1st regeneration



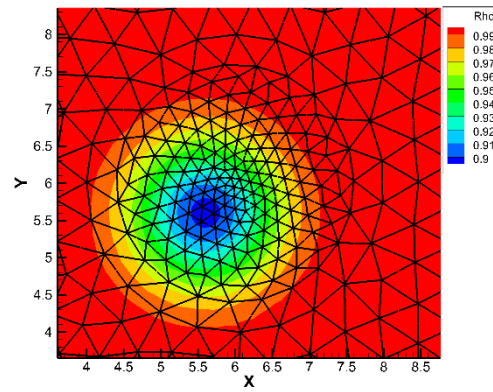
Vortex and grid position after the 1st regeneration



Vortex and grid position before the 5th regeneration



Vortex and grid position after the 5th regeneration



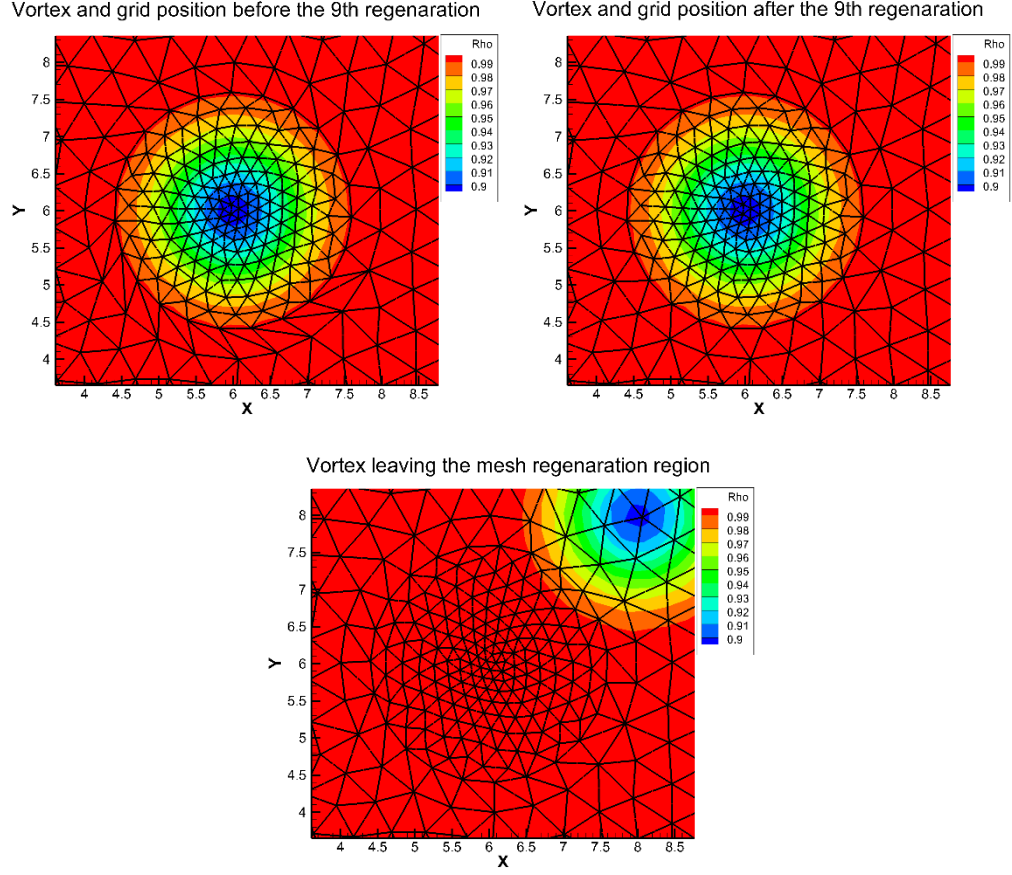


Figure 6. Vortex and grid position with the mesh regeneration.

Furthermore, the shape of vortex can be maintained well in long-time numerical simulation. In Figure 7, we present the density field of the vortex after it travels over the whole domain four times.

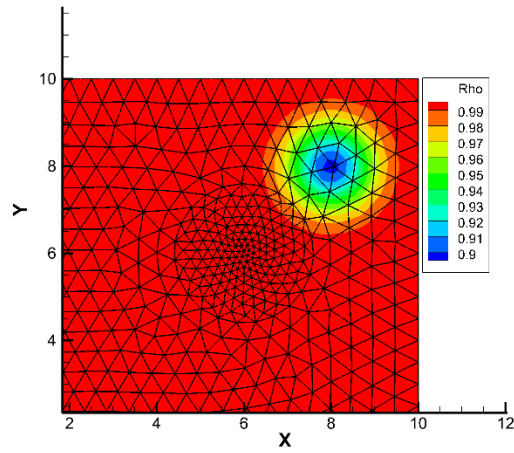


Figure 7. Shape of vortex after four periods.

