4. ANKARA INTERNATIONAL AEROSPACE CONFERENCE **AIAC-2007-017**
10-12 September, 2007 - METU, Ankara

NUMERICAL SOLUTION OF UNSTEADY COMPRESSIBLE FLOWS ON MOVING GRIDS USING DUAL TIME STEPPING

Murat Uygun*
Turkish Air Force Academy
Istanbul, Turkey

Kadir Kirkkopru†
Istanbul Technical University
Istanbul, Turkey

ABSTRACT

The formulation, implementation and validation of second-order accurate time stepping scheme for solving flow problems with moving boundaries on structured grids are presented. The time dependent equations are written for an Arbitrary Lagrangian-Eulerian (ALE) system and are solved with dual time stepping method. Perturbation method based on algebraic Transfinite Interpolation (TFI) is used for generation of new grid at each time step. ALE time-stepping scheme, which complies the Geometric Conservation Law (GCL) and retains the design time-accuracy, is obtained from the time-stepping scheme on non-moving grids by evaluating the cell volumes and the cell face velocities at locations in time as proposed by the dual time stepping method. Inviscid and viscous flows past NACA 0012 airfoil pitching about its quarter chord are computed as test cases. Computed results agree well with the experimental data.

INTRODUCTION

The fast and accurate computation of time accurate flows on moving grids is essential, since many flow problems such as flutter, store separation, control surface movement, deformation and burning of propellant in solid rocket motors, and flows in turbomachines involve unsteadiness and body motion. In such cases, a selection for kinematical description of continuum becomes important. Lagrangian description, where grid points follow the fluid particles in motion, results in the numerical algorithms that can easily track fluid-structure interfaces, but fails in case of large twist of computational domain. Eulerian description, where the computational grid is fixed in space and the fluid particles move with respect to the grid, results in numerical algorithms that can easily handle the large twist of computational domain, but fails in case of low resolution of interface [3]. In order to benefit from both Lagrangian and Eulerian algorithms, Arbitrary Lagrangian-Eulerian (ALE) method, which uses a grid with vertices that may be stationary, or be moving, was defined by Hirt et al. [36].

Time-integration in an ALE numerical scheme is simply carried out using a time-stepping scheme, which is developed for non-moving grids. The use of such an ALE time-stepping scheme does neither necessarily preserve the design time-accuracy established on non-moving grids nor satisfy the geometric conservation law (GCL). The GCL relates the change in volume of a cell to the motion of its faces [34]. GCL is a conservation law, which is necessary and sufficient for retaining on moving grids the nonlinear stability achieved on non-moving grids [8]. However, although satisfying the GCL is a sufficient condition for achieving first order time-accuracy [10], it is neither a necessary nor a sufficient condition for retaining on moving grids the design time-accuracy achieved on non-moving grids [5]. In this study, a GCL compliant ALE time-stepping scheme, which retains on moving grids the nonlinear stability and the second-order time-accuracy achieved on non-moving grids, is employed using the dual time stepping method (DTS) [24]. The cell face velocities and cell volumes are evaluated at the n+1 time level as proposed by the time-stepping scheme [1, 15].

Either explicit or implicit methods can be utilized for time integration. However, due to stability issues, explicit methods limit the time step to a value, which is much smaller than the one required for numerical accuracy. Hence, the use of implicit methods becomes practical for the computation of unsteady viscous flows since, they allow larger time steps. However, higher computational cost is required in the latter case for performing matrix operations at each time step. For the solution of time dependent Navier-Stokes equations, Jameson

* MSc. in Aero.Eng. Department, Email: m.uygun@hho.edu.tr
† Prof. in Mech.Eng. Department, Email: kirkkopruk@itu.edu.tr

described a dual time stepping method [24], where a modified steady problem with implicit discretization in physical time is solved by advancing in pseudo time with explicit time stepping scheme at each physical time step. The use of dual time stepping lets the physical time step be not limited by the stability issues and be selected based on the numerical accuracy requirement only. This is particularly useful in the computation of viscous flows, since the high aspect ratio cells in the computational grid limit the physical time step due to numerical stability. DTS was successfully implemented on rigid [24] and moving structured grids [20] for the computation of inviscid flows. DTS was also utilized in order to solve viscous flows on moving grids with turbulence closure [7, 12, 18].

In order to solve unsteady flow problems with moving boundaries, grid has to be regenerated at each time step, such that it fits the body motion. Hence, an efficient grid regeneration algorithm, which is computationally fast and cheap, is required in order to shorten the overall computation time. In this study, a perturbation method based on algebraic Transfinite Interpolation (TFI) is used for rapid generation of structured moving grids [6, 9, 11]. The grid points are modified using the displacements of corner points and block boundaries. The use of perturbation method guarantees that quality of the original grid is retained and allows each boundary to move independently.

In the current work, cell-centered finite volume discretization technique is adopted. Convective terms are evaluated using central differencing scheme [33]. The flux vectors at the midpoint of a cell face are computed by arithmetic averaging of flow variables at two neighboring cells. The flow solution is advanced at the local maximum speed by using local time stepping [17]. Larger time steps are allowed due to implicit residual smoothing [26, 28, 31]. Multigrid method based on Full Approximation Storage (FAS) scheme [27, 29, 32] is implemented in order to accelerate convergence of numerical solution. The variables, which are required for the computation of viscous terms, are also averaged at a cell face. Gradients at the midpoints of a cell face are computed by means of Green's theorem [21].

Details regarding the formulation, implementation and validation of second-order accurate time stepping scheme for solving flow problems with moving boundaries on structured grids are presented. Inviscid and viscous flows past NACA 0012 airfoil pitching about its quarter-chord are studied as test cases. A sensitivity study is carried out including the effects of both grid density and physical time step. Grid and time step independent results are presented only. Computed results agree well with the experimental data [13].

GOVERNING EQUATIONS

Application of Reynolds (or time) averaging (denoted by overbar) to density and pressure and of Favre (or mass) averaging (denoted by tilde) to the remaining flow variables in the compressible Navier-Stokes equations yields the so called Favre Averaged Navier-Stokes (FANS) equations [23]. FANS equations written in ALE form account for the relative motion of the grid with respect to the fluid. FANS equations are cast into integral form as

$$\iint_{\Omega(t)} \frac{\partial \vec{C}}{\partial t} d\Omega + \oint_{\partial\Omega(t)} \vec{F}\vec{n} dS = 0 \,, \tag{1}$$

where Ω denotes for the control volume surrounded by the control surface $\partial\Omega$. $\vec{n} = n_x\vec{i} + n_y\vec{j}$ is the outward vector normal to the control surface $\partial\Omega$. dS is the elementary surface area. $\vec{C}$ is the vector of conservative variables and $\vec{F} = \vec{F}_C - \vec{F}_V$ is the flux vector, which can be split into a convective part $\vec{F}_C$ and a viscous part $\vec{F}_V$:

$$\vec{C} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{E} \end{bmatrix} \qquad \vec{F}_C = \begin{bmatrix} \bar{\rho}\left(\vec{V} - \vec{V}_{cf}\right) \\ \bar{\rho}\tilde{u}\left(\vec{V} - \vec{V}_{cf}\right) + \bar{p}\vec{i} \\ \bar{\rho}\tilde{v}\left(\vec{V} - \vec{V}_{cf}\right) + \bar{p}\vec{j} \\ \bar{\rho}\tilde{H}\left(\vec{V} - \vec{V}_{cf}\right) + \bar{p}\vec{V}_{cf} \end{bmatrix} \qquad \vec{F}_V = \begin{bmatrix} 0 \\ \tilde{\tau}_{xx}\vec{i} + \tilde{\tau}_{xy}\vec{j} \\ \tilde{\tau}_{yx}\vec{i} + \tilde{\tau}_{yy}\vec{j} \\ \tilde{\Pi}_x\vec{i} + \tilde{\Pi}_y\vec{j} \end{bmatrix} \tag{2}$$

