Evaluation of geometric conservation law using pressure-based fluid solver and moving grid technique

Ramji Kamakoti and Wei Shyy
Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, USA

Received February 2003
Revised September 2003
Accepted December 2003

Keywords *Elastic properties (fluids), Differential equations, Finite volume methods, Flow*

Abstract *The geometric conservation law (GCL) is an important concept for moving grid techniques because it directly regulates the treatments of the fluid flow and grid movement. With the grid movement at every time instant, the Jacobian, associated with the volume of each element in curvilinear co-ordinates, needs to be updated in a conservative manner. In this study, alternative GCL schemes for evaluating the Jacobian have been investigated in the context of a pressure-based Navier-Stokes solver, utilizing moving grid and the first-order implicit time stepping procedure as well as the PISO scheme. GCL-based on first and second-order, implicit as well as time-averaged, time integration schemes were considered. Accuracy and conservative properties were tested on steady-state, laminar flow inside a 2D channel and time dependent, turbulent flow around a 3D elastic wing; both treated with moving grid techniques. It seems that the formal order of accuracy is not a decisive indicator. Instead, the speed of grid movement and the interplay between the flow solver and the GCL treatments make a more noticeable impact.*

Nomenclature

a, b	= parameters influencing grid movement	Δt	= time step ($t^{n+1} - t^n$)
		Δx	= artificial grid movement of nodes
i	= ith grid point		
J	= Jacobian of the inverse transformation	ξ, η, ζ	= curvilinear co-ordinate directions
		ρ	= density of fluid
S	= surface area of element		
t	= time		
V	= volume of element	*Subscripts and superscripts*	
v	= velocity of fluid flow	t	= derivative with respect to time
$\mathbf{w}_s$	= local velocity of cell boundary	ξ	= derivative in this direction
x, y, z	= Cartesian co-ordinate directions	η	= derivative in this direction
$\dot{x}, \dot{y}, \dot{z}$	= grid velocity in the respective directions	ζ	= derivative in this direction
		n	= nth time step

1. Introduction

In many fluid flow and heat/mass transfer problems, situations arise when one or more boundaries move with respect to time. There are alternative techniques to handle such boundary movements, including Lagrangian, Eulerian and combined approaches (Shyy *et al.*, 1996, 2001). The Lagrangian approach often devise grid to move in time

This work has been partially supported by NASA, under the URETI project, and US Air Force.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 14 No. 7, 2004
pp. 851-865

0961-5539
DOI 10.1108/09615530410546254

and to conform to the change in boundary geometry. In such problems, we need to take care of certain geometric quantities involving the new mesh position and velocity. This can be taken care of by the geometric conservation law (GCL) as coined by Thomas and Lombard (1979). In the numerical perspective, a discrete version of GCL (DGCL) will be implemented.

The DGCL states that the computation of the geometric quantities associated with a moving grid should be computed in such a way that, independent of the mesh movement, the numerical scheme used for integrating the flow equations must preserve a uniform flow field (Guillard and Farhat, 2000). This is in conjunction with the fact that preserving uniform field implies first-order accuracy. In addition, Guillard and Farhat (2000) showed that for a p-order time-accurate scheme on a fixed mesh, satisfying the corresponding p-order DGCL is a sufficient condition for the scheme to be at least first-order time accurate on a moving mesh. They established the requirement that preserving the uniform flow field on moving grids is related to a consistency condition. It has also been proven that not satisfying the DGCL introduces a weak instability in the numerical solution on moving grids (Lesoinne and Farhat, 1996).

Substantial evidence exists showing that not satisfying the GCL leads to erroneous solutions or spurious oscillations in the solution (Farhat *et al.*, 2001, 2003; Guillard and Farhat, 2000; Lesoinne and Farhat, 1996). For example, Shyy *et al.* (1996) demonstrated that without explicitly enforcing GCL, O(1) error could be induced in the computation simply due to the grid movement effect. It has also been shown that satisfying the DGCL can improve the time-accuracy of computations on moving grids (Koobus and Farhat, 1999). One of the widely used methods for fluid-structure interaction problems is the arbitrary Lagrangian-Eulerian (ALE) formulation. It formulates the Navier-Stokes equations in three co-ordinate systems viz., material or Lagrangian (for structure motion), spatial or Eulerian (for fluid motion) and referential (for grid movement). Farhat *et al.* (2001, 2003) showed that for ALE schemes, satisfying the DGCL leads to a necessary and sufficient condition for the numerical scheme to preserve non-linear stability on a fixed grid. However, there have been a few cases where satisfying or not satisfying the GCL produced the same results (Morton *et al.*, 1998).

It should be noted that since GCL arises due to the numerical procedures devised based on grid movement, its implications are expected to be scheme dependent. Alternative forms of the GCL have been implemented over the years to study its impact on solution accuracy. Thomas and Lombard (1979) implemented the GCL for density-based finite difference schemes on structured meshes by updating the value of the Jacobian at each time step. Shyy *et al.* (1996, 2001) implemented the GCL along the lines of Thomas and Lombard for pressure-based finite volume schemes by updating the Jacobian values after every time step using a first-order backward Euler time-integration scheme. Lesoinne and Farhat (1996) developed a first-order, time accurate scheme preserving the GCL using the density-based ALE finite volume as well as finite element schemes on unstructured grids. Koobus and Farhat (1999) proposed a GCL scheme for second-order time-accurate density-based ALE finite volume schemes. Farhat *et al.* (2001) summarized six different time-integration schemes based on ALE formulation, some of them preserving the DGCL and some of them that do not, and showed the impact the different schemes have on solution accuracy. In this effort, we assess selected approaches for multi-block structured grids based on finite volume formulation and do a comparative study on these methods.

Most previously conducted studies employed the density-based fluid flow solver; in the present effort, the pressure-based fluid flow solver (Shyy, 1994; Shyy *et al.*, 1997; Thakur and Wright, 2002) is utilized. The implications of different implementation of GCL and the fluid flow solver are of main interest. Together with the previously cited references, the present work offers a more complete assessment of the GCL.

In the following, we first derive, from first principles, the equations for GCL and look at the numerical procedure for implementing it. Then, test problems will be devised to evaluate the performance of these alternative approaches directly.