L_2 error estimation has also been performed for both moving grid with mesh regeneration and stationary grid (with the same mesh) during the vortex passing over the regeneration region. The results are shown in Figure 8. It is observed that the L_2 error from the moving grid with mesh regeneration is slightly higher than the stationary grid.

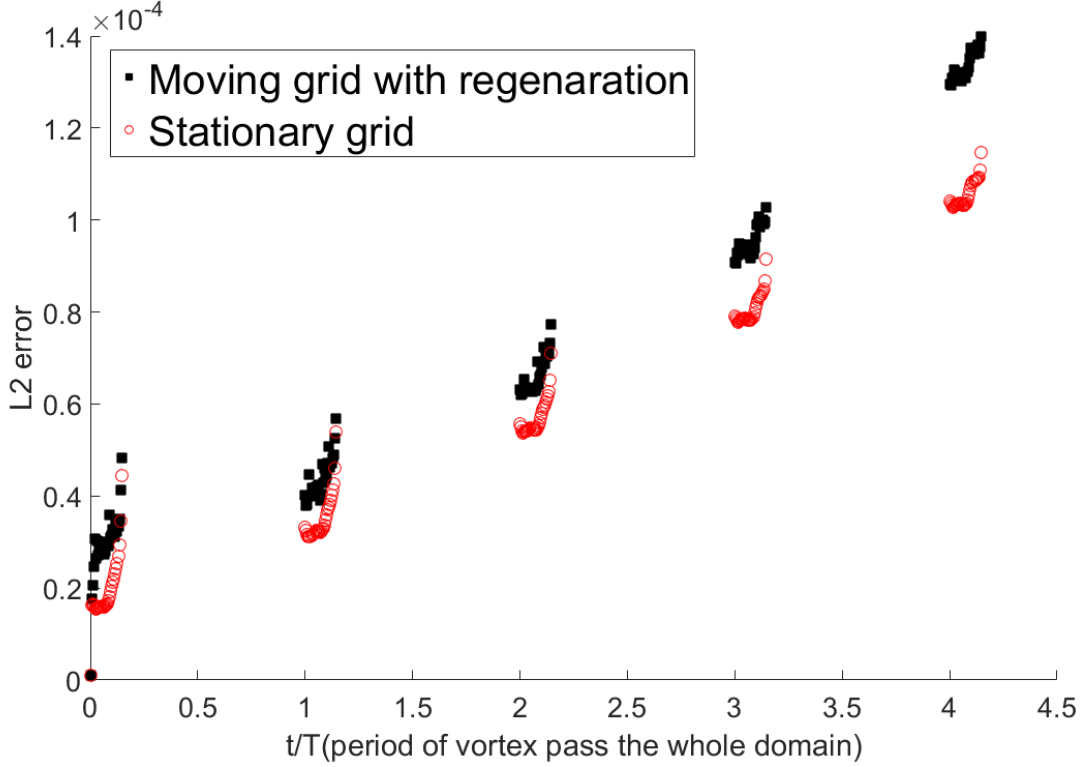


Figure 8. L_2 error when vortex passing through the mesh regeneration domain.

4. Conclusion and future work

This work demonstrates the possibility of using mesh regeneration in high-order CFD simulation of unsteady flows on moving domains with large deformation. The ‘dist-mesh’ technique has been adopted to handle mesh regeneration, and a local Lagrange-polynomial-based interpolation approach is used to interpolate solutions from the old mesh to the new one. It is found that this technique does not deteriorate the high-order accuracy of the FR method in unsteady flow simulation. We also observed that similar to that from simulation with quadrilateral elements, spatial super convergence can occur on triangular elements during long-time unsteady flow simulation; however, much longer time needs to be taken before super convergence shows up on meshes tessellated with triangular elements. Future work will focus on applying this approach in simulation of unsteady wall-bounded flows on domains with large deformation.