The variables $x-y$, t, $\bar{p}$, $\bar{\rho}$, $\tilde{T}$, $\tilde{u}-\tilde{v}$ are cartesian coordinates, time, pressure, density, temperature and velocity components, respectively. $\vec{i}-\vec{j}$ are unit vectors associated with the cartesian coordinates. $\tilde{E}$ and $\tilde{H}$ denote total energy and total enthalpy. $\vec{V} = \tilde{u}\vec{i} + \tilde{v}\vec{j}$ is the velocity vector and $\vec{V}_{cf}$ is the cell face velocity. The terms describing the work of viscous stresses and the heat conduction are defined as

$$\tilde{\Pi}_x = \tilde{u}\tilde{\tau}_{xx} + \tilde{v}\tilde{\tau}_{xy} + \bar{k}_L \frac{\partial \tilde{T}}{\partial x}$$
$$\tilde{\Pi}_y = \tilde{u}\tilde{\tau}_{yx} + \tilde{v}\tilde{\tau}_{yy} + \bar{k}_L \frac{\partial \tilde{T}}{\partial y} \quad (3)$$

where

$$\tilde{\tau}_{xx} = \frac{2}{3}\bar{\mu}_L \left(2\frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{v}}{\partial y} \right)$$
$$\tilde{\tau}_{yy} = \frac{2}{3}\bar{\mu}_L \left(2\frac{\partial \tilde{v}}{\partial y} - \frac{\partial \tilde{u}}{\partial x} \right) . \quad (4)$$
$$\tilde{\tau}_{xy} = \tilde{\tau}_{yx} = \bar{\mu}_L \left(\frac{\partial \tilde{u}}{\partial y} + \frac{\partial \tilde{v}}{\partial x} \right)$$

The viscous stresses and the vector of viscous fluxes do not involve the cell face velocity. That is, the grid motion does not change the way viscous fluxes are computed. In case of turbulent flows, the effective viscosity

$$\mu = \bar{\mu}_L + \mu_T \quad (5)$$

replaces the laminar viscosity and the effective thermal conductivity coefficient

$$k = \bar{k}_L + k_T = c_p \left(\frac{\bar{\mu}_L}{\mathrm{Pr}_L} + \frac{\mu_T}{\mathrm{Pr}_T} \right) \quad (6)$$

replaces the laminar thermal conductivity coefficient. c_p denotes the specific heat coefficient at constant pressure. Coefficient of laminar viscosity $\bar{\mu}_L$ is computed using the Sutherland formula:

$$\bar{\mu}_L = \frac{1.45\tilde{T}^{3/2}}{\tilde{T} + 110} 10^{-6} . \quad (7)$$

Laminar Prandtl number Pr_L can be taken as constant. Turbulent viscosity μ_T and turbulent Prandtl number Pr_T are provided by means of turbulence model [35]. Assuming air as an ideal gas, the equation of state is used to calculate the pressure and temperature;

$$\bar{p} = (\gamma - 1)\bar{\rho}\left[\tilde{E} - \frac{\tilde{u}^2 + \tilde{v}^2}{2} \right], \quad \tilde{T} = \frac{\bar{p}}{\bar{\rho} R} . \quad (8)$$

γ is the ratio of specific heats and R is the gas constant.

SPATIAL DISCRETIZATION

Integral form of FANS equations is solved using cell-centered finite volume method. Computational domain is divided into quadrilateral control volumes (figure 1). In the cell-centered scheme, flow quantities are associated with the center of a control volume (cell). Finite volume method requires the evaluation of the convective and viscous fluxes, which are assumed to be constant along the individual cell face. In this work, convective fluxes are evaluated by means of central differencing scheme [33] and the gradients, which are required in the computation of viscous fluxes, are evaluated using Green's theorem [21].

After writing eqn. (1) for all cells and employing method of lines, where spatial and temporal terms are discretized separately, a system of first order ODE is obtained. Approximating the integrals with the mean value theorem, eqn. (1) for a particular moving cell becomes

$$\frac{d}{dt}\left(\Omega_{I,J} \vec{C}_{I,J} \right) + \vec{R}_{I,J}\left(\vec{C} \right) = 0 , \quad (9)$$

where I and J address the particular cell. $\vec{R}_{I,J}\left(\vec{C} \right)$ is called the residual, which is defined as

$$\vec{R}_{I,J}\left(\vec{C} \right) = \sum_{ncf=1}^{4} \vec{F}_{ncf} \left(\vec{n} \Delta S \right)_{ncf} , \quad (10)$$

where ncf identifies four cell faces and ΔS denotes the area of the cell face.

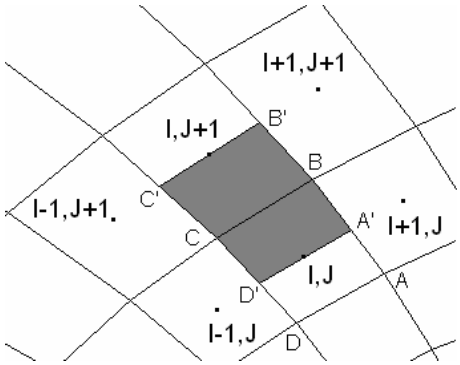

Figure 1: *Control volume (cell), control surface (cell face) and auxiliary control volume*

In regions, where viscous effects are negligible, physical diffusion is not adequate to prevent odd-even decoupling of the cell-centered schemes. In order to avoid odd-even decoupling of the solution and of the oscillations near shocks, stagnation points and boundary layer edges, finite volume scheme requires the addition of artificial dissipation term $\vec{D}$ to the discretized equations. Then, eqn. (10) is written as

$$\vec{R}_{I,J}\left(\vec{C}\right)=\sum_{ncf=1}^{4}\left[\vec{F}\vec{n}\Delta S-\vec{D}\right]_{ncf}. \tag{11}$$

The net flux through the faces of a particular cell is calculated as

$$\begin{aligned}\sum_{ncf=1}^{4}\vec{F}_{ncf}\left(\vec{n}\Delta S\right)_{ncf}&=\left(\vec{F}\vec{n}\Delta S\right)_{I+1/2,J}-\left(\vec{F}\vec{n}\Delta S\right)_{I-1/2,J}\\&+\left(\vec{F}\vec{n}\Delta S\right)_{I,J+1/2}-\left(\vec{F}\vec{n}\Delta S\right)_{I,J-1/2}\end{aligned}. \tag{12}$$

