Flowfield-Dependent Variation Method for Moving-Boundary Problems

Mohd Fadhlį*

University Tun Hussein Onn Malaysia, 86400 Batu Pahat, Malaysia

Ashraf A. Omar†

University of Tripoli, Sidi Almasry, Al Furnaj Road, Tripoli, Libya

and

Waqar Asrar‡

International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia

DOI: 10.2514/1.J053353

A novel numerical scheme using the combination of flowfield-dependent variation method and arbitrary Lagrangian–Eulerian method is developed. This method is a mixed explicit–implicit numerical scheme, and its implicitness is dependent on the physical properties of the flowfield. The scheme is discretized using the finite-volume method to give flexibility in dealing with complicated geometries. The formulation itself yields a sparse matrix, which can be solved by using any iterative algorithm. Several benchmark problems in two-dimensional inviscid and viscous flow have been selected to validate the method. Good agreement with available experimental and numerical data in the literature has been obtained, thus showing its promising application in complex fluid–structure interaction problems.

Nomenclature

\(a\) = Jacobian of convection flux
\(b\) = Jacobian of diffusion flux
\(c_{ij}\) = Jacobian of gradient diffusion flux
\(F\) = vector of convection flux
\(G\) = vector of diffusion flux
\(I, J\) = vertex index
\(M\) = Mach number
\(n\) = normal vector
\(Re\) = Reynolds number
\(s_{a, s_b}\) = implicitness parameters
\(s_{a, s_b}\) = convection flowfield-dependent variation parameters
\(s_{3, s_4}\) = diffusion flowfield-dependent variation parameters
\(t\) = time
\(U\) = vector of conservative variables
\(\nu_m\) = mesh velocity
\(x\) = spatial coordinate
\(\Gamma\) = control surface
\(\Omega\) = control volume

Subscripts

\(i, j, k\) = spatial dimension index

Superscript

\(n\) = time level

I. Introduction

The study of fluid phenomena is highly dependent on numerical simulations because computational approaches are usually more economical. In fluid mechanics, such an approach is computational fluid dynamics, and its development began with the advent of the computer in the 1950s. Continuous development of computational power has enabled researchers to study fluid flow around complete structures such as helicopters, airplanes, and spacecraft. However, algorithm efficiency of numerical solvers has become an issue as the size and the complexities of the problems increased. Moreover, multidisciplinary problems such as moving-boundary or fluid–structure interaction problems become more relevant as the accuracy of the solution to the real-world application has becomes more demanding than ever before.

Recently, the so-called flowfield-dependent variation (FDV) method has been developed [1] as an approach to solve complex fluid-flow problems with a single numerical solver. Based on earlier works of Yoon and Chung [2] and Yoon et al. [3], the FDV method was devised by Chung [1] to deal with a domain that contains all speed flows and various physical properties. The FDV method was formulated using the expansion of conservative variables in a special form of Taylor series that included parameters that govern the implicitness of the formulation. Furthermore, these parameters vary locally and made dependent on the physical properties of the flow instead of predetermined like other existing schemes and hence are named as FDV parameters. In particular, these parameters are evaluated based on the local gradients of certain physical parameters such as Mach and Reynolds numbers in a computational domain. As a result, the FDV method becomes a mixed explicit–implicit numerical scheme that will adjust accordingly for every computational point based on the flow properties in that domain [1–4].

Chung [1] applied finite-element method to approximate FDV formulation, but later Schunk et al. [5] addressed the FDV method as a strategy toward unification of finite-element, finite-volume, and finite-difference methods. Spatial discretization is simply the option to discretize between adjacent grid points or within an element but does not dictate the physics because all the physical phenomena are taken into account in FDV equations. Recently, Elfgahi et al. [6,7] has successfully combined the finite-difference form of FDV method with higher-order compact method, which makes the method more efficient in obtaining high-accuracy solutions. Moreover, the FDV method has been expanded to high-energy astrophysics application by Richardson et al. [8]. They found that the FDV method is...
numerically stable and comparable with Yee's total variation diminishing (TVD) method [9] in terms of accuracy.

On the other hand, moving-boundary problems or fluid–structure interaction problems are the cases where the structure interacts with its surrounding fluid through the movement of its boundaries or the shape and location of the original structure changes due to the effect of the fluid flow. Some examples are rotating propellers, wing flutter, parachute openings, flapping wings, accelerating aircraft or cars, reciprocating engines, vibration of suspension bridges, pulsating blood vessels, etc. Some movements of the boundaries are relatively small, but when they undergo large displacements, rotations, or deformations, the effects of fluid–structure interaction cannot be ignored. The need to solve such flow problems using dynamic mesh has attracted many researchers to develop various kinds of moving-grid-interpolation techniques, and one of them is the arbitrary Lagrangian–Eulerian (ALE) method [10]. The ALE method was originally introduced in a finite-difference formulation [11] and has been successfully implemented in finite-volume (Xia and Lin [12], Guardone et al. [13] and Habchi et al. [14]) and finite-element formulations (Feistauer et al. [15,16] and Sun et al. [17]). In addition, many researchers have enforced the geometric conservation law (GCL) [18,19] due to the influence of remeshing technique in the stability and accuracy of the ALE method. However, development of numerical schemes for moving-boundary cases still remain specifically on the physical properties of the governing flows. For instance, the ALE application developed by Guardone et al. [13] and Feistauer et al. [16] is based on the compressible flow model, whereas Xia and Lin [12], Sun et al. [17], and Habchi et al. [14] are based on the incompressible flow model.

This paper proposes a novel technique based on a combination of the ALE and FDV methods as a way to overcome difficulties when dealing with complex flow interactions due to the moving boundary. Our objective to extend the FDV method for multidisciplinary applications is motivated by the capability of the FDV method in dealing with the transition and interaction of complex fluid flows. Although some researchers have implemented the FDV method as an error indicator for adaptive mesh refinement [20], to the authors' knowledge, to date this method has not been fully incorporated into a fluid–structure application that involves dynamic meshes.

Details of the FDV theory and the development of the ALE–FDV method for two-dimensional applications are written in Sec. II. The applicability of the ALE–FDV method for moving boundaries in two-dimensional inviscid and viscous flow problems is demonstrated in Sec. III as well as the discussion about its accuracy and validity. We conclude the paper in Sec. IV. The ALE–FDV method is expected to provide a new technique in resolving the interaction of arbitrary bodies in complex flowfields.

II. Arbitrary Lagrangian–Eulerian Form of Flowfield-Dependent Variation Formulation

A. Flowfield-Dependent Variation Method

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

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

where the definition of vectors $\mathbf{U}$, $\mathbf{F}$, and $\mathbf{G}$ are given as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_i \\ \rho u_j \\ \rho u_k \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u_i \\ \rho u_i u_j + p \delta_{ij} \\ \rho u_i u_j + p \delta_{ij} \\ \rho u_i u_k + p \delta_{ik} \\ \rho u_i (\rho E + p) u_j \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ -\frac{1}{Re} \tau_{ij} \end{bmatrix}$$

Here, $\rho$ and $p$ are density and pressure (all flow properties and quantities are dimensionless), respectively, and $u_i$ is the velocity element in Cartesian coordinate. $E$ and $e$ are the total energy and internal energy per unit mass, respectively, $Re_{\infty}$ is the reference (e.g., free-stream) Reynolds number defined as $Re_{\infty} = \rho_{\infty} V_{\infty} L / \mu_{\infty}$, where $\rho_{\infty}$, $V_{\infty}$, and $\mu_{\infty}$ are reference density, reference velocity, and reference dynamic viscosity, respectively; and $L$ is the characteristic length. $Pr$ is the Prandtl number, and $\gamma$ is 1.4 for ideal gases. For Newtonian fluids, viscous stresses $\tau_{ij}$ are proportional to the velocity gradients:

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \frac{\partial u_i}{\partial x_i} \delta_{ij}$$

where $\delta_{ij}$ is the Kronecker delta, $\mu$ is the dynamic viscosity coefficient, and $\lambda$ is approximated as $\lambda = -2/3\mu$ based on Stokes's hypothesis. Chung et al. [1] developed the FDV method by introduced parameters $s_a$ and $s_b$ into a special form of Taylor series of conservative variables $U$ and then applied the governing Eq. (1) into such a series. Consider the expansion of $U$ in the special form of Taylor series up to the second-order time derivatives as

$$\Delta U^{n+1} = \Delta t \left( \frac{\partial \mathbf{U}}{\partial t} + s_a \frac{\partial \Delta U^{n+1}}{\partial t} \right) + \frac{\Delta t^2}{2} \left( \frac{\partial^2 \mathbf{U}}{\partial t^2} + s_b \frac{\partial^2 \Delta U^{n+1}}{\partial t^2} \right) + O(\Delta t^3)$$

and second-order time derivatives of $U$ as

$$\frac{\partial^2 \mathbf{U}}{\partial t^2} = -\frac{\partial}{\partial x_i} \left( (a_i + b_i) \frac{\partial F_i}{\partial x_j} + c_{ij} \frac{\partial F_i}{\partial x_j} \right)$$

where $\Delta U^{n+1}$ is defined as $U^{n+1} - U^n$, and the Jacobians $a_i$, $b_i$, and $c_{ij}$ are defined as $\partial F_i / \partial U$, $\partial G_i / \partial U$, and $\partial G_i / (\partial U / \partial x_j)$, respectively. Substituting Eqs. (1) and (5) into Eq. (4), and assuming third-order derivatives to be negligible, yields

$$\Delta U^{n+1} = \Delta t \left( -\frac{\partial F_i}{\partial x_j} + \frac{\partial G_i}{\partial x_j} - s_a \left( \frac{\partial F_i}{\partial x_j} + \frac{\partial G_i}{\partial x_j} \right) \right) + \frac{\Delta t^2}{2} \left( \frac{\partial}{\partial x_i} \left( (a_i + b_i) \frac{\partial F_i}{\partial x_j} + c_{ij} \frac{\partial F_i}{\partial x_j} \right) \right)$$