References

- [1] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics - theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375-389, 1977.
- [2] J.-S. Chen, M. Hillman and S.-W. Chi, "Meshfree Methods: Progress Made after 20 Years," *Journal of Engineering Mechanics*, vol. 143, p. 04017001, 2017.
- [3] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252-271, 1972.
- [4] Rajat Mittal, Gianluca Iaccarino, "IMMERSED BOUNDARY METHODS," *Annual Review of Fluid Mechanics*, vol. 37, pp. 239-261, 2005.
- [5] J. Donea, S. Giuliani and J. P. Halleux, "An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions," *Computer Methods in Applied Mechanics and Engineering*, vol. 33, pp. 689-723, 1982.
- [6] Peter R. Eiseman, Robert E. Smith, "Mesh generation using algebraic techniques," *Institute for Computer Applications in Science and Engineering*, 1980.
- [7] X. Liu, N. Qin and H. Xia, "Fast dynamic grid deformation based on Delaunay graph mapping," *Journal of Computational*, vol. 211, no. 1, pp. 405-423, 2006.
- [8] J. T. BATINA, "Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis," *AIAA Journal*, vol. 29, no. 3, pp. 327-333, 1991.
- [9] C. Degand, C. Farhat, "A three-dimensional torsional spring analogy method for unstructured dynamic meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 80, no. 3-4, pp. 305-316, 2002.
- [10] F. J. Blom, "Considerations on the spring analogy," *International Journal for Numerical Methods in Fluids*, vol. 32, no. 6, pp. 647-668, 2000.
- [11] Per-Olof Persson, Gilbert Strang, "A Simple Mesh Generator," *SIAM REVIEW*, vol. 46, no. 2, pp. 329-345, 2004.
- [12] M. L. Yu, Z. J. Wang and H. Hu, "A high-order spectral difference method for unstructured dynamic grids," *Computers & Fluids*, vol. 48, pp. 84-97, 2011.
- [13] D. A. Field, "Qualitative measures for initial meshes," *Internat. J. Numer. Methods Engrg.*, vol. 47, pp. 887-906, 2000.
- [14] H. T. Huynh, "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods," in *the 18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida, AIAA Paper No. 2007-4079, 2007.
- [15] Z. J. Wang and H. Y. Gao, "A unifying lifting collocation penalty formulation for the Euler equations," *AIAA Paper 2009-401*.
- [16] P. E. Vincent, P. Castonguay and A. Jameson, "A new class of high-order energy stable flux reconstruction schemes," *J. Sci. Comput.*, vol. 47, no. 1, pp. 50-72, 2011.
- [17] M. L. Yu and Z. J. Wang, "On the connection between the correction and weighting functions in the correction procedure via reconstruction method," *J. Sci. Comput.*, vol. 54, no. 1, pp. 227-244, 2013.
- [18] Lai Wang, Meilin Yu, "A High-Order Accurate Preconditioned Flux Reconstruction

- Formulation for Low-Mach-Number Flow Simulation with Dynamic Meshes," in *55th AIAA Aerospace Sciences Meeting*, Grapevine, Texas, 2017.
- [19] M. L. Yu and L. Wang, "A high-order flux reconstruction/correction procedure via reconstruction formulation for unsteady incompressible flow on unstructured moving grids," *Computers & Fluids*, vol. 139, pp. 161-173, 2016.
 - [20] Cesar A. Acosta Minoli, David A.Kopriva, "Discontinuous Galerkin spectral element approximations on moving meshes," *Journal of Computational Physics*, vol. 230, no. 5, pp. 1876-1902, 2011.
 - [21] X.D. Niu, C. Shu, Y.T. Chew, Y.Peng, "A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows," *Physics Letters A*, vol. 354, no. 3, pp. 173-182, 2006.
 - [22] K. H. A. C. T. L. Mark C.Thompson, "Hydrodynamics of a particle impact on a wall," *Applied Mathematical Modelling*, vol. 30, no. 11, pp. 1356-1369, 2006.
 - [23] P. L. Roe, "Approximate riemann solvers, parameter vectors and difference schemes," *Journal of Computational Physics*, vol. 43, pp. 357-372, 1981.
 - [24] F. Bassi and S. Rebay, "A High-Order Accurate Discontinuous Finite Element Method for the Numbical Solution of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, vol. 131, no. 2, pp. 267-279, 1997.
 - [25] P.-O. Persson, "Mesh Generation for Implicit Geometries," MASSACHUSETTS INSTITUTE OF TECHNOLOGY, Boston, 2005.
 - [26] Kartikey Asthana, Jerry Watkins, AntonyJameson, "On consistency and rate of convergence of Flux Reconstruction for time-dependent problems," *Journal of Computational Physics*, vol. 334, pp. 367-391, 2017.
 - [27] Slimane Adjerd, Karen D. Devine, Joseph E. Flaherty, Lilia Krivodonova, "A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1097-1112, 2002.