2. Theory and numerical procedure

The GCL is derived from the conservation of mass by setting $\rho = 1$ and $v = 0$. It can be written as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \mathrm{d}V = \int_S \mathbf{w}_\mathrm{s}\cdot\mathrm{d}S \tag{1}$$

In words, it can be understood as the change in volume of each control volume between two time instants, t^n and t^{n+1}, must be equal to the volume swept by the cell boundary during that time $\Delta t = t^{n+1} - t^n$.

The above expression is referred to as the integral form of GCL. A differential statement of the GCL can be derived from the integral statement of GCL. Specifically, we first perform a transformation from the Cartesian co-ordinate system (x,y,z) to the body-fitted co-ordinate system (ξ, η, ζ), which leads to the following form of the integral statement:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V J\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta = \int_V (\nabla\cdot\mathbf{w}_\mathrm{s})J\,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{2}$$

Here, J represents the volume element in the transformed co-ordinate system hence each node is associated with a particular value of J. Therefore, the computed value of J must be consistent with the value of ΔV implied by the numerical scheme used for solving the flow equations. Earlier, arbitrary procedures were used to compute J, e.g. instantaneous mesh distribution at a given time instant was used to evaluate J at that particular time, which lead to an erroneous solution.

Expanding the right hand side of equation (2) and after performing necessary manipulations, we arrive at the following form of the differential statement of GCL.

$$J_t + (\xi_t)_\xi + (\eta_t)_\eta + (\zeta_t)_\zeta = 0 \tag{3}$$

where, ξ_t, η_t, ζ_t are the metric terms given by

$$\begin{aligned}
\xi_t &= -[\dot{x}(y_\eta z_\zeta - y_\zeta z_\eta) + \dot{y}(z_\eta x_\zeta - z_\zeta x_\eta) + \dot{z}(x_\eta y_\zeta - x_\zeta y_\eta)] \\
\eta_t &= -[\dot{x}(y_\zeta z_\xi - y_\xi z_\zeta) + \dot{y}(z_\zeta x_\xi - z_\xi x_\zeta) + \dot{z}(x_\zeta y_\xi - x_\xi y_\zeta)] \\
\zeta_t &= -[\dot{x}(y_\xi z_\eta - y_\eta z_\xi) + \dot{y}(z_\xi x_\eta - z_\eta x_\xi) + \dot{z}(x_\xi y_\eta - x_\eta y_\xi)]
\end{aligned} \tag{4}$$

Here, $\dot{x}, \dot{y}, \dot{z}$ are the grid velocities in the x-, y- and z-directions, respectively. Equation (3) is solved numerically to update the Jacobian values at each time step. The numerical

solution for equation (3) requires only an initial condition, which is obtained from the initial fixed grid and is given by

$$J = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\xi - x_\zeta y_\eta z_\xi - x_\eta y_\xi z_\zeta \tag{5}$$

We will now look at how to calculate the Jacobian values in a consistent fashion. In this regard, we will implement four different time-integration schemes for evaluating the Jacobian values, they being, a first-order implicit scheme, first-order time-averaged scheme, second-order implicit scheme and a second-order time-averaged scheme. The formulation for the different GCL schemes is as follows.

2.1 First-order implicit scheme

This scheme was initially suggested by Thomas and Lombard (1979) for density-based finite difference schemes and later was implemented using a pressure-based finite volume method by Shyy *et al.* (1996, 2001). The difference equation for the fully implicit scheme is given by

$$J^{n+1} = J^n - \Delta t\left[(\xi_t)_\xi^{n+1} + (\eta_t)_\eta^{n+1} + (\zeta_t)_\zeta^{n+1}\right] \tag{6}$$

where the metric coefficients are calculated from equation (4) at the $(n+1)$th time step and the grid velocities are calculated as follows

$$\dot{x}^{n+1} = \frac{x^{n+1} - x^n}{\Delta t},\ \ \dot{y}^{n+1} = \frac{y^{n+1} - y^n}{\Delta t},\ \ \dot{z}^{n+1} = \frac{z^{n+1} - z^n}{\Delta t} \tag{7}$$

2.2 First-order time-averaged scheme

It was suggested (Farhat *et al.*, 2001) that a time-averaged evaluation of the metric terms leads to a more consistent evaluation of the Jacobian especially when the time step is large. Hence, we reformulate the governing equation (3) by time-averaging the evaluation of metrics over more than one mesh configuration as opposed to evaluation of metrics on a single mesh configuration (equation (6)). It can be written as follows:

$$\frac{J^{n+1} - J^n}{\Delta t} + \left[(\xi_t)_\xi^{n+\frac{1}{2}} + (\eta_t)_\eta^{n+\frac{1}{2}} + (\zeta_t)_\zeta^{n+\frac{1}{2}}\right] = 0 \tag{8}$$

Here

$$\begin{aligned}
(\xi_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\xi_t)_\xi^{n+1} + (\xi_t)_\xi^n\right] \\
(\eta_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\eta_t)_\xi^{n+1} + (\eta_t)_\xi^n\right] \\
(\zeta_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\zeta_t)_\xi^{n+1} + (\zeta_t)_\xi^n\right]
\end{aligned} \tag{9}$$

As can be seen from the above expression that the metrics are evaluated at a mesh configuration, which is in between the mesh configurations at t^n and t^{n+1}. The grid velocities for the metric term is evaluated as follows:

$$\dot{x}^{n+\frac{1}{2}} = \frac{x^{n+1} - x^n}{\Delta t}, \quad \dot{y}^{n+\frac{1}{2}} = \frac{y^{n+1} - y^n}{\Delta t}, \quad \dot{z}^{n+\frac{1}{2}} = \frac{z^{n+1} - z^n}{\Delta t} \tag{10}$$

2.3 Second-order implicit scheme

The difference equation for a second-order implicit scheme (three-point backward), as suggested by Koobus and Farhat (1999) is used here. For structured meshes, it can be written as

$$\frac{\frac{3}{2}J^{n+1} - 2J^n + \frac{1}{2}J^{n-1}}{\Delta t} + \left[(\xi_t)_\xi^{n+1} + (\eta_t)_\eta^{n+1} + (\zeta_t)_\zeta^{n+1}\right] = 0 \tag{11}$$

Here, the metrics are evaluated in a similar fashion to the one used for first-order implicit scheme at the $(n+1)$th grid configuration. The grid velocity is also calculated in a similar fashion as given by equation (7).

2.4 Second-order time-averaged evaluation of Jacobian

Here we use a time-averaged procedure to evaluate the fluxes. Koobus and Farhat (1999) employed such a procedure for unstructured meshes and we extend such a scheme to a structured mesh here. The scheme can be written as follows:

$$\begin{aligned}\frac{\frac{3}{2}J^{n+1} - 2J^n + \frac{1}{2}J^{n-1}}{\Delta t} + \frac{1}{2}\left[(\xi_t)_\xi^{n+\frac{1}{2}} + (\eta_t)_\eta^{n+\frac{1}{2}} + (\zeta_t)_\zeta^{n+\frac{1}{2}}\right] \\ + \frac{1}{2}\left[(\xi_t)_\xi^{n-\frac{1}{2}} + (\eta_t)_\eta^{n-\frac{1}{2}} + (\zeta_t)_\zeta^{n-\frac{1}{2}}\right] = 0\end{aligned} \tag{12}$$

where

$$\begin{aligned}(\xi_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\xi_t)_\xi^{n+1} + (\xi_t)_\xi^n\right]; \quad (\xi_t)_\xi^{n-\frac{1}{2}} = \frac{1}{2}\left[(\xi_t)_\xi^n + (\xi_t)_\xi^{n-1}\right] \\ (\eta_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\eta_t)_\xi^{n+1} + (\eta_t)_\xi^n\right]; \quad (\eta_t)_\xi^{n-\frac{1}{2}} = \frac{1}{2}\left[(\eta_t)_\xi^n + (\eta_t)_\xi^{n-1}\right] \\ (\zeta_t)_\xi^{n+\frac{1}{2}} &= \frac{1}{2}\left[(\zeta_t)_\xi^{n+1} + (\zeta_t)_\xi^n\right]; \quad (\zeta_t)_\xi^{n-\frac{1}{2}} = \frac{1}{2}\left[(\zeta_t)_\xi^n + (\zeta_t)_\xi^{n-1}\right]\end{aligned} \tag{13}$$

Here, the averaging is done in between configurations $[t^n, t^{n+1}]$ and $[t^n, t^{n-1}]$ and the mesh velocities for each of these metric terms are evaluated as follows:

$$\begin{aligned}\dot{x}^{n+\frac{1}{2}} &= \frac{x^{n+1} - x^n}{\Delta t}, \quad \dot{y}^{n+\frac{1}{2}} = \frac{y^{n+1} - y^n}{\Delta t}, \quad \dot{z}^{n+\frac{1}{2}} = \frac{z^{n+1} - z^n}{\Delta t} \\ \dot{x}^{n-\frac{1}{2}} &= \frac{x^n - x^{n-1}}{\Delta t}, \quad \dot{y}^{n-\frac{1}{2}} = \frac{y^n - y^{n-1}}{\Delta t}, \quad \dot{z}^{n-\frac{1}{2}} = \frac{z^n - z^{n-1}}{\Delta t}\end{aligned} \tag{14}$$

As can be seen from the above equations that three different mesh configurations were used to arrive at this scheme. Having formulated the various schemes to implement GCL (calculation of Jacobians in our case), we now demonstrate the schemes by employing

two different mesh configurations: a 2D channel flow bounded by fixed, rigid solid boundaries, and flow over a 3D elastic wing. For all cases, the flow solver is based on the second-order spatial treatments of all terms, second-order upwind for convection, and central differences for pressure and viscous terms. Two different time marching schemes will be used to test the GCL for the 2D channel flow case, they being, the fully implicit scheme and the pressure implicit splitting of operators (PISO) scheme (Shyy, 1994; Thakur and Wright, 2002). The first-order fully implicit scheme for flow solver along with different approaches for GCL formulation will be used for testing the 3D case.

The channel flow is a fixed-geometry, steady-state problem. It was solved using artificially generated moving grid in a time dependent manner, with the aim of examining the error induced by the moving grid along with GCL implementations. The aeroelastic wing problem does involve a coupled fluid and structure interaction, resulting in time dependency in both wing shape and fluid flow. More details will be offered when describing the individual problem.

3. Numerical results

3.1 Two-dimensional channel flow: first-order backward Euler flow solver

We first demonstrate the above-mentioned methods for a 2D channel flow with fixed dimensions of 15×1. We do an incompressible, laminar flow calculation with a Reynolds number of 100 based on inlet length. The original grid for the channel is a uniform grid with 151 points in the x-direction and 11 points in the y-direction as shown in Figure 1(a). We perform unsteady flow calculations by moving the grid

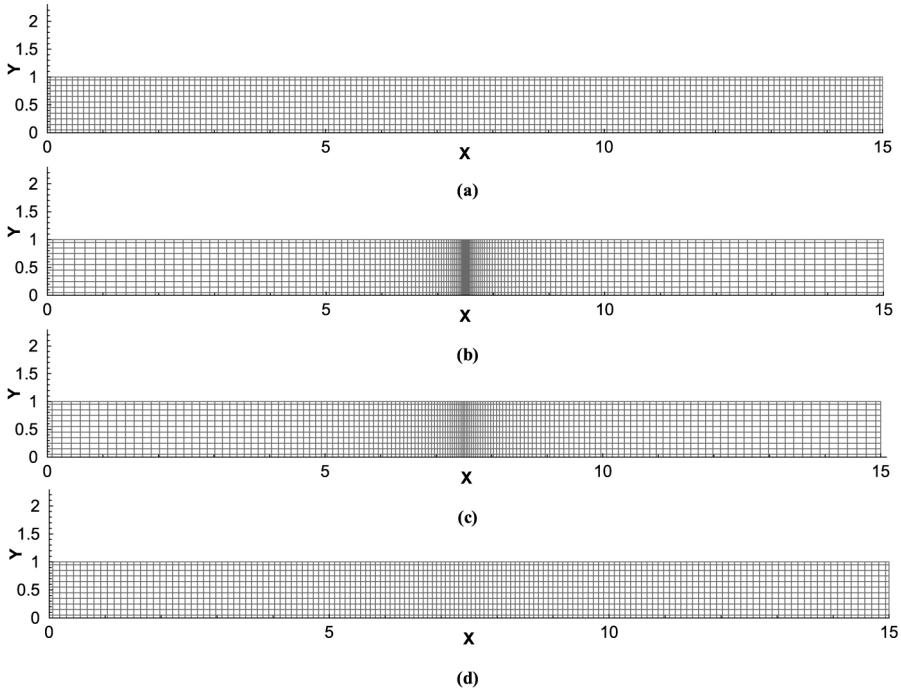

Figure 1.
(a) Original grid for the channel, (b)-(d) Grid distribution at different time instants

towards the center with varying stiffness after each time step. We ensure convergence of fully developed solution at every time step. In other words, steady-state is reached at every time step in spite of the artificial grid movement. The grid is moved in the following manner.

$$\Delta x(i) = a(t) + (i - 1)b(t)$$

where $\Delta x(i)$ represents the grid movement for the ith node; $a(t)$ and $b(t)$ are time varying quantities representing the spacing between the first two nodes and the spacing between the last two nodes for the first half of the grid. Similar procedure is followed for the second half of the grid. Such a gird movement produces different grid velocities at each time step. The grid snapshot at different time steps is shown in Figure 1(b)-(d).

The velocity profile at different time instants for each of the schemes is shown in Figure 2. The steady-state velocity profile is shown as a solid line and as can be seen from the figure, deviation from steady-state solution is observed for higher order schemes as well as the first-order time-averaged scheme. The first-order fully implicit

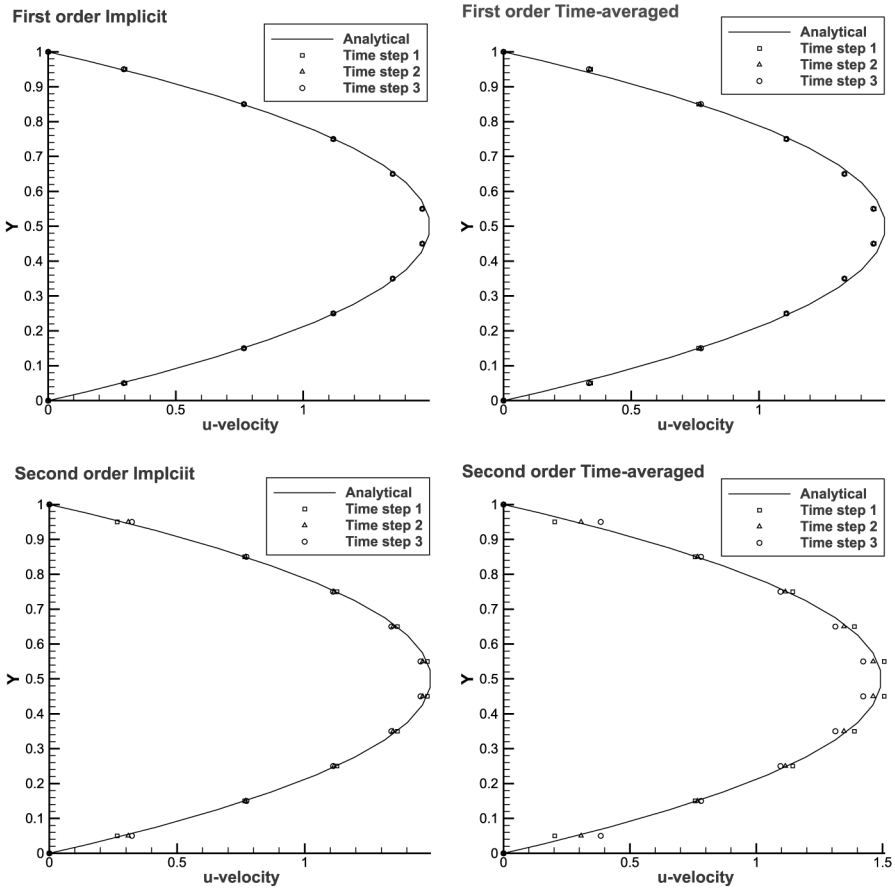

Figure 2.
Velocity profile for channel flow with Re = 10 at different time instants for coarse grid (151 × 11) using backward Euler method

scheme was found to produce the most accurate solution among all GCL schemes considered. We also computed the error norm by assuming the steady-state solution for channel flow as the exact solution and plot the error norm versus grid velocity for each of the four schemes. This is shown in Figure 3. As can be seen from the plot, the error norm is within comparable ranges for all cases except for the second-order time-averaged scheme. This is consistent with the velocity profile plot shown in Figure 2. It can clearly be seen that a first-order implicit scheme produces the least error norm compared to all other schemes. A grid refinement study was done on the same test case by doubling the number of grid points in both directions. A similar grid movement pattern was used here. The plot of velocity profiles is shown in Figure 4. The error norm versus grid velocity is shown in Figure 5 and as can be seen from the figure, the pattern is very similar as compared to the coarse grid case, however, the magnitude of error is less for fine grid case than coarse grid case as expected. Thus, we can infer that there is not much of an advantage of using a higher order GCL scheme as compared to a first-order implicit scheme. This observation can be understood with the view that GCL could be related to Guillard and Farhat's (2000) statement that for a p-order time-accurate scheme on a fixed mesh, satisfying the p-order accurate DGCL is a sufficient condition for the scheme to be first-order time accurate on a moving mesh. The above statement coupled with the fact that our flow solver is first-order implicit in time leads to the observation that the first order GCL scheme performs the best among the schemes considered.

3.2 Two-dimensional channel flow: PISO algorithm

The main reason for testing GCL for the PISO algorithm is due to the non-iterative nature of the PISO algorithm, which makes it much faster than the fully implicit Euler method. However, there is a limitation in the choice of time step since the PISO algorithm is a semi-implicit algorithm, hence confined to smaller CFL numbers. The test case for the channel flow case using the PISO flow solver is identical to the Euler

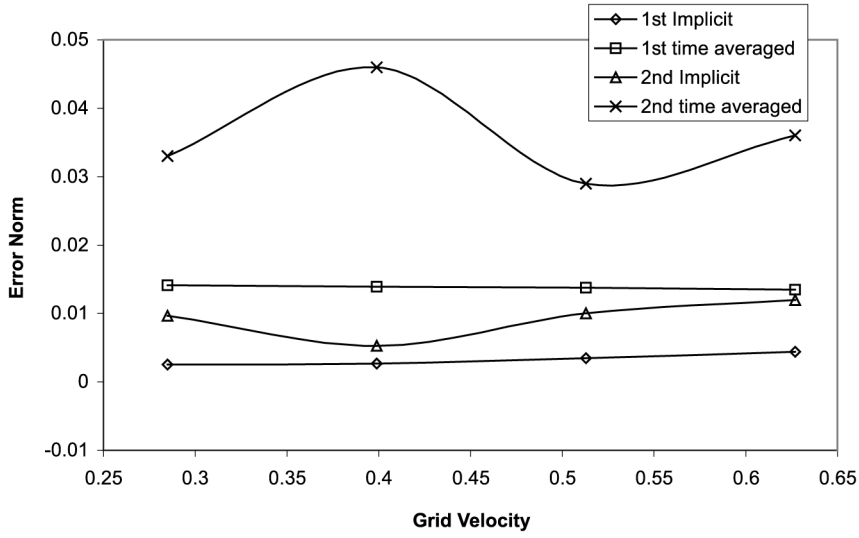

Figure 3.
Error norm vs grid velocity using various schemes for channel flow for 151×11 grid using Backward Euler method

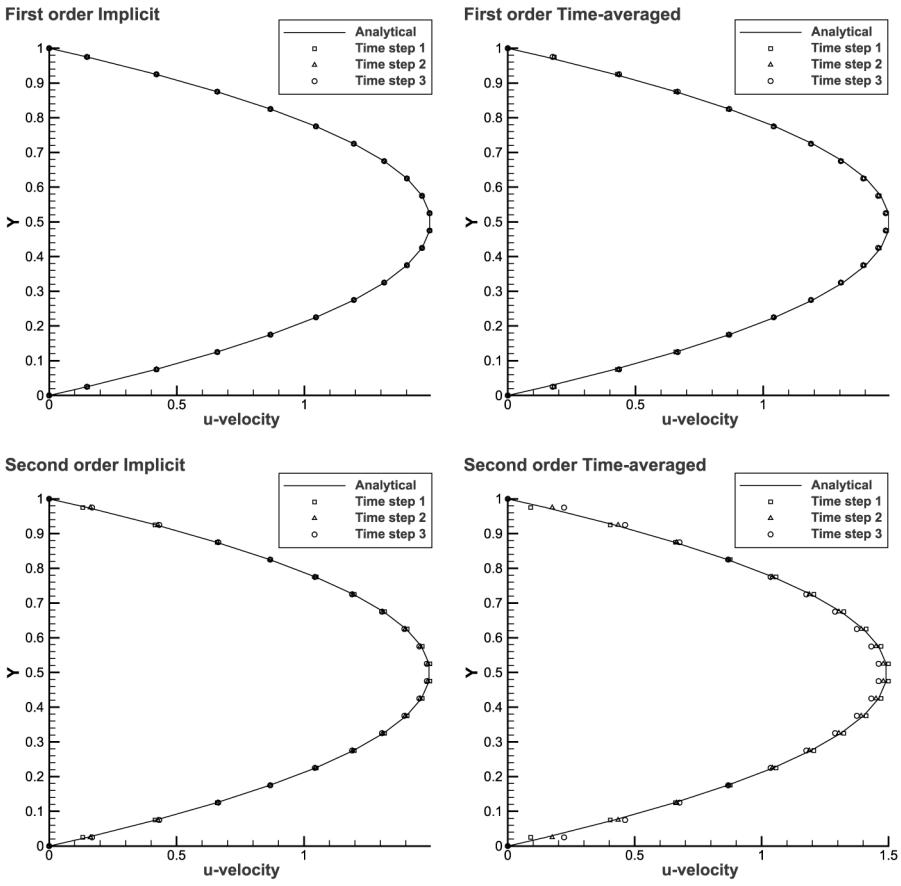

Figure 4.
Velocity profile for channel flow at different time instants for fine grid (301×21) case using backward Euler method

case except for the magnitude of grid movement since CFL number needs to be preserved. The CFL number is defined as

$$\mathrm{CFL} = \frac{U\Delta t}{\Delta x}.$$

As can be seen from the above expression, both time step size and grid spacing have an effect in deciding the CFL number. Too small a grid spacing will lead to undesirably high CFL numbers, hence the reason for smaller magnitude grid movement for the PISO algorithm test case. Other than the magnitude, the grid movement follows the identical pattern as mentioned for the Euler case. Similar velocity profile plot at different time instants for each of the GCL schemes is shown in Figure 6. The first-order fully implicit scheme was found to produce the most accurate solution among all GCL schemes considered even though it is not very clear from the plot. Similar error norms were calculated using the benchmark steady-state solution as the exact solution. The error norms for the four schemes are tabulated in Table I. From the table, it can be seen that first-order implicit scheme for GCL produces the least error

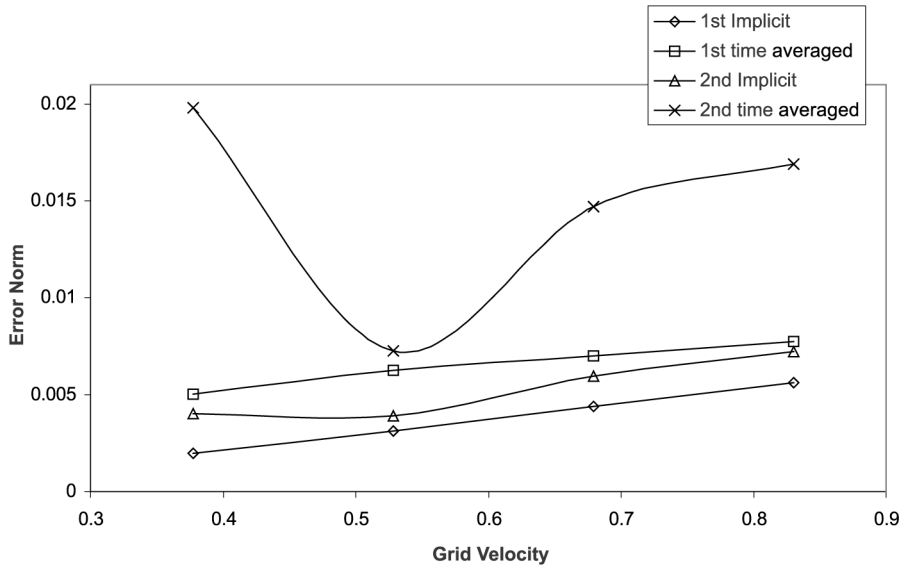

Figure 5.
Error norm vs grid velocity using various schemes for channel flow for 301×21 grid using backward Euler method

norm compared to other cases. This again shows that the first-order implicit scheme to evaluate the Jacobians performs best in the context of flow solver under consideration. We will now demonstrate a 3D example.