Parameters $s_a$ and $s_b$ have a value between 0 and 1 and are called implicitness parameters because the equation becomes fully explicit if both parameters are 0 or fully implicit if they are 1. If their values are between 0 and 1, they act as a weight factor between explicit and implicit equations. The principal idea in the FDV method is that implicitness parameters are flowfield-dependent (i.e., calculated automatically from physical changes in the flow instead of predetermined manually to a single number irrespective of the variation of the local flowfield). Therefore, within Eq. (6), parameters $s_a$ and $s_b$ are defined to be flowfield-dependent parameters and split into several convection and diffusion parameters $s_1-S_4$ [1] as follows:

$$s_a \left( \Delta F + \Delta G \right) \Rightarrow s_1 \Delta F + s_2 \Delta G$$

$$s_b \left( \Delta F + \Delta G \right) \Rightarrow s_3 \Delta F + s_4 \Delta G$$

Parameters $s_1$ and $s_2$ are named as first-order and second-order convection FDV parameters, whereas $s_3$ and $s_4$ are named as first-order and second-order diffusion FDV parameters, respectively. As shown in the following equation, convection and diffusion FDV parameters are defined to be dependent on the gradient of the local
Mach number and local Reynolds number, respectively, within adjacent computational nodes:

\[
s_1 = \begin{cases} 
    \min(r_e, 1) & r_e > \alpha \\
    0 & r_e < \alpha \\
    M_{\min} & M_{\min} \neq 0 \\
    1 & M_{\min} = 0 
\end{cases}
\]

\[s_2 = \frac{1}{2}(1 + s_1^2)
\]

\[
s_3 = \begin{cases} 
    \min(r_d, 1) & r_d > \alpha \\
    0 & r_d < \alpha \\
    R_{\min} & R_{\min} \neq 0 \\
    1 & R_{\min} = 0 
\end{cases}
\]

\[s_4 = \frac{1}{2}(1 + s_3^2)
\]

In Eq. (8), the constant \(\alpha\) is chosen typically about 0.01, whereas \(\eta\) is chosen appropriately between 0.05 and 0.2, depending on the problem being solved [4,7]. For the sake of simplicity, \(\eta\) is set as 0.01 and \(\alpha\) is chosen as 0.1 for all problems investigated in this study. If the problem being solved involves high temperature gradients, such as in high-speed flows, the Péclet number can be used instead of the Reynolds number.

As a result, changing implicitness parameters \(s_a\) and \(s_b\) in Eq. (6) into FDV parameters through Eq. (7) yields

\[
\Delta U^{n+1} = -\Delta t \left[ s_1 \left( \frac{\Delta F^{n+1}}{\Delta x_i} + s_3 \left( \frac{\Delta G^{n+1}}{\Delta x_i} \right) \right) \right] + \Delta t \frac{\partial}{\partial x_i} (D_i \Delta U^n) + \Delta t \left[ \frac{\partial^2}{\partial x_i \partial x_j} (E_{ij} \Delta U^n) - \Delta t \frac{\partial Q^n}{\partial x_i} \right] + O(\Delta t^3)
\]

Rearranging the equation further by replacing unknown flux terms at time level \(n + 1\) with their associated Jacobians and assuming third-order derivatives to be negligible yields the final form of the FDV method:

\[
\Delta U^{n+1} = -\frac{\partial}{\partial x_i} (D_i \Delta U^n) - \frac{\partial^2}{\partial x_i \partial x_j} (E_{ij} \Delta U^n) - \Delta t \frac{\partial Q^n}{\partial x_i}
\]

(10)

where

\[D_i = s_2 a_i + s_3 b_i\]

\[E_{ij} = s_3 c_{ij} - \frac{\Delta t}{2} (a_i + b_i) \left( s_2 a_j + s_3 b_j \right)\]

\[Q = F_i + G_i - \frac{\Delta t}{2} (a_i + b_i) \frac{\partial}{\partial x_j} (F_j + G_j)\]

(11)

B. Arbitrary Lagrangian–Eulerian Form of Flowfield-Dependent Variation Method

The fluid is described in an Eulerian manner in fluid mechanics because fluid particles are moving with respect to the fixed computational mesh. In solid mechanics, however, the solid is described in a Lagrangian manner because each nodal point of the computational domain follows the movement of associated structures. The arbitrary Lagrangian–Eulerian (ALE) technique combines Lagrangian and Eulerian descriptions of a continuum (i.e., fluid and solid) in a single numerical scheme. As a result, ALE-type numerical schemes allow a computational mesh to move arbitrarily as well as following adequately the movement of existing structures in the fluid, whereas the fluid is still seen in the Eulerian manner.

To apply the FDV method to solve moving-boundary problems, we propose to develop the FDV method in an ALE formulation, which we name as the ALE–FDV method. In addition, because we intend to make the proposed method applicable in solving flow problems involving complicated geometries, we discretize it using the finite-volume method. To our knowledge, up to this point, the FDV method has only been formulated in the Eulerian manner. The finite-volume form of FDV method can be obtained by volume integration of Eq. (10) over a control volume \(\Omega\) [4]. Considering a moving control volume at time instant \(t\) as \(\Omega(t)\) and its boundaries as \(\Gamma(t)\), the integral form of FDV formulation is

\[
\int_{\Omega(t)} \frac{\Delta U^{n+1}}{\Delta t} d\Omega = -\int_{\Gamma(t)} (D_i \Delta U^n + E_{ij} \Delta U^n) + Q^n \cdot n d\Gamma
\]

(12)

Moreover, because the control volume is not fixed in space for a numerical method involving deforming mesh, the rate of change of a control volume must be taken into account. To determine scalar quantities (i.e., density, pressure, and velocity components, etc.), inside the control volume, ALE-type numerical scheme uses Reynolds's transport theorem [10] to connect the rate of change of the control volume with the changes of scalar quantities inside the control volume. Consider the boundary of the control volume \(\Omega(t)\) move with velocity \(v_m\) and \(\varphi\) is a scalar quantity inside the control volume; Reynolds's transport theorem gives the time rate of change of the control volume as

\[
\frac{d}{dt} \int_{\Omega(t)} \varphi \, d\Omega = \int_{\Omega(t)} \frac{\partial \varphi}{\partial t} d\Omega + \int_{\Gamma(t)} \varphi v_m \cdot n \, d\Gamma
\]

(13)

Replacing \(\varphi\) with fluid density \(\rho\), momentum \(\rho u_i\), and specific total energy \(\rho E\), Eq. (13) can be rewritten in vector and semidiscrete form as

\[
\frac{d}{dt} \int_{\Omega(t)} U \, d\Omega = \int_{\Omega(t)} \frac{\partial U}{\partial t} d\Omega + \int_{\Gamma(t)} U^n v_m \cdot n \, d\Gamma
\]

(14)

The FDV method is applied by substituting the first term of the right-hand side of Eq. (14) with the FDV formulation written in Eq. (12), which yields

\[
\frac{\Delta}{\Delta t} \int_{\Omega(t)} U \, d\Omega = -\frac{\partial}{\partial t} \int_{\Omega(t)} (D_i \Delta U^n) + E_{ij} \Delta U^n + (Q^n - U^n v_m) \cdot n \, d\Gamma
\]

(15)

Equation (15) is linearized by applying the chain rule on the left-hand side term:

\[
\Delta \left[ \int_{\Omega(t)} U \, d\Omega \right] = \Delta U^{n+1} \left[ \int_{\Omega(t)} d\Omega \right] + U^n \Delta \left[ \int_{\Omega(t)} d\Omega \right] - \Delta \left[ \int_{\Gamma(t)} (Q^n - U^n v_m) \cdot n \, d\Gamma \right]
\]

(16)

The right-hand side of Eq. (15) is linearized by lagging the \(D\) and \(E\) terms one time step behind, which yields the ALE form of the FDV method:

\[
\Delta U^{n+1} \left[ \int_{\Omega(t)} d\Omega \right] = \Delta t \left[ D_i (\Delta U^n) + E_{ij} \frac{\partial}{\partial x_j} (\Delta U^n) \right] - \Delta t \int_{\Gamma(t)} (Q^n - U^n v_m) \cdot n \, d\Gamma
\]

(17)

Equation (17) is applicable for three-dimensional flow problems, and any standard finite-volume technique can be used to discretize the volume and surface integrals in it. However in this paper, the application of the ALE–FDV method is focused on two-dimensional moving-boundary problems, and we adopted vertex-centered finite-
volume method to discretize Eq. (17). Vertex-centered finite volume was chosen because the implementation of the wall-boundary condition can be defined explicitly (e.g., the 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, as in the case of cell-centered finite volume, where usually additional cells called ghost cells are used to apply the wall-boundary condition.

Consider a vertex \( I \) surrounded by several triangular and quadrilateral elements \( e_j \), as shown in Fig. 1. The cell (i.e., control volume) of vertex \( I \) is defined by joining centroid of its surrounding elements. Face \( \Delta \Omega_{IJ}^1 \) or \( \Delta \Omega_{IJ}^2 \) 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 \( J \). On the other hand, the subvolume of cell \( I \), \( \Delta \Omega_{IJ}^3 \), is formed by joining the median of line \( I - I(J - 1) \), centroid elements \( e_j \), the median of line \( I - I(J) \), and vertex \( I \).

Based on Fig. 1, ALE–FDV formulation can be discretized as:

\[
\begin{align*}
\Delta U_{ij}^{n+1} &= \sum_{j=1}^{N} \Delta \Omega_{ij}^3 \left[ D_f \left( \frac{\partial}{\partial x_j} \Delta U \right)^{n+1} - n \Delta \Gamma \right] \\
&+ \Delta \sum_{j=1}^{N} \left[ D_f \left( \frac{\partial}{\partial x_j} \Delta U \right)^{n+1} - n \Delta \Gamma \right]_{ij} = R_i
\end{align*}
\]

where known terms on the right-hand side are:

\[
R_i = -\Delta \sum_{j=1}^{N} \left( Q^n - U^n v_m \cdot n \right) \Delta \Omega_{ij}^3 - U_i \Delta \left[ \sum_{j=1}^{N} \Delta \Omega_{ij}^3 \right]^{n+1}
\]

In Eqs. (18) and (19), interface variables can be approximated by using the centroid value of adjacent cells. Because interface values are defined at the median of line \( I - I(J) \), as shown in Fig. 1, interface variables can be easily determined by the average between two connecting cell centroid values. On the other hand, the gradients of interface variables can be approximated by either the technique called least-square gradient reconstruction (LSGR) method [21] or by the Green–Gauss method [22] as follows:

\[
\left( \frac{\partial}{\partial x_j} \Phi \right)_j = \frac{1}{V} \int n \cdot \Phi \, dS
\]

where, in two-dimensional space, \( V \) is a small area where the interface value \( \Phi \) resides, and \( S \) is the boundary or perimeter of such an area. In this study, we adopted the Green–Gauss method to approximate the gradient of interface \( \Delta U \) [third term in Eq. (18)] because it is much simpler than the LSGR method that requires matrix manipulation. For example, in Fig. 1, the gradient of interface \( \Delta U \) at the median of line \( I - I(J) \) (i.e., \( \Delta U/J \)) is determined by taking \( V \) as the combined area of two adjacent elements \( e_j \) and \( e_{j+1} \), and \( \Phi \) is the average value of \( \Delta U \) at each boundary \( dS \), namely line \( I - I(J - 1), I(J - 1) - I(J), I(J) - I(J + 1), \) and \( I(J + 1) - I \).

Similarly, the interface gradient of the velocity as well as the internal energy in the viscous flux \( G_i \) are interpolated using the Green–Gauss method. On the other hand, the inviscid flux \( F_i \) is approximated as the numerical flux by the high-order monotonic upstream-centered scheme for conservation laws (MUSCL) scheme for the unstructured grid [23], as shown in Eq. (21), instead of taking an average of analytical flux between two cell centroids, which is only first-order accurate:

\[
(F - U v_m) \cdot n = \frac{1}{2} (F(U_R) + F(U_L) - (U_R + U_L) v_m) \cdot n
\]

\[
- \frac{1}{2} A \left( U_R, U_L, n \right) - I \left( v_m \cdot n \right) [(U_R - U_L) \cdot n]
\]

In Eq. (21), \( A \) is the convection Jacobian matrix evaluated based on Roe’s average properties, and \( I \) is the identity matrix. Interface gradients of fluxes in \( Q \) [i.e., \( \partial / \partial x_j (F_i + G_i) \) in Eq. (11)] can be approximated by averaging the gradients of fluxes in cell \( I \) and its adjacent cell \( I(J) \), as shown in Fig. 1, where gradients in each cell are determined by the Green–Gauss method using the \( F_i \) and \( G_i \) at the cell interfaces. Besides conservation laws of mass, momentum, and energy, the geometric conservation law (GCL) [18] is required for ALE-type numerical schemes. To make the formulation GCL-compliant, the cell volumes in Eqs. (18) and (19) are evaluated by its boundary velocity as follows:

\[
\Delta \left[ \sum_{j=1}^{N} \Delta \Omega_{ij}^3 \right]^{n+1} = \Delta \left[ \sum_{j=1}^{N} \sum_{k=1}^{N} (v_m \cdot n) \Delta \Omega_{ij}^3 \right]^{n+1/2}
\]

Finally, the resulting equation written in compact form is:

\[
\begin{align*}
\Delta \left[ \sum_{j=1}^{N} \Delta \Omega_{ij}^3 \right]^{n+1} &- \Delta \left[ \sum_{j=1}^{N} \sum_{k=1}^{N} (v_m \cdot n) \Delta \Omega_{ij}^3 \right]^{n+1/2} = R_i
\end{align*}
\]
Table 1: Velocities maximum error norm $L_{\infty}$ ($u_i$)