The total flux at the cell face AB (figure 1) is approximated by

$$\left(\vec{F}\vec{n}\Delta S\right)_{I+1/2,J}=\left[\vec{F}_C\left(\vec{C}_{I+1/2,J}\right)-\vec{F}_V\left(\vec{U}_{I+1/2,J}\right)\right]\left(\vec{n}\Delta S\right)_{I+1/2,J}, \tag{13}$$

where $\vec{C}_{I+1/2,J}=\left(\vec{C}_{I,J}+\vec{C}_{I+1,J}\right)/2$ and $\vec{U}_{I+1/2,J}=\left(\vec{U}_{I,J}+\vec{U}_{I+1,J}\right)/2$. (14)

$\vec{U}$ represents the flow variables $\tilde{u},\tilde{v},\overline{k},\overline{\mu}$, which are required for the computation of the viscous terms and of the stresses. Gradients at the midpoint of the cell face BC are evaluated using Green's theorem with the aid of an auxiliary control volume Ω^{BC}, which is defined by the curve $A'B'C'D'$ (figure 1). The derivative of temperature with respect to y coordinate is calculated as

$$\left(\frac{\partial\tilde{T}}{\partial y}\right)_{BC}=\frac{1}{\Omega^{BC}}\iint_{\Omega}\left(\frac{\partial\tilde{T}}{\partial y}\right)d\Omega=\frac{1}{\Omega^{BC}}\int_{\partial\Omega}\tilde{T}dS_y^{BC}\approx\frac{1}{\Omega^{BC}}\sum_{ncf=1}^{4}\left(\tilde{T}\Delta S_y^{BC}\right)_{ncf}, \tag{15}$$

where $\Omega^{BC}=\left(\Omega_{I,J}+\Omega_{I,J+1}\right)/2$. (16)

Temperature at cell faces is obtained as cell-centered values. The use of eqn. (15) yields a second order accurate scheme for smoothly stretched grids.

The net dissipation flux through the faces of a particular cell is calculated as

$$\sum_{ncf=1}^{4}\vec{D}=\vec{D}_{I+1/2,J}-\vec{D}_{I-1/2,J}+\vec{D}_{I,J+1/2}-\vec{D}_{I,J-1/2}. \tag{17}$$

The dissipation flux at the cell face AB (figure 1) is defined as [33]

$$\vec{D}_{I+1/2,J}=\alpha_{I+1/2,J}\left[\varepsilon^{(2)}_{I+1/2,J}\delta^{(1)}_I\vec{C}_{I,J}-\varepsilon^{(4)}_{I+1/2,J}\delta^{(3)}_I\vec{C}_{I,J}\right]. \tag{18}$$

where $\delta^{(1)}$ and $\delta^{(3)}$ are 1st and 3rd order difference operators, which are defined as

$$\begin{aligned}\delta^{(1)}_I\vec{C}_{I,J}&=\vec{C}_{I+1,J}-\vec{C}_{I,J}\\\delta^{(3)}_I\vec{C}_{I,J}&=\vec{C}_{I+2,J}-3\vec{C}_{I+1,J}+3\vec{C}_{I,J}-\vec{C}_{I-1,J}\end{aligned}. \tag{19}$$

α is a scaling factor, which is written for I direction as [19]

$$\alpha_{I+1/2,J}=\left[\left(\Lambda^I_C\right)_{I,J}+\left(\Lambda^I_C\right)_{I+1,J}\right]/2, \tag{20}$$

where

$$\Lambda_C^I = \left(\left| \vec{V}\vec{n} - \vec{V}_{cf}\vec{n} \right| + c \right) \Delta S \ . \tag{21}$$

Λ_C^I is the spectral radii of the convective flux Jacobian matrix. c is the local speed of sound. The coefficients are computed as

$$\begin{aligned} \varepsilon_{I+1/2,J}^{(2)} &= k^{(2)} \max\left(\nu_{I,J}, \nu_{I+1,J} \right) \\ \varepsilon_{I+1/2,J}^{(4)} &= \max\left[0, \left(k^{(4)} - \varepsilon_{I+1/2,J}^{(2)} \right) \right] \end{aligned} \tag{22}$$

with the pressure sensor given as

$$\nu_{I,J} = \left| \frac{\overline{p}_{I-1,J} - 2\,\overline{p}_{I,J} + \overline{p}_{I+1,J}}{\overline{p}_{I-1,J} + 2\,\overline{p}_{I,J} + \overline{p}_{I+1,J}} \right| . \tag{23}$$

Pressure sensors activate the second-difference dissipation in regions of large pressure gradients and switch it off in smooth regions of flow. $k^{(2)} \approx 1/2$ and $k^{(4)} \approx 1/96$ are typical values as dissipation coefficients. Dissipation flux is of first order in regions of large pressure gradients and is of third order in smooth regions of flow.

TIME INTEGRATION

Dual time stepping, where a modified steady problem with implicit discretization in physical time is solved by advancing in pseudo time with explicit time stepping scheme at each physical time step is utilized in this work [24]. Time derivative in eqn. (9) can be discretized as

$$\frac{(1+\phi)\left(\Omega_{I,J}^{n+1}\vec{C}_{I,J}^{n+1} - \Omega_{I,J}^{n}\vec{C}_{I,J}^{n} \right) - \phi\left(\Omega_{I,J}^{n}\vec{C}_{I,J}^{n} - \Omega_{I,J}^{n-1}\vec{C}_{I,J}^{n-1} \right)}{\Delta t} + \vec{R}_{I,J}\left(\vec{C}^{n+1} \right) = 0 \tag{24}$$

where the superscripts indicate physical time level. $\phi = 0$ results in 1st order time accurate difference formula and $\phi = 1/2$ results in 2nd order time accurate, three point backward difference formula (BDF).

In dual time stepping concept, pseudo-time derivative is appended to the eqn. (9):

$$\frac{d}{d\tau}\left(\Omega_{I,J}\vec{C}_{I,J} \right) + \frac{d}{dt}\left(\Omega_{I,J}\vec{C}_{I,J} \right) + \vec{R}_{I,J}\left(\vec{C} \right) = 0 \, . \tag{25}$$

Discretizing eqn. (25) with second order time accurate BDF $\left(\phi = 1/2\right)$ for the physical time derivative results in a new residual $\vec{R}^*$, which is called an unsteady residual;

$$\frac{d}{d\tau}\left(\Omega_{I,J}^{n+1}\vec{C}_{I,J}^{n+1} \right) + \vec{R}_{I,J}^{*}\left(\vec{C}^{n+1} \right) = 0 \, , \tag{26}$$

where

$$\vec{R}_{I,J}^{*}\left(\vec{C}^{n+1} \right) = \vec{R}_{I,J}\left(\vec{C}^{n+1} \right) + \frac{3\Omega_{I,J}^{n+1}\vec{C}_{I,J}^{n+1} - 4\Omega_{I,J}^{n}\vec{C}_{I,J}^{n} + \Omega_{I,J}^{n-1}\vec{C}_{I,J}^{n-1}}{2\Delta t} . \tag{27}$$

Eqn. (26) is fully implicit. Discretization of eqn. (26) with $\phi = 0$ for the pseudo-time derivative results in

$$\frac{\vec{C}_{I,J}^{k+1} - \vec{C}_{I,J}^{k}}{\Delta\tau_{I,J}} + \frac{1}{\Omega_{I,J}^{n+1}}\vec{R}_{I,J}^{*}\left(\vec{C}^{n+1} \right) = 0 \, , \tag{28}$$

where k is the iteration count in pseudo-time. The unsteady residual is driven to zero at each physical time step. Time stepping in eqn. (28) is done using M-stage Runge-Kutta (R-K) scheme with point-implicit calculation of the unsteady terms [4, 16,22]:

$$\begin{aligned} &\vec{C}_{I,J}^{(0)} = \vec{C}_{I,J}^{(k)} \\ &\vdots \\ &\vec{C}_{I,J}^{(m)}\left[1 + \alpha_m\beta\frac{3}{2}\frac{\Delta\tau_{I,J}}{\Delta t} \right] = \vec{C}_{I,J}^{(0)} - \alpha_m\frac{\Delta\tau_{I,J}}{\Omega_{I,J}^{n+1}}\vec{R}_{I,J}^{*}\left(\vec{C}^{(m-1)} \right) + \alpha_m\beta\frac{3}{2}\frac{\Delta\tau_{I,J}}{\Delta t}\vec{C}_{I,J}^{(m-1)} \qquad m = 1, 2, \ldots, M \\ &\vdots \\ &\vec{C}_{I,J}^{(k+1)} = \vec{C}_{I,J}^{(M)} \end{aligned} \tag{29}$$

where unsteady residual is

$$\vec{R}^{*}_{I,J}\left(\vec{C}^{(m-1)}\right)=\vec{R}_{I,J}\left(\vec{C}^{(m-1)}\right)+\frac{3}{2}\frac{\Omega^{n+1}_{I,J}}{\Delta t}\vec{C}^{(m-1)}_{I,J}-\vec{Q}_{I,J}\,, \tag{30}$$

with $\vec{Q}_{I,J}=\dfrac{1}{2\Delta t}\left(4\Omega^{n}_{I,J}\vec{C}^{n}_{I,J}-\Omega^{n-1}_{I,J}\vec{C}^{n-1}_{I,J}\right)$. (31)

$\vec{Q}_{I,J}$ contains the terms, which remain constant throughout the R-K iteration. β is a constant taken as $\beta \geq 2$ [15].

Table 1 presents optimized Runge-Kutta coefficients for maximum stability of a centrally discretized scheme. Convergence rate of the explicit time-stepping scheme is accelerated using local time stepping [17], implicit residual smoothing [26, 28, 31] and multigrid [27, 29, 32]. In this work, V-cycle procedure with three grid levels is used to execute the multigrid strategy. One Runge-Kutta time step before the restriction and no Runge-Kutta time step after prolongation are done. Residuals and flow variables are restricted from fine to coarse grid by a weighted average. A forcing function is introduced into the time-stepping scheme. Solution corrections are prolongated from coarse to fine grid by bilinear interpolation. Implicit smoothing of solution corrections with constant coefficients is used in order to damp the high frequency errors, which are introduced by interpolation of the solution corrections. The artificial dissipation model with constant coefficient, second-order differences is used on the coarse grids to reduce computational effort. The Courant-Friedrichs-Lewy (CFL) number of 7.5 is used on all grids so that larger time steps are used on coarser grids. On each grid, boundary conditions are treated in the same way.

Table 1. Optimized stage coefficients

Central Differencing Scheme					
m	1	2	3	4	5
α	1/4	1/6	3/8	1/2	1

GEOMETRIC CONSERVATION LAW

GCL is a conservation law, which is necessary and sufficient for retaining on moving grids the nonlinear stability achieved on non-moving grids [8]. However, although satisfying the GCL is a sufficient condition for achieving first order time-accuracy [10], it is neither a necessary nor a sufficient condition for retaining on moving grids the second order time-accuracy achieved on non-moving grids [5].

GCL in integral form reads [34]

$$\frac{\partial}{\partial t}\iint_{\Omega(t)}d\Omega-\oint_{\partial\Omega(t)}\vec{V}_{cf}\,\vec{n}dS=0\,. \tag{32}$$

The GCL relates the change in volume of a cell to the motion of its faces. The use of GCL guarantees that grid motion does not disturb the uniform flow. The GCL is discretized temporally using three-point backward difference formula;

$$\frac{3\Omega^{n+1}-4\Omega^{n}+\Omega^{n-1}}{2\Delta t}=\sum_{ncf=1}^{4}\left[\vec{V}_{cf}\,\vec{n}\Delta S\right]^{n+1}_{ncf}\,. \tag{33}$$

The change in volume of a cell between two consecutive time steps (t^{n},t^{n+1}) is defined as

$$\Delta\Omega^{n+1}=\Omega^{n+1}-\Omega^{n}=\sum_{ncf=1}^{4}\delta\Omega^{n+1}_{ncf}\,, \tag{34}$$

where $\delta\Omega^{n+1}_{ncf}$ is the volume swept by the cell face ncf during time interval $\Delta t=t^{n+1}-t^{n}$. Then, eqn. (33) can be written as

$$\frac{3\Delta\Omega^{n+1}-\Delta\Omega^{n}}{2\Delta t}=\sum_{ncf=1}^{4}\left[\vec{V}_{cf}\,\vec{n}\Delta S\right]^{n+1}_{ncf}\,. \tag{35}$$

Substituting eqn. (34) into eqn. (35) yields

$$\sum_{ncf=1}^{4}\frac{3\delta\Omega^{n+1}_{ncf}-\delta\Omega^{n}_{ncf}}{2\Delta t}=\sum_{ncf=1}^{4}\left[\vec{V}_{cf}\,\vec{n}\Delta S\right]^{n+1}_{ncf}\,. \tag{36}$$

The face normal integrated velocity is defined as

$$\left[\vec{V}_{cf}\,\vec{n}\Delta S\right]^{n+1}_{ncf}=\frac{3\delta\Omega^{n+1}_{ncf}-\delta\Omega^{n}_{ncf}}{2\Delta t}\,. \tag{37}$$

The volume swept by each cell face during Δt is equal to the area of the quadrilateral formed by joining the grid points at two consecutive time levels. The use of eqn. (37) together with the cell volumes and cell face normal vectors, which are computed at the n+1 time level, in the computation of convective and viscous fluxes guarantees that temporal discretization is of second order on moving grids. GCL needs no further treatment for the computation of viscous flows.

MOVING GRID ALGORITHM

In order to solve unsteady flow problems with moving boundaries, grid has to be regenerated at each time step, such that it fits the body motion. In this work, a perturbation method based on algebraic TFI is used for rapid generation of structured moving grids [6, 9, 11]. The grid points are modified using the displacements of corner points and block boundaries. The use of perturbation method guarantees that quality of the original grid is retained and allows each boundary to move independently. Grid perturbation algorithm proceeds as follows:

Compute the displacement of corner points

Grid points at the block corners are moved from their initial locations to the following positions at n+1 time level according to a prescribed motion. Then, displacement dx of each corner point is computed.

Compute the displacement of grid points along the block boundaries

The displacements of the block boundaries are interpolated using the displacements of block corners and the grid coordinates from the previous time step. Displacement dx of any grid point in I direction is defined by one dimensional TFI as

$$dx_{I,1} = \left(1 - L^I_{I,1}\right) dx_{1,1} + L^I_{I,1} dx_{\mathrm{IMAX},1} \tag{38}$$

where $dx_{1,1}$ and $dx_{\mathrm{IMAX},1}$ are the displacements of two corner points. L^I is the parameterized arclength of the grid points in I direction:

$$L^I_{I,1} = \frac{\sum_{k=2}^{I} \sqrt{\left(x_{k,1} - x_{k-1,1}\right)^2 + \left(y_{k,1} - y_{k-1,1}\right)^2}}{\sum_{k=2}^{\mathrm{IMAX}} \sqrt{\left(x_{k,1} - x_{k-1,1}\right)^2 + \left(y_{k,1} - y_{k-1,1}\right)^2}}, \tag{39}$$

where $x - y$ are grid coordinates from the previous time step.

Compute the displacement of interior grid points

The displacements of the interior grid points are interpolated using the displacements of the block boundaries and the interior grid coordinates from the previous time step. Two dimensional TFI for grid displacements is defined as [6, 37]

$$\begin{aligned} dx_{I,J} = {} & \phi^1_{I,J} dx_{I,1} + \phi^2_{I,J} dx_{I,\mathrm{JMAX}} \\ & + \psi^1_{I,J} dx_{1,J} + \psi^2_{I,J} dx_{\mathrm{IMAX},J} \\ & - \phi^1_{I,J} \psi^1_{I,J} dx_{1,1} - \phi^2_{I,J} \psi^1_{I,J} dx_{1,\mathrm{JMAX}} \\ & - \phi^1_{I,J} \psi^2_{I,J} dx_{\mathrm{IMAX},1} - \phi^2_{I,J} \psi^2_{I,J} dx_{\mathrm{IMAX,JMAX}} \end{aligned} \tag{40}$$

where $dx_{1,1}$, $dx_{1,\mathrm{JMAX}}$, $dx_{\mathrm{IMAX},1}$, $dx_{\mathrm{IMAX,JMAX}}$ denotes the displacements of corner points and $dx_{I,1}$, $dx_{1,\mathrm{JMAX}}$, $dx_{1,J}$, $dx_{\mathrm{IMAX},J}$ denotes the displacements of grid points along the block boundaries. Blending functions are defined as [30]

$$\begin{aligned} \phi^1_{I,J} &= 1 - \eta_{I,J} \\ \phi^2_{I,J} &= \eta_{I,J} \\ \psi^1_{I,J} &= 1 - \xi_{I,J} \\ \psi^2_{I,J} &= \xi_{I,J} \end{aligned}, \tag{41}$$

where

$$\xi_{I,J} = \frac{L^J_{1,J}\left(L^I_{I,\mathrm{JMAX}} - L^I_{I,1}\right) + L^I_{I,1}}{1 - \left(L^I_{I,\mathrm{JMAX}} - L^I_{I,1}\right)\left(L^J_{\mathrm{IMAX},J} - L^J_{1,J}\right)}, \tag{42}$$

$$\eta_{I,J} = \frac{L^I_{I,1}\left(L^J_{\mathrm{IMAX},J} - L^J_{1,J}\right) + L^J_{1,J}}{1 - \left(L^I_{I,\mathrm{JMAX}} - L^I_{I,1}\right)\left(L^J_{\mathrm{IMAX},J} - L^J_{1,J}\right)}, \tag{43}$$

with L^J representing the parameterized arclength of grid points in J direction.

Update grid points

The coordinates of the new grid points are obtained by adding the displacements onto the grid positions from the previous time step.

$$x^{n+1}_{I,J} = x^n_{I,J} + dx_{I,J}\,. \tag{44}$$

A computational block in two dimensions includes four boundaries. In case, any boundary involves multi-segments, the parameterized arclength of grid points on each segment is calculated separately. Any boundary (segment), which connects moving and stationary boundaries (segments), is allowed to adapt itself to the motion of accompanying boundaries (segments). In case of large deformations of viscous computational grid, perturbation method based on algebraic TFI results crossover of cells near solid boundaries and wake cut. In this work, the crossover of cells is prevented by setting the displacement of grid points in the boundary layer and in the wake equal to those along the solid surface and wake cut.

IMPLEMENTATION

For a block with moving boundaries, implementation of ALE method proceeds according to following steps:

1. Calculate steady state solution to use as an initial guess, $\vec{C}^n_{I,J}$.
2. Deform the grid to next time step;
   a. move corner points,
   b. distribute the displacements onto segments of the moving boundary,
   c. distribute the displacements onto the interior grid points,
   d. update the interior grid coordinates, $x^{n+1}_{I,J}$.
3. Calculate the volumes swept by each cell faces, $\delta\Omega^{n+1}_{ncf}$.
4. Calculate the cell face normal integrated velocities, $\left(\vec{V}_{cf}\vec{n}\Delta S\right)^{n+1}_{ncf}$.
5. Calculate cell volumes, Ω^{n+1} and face normal vectors, $\left(n\Delta S\right)^{n+1}$.
6. Solve FANS equations in ALE form.
7. Repeat from step 2.

BOUNDARY CONDITIONS

Boundary conditions on all multigrid levels are treated in the same way. In case of inviscid flows, no flow across the wall condition is enforced at the solid walls. This requires the condition $\left(\vec{V} - \vec{V}_{cf}\right)\vec{n} = 0$ to be satisfied. Then, the vector of convective fluxes has only the terms including wall pressure and wall velocity. The wall pressure is extrapolated from the interior cells. The wall velocity is computed by eqn. (37). In case of viscous flows, “no slip” condition is enforced. It results in a fluid velocity at the wall to be equal to wall velocity and the velocity components are extrapolated into ghost cells using the following equations:

$$\begin{aligned} \tilde{u}_{ghost} &= -\tilde{u} + 2\tilde{u}_{cf} \\ \tilde{v}_{ghost} &= -\tilde{v} + 2\tilde{v}_{cf} \end{aligned} \tag{45}$$

The pressure and density at the ghost cell are set to be equal to those of boundary cell. The non-reflecting boundary condition, which is based on Riemann invariants, is applied to the farfield boundary.

COMPUTATIONAL RESULTS

The numerical results presented in this section validate the accuracy and computational efficiency of the present solver, which employs ALE form of time dependent governing equations and GCL, for solving flow problems with moving boundaries on deforming grids. Inviscid and viscous flows past a NACA 0012 airfoil pitching about its quarter-chord are studied as test cases [13]. The airfoil incidence is defined as a function of time as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t)\,, \tag{46}$$

where α_m is the mean incidence, α_0 is the pitch amplitude and ω is the angular frequency given by

$$\omega = 2kU_\infty / c \qquad [rad/s]. \tag{47}$$

k is the reduced frequency, c is the chord length and U_∞ is the free stream velocity. AGARD test cases presented in table 2 are validated in this study.

Table 2. AGARD Test Cases

AGARD CASE	M_∞	Re_∞	α_m (deg.)	α_0 (deg.)	k
CT 1	0.6	$4.8\text{x}10^6$	2.89	2.41	0.0808
CT 5	0.755	$5.5\text{x}10^6$	0.016	2.51	0.0814

For each AGARD test cases, a sensitivity study is carried out including the effects of both grid density and physical time step. Grid and time step independent results are presented only. Numerical results presented in this section are obtained using C-type grid with the farfield boundary located at a distance of 15 chords away from the airfoil (figure 2). Inviscid grid involves 160x32 cells with 96 cells on the airfoil. The spacing between the first grid point and the airfoil surface is 0.003c at the leading edge and is 0.01c at the trailing edge. Viscous grid involves 320x64 cells with 192 cells on the airfoil. The spacing between the first grid point and the airfoil surface is $3\text{x}10^{-6}$c at the leading edge and is $5\text{x}10^{-6}$c at the trailing edge.

A physical time step, which corresponds to about 75 steps per complete cycle of the airfoil, is used to obtain time-dependent solution. Larger physical time steps corresponding to a minimum of 40 steps per cycle can be used without loss of solution accuracy. Steady flow solution computed at the mean incidence is used as an initial guess to unsteady flow computations. V-cycle procedure with three grid levels is used to execute the multigrid strategy. At each time step, farfield boundary is kept stationary, while the airfoil is pitching. Convergence criterion in pseudo-time is based on L2 norm of density and it is set to 10^{-3}. Further reduction in convergence criterion didn't improve the solution accuracy noticeably. Computations are done on a PC having 512 Mbytes memory and operating Windows XP.

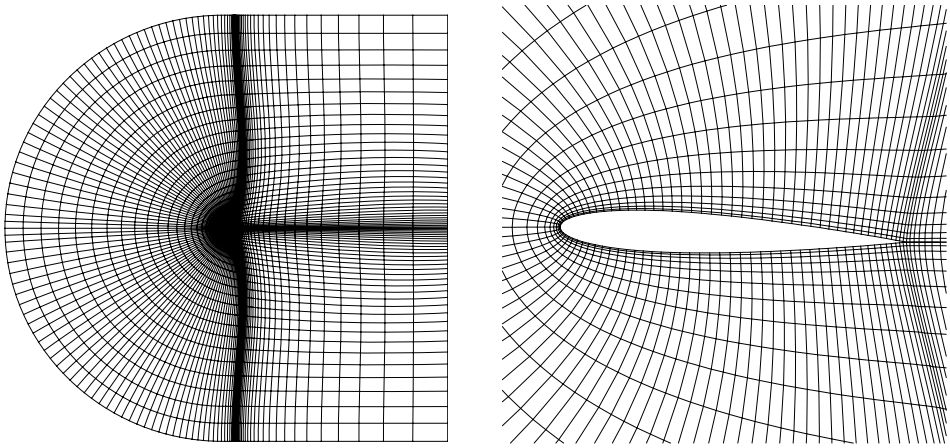

Figure 2: *Computational grid at 2.526 degrees of incidence*

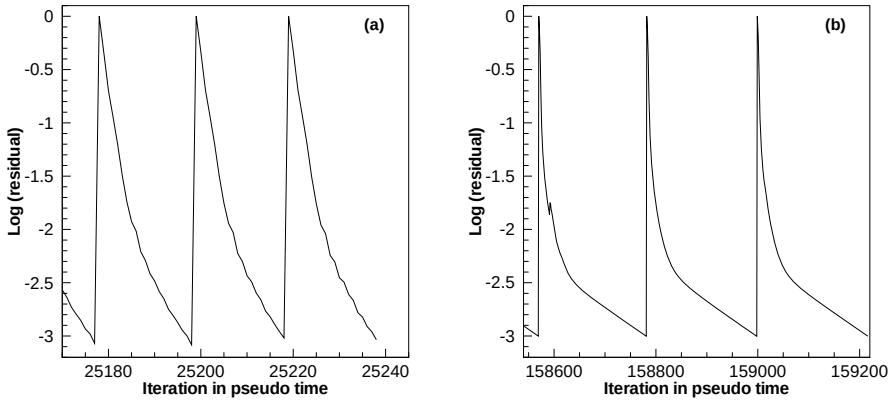

Figure 3: *Convergence history – inviscid (a) and viscous flows (b) (CASE 1)*

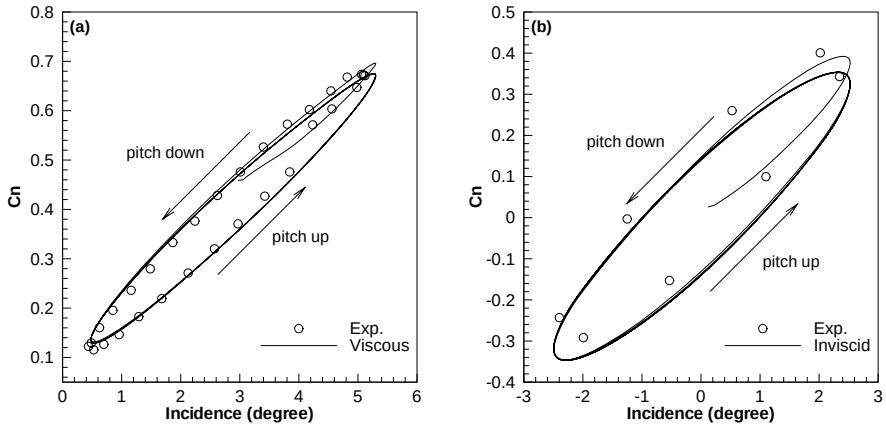

Figure 4: *Normal force vs. incidence angle – CASE 1 (a) and CASE 5 (b)*

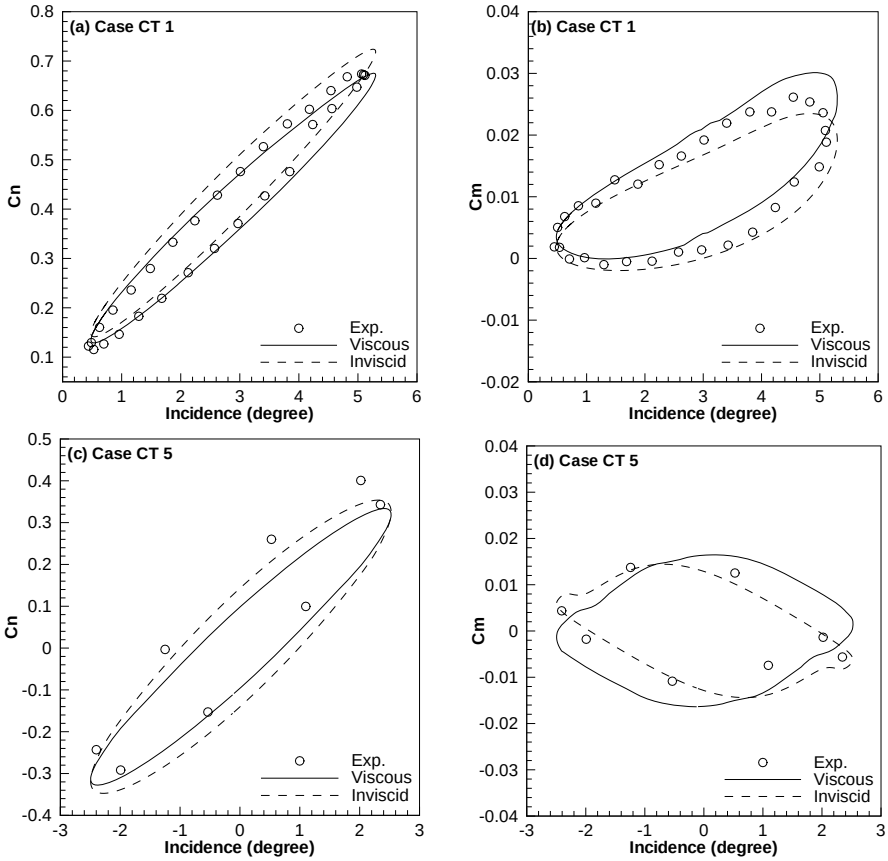

Figure 5: *Normal force (a, c) and pitching moment (b, d) vs. incidence angle*

Figure 3 indicates the convergence histories at the last three time steps of the fourth cycle of Euler and Navier-Stokes computations. The fast convergence rate in the computations is obtained owing to the use of multigrid method. Euler and Navier-Stokes solutions become periodic after an initial hysterisis and the following cycle (figure 4). Normal force and pitching moment coefficients have two values at a given incidence depending on whether the airfoil is pitching up or down.

Figures 5a and 5b present the normal force and pitching moment coefficients (about 0.273c) computed at the fourth cycle of Euler and Navier-Stokes computations for Case 1. Normal force is overestimated in case of Euler computations. The difference in the normal force between computation and experiment can be attributed to the absence of viscous effects. In case of Navier-Stokes computations, normal force and moment coefficient are in good agreement with experiment. Slight differences between computation and experiment appearing at high incidences may be attributed to the use of algebraic turbulence model. While the airfoil incidence is increasing, a shock wave appears on the upper surface (figure 6).

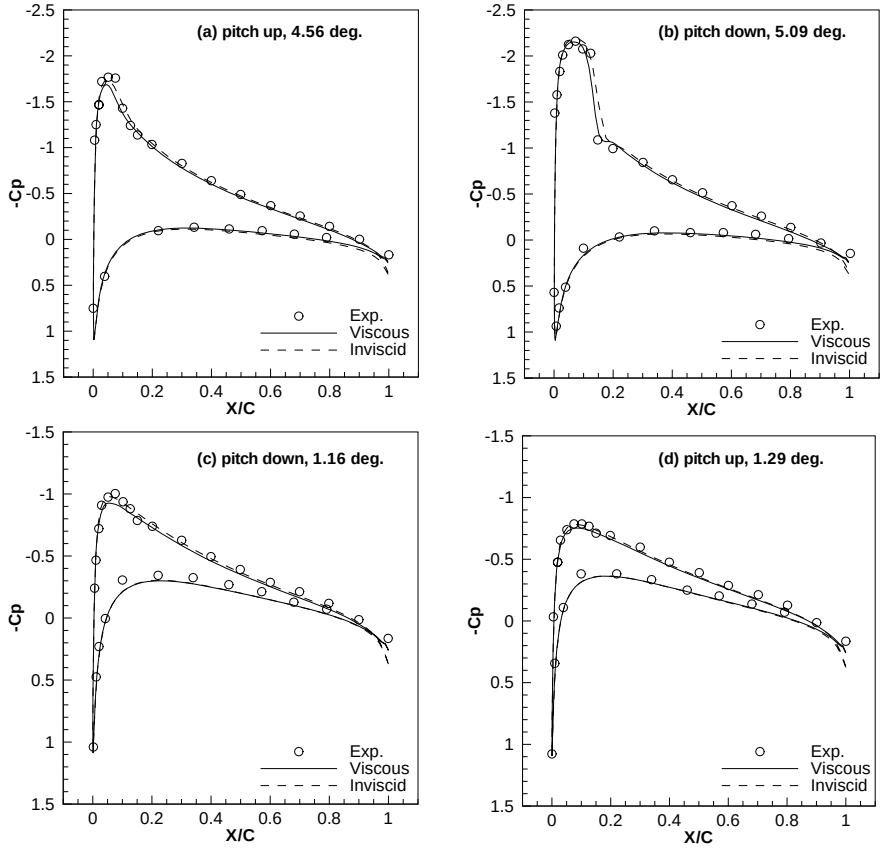

Figure 6: *Pressure distributions in various instants (CASE 1)*

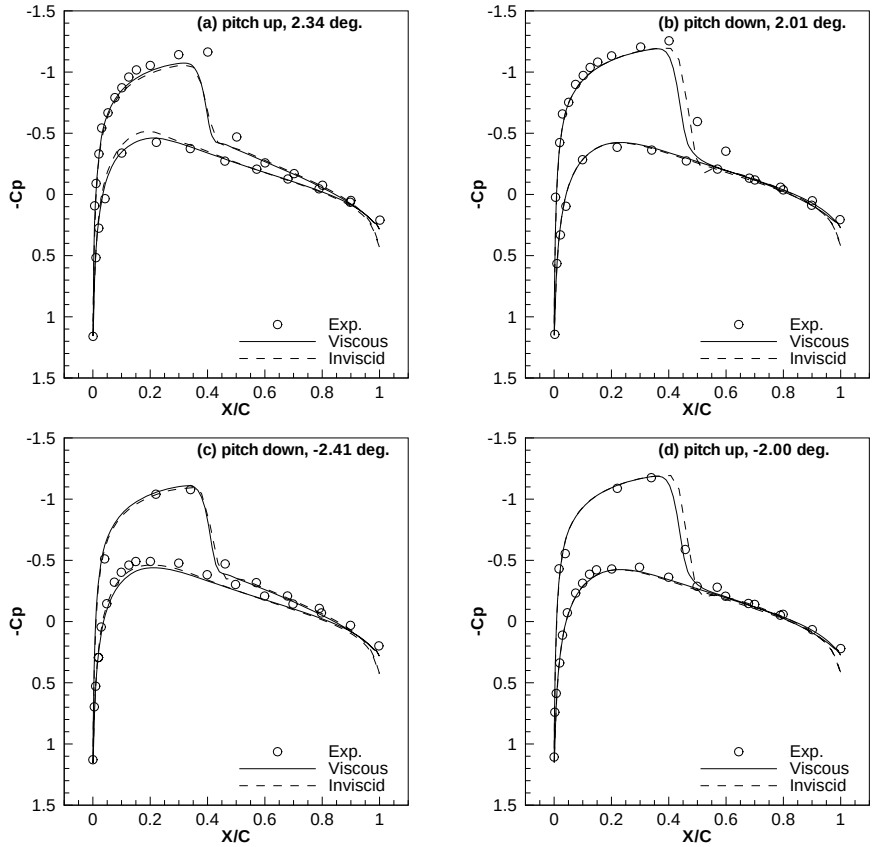

Figure 7: *Pressure distributions in various instants (CASE 5)*

Shock wave is captured well and pressure distributions past the airfoil agree well with experiment at all incidences. The present results agree well with the numerical data presented in ref. [14, 20], although they are not shown here.

Figures 5c and 5d present the normal force and pitching moment coefficients (about 0.25c) computed at the fourth cycle of Euler and Navier-Stokes computations for Case 5. This is a challenging test case, since viscous effects are significant due to high free stream Mach number. Normal force is underestimated in all computations. The difference in normal force and moment coefficient between computation and experiment may be attributed to far upstream prediction of shock location (figure 7). While the airfoil is pitching, a shock wave appears on the upper and lower surfaces in succession and shock wave location moves around 0.25c. The shock waves get sharper as the incidence angle increases. The present results agree well with the numerical data presented in ref. [2, 14, 25], although they are not shown here.

CONCLUSION

The time-dependent equations, which are written for an ALE system, were solved using an implicit dual time-stepping method. GCL compliant ALE time-stepping scheme, which retains on moving grids the nonlinear stability and the second order time-accuracy achieved on non-moving grids, was used. GCL was solved concurrently with the time-dependent equations. The face normal integrated velocities, cell volumes and cell face normal vectors are evaluated at the n+1 time level as proposed by dual time stepping method. A perturbation method based on algebraic Transfinite Interpolation (TFI) was used for rapid generation of structured moving grids at each time step.

Numerical results from Euler and Navier-Stokes computations for flows past a NACA 0012 airfoil pitching about its quarter-chord agree experimental data. Grid deformation and GCL compliant ALE time-stepping scheme have no detrimental effect on the solution accuracy. Time step is selected based on the numerical accuracy considerations and not stability restrictions. The computation of unsteady viscous flows with the GCL compliant ALE time-stepping scheme involves no further treatment other than boundary condition at no-slip walls, since the viscous fluxes are not affected by the grid motion and the GCL needs no further treatment for viscous flows.

References

[1] Mavriplis, D.J., and Yang, Z., *Construction of the Discrete Geometric Conservation Law for High-Order Time-Accurate Simulations on Dynamic Meshes*, Journal of Computational Physics, Vol. 213, pp: 557-573, 2006.

[2] Jahangirian, A., and Hadidoolabi, M., *Unstructured Moving Grids for Implicit Calculation of Unsteady Compressible Viscous Flows,* International Journal for Numerical Methods in Fluids, Vol.47, pp: 1107-1113. 2005.

[3] Donea, J., Huerta, A., Ponthot, J., and Rodriguez-Ferran, A., *Arbitrary Lagrangian-Eulerian Methods*, Encyclopedia of Computational Mechanics, Volume 1: Fundamentals, Chapter 14, John Wiley&Sons Ltd., 2004.

[4] Turkel, E., and Vatsa, V.N.; *Choice of Variables and Preconditioning for Time Dependent Problems*, AIAA Paper 2003-3692, 2003.

[5] Geuzaine, P., Crandmont, C., and Farhat, C.; *Design and Analysis of ALE Schemes with Provable Second-order Time-accuracy for Inviscid and Viscous Flow Simulations*, Journal of Computational Physics, Vol. 191, pp: 206-227, 2003.

[6] Farrashkhalvat, M., Miles, J.P., *Basic Structured Grid Generation with an Introduction to Unstructured Grid Generation*, Butterworth-Heinemann, 2003.

[7] Chassaing, J.C., Gerolymos, G.A., and Vallet, I., *Reynolds-Stress Model Dual-Time-Stepping Computation of Unsteady 3-D Flows*, AIAA Journal, Vol.41, No.10, pp: 1882-1894, 2003.

[8] Farhat, C., Geuzaine, P., and Crandmont, C., *The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution Flow Problems on Moving Grids*, Journal of Computational Physics, Vol. 174, pp: 669-694, 2001.

[9] Wong, A.S.F., Tsai, H.M., Cai, J., Zhu, Y., and Liu, F., *Unsteady Flow Calculations with a Multi-Block Moving Mesh Algorithm*, AIAA Journal, Vol. 39, No. 6, pp: 1021-1029, June 2001.

[10] Guillard, H., and Farhat, C., *On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes*, Computer Methods in Applied Mechanics and Engineering, Vol.190, pp: 1467-1482, 2000.

[11] Dubuc, L., Cantariti, F., Woodgate, M., Gribben, B., Badcock, K.J., and Richards, B.E., *A Grid Deformation Technique for Unsteady Flow Computations*, International Journal of Numerical Methods in Fluids Vol.32, pp: 285-311, 2000.

[12] Badcock, K.J., Cantariti, F., Hawkins, I., Woodgate, M., Dubuc, L., and Richards, B.E., *Simulation of Unsteady Turbulent Flows Around Moving Aerofoils Using the Pseudo-Time Method,* International Journal of Numerical Methods in Fluids, Vol.32, pp: 585-604, 2000.

[13] Landon, R.H., *NACA0012 Oscillatory and Transient Pitching*, in Verification and Validation Data for Computational Unsteady Aerodynamics, RTO-TR-26, October 2000.

[14] Dubuc, L., Cantariti, F., Woodgate, B., Gribben, I., Badcock, K.J., and Richards, B.E., *Solution of the Euler Equations Using Deforming Grids,* Aero. Report 9704, March 1997.

[15] Venkatakrishnan, V. and Mavriplis, D.J., *Implicit Method for the Computation of Unsteady Flows on Unstructured Grids*, Journal of Computational Physics, Vol.127, pp: 380-397, 1996.

[16] Melson, N.D., and Sanetrik, M.D., *Multigrid Acceleration of Time-Accurate Navier-Stokes Calculations*, 7th Cooper Mountain Conference on Multigrid Methods, 1995.

[17] Arnone, A., Liou, M.S. and Povinelli, L.A., *Integration of Navier-Stokes Equations Using Dual Time Stepping and a Multigrid Method*, AIAA Journal, Vol.33, No. 6, pp: 985-990, June 1995.

[18] Alonso, J., Martinelli, L., and Jameson, A., *Multigrid Unsteady Navier-Stokes Calculation with Aeroelastic Application,* AIAA paper 95-0048, 1995.

[19] Blazek, J., *Verfahren zur Beschleunigung der Lösung der Euler- und Navier–Stokes Gleichungen bei stationaren Über-und Hyperschallströmungen*, Ph.D. Thesis, University of Braunschweig, 1994.

[20] Giatonde, A.L. *A Dual-Time Method for the Solution of the Unsteady Euler Equations,"* Aeronautical Journal, No.2049, pp: 283-291, 1994.

[21] Rizzi, A., Eliasson, P., Lindbland, I., Hirsch, C., Lacor, C., and Haeuser, J., *The Engineering of Multiblock/Multigrid Software for Navier-Stokes Flows on Structured Meshes*, Computers& Fluids, Vol.22, No.2/3, pp: 341-367, 1993.

[22] Melson, N.D., Sanetrik, M.D., and Atkins, H.L., *Time-Accurate Navier-Stokes Calculations with Multigrid Accelerations*, 6th Cooper Mountain Conference on Multigrid Methods, 1993.

[23] Wilcox, D.C., *Turbulence Modeling for CFD*, DCW Industries Inc., La Canada, California, USA, 1993.

[24] Jameson, A., *Time Dependent Calculations Using Multigrid with Application to Unsteady Flows past Airfoils and Wings*, AIAA Paper 91-1596, 1991.

[25] Batina, J.T., *Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes,* AIAA 89-0115, Jan. 1989.

[26] Martinelli, L., and Jameson, A., *Validation of a Multigrid Method for the Reynolds Averaged Equations*, AