ARBITRARY LAGRANGIAN-EULERIAN FORM OF FLOW FIELD DEPENDENT VARIATION METHOD FOR FLUID-STRUCTURE INTERACTION APPLICATION

Mohd Fadhli[1], Syariful Syafiq Shamsudin[1], Ashraf Ali Omar[2] and Waqar Asrar[3]
[1]Universiti Tun Hussein Onn Malaysia, BatuPahat, Johor, Malaysia
[2]University of Tripoli, Tripoli, Libya
[3]International Islamic University Malaysia, Kuala Lumpur, Malaysia
E-Mail: fadhli@uthm.edu.my

ABSTRACT

In this study, Flowfield Dependent Variation (FDV) method is coupled with Arbitrary Lagrangian-Eulerian (ALE) method in order to solve fluid-structure interaction problems. FDV method is a mixed explicit-implicit numerical scheme where its implicitness is determined by several parameters that are dependent on the physical properties of the local flow. The scheme which is called as ALE-FDV method is discretized using finite volume method to give flexibility in dealing with complicated geometries. The formulation itself yields a sparse matrix, which can be solved using any iterative algorithm. Several numerical tests have been conducted and the results are in good agreement with exact and available numerical solutions in the literature.

Keywords: arbitrary lagrangian-eulerian, flow field dependent variation, fluid-structure interaction.

INTRODUCTION

The Flow field Dependent Variation (FDV) method has been developed by Chung [1, 2] in order to resolve complex flow problems with a single numerical scheme. This method is a mixed explicit-implicit numerical scheme that changes its implicitness due to the physical properties of the local flow regions. On the other hand, fluid-structure interaction problems in CFD applications are consist of numerical cases where boundary of the bodies are moving and/or deforming and interacting with the flow (e.g. wing flutter and rotating propellers). Some movements of the boundary are relatively small but when they undergo large displacements, rotations or deformations, the effects of fluid-structure interaction cannot be ignored.

The advantages of FDV method in dealing with the transition and interaction of complex flows motivated us to extend its capability into the flow-structure interaction problems. We propose a new technique based on a combination of the FDV and ALE methods [3] for solving moving boundary problems in inviscid and viscous flows. The finite volume method is used for spatial discretization to add flexibility to the proposed method in dealing with complicated geometries.

BASIC EQUATION

FDV Theory

Governing equations of three-dimensional Newtonian fluid in conservative form (without the source term) can be written as,

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}_i}{\partial x_i}+\frac{\partial \boldsymbol{G}_i}{\partial x_i}=0 \qquad (1)$$

where *U*, *F* and *G* are conservative variables, convection flux and diffusion flux vectors, respectively. As introduced by Chung [1, 2], FDV formulation was derived by substituting governing Eqn. (1) into a special form of Taylor series that includes implicitness parameters as shown in Eqn. (2). Instead of being predetermined by users, these parameters are automatically computed based on the physical properties of the local flow field in the computational domains. These parameters, namely convection FDV parameters, s_1 and s_2 are determined based on the gradient of the local Mach number, while diffusion FDV parameters s_3 and s_4 are determined based on the gradient of either local Reynolds number or Peclet number for cases involving high temperature gradients [1]. Full FDV formulation in compact form can be write as,

$$\Delta \boldsymbol{U}^{n+1}=\frac{\partial}{\partial x_i}\left(\Delta t \boldsymbol{D}_i \Delta \boldsymbol{U}\right)^{n+1}-\frac{\partial^2}{\partial x_i \partial x_j}\left(\Delta t \boldsymbol{E}_{ij} \Delta \boldsymbol{U}\right)^{n+1} + \Delta t \frac{\partial \boldsymbol{Q}^n}{\partial x_i}+O\left(\Delta t^3\right) \qquad (2)$$

where $\Delta U^{n+1} = U^{n+1}-U^n$ and

$$\begin{aligned} \boldsymbol{D}_i &= s_1 \boldsymbol{a}_i + s_3 \boldsymbol{b}_i \\ \boldsymbol{E}_{ij} &= s_3 \boldsymbol{c}_{ij} - \frac{\Delta t}{2}\left(\boldsymbol{a}_i+\boldsymbol{b}_i\right)\left(s_2 \boldsymbol{a}_j + s_4 \boldsymbol{b}_j\right) \\ \boldsymbol{Q} &= \boldsymbol{F}_i + \boldsymbol{G}_i - \frac{\Delta t}{2}\left(\boldsymbol{a}_i+\boldsymbol{b}_i\right)\frac{\partial}{\partial x_j}\left(\boldsymbol{F}_j+\boldsymbol{G}_j\right) \end{aligned} \qquad (3)$$

while a, b, and **c** are Jacobians of convection, diffusion, and diffusion gradient, respectively. For inviscid flow problems, the terms related to diffusion effect in the Eqn. (3) can be ignored.