3.3 Three-dimensional elastic wing

As a next case study, we carry out a uniform flow test for a 3D case. We will consider flow over a 3D AGARD 445.6 wing to illustrate the effect of the various schemes. The AGARD 445.6 wing is a 3D elastic wing constructed with a NACA65A004 airfoil with a root cord of 1.833 ft, a semi-span of 2.5 ft and a taper ratio of 0.66. It is placed in the middle of a CFD domain with fixed dimensions of $10 \times 5 \times 5$. A schematic of the top view of the domain, along with the boundary conditions, is shown in Figure 7.

Similar to the channel flow case, two different examples are presented. First, we simply employed the wing geometry to define the mesh system. The fluid flow is considered to be uniform and the solid boundary is ignored. In this example, again, we were only interested in examining the impact of grid movement and the implementation of GCL on the numerical outcome of a trivial analytical solution. Second, we conducted a time dependent, turbulent flow computations around a solid, elastic wing.

For the first case, the mesh on the surface of the wing was arbitrarily moved in a direction normal to the surface with a linear variation in the spanwise direction with the root being fixed. A plot of the spanwise deflections at different time instants is shown in Figure 8. Uniform flow calculation was performed after moving the entire grid after each time instant and error norm calculated based on the trivial solution for pressure on the surface of the wing. All four schemes were tested and the corresponding error norm tabulated in Table II. A plot of these error norms versus grid velocity is shown in Figure 9. As can be seen from the figure, there is hardly any

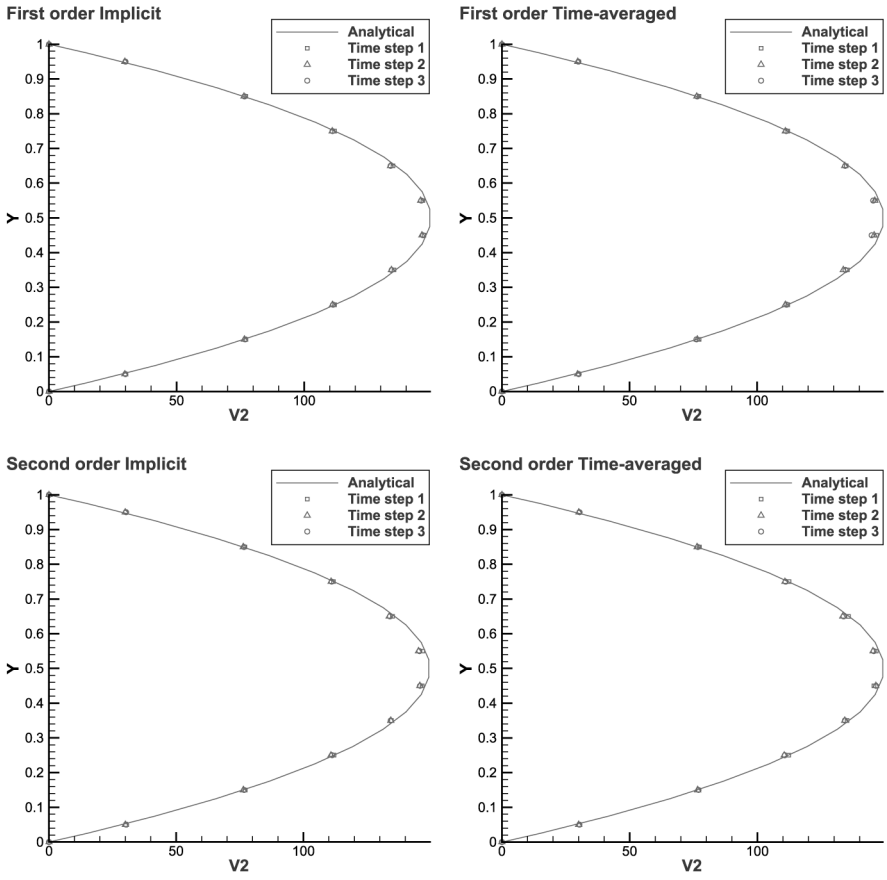

Figure 6.
Velocity profile for channel flow at different time instants for 151×11 grid using PISO method

Time step no.	First-order implicit	First-order time-averaged	Second-order implicit	Second-order time-averaged
1	0.0184	0.0183	0.0185	0.0184
2	0.0228	0.0227	0.0227	0.0231
3	0.0223	0.0224	0.0225	0.0228

Table I.
Error norm versus grid velocity for the four GCL schemes for 3D wing case using backward Euler method

difference between the four schemes implemented to preserve GCL. Although the plots look indistinguishable, it can be seen from the table that there are minute differences between each schemes, however, they are insignificant. This is in contrary to the results obtained by Farhat *et al.* (2001), where they showed that considerable differences were obtained between various schemes employed to preserve GCL. This could be due to the choice of first-order fully implicit fluid flow solver used to perform these computations. Apparently, depending on the fluid flow solver, different outcome can be realized in terms of the performance of the implementation of GCL.

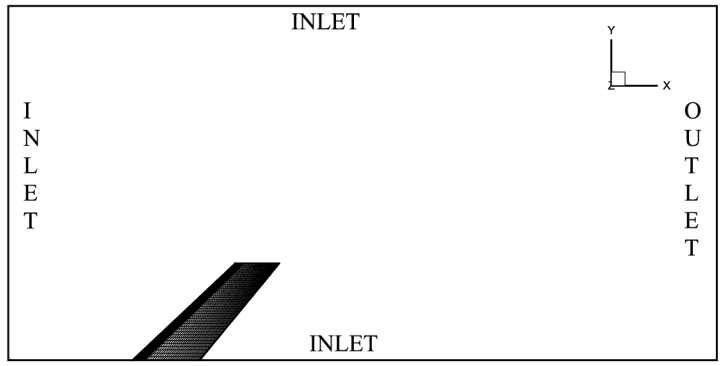

Figure 7.
Schematic of AGARD wing in CFD domain along with boundary conditions

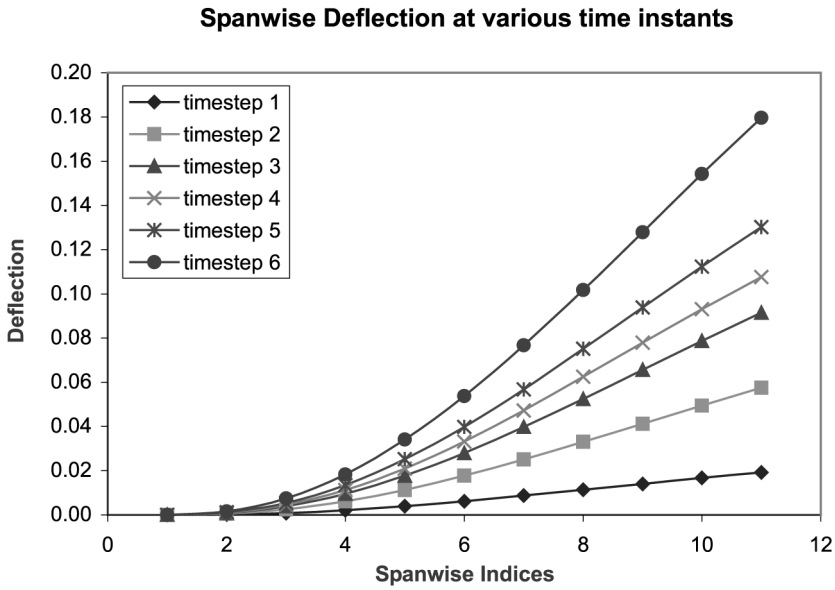

Figure 8.
Plot depicting the arbitrary movement of the wing in the spanwise direction

Table II.
Error norm versus grid velocity for the four GCL schemes for 3D wing case

	Error norm			
Grid velocity	First-order implicit	First-order time averaged	Second-order implicit	Second-order time averaged
0.016	2.583×10^{-5}	2.583×10^{-5}	2.584×10^{-5}	2.584×10^{-5}
0.023	1.475×10^{-4}	1.479×10^{-4}	1.474×10^{-4}	1.478×10^{-4}
0.034	1.991×10^{-4}	1.992×10^{-4}	1.992×10^{-4}	1.986×10^{-4}
0.038	1.907×10^{-4}	1.907×10^{-4}	1.907×10^{-4}	1.907×10^{-4}

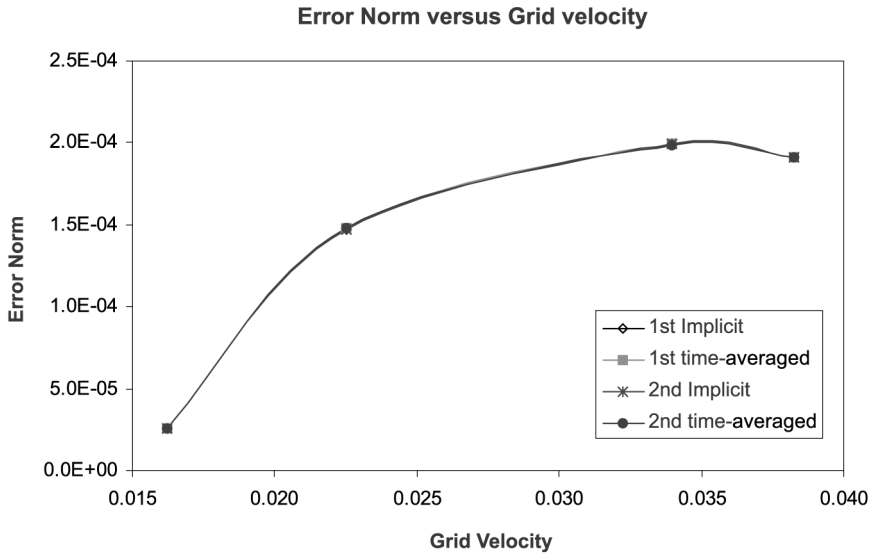

Figure 9.
Various schemes for AGARD wing using backward Euler method

It has been shown by Farhat *et al.* (2001, 2003) and Lesoinne and Farhat (1996) that the choice of GCL scheme does affect the accuracy of coupled fluid-structure solutions. Therefore, we have examined a 3D aeroelastic case with different GCL schemes. Time dependent, turbulent aeroelastic calculation based on the aforementioned AGARD wing geometry was adopted. The structure solver used here is a time-invariant model based on beam finite elements (Kamakoti *et al.*, 2002) with an explicit time integrator; hence, the structure solver was iterated several times to synchronize with the fully implicit flow solver. The grid movement is taken care of with the help of a combination of perturbation method and transfinite interpolation method (Lian *et al.*, 2003). Calculations were performed for a turbulent Reynolds number of 107,000/ft, based on root chord, with a 5° angle of attack. Details of the boundary and initial conditions, geometric information and the structural and fluid flow solvers can be found in the work of Kamakoti *et al.* (2002). The spanwise deflection at the first four time instants, for the first-order fully implicit GCL scheme, is shown in Figure 10. Since the results are very similar for the remaining schemes, they are not shown in the same plot, instead, the tip deflection at two different time instants is tabulated in Table III. As can be seen from the table, the values are virtually identical. This leads us to believe that the choice of GCL scheme does not really affect the aeroelastic solution as well in our case. Such an observation for the 3D wing case can be explained as follows. Comparing Figure 3 and Table II in the context of grid velocity, we can see that the grid velocity associated with the 3D wing movement is an order of magnitude less than that of the channel flow case. It has been proved by previous authors (Farhat *et al.*, 2001) that, for smaller grid velocity, the impact of GCL is minimal on solution accuracy. Hence, it seems clear that both the detailed schemes employed by the flow solver and the magnitude of grid movement directly influence the performance of the GCL scheme for moving boundary computations.

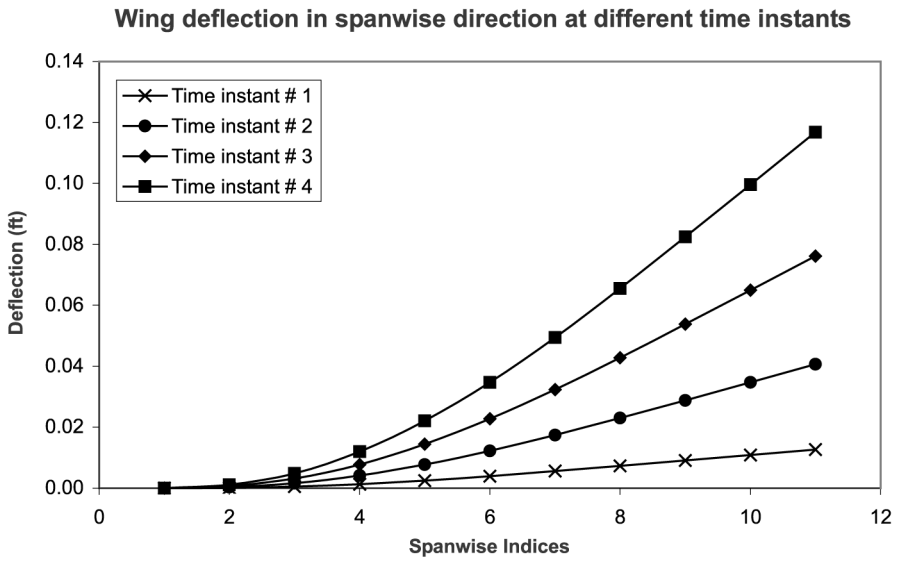

Figure 10.
Spanwise deflection of AGARD wing at four different time instants

Scheme	Time instant no. 2	Time instant no. 4
First implicit	7.613×10^{-2}	1.168×10^{-1}
First semi-implicit	7.616×10^{-2}	1.170×10^{-1}
Second implicit	7.621×10^{-2}	1.171×10^{-1}
Second semi-implicit	7.620×10^{-2}	1.171×10^{-1}

Table III.
Tip deflection at two different time instants for different GCL schemes for 3D wing case

4. Summary and conclusions

Four different schemes for updating Jacobian values that preserve the GCL were examined in an attempt to arrive at a robust choice for performing moving grid computations along with a pressure-based finite volume fluid flow solver. Both first- and second-order fully implicit as well as time-averaged evaluation of metrics were considered with the aid of finite volume method for CFD computations. Three test cases, two using the 2D channel flow one using the 3D elastic wing, with different Reynolds number, time dependency, and geometry movement were carried out to assess the performance of the different GCL schemes implemented. The first-order fully implicit method was found to produce the least error norm for the uniform flow test cases considered as well as the aeroelastic test case. The results have demonstrated that the impact of GCL on the solution accuracy is not simply governed by the formal order of accuracy of the discretization scheme. The results also show that the choice of fluid flow solver and magnitude of grid movement strongly affect the GCL scheme for performing aeroelastic computations. Since the first-order fully implicit GCL scheme simplifies the data structure in the code development effort, we conclude that this scheme is appropriate in the context of the fluid flow solver employed.

References

Farhat, C., Geuzaine, P. and Brown, G. (2003), "Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter", *Comput. Fluids*, Vol. 32, pp. 3-29.

Farhat, C., Geuzaine, P. and Grandmont, C. (2001), "The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids", *J. Comput. Phys.*, Vol. 174, pp. 669-94.

Guillard, H. and Farhat, C. (2000), "On the significance of the geometric conservation law for flow computations on moving meshes", *Comput. Methods Appl. Mech. Eng.*, Vol. 190, pp. 1467-82.

Koobus, B. and Farhat, C. (1999), "Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes", *Comput. Methods Appl. Mech. Eng.*, Vol. 170, pp. 103-29.

Kamakoti, R., Lian, Y., Regisford, S., Kurdila, A. and Shyy, W. (2002), "Computational aeroelasticity using a pressure-based solver", *Comput. Model. Eng. Sci.*, Vol. 3 No. 6, pp. 773-90.

Lesoinne, M. and Farhat, C. (1996), "Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations", *Comput. Methods Appl. Mech. Eng.*, Vol. 134, pp. 71-90.

Lian, Y., Steen, J., Trygg-Wilander, M. and Shyy, W. (2003), "Low Reynolds number turbulent flows around a dynamically shaped airfoil", *Comput. Fluids*, Vol. 32, pp. 287-303.

Morton, S.A., Melville, R.B. and Visbal, M.R. (1998), "Accuracy and coupling issues of aeroelastic Navier-Stokes solutions on deforming meshes", *J. Aircraft*, Vol. 35 No. 5, pp. 798-805.

Shyy, W. (1994), *Computational Modeling for Fluid Flow and Interfacial Transport*, Elsevier, Amsterdam.

Shyy, W., Udaykumar, H.S., Rao, M.M. and Smith, R.W. (1996), *Computational Fluid Dynamics with Moving Boundaries*, Taylor and Francis, Washington, DC.

Shyy, W., Francois, M., Udaykumar, H.S., N'dri, N. and Tran-Son-Tay, R. (2001), "Moving boundaries in micro-scale biofluid dynamics", *Appl. Mech. Rev.*, Vol. 54, pp. 419-53.

Shyy, W., Thakur, S.S., Ouyang, H., Liu, J. and Blosch, E. (1997), *Computational Techniques for Complex Transport Phenomena*, Cambridge University Press, New York, NY.

Thakur, S. and Wright, J. (2002), *STREAM: A Computational Fluid Dynamics and Heat Transfer Navier-Stokes Solver: Theory and Applications*, Streamline Numerics, Inc., Gainesville, FL.

Thomas, P.D. and Lombard, K. (1979), "Geometric conservation law and its application to flow computations on moving grids", *AIAA J.*, Vol. 17 No. 10, pp. 1030-7.