<table>
<thead>
<tr>
<th>Time</th>
<th>$L_{\infty}(u_1)$</th>
<th>$L_{\infty}(u_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5T</td>
<td>1.33E-15</td>
<td>1.94E-15</td>
</tr>
<tr>
<td>T</td>
<td>1.77E-15</td>
<td>1.44E-15</td>
</tr>
</tbody>
</table>

$$K_i \Delta U_i^{n+1} + K_{i(1)} \Delta U_{i(1)}^{n+1} + \cdots + K_{i(N)} \Delta U_{i(N)}^{n+1} = R_i$$

(23)

where $K_i$ and $K_{i(j)}$ denote the collective sum of contributions for the main cell and its surrounding cells, respectively. As far as the unstructured mesh is concerned, combining Eq. (23) for all cells in the computational domain will result in a linear system of equations with large sparse matrix, namely $[K]$:

$$[K][\Delta U^{n+1}] = [R]$$

(24)

To solve the linear system of Eq. (24), we used an iterative method: the restart general minimal residual (restart GMRES) algorithm [24]. In addition, restart GMRES is combined with the left-block Gauss-Seidel preconditioner [25] to accelerate the convergence of solutions. Although other preconditioners that converge faster exist, the Gauss-Seidel preconditioner was chosen because it is relatively simple.

**III. Results and Discussions**

A. Freestream Preservation

To verify whether the ALE–FDV method satisfied the GCL, we conducted a freestream preservation test on a mesh that deformed like a wave, as depicted in Fig. 2. The mesh is a square domain with a size of $1 \times 1$ unit squared, and as shown in Fig. 2a, the mesh has initially 512 uniform triangular elements. The motion of the mesh is defined as [26]

$$x_i^{n+1} = x_i^n + \Delta x_i$$

$$\Delta x_i(t) = A_i L_i \left( \frac{\Delta t}{T} \right) \sin \left( \frac{c_1}{L_1} \right) \sin \left( \frac{c_2}{L_2} \right) \sin \left( \frac{c_3}{T} \right)$$

(25)

with $i = 1, 2$ for two-dimensional cases. The amplitude $A_i$ is set as 0.1; the reference length $L_i$ and time $T$ are set as 1.0 and 20, respectively; constants $c_1$ and $c_2$ are 4$\pi$; and constant $c_3$ is $\pi$. The test has been performed using a time interval $\Delta t = 0.01$. Inviscid uniform flow ($u_1 = 1.0, u_2 = 0$) was set as an initial and boundary condition. The solution was advanced in time until $t = T$, where the mesh is at its maximum distortion, as depicted in Fig. 2c.

Results of maximum error norm $L_{\infty}$ of velocities $u_1$ and $u_2$ at time instant $t = 0.5T$ and $t = T$ are reported in Table 1. 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, freestream preservation is achieved, and the ALE-FDV method is verified to be GCL-compliant.

B. Propagating Isentropic Vortex

The accuracy of the proposed ALE-FDV method for a deformable mesh is verified by the following, propagating isentropic vortex in two-dimensional inviscid flow. Initially, the flows are perturbed by an isentropic vortex centered at $(0, 0)$. The perturbed values of velocity, density, and pressure are given [26,27] by

\begin{align*}
  u_r &= U_{\text{max}} \frac{r}{b} \exp \left[ \frac{1}{2} \left( 1 - \frac{r^2}{b^2} \right) \right] \\
  \rho_r &= \left[ 1 - 1/2 (r - 1) U_{\text{max}}^2 \exp \left( 1 - \frac{r^2}{b^2} \right) \right]^{1/(r-1)} \\
  \rho_r &= \frac{1}{r \rho^*} \tag{26}
\end{align*}

where $r$ is the radial distance of any point in the computational domain from the vortex center at time $t$, and $U_{\text{max}}$ is the maximum velocity at distance $r = b$. The radial distance $r$ is defined as

\begin{align*}
  u_r &= U_{\text{max}} \frac{r}{b} \exp \left[ \frac{1}{2} \left( 1 - \frac{r^2}{b^2} \right) \right] \\
  \rho_r &= \left[ 1 - 1/2 (r - 1) U_{\text{max}}^2 \exp \left( 1 - \frac{r^2}{b^2} \right) \right]^{1/(r-1)} \\
  \rho_r &= \frac{1}{r \rho^*} \tag{26}
\end{align*}

Fig. 6 Pressure contour at several angles of attack (left figures: present, right figures: Murman et al. [28]).
\[ r = \sqrt{(x_1 - U_0 t)^2 + (x_2 - V_0 t)^2} \]  

(27)

The exact solutions are then given by

\[
\begin{bmatrix}
\rho \\
u_1 \\
u_2 \\
p
\end{bmatrix}
= \begin{bmatrix}
0 \\
U_0 \\
V_0 \\
0
\end{bmatrix}
+ \begin{bmatrix}
\rho_r \\
-u_r \sin \theta \\
u_r \cos \theta \\
p_r
\end{bmatrix}
\]

(28)

i.e., the vortex will retain its shape while convecting in the domain with freestream velocity \((U_0, V_0)\) through time. The initial condition is given by \(U_0 = 0.5, V_0 = 0.0, U_{\text{max}} = 0.5U_0\), and \(b = 0.2\). The computational domain is set as \(4 \times 4\) unit squared, centered at \((0, 0)\).

Two kinds of mesh are employed in this case, namely uniform quadrilateral and uniform triangular mesh, to demonstrate the applicability of the proposed method for unstructured mesh. Both kinds of mesh are deformed in time by the motion defined in Eq. (25). In this case, the amplitude \(A_p\) is set as 0.2; the reference length \(L_1\) and time \(T\) are set as 4.0 and 0.1, respectively; constants \(c_1\) and \(c_2\) are \(2\pi\); and the constant \(c_3\) is \(\pi\). In addition, similar to [26], a very small time interval is chosen \((\Delta t = 5 \times 10^{-5})\) to eliminate the effect of time integration error on the accuracy.

The accuracy of the proposed method is analyzed at time instant \(t = 0.1\) because the mesh has the largest deformation at this time regardless of the type of the mesh. Numerical results of pressure coefficient \(c_p\) for both meshes are in good agreement with the exact solution, as shown in Fig. 3. On the other hand, as shown in Fig. 4, the vortex as depicted by the density contour for the triangular mesh (Fig. 4b) qualitatively preserves its shape better than the vortex in a quadrilateral mesh (Fig. 4a). However, both meshes do not preserve the shape of the vortex exactly as predicted by the exact solution (i.e., a circular shape) due to the dissipation error of the second-order MUSCL scheme used in the proposed method. Furthermore, the numerical error of the density distribution in terms of average squared error norm \(L_2\) and maximum error norm \(L_{\infty}\) for three different grid resolutions with the number of nodes \(N = 1600, 6400,\) and 14,400 are plotted in Fig. 5. As a comparison, the errors for a stationary mesh are plotted in the same figure. Overall, slope of the errors for each case is close to –2, implying that the method is second-order accurate in space. In particular, the errors produced by the quadrilateral and triangular meshes are the same, but the errors due to the motion of the mesh are slightly higher than the stationary mesh.

### C. Oscillating NACA 0012 Airfoil

This case was first investigated experimentally by Landon [28] and since then has been used by many researchers as a benchmark problem to validate numerical solutions of moving bodies inside inviscid compressible flows. In this case a NACA 0012 airfoil which oscillates harmonically about its quarter chord in a flow with free stream Mach number \(M = 0.75\) is considered. The oscillating motion is defined by the angle-of-attack function:

---

**Table 2** Summary of unstructured meshes

<table>
<thead>
<tr>
<th>Name</th>
<th>Nodes</th>
<th>Smallest element size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>12,909</td>
<td>0.00048</td>
</tr>
<tr>
<td>Medium</td>
<td>18,570</td>
<td>0.00036</td>
</tr>
<tr>
<td>Fine</td>
<td>24,170</td>
<td>0.00025</td>
</tr>
</tbody>
</table>

---

**Fig. 7** Comparison of lift coefficient.

**Fig. 8** Comparison of lift coefficient \(c_l\) and drag coefficient \(c_d\).

**Fig. 9** Comparison of vorticity profile at 44 deg angle of attack.
\[ a(t) = 0.016 + 2.51 \sin(wt) \quad (29) \]

where the angular velocity is given by \( w = 2kU_\infty/c \), and the reduced frequency is \( k = 0.0814 \). The numerical simulation was conducted by using an unstructured mesh with 6700 cells. To properly capture the shock, the MUSCL scheme for the unstructured mesh combined with minmod limiters [23] has been used to approximate the inviscid flux. Figure 6 shows the instantaneous pressure contour around the airfoil at several angles of attack. The strength of the shock wave decreased, and its location shifted from the lower to the upper surface as the airfoil pitches up, qualitatively similar to the results shown by Murman et al. [29].

As shown in Fig. 7, the hysteresis of the lift coefficient obtained by the present method is in good agreement with the numerical solutions reported by Venkatakrishnan and Mavriplis [30] and Schneiders et al. [31] but lack agreement with the experimental results of Landon [28]. The discrepancy between numerical and experimental data is perhaps due to the viscous effect of the flow in the experiment. Mohaghegh and Jafarian [32] reported that the interaction of the boundary layer and the shock wave produced by the airfoil has reduced the strength of the shock compared with the inviscid simulation, resulting in differences between experimental data and inviscid numerical solutions.

D. Rapidly Pitching NACA 0015 Airfoil

In this section, we consider a laminar viscous flow past a NACA 0015 airfoil that is pitched rapidly from 0 deg angle of attack at a constant pitch rate:

\[ \omega(t) = \omega_0 \left[ 1 - \exp \left( -\frac{4.6}{t_0} \right) \right] \quad (30) \]

about the axis located at the quarter-chord. The final pitch rate is set as \( \omega_0 = 0.6 \), and the time taken by the airfoil to reach 99% of \( \omega_0 \) is given by \( t_0 = 1.0 \). The freestream Mach number and Reynolds number are \( M = 0.2 \) and \( Re = 10,000 \), respectively. The numerical simulation was conducted using an unstructured mesh with 28,134 grid points. This case was first studied by Visbal and Shang [33] and later was used for validation of numerical solutions by Lomtev et al. [34] and Schneiders et al. [31]. Figure 8 shows the comparison of drag and lift coefficients obtained by the present method with others. The trends of

Fig. 10  \( \omega = 1.0 \) (left figures: present, right figures: Sen et al. [35]).
lift and drag coefficient from the present results match quite well with those available in the literature.

The vorticity profile at 44 deg angle of attack obtained by the proposed method is shown in Fig. 9a. As a comparison, Fig. 9b also shows the same vorticity results obtained by Schneiders et al. [31]. Both figures show a 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.

E. Rotating Cylinder

In this problem, a laminar viscous flow with $Re = 1000$ past a circular cylinder that rotates impulsively from rest is considered. Two angular speeds are used: $\omega = 1.0$ and $\omega = 3.0$. The unstructured mesh with 18,570 cells was used for both cases. In addition, for the purpose of the grid-independence study, two other meshes with different number of cells were generated from the original mesh. Information on all meshes is summarized in Table 2.

Fig. 11 $\omega = 3.0$ (left figures: present, right figures: Sen et al. [35]).

Fig. 12 Experimental results by Badr et al. [36] (left: $\omega = 1.0$, right: $\omega = 3.0$).
Comparisons of the flowfield obtained using the medium-sized cells with the results of Sen et al. [35] are shown in Fig. 10 for \( \omega = 1.0 \) and Fig. 11 for \( \omega = 3.0 \). Overall, both results agree qualitatively. However, one can see some minor differences, such as at \( t = 8.0 \) in Fig. 10, where the utmost right vortex is weaker, and at \( t = 1.5 \) in Fig. 11, where the location of a vortex near the cylinder surface is unmatched and the formation of a secondary vortex does not exist when compared to the results of Sen et al. [35]. Nevertheless, similar flowfield structures can be observed by comparing the current results (\( t = 3.5, \omega = 1.0 \) and \( t = 5.0, \omega = 3.0 \)) and the experimental results of Baer et al. [36], as shown in Fig. 12.

Next, we compare the results of the case of \( \omega = 1.0 \) with the results obtained using the coarse mesh and the fine mesh. As shown in Fig. 13, the coarse mesh and fine mesh gave a similar pattern of flowfield, with the results of the medium-sized mesh shown in Fig. 10b, indicating that all meshes gave consistent results. Figure 14 shows plots of the averaged lift coefficient \( c_l \) and the drag coefficient \( c_d \) versus the smallest element size of each mesh. The average \( c_l \) and \( c_d \) were determined based on the time period of \( t = 5.5 \) to \( t = 15.0 \). The plot shows that good linear grid convergence with small differences of \( c_l \) and \( c_d \) from all meshes is obtained. Therefore, we can conclude that the effect of the mesh on the results is small, and the solutions obtained by the medium mesh shown in Figs. 10 and 11 are grid-independent.

**IV. Conclusions**

The capability of the ALE-FDV method in solving two-dimensional moving-boundary problems has been presented in this paper. The FDV method is formulated in ALE form for moving mesh applications, motivated by an interest to extend its capability to complex fluid-structure interaction, and the formulation is discretized using vertex-centered finite-volume method. Two cases of prescribed boundary motion in inviscid flow and another two cases in viscous flow have been chosen as benchmark problems to validate the ALE-FDV method. The results obtained are in good agreement with other numerical/experimental methods, thus confirming its applicability in solving moving-boundary problems.

**References**


doi:10.1016/j.jcp.2011.06.026

doi:10.1016/j.compfluid.2012.11.004

doi:10.1016/j.cam.2013.03.028

doi:10.1016/j.matcom.2009.01.020


doi:10.2514/3.61273

doi:10.1016/S0045-7825(00)00173-0


doi:10.2514/1.44940

doi:10.1016/S0017-9310(02)00330-7

doi:10.1137/0907058

doi:10.1137/070692108

doi:10.1016/j.compfluid.2010.11.015

doi:10.1016/j.jcp.2008.01.010


doi:10.1006/jcph.1996.0182

doi:10.1016/j.jcp.2012.09.038


doi:10.2514/3.10219

doi:10.1006/jcph.1999.6331

doi:10.1016/j.compfluid.2013.05.016

doi:10.1017/S0022112090003342

Z. J. Wang
Associate Editor