ALE-FDV Method

The proposed numerical scheme combines the FDV method with the ALE technique and discretize it using the finite volume method. ALE technique combines Lagrangian and Eulerian descriptions of a continuum (i.e. fluid and solid) in one numerical scheme for solving moving boundary problems. Finite volume form of FDV method can be obtained by volume integration of the conservative variables $\boldsymbol{U}$ over a moving control volume $\Omega(t)$ as follows,

$$\int_{\Omega(t)} \frac{\Delta \boldsymbol{U}^{n+1}}{\Delta t} \mathrm{d}\Omega = -\int_{\Omega(t)} \left[\frac{\partial}{\partial x_i} \left(\boldsymbol{D}_i \Delta \boldsymbol{U} \right)^{n+1} + \frac{\partial^2}{\partial x_i \partial x_j} \left(\boldsymbol{E}_{ij} \Delta \boldsymbol{U} \right)^{n+1} - \frac{\partial \boldsymbol{Q}^n}{\partial x_i} \right] \mathrm{d}\Omega \tag{4}$$

The starting point in deriving an ALE type numerical scheme is implementation of the Reynolds transport theorem to the control volume $\Omega(t)$ where its boundary $\Gamma(t)$ moves with velocity $\boldsymbol{v}_m$ [3]. Applying the theorem to Eqn. (4) yields,

$$\frac{\Delta}{\Delta t} \left[\int_{\Omega(t)} \boldsymbol{U} \mathrm{d}\Omega \right]^{n+1} = -\int_{\Gamma(t)} \left[\left(\boldsymbol{D}_i \Delta \boldsymbol{U} \right)^{n+1} - \frac{\partial}{\partial x_j} \left(\boldsymbol{E}_{ij} \Delta \boldsymbol{U} \right)^{n+1} + \left(\boldsymbol{Q}^n - \boldsymbol{U}^n \boldsymbol{v}_m \right) \right] \cdot \boldsymbol{n} \mathrm{d}\Gamma \tag{5}$$

where the volume integral on the right hand side of Eqn. (4) has been changed into a surface integral using the divergence theorem.

ALE-FDV Method in finite volume formulation

In two-dimensional problems, the vertex-centered finite volume method is adopted to discretize ALE-FDV formulation (5). Consider a vertex I surrounded by several triangular and quadrilateral elements, $e_J$as shown inFigure-1. Cell (i.e. control volume) of vertex I is defined by joining centroid of its surrounding elements. Face $\Delta\Gamma_{I(J)}{}^1$ or $\Delta\Gamma_{I(J)}{}^2$ is formed either by connecting centroid element e_J or element e_{J+1} with the median of line connecting vertex I and its neighbor vertex $I(J)$. On the other hand, sub-volume of cell I, $\Delta\Omega_{I(J)}$ is formed by joining the median of line I–$I(J$-1), centroid elements e_J, median of line I-$I(J)$ and vertex I.

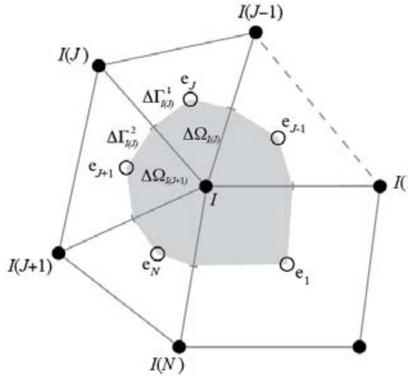


Figure-1. Cell*I* for vertex-centered finite volume.

Based on Figure-1, ALE-FDV formulation in Eqn. (5) can be discretized as,

$$\Delta \boldsymbol{U}_I^{n+1} \left[\sum_{J=1}^{N} \Delta\Omega_{I(J)} \right]^{n+1} + \Delta t \sum_{J=1}^{N} \left[\boldsymbol{D}_i^n \left(\Delta \boldsymbol{U} \right)^{n+1} \cdot \boldsymbol{n} \Delta\Gamma + \boldsymbol{E}_{ij}^n \left(\frac{\partial}{\partial x_j} \Delta \boldsymbol{U} \right)^{n+1} \cdot \boldsymbol{n} \Delta\Gamma \right]_{I(J)} = \boldsymbol{R} \tag{6}$$

where known terms on the right hand side are,

$$\boldsymbol{R} = -\Delta t \sum_{J=1}^{N} \left[\left(\boldsymbol{Q}^n - \boldsymbol{U}^n \boldsymbol{v}_m \right) \cdot \boldsymbol{n} \Delta\Gamma \right]_{I(J)} - \boldsymbol{U}_I^n \Delta \left[\sum_{J=1}^{N} \Delta\Omega_{I(J)} \right]^{n+1} \tag{7}$$

Vertex-centered finite volume was chosen because the implementation of wall boundary condition can be defined explicitly (e.g. velocity of the flow on some point of the wall is same as the wall velocity of that particular point for moving boundary problems) without using some additional points. In the case of cell-centered finite volume, usually additional cells called ghost cells are used for wall boundary condition.

For three-dimensional moving boundary problems, the cell-centered finite volume method is adopted to discretize ALE-FDV formulation in Eqn. (5).

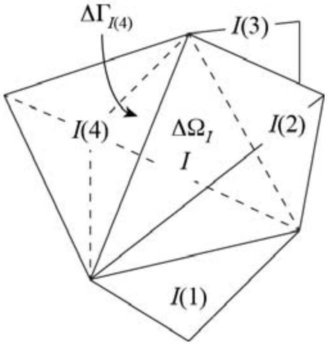


Figure-2. Cell I for cell-centered finite volume.

Consider a tetrahedron element *I* surrounded by other elements *I*(*J*) as shown in Figure-2. In cell-centered finite volume method, an element is the control volume or the cell. The volume of the cell is $\Delta\Omega_I$ and each of its faces is $\Delta\Gamma_{I(J)}$. Based on this figure, ALE-FDV formulation in Eqn. (5) is discretized as

$$\Delta \boldsymbol{U}_I^{n+1}\left[\Delta\Omega_I\right]^{n+1} + \Delta t \sum_{J=1}^{N} \Big[\boldsymbol{D}_i^n \left(\Delta \boldsymbol{U}\right)^{n+1} \cdot \boldsymbol{n}\Delta\Gamma + \boldsymbol{E}_{ij}^n \left(\frac{\partial}{\partial x_j}\Delta \boldsymbol{U}\right)^{n+1} \cdot \boldsymbol{n}\Delta\Gamma \Big]_{I(J)} = \boldsymbol{R} \qquad (8)$$

where known terms on the right hand side are,

$$\boldsymbol{R} = -\Delta t \sum_{J=1}^{N} \Big[\left(\boldsymbol{Q}^n - \boldsymbol{U}^n \boldsymbol{v}_m\right) \cdot \boldsymbol{n}\Delta\Gamma \Big]_{I(J)} - \boldsymbol{U}_I^n \Delta\left[\Delta\Omega_I\right]^{n+1} \qquad (9)$$

The cell-centered finite volume form of ALE-FDV method is similar with its vertex-centered form shown in Eqn. (6) and (7). However, the number of faces/boundaries and adjacent cells of the cell-centered finite volume form of ALE-FDV are lower than the vertex-centered form, and the volume for the cell-centered is evaluated directly from the element vertices. On the other hand, cell-centered finite volume method requires additional cells called ghost cells to evaluate boundary condition. However, since ghost cells are added only at the computational boundaries, overall cell-centered form require less computational works and memory than vertex-centered form. The computational issues in three-dimensional computation involve long computational time and high usage of memory, thus cell-centered finite volume is typically preferable than the vertex-centered method in three-dimensional computation cases.

In both vertex-centered and cell-centered formulation, interface variables defined at the center of each faces can be interpolated using known variables from the cell's centroid and the centroid of adjacent cells next to each faces. Similarly, gradients of interface variables can be interpolated using the gradient of variables of each cell's centroid which can be approximated using the Green-Gauss method [4]. Meanwhile, third-order accurate MUSCL scheme for unstructured mesh with minmod limiters [5] is used to approximate the inviscid flux vectors. Finally, the resulting equation written in compact form is given as,

$$\boldsymbol{K}_I \Delta \boldsymbol{U}_I^{n+1} + \cdots \boldsymbol{K}_{I(J)} \Delta \boldsymbol{U}_{I(J)}^{n+1} \cdots + \boldsymbol{K}_{I(N)} \Delta \boldsymbol{U}_{I(N)}^{n+1} = \boldsymbol{R}_I \qquad (11)$$

where K_I and $K_{I(J)}$ denote the collective sum of contributions for the main cell and its surrounding cells respectively. In order to solve the linear equation in Eqn. (11), we used an iterative method called Restart General Minimal Residual (Restart GMRES) algorithm [6]. As improvement to the convergence rate of the Restart GMRES algorithm, the algorithm has been combined with Left Block Gauss-Seidel pre-conditioner [7] to accelerate the convergence of solutions. While there exists other type of pre-conditioners that converge faster, the Gauss-Seidel pre-conditioner was chosen because it is relatively simpler and easy to use for parallel computing.

Numerical algorithm

In the main program flowchart shown inFigure-3, the program starts the calculation routine by reading the geometry file that has information about the computational mesh and the connection of cells in the mesh. Next, a suitable initial condition of the problem under consideration is set, and an iterative process (for steady problem) or time advancement process (for unsteady problems) will start. The iterative or time advancement process begins with appropriate boundary conditions setup and the ALE-FDV sub-program will determine the solution at each iteration or time step until the solutions are converged or the process meets the stopping criteria. At the same time, intermediate solutions will also be produced at desired times or step intervals.

As shown in Figure-4, the ALE-FDV sub-program will produces solution for mesh movement and flow field. First, the mesh movement will be computed based on some movement function or fluid forces determined from the solutions of the previous time step. Next, the inviscid and viscous flux as well as the FDV parameters will be computed. Then vector *R* and matrix *K* will be computed, and the resulting linear system of equations is solved using Restart GMRES algorithm. Next, the solutions will be updated to the current time step and will be used to compute the fluid forces that work on the body surfaces. Finally, the sub-program will return to the main program and the whole process will be repeated again until the main program meets predefined stopping criteria.

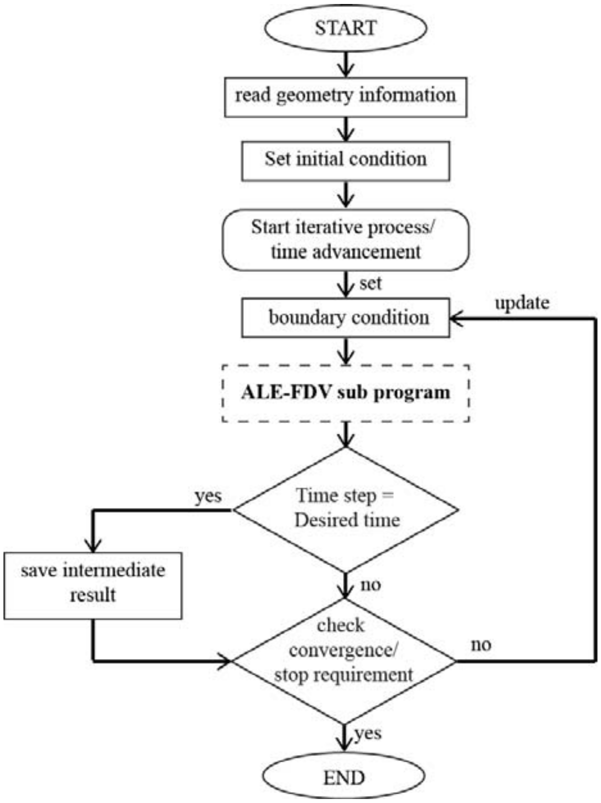


Figure-3. Flowchart of ALE-FDV method's main program.

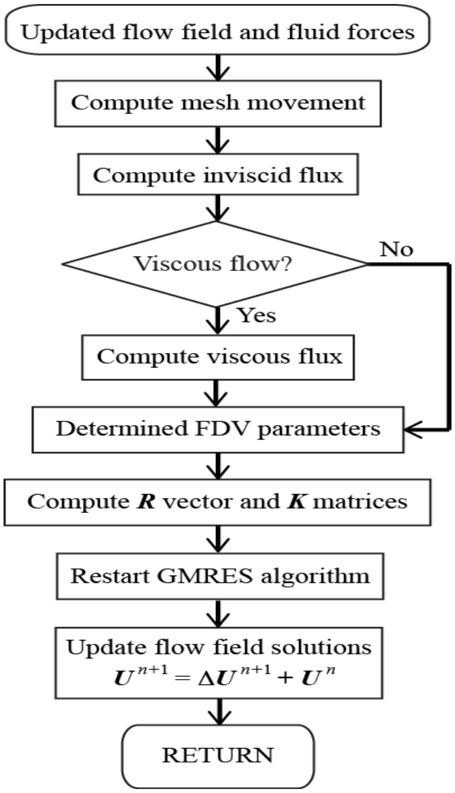

Updated flow field and fluid forces
Compute mesh movement
Compute inviscid flux
Viscous flow?
No
Yes
Compute viscous flux
Determined FDV parameters
Compute R vector and K matrices
Restart GMRES algorithm
Update flow field solutions
$U^{n+1} = \Delta U^{n+1} + U^n$
RETURN

Figure-4. Flowchart of ALE-FDV method's sub-program.

RESULTS AND DISCUSSIONS

Free-Stream preservation test

In order to verify whether the ALE-FDV method satisfies the Geometric Conservation Law (GCL) [8, 9], a free-stream preservation test is conducted on the two and three-dimensional wavy mesh that deformed under the following motion [10]:

$$x_i^{n+1} = x_i^n + \Delta x_i$$
$$\Delta x_i(t) = A_i L_i \left(\frac{\Delta t}{T}\right)\sin\left(x_j \frac{c_j}{L_j}\right)\sin\left(x_k \frac{c_k}{L_k}\right)\sin\left(t\frac{c_t}{T}\right) \qquad (12)$$

with i= 1, 2,j = 1 and k = 2 for two-dimensional cases while i, j, k = 1, 2, 3 and $i \neq j$, $j \neq k$, $i \neq k$ for three-dimensional wavy mesh. The parameters used for both mesh motion are A_i = 0.1, L_i = 1.0 and T = 20, respectively, while all constants c_i= 4π and constant c_t-=π.As shown in Figure-5, two-dimensional mesh is a square domain with a size of 1.0 unit^2 and the mesh has initially 512 uniform triangular elements. On the other hand, three-dimensional mesh is a cube with a size of 1.0 unit^3 and has initially 8000 uniform quadrilateral elements. Both cases produce solution up until t = T where the mesh is at its maximum distortion as depicted in Figure-5 and 6.

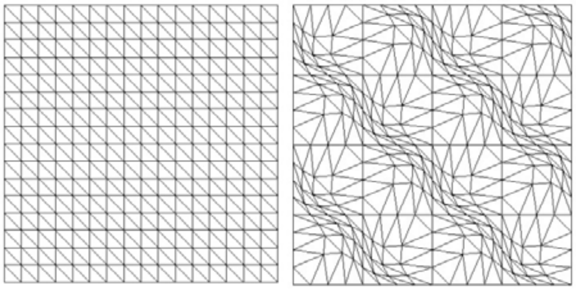


Figure-5. Initial mesh (left) and mesh at t =T (right) (2D).

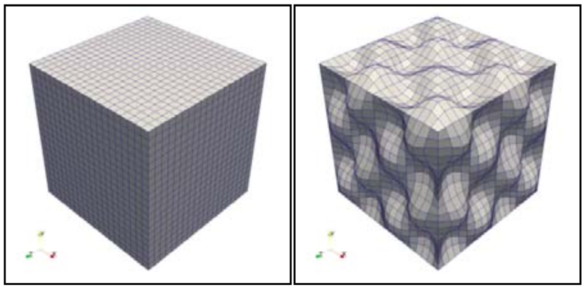


Figure-6. Initial mesh (left) and mesh at t = T (right) (3D).

The velocity u_1 are expected to remain at 1.0 while velocity u_2 and u_3 are expected to remain zero if this method satisfied GCL. Results of maximum error norm, $L\infty$ of velocities at time instant t = 0.5T and t = T are reported in Table-1 for two-dimensional case and Table-2 for three-dimensional case. The present results show that the ALE-FDV method has errors around 10^{-15}, which is closed to machine zero for double precision computation. Therefore, the free-stream preservation is achieved and the ALE-FDV method is verified to be GCL compliant.

Table-1. Velocities maximum error norm for two-dimensional wavy mesh.

time	$L\infty(u_1)$	$L\infty(u_2)$
0.5*T*	1.33E-15	1.94E-15
T	1.77E-15	1.44E-15

Table-2. Velocities maximum error norm for three-dimensional wavy mesh.

time	$L\infty(u_1)$	$L\infty(u_2)$	$L\infty(u_3)$
0.5*T*	1.84E-16	1.57E-16	1.64E-16
T	1.27E-15	4.34E-16	5.04E-16

Rapidly pitching NACA 0015 airfoil

In this section, we consider a laminar flow past a NACA 0015 airfoil that is pitched rapidly from 0° angle of attack at a constant rate about the axis located at the quarter chord. The pitching rate formulation is given as$\omega(t) = \omega_0[1 - \exp(-4.6\ t/t_0)]$. The final pitch rate is set at$\omega_0 = 0.6$ while the time taken by the airfoil to reach 99% of ω_0 is given by $t_0 = 1.0$. The free stream Mach number and Reynolds number is $M = 0.2$ and $Re = 10000$, respectively. The time interval is set as $\Delta t = 5 \times 10^{-5}$. As shown in Figure-7, the mesh used in this problem is an unstructured triangular mesh with 28314 mesh points.

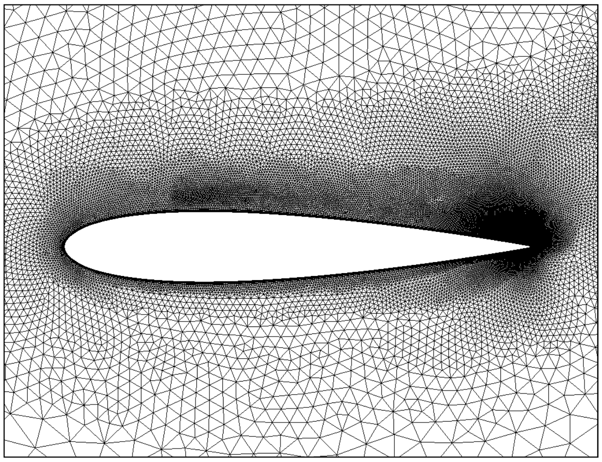
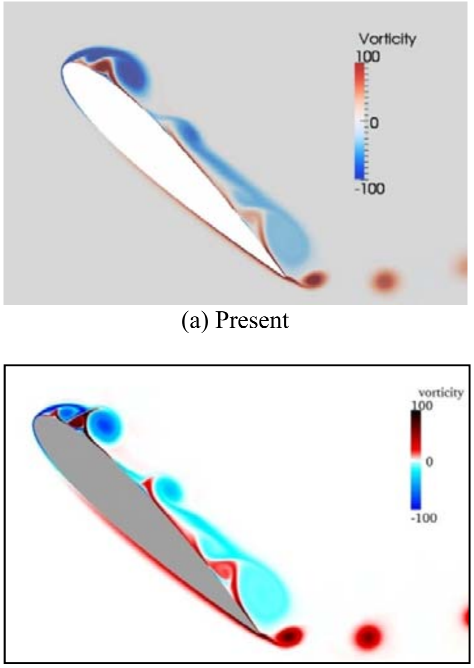


Figure-7. Unstructured mesh around NACA 0015 airfoil.

This case was first studied by Visbal and Shang [11] and later was used for validation of numerical solutions by Lomtev, Kirby, and Karniadakis [12] and Schneiders *et al*. [13].Vorticity profile at angle of attack, AoA = 44° obtained by the proposed method is shown in Figure-8(a). As comparison, Figure-8(b) also shows the same vorticity results obtained by Schneiders *et al*. [13]. Both figures show similar pattern of vortex structures such as the leading edge vortex and trailing edge vortex detached from the airfoil and several shear layer vortices existing on the upper surface of the airfoil.



(a) Present

(b) Schneiders *et al*. [13]

Figure-8. Comparison of vorticity profile (AoA = 44 deg).

Figure-9 shows the comparison of drag and lift coefficient obtained by the proposed method and the above mentioned references. The trends of lift and drag coefficient from the present results match relatively well with the results from the literatures.

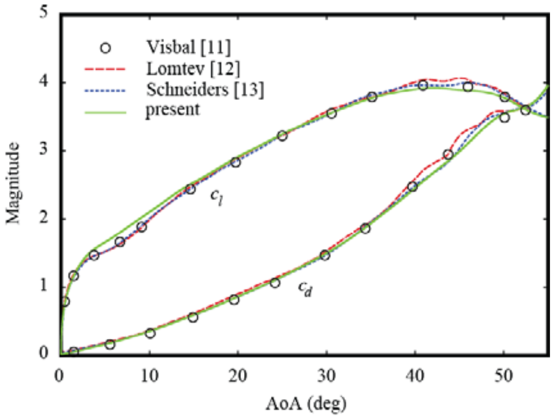
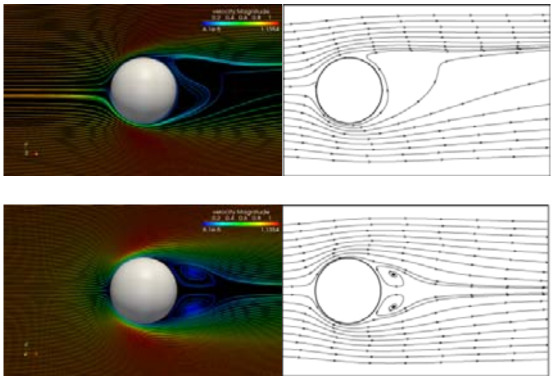


Figure-9. Comparison of lift coefficient, $c_l$and drag coefficient, c_d.

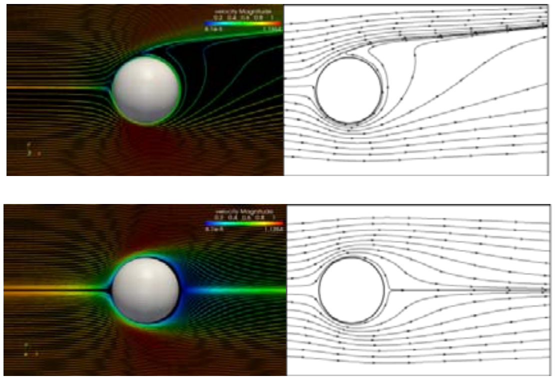
Viscous flow over rotating sphere

Three-dimensional simulation of a viscous flow past a rotating sphere has been conducted using the proposed method. In the present case, Reynolds number of the free stream flow is set at 100, thus is considered laminar. Meanwhile, the sphere of radius R is rotated about an axis perpendicular to the free stream direction. Two non-dimensional rotational speeds ($\omega = \omega^{*}R/U_{\infty}$) at $\omega = 0.25$ and $\omega = 0.6$ are considered in this simulation. The flow over a stationary sphere ($\omega = 0.0$) is first simulated, and its converged solutions are then used as an initial flow field for the rotating sphere. The simulation was conducted using unstructured mesh with 48840 elements and a time interval of 0.0005(Courant number is approximately 1.25). With the same initial condition, Kim [14] used the immersed boundary method to simulate the flow over a rotating sphere. As comparison, the results obtained from the proposed method and the results from Kim [14] are shown in Figure-10. Figure-10(a) and 11(b) shows the streamlines on the x-y and x-z plane when the sphere is rotating at a speed of $\omega = 0.25$ and $\omega = 0.6$, respectively. Similar to the results of Kim [14], the results show that the flow loses its axisymmetric feature and becomes planar-symmetric as the sphere is rotated. As shown in Figure-10(a), the wake is deflected upward (x-y plane) while the recirculation region behind the sphere becomes smaller (x-z plane) at rotational speed of $\omega = 0.25$. Furthermore, Figure-10(b) shows the deflection of the wake becomes significant (x-z plane) and the recirculation region behind the sphere disappears as the rotational speed is increased to $\omega = 0.6$.

The CPU time per iteration, total iteration as well as the total CPU time taken by the proposed method to converge for the case of rotational speed, $\omega = 0.6$ is shown in Table-3. As comparison, the CPU time per iteration taken by the commercial software, ANSYS Fluent is shown in the same table. The proposed method takes around 3% to 5% higher CPU time per iteration as expected because of the additional terms involving the FDV parameters that need to be calculated. However, the total iteration taken by the present method to converge is 8% lower, thus total CPU time taken by the present method is shorter than the ANSYS Fluent software. Therefore, the proposed method solves the problem slightly faster even though it takes higher CPU time per iteration.



(a) $\omega = 0.25$ (above: x-y plane, below: x-z plane)



(b) $\omega = 0.6$ (above: x-y plane, below: x-z plane)

Figure-10. Streamline profile of present method (left) and Kim [14] (right) at different rotation speed.

Table-3. Comparison of CPU time taken by ALE-FDV and ANSYS Fluent.

	ALE-FDV	*FLUENT*
CPU time (sec)	4.904	4.776
Total iteration	17280	18830
Total CPU time (hours)	23.539	24.981

CONCLUSIONS

The capability of the ALE-FDV method in solving two and three-dimensional fluid-structure interaction problems has been presented in this study. We have formulated the FDV method in ALE form for moving mesh applications, motivated by our interest to extend its capability to fluid-structure interaction. In all test problems, the proposed ALE-FDV method has shown its capability to operate with relatively large time step and Courant number higher than one. The results obtained are in good agreement with other experimental and numerical methods thus confirming its applicability in solving complex flow interaction problems.

REFERENCES

[1] T.J. Chung. 1999. Transitions and interactions of inviscid/ viscous, compressible/ incompressible and laminar/ turbulent flows. International Journal for Numerical Methods in Fluids. 31: 223-246.

[2] T.J. Chung. 2002. Computational Fluid Dynamics. Cambridge University Press, New York, USA.

[3] J.Donea, A. Huerta, J.P. Ponthot and A. Rodríguez-Ferran. 2004. Arbitrary Lagrangian-Eulerian Methods. Encyclopedia of Computational Mechanics.

[4] B. Diskin, J.L. Thomas, E. J. Nielsen, H. Nishikawa and J.A. White. 2010. Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume

Discretizations: Viscous Fluxes. AIAA Journal 48: 1326-1338.

[5] [5] M.S. Darwish and F. Moukalled. 2003. TVD schemes for unstructured grids. International Journal of heat and mass transfer, 46: 599-611.

[6] Y.Saad and M.H.Schultz.1986. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing. 7:856-869.

[7] P.O. Perssonand J. Peraire. 2008. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. SIAM Journal on Scientific Computing. 30: 2709-2733.

[8] P.D. Thomas and C.K. Lombard. 1979. Geometric conservation law and its application to flow computations on moving grids. AIAA Journal. 17: 1030-1037.

[9] H. Guillardand C. Farhat. 2000. On the significance of the geometric conservation law for flow computations on moving meshes. Computer Methods in Applied Mechanics and Engineering. 190: 1467-1482.

[10] T. Nonomura, N. Iizuka and K. Fujii. 2010. Freestream and vortex preservation properties of high-order WENO and WCNS on curvilinear grids. Computers & Fluids. 39: 197-214.

[11] M.R. Visbal and J.S. Shang.1989. Investigation of the flow structure around a rapidly pitching airfoil. AIAA journal.27: 1044-1051.

[12] I. Lomtev, R.M. Kirby and G.E. Karniadakis. 1999. A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. Journal of Computational Physics. 155: 128-159.

[13] L. Schneiders, D. Hartmann, M. Meinke and W. Schröder. 2013. An accurate moving boundary formulation in cut-cell methods. Journal of Computational Physics. 235: 786-809.

[14] D. Kim. 2009. Laminar flow past a sphere rotating in the transverse direction. Journal of Mechanical Science and Technology. 23: 578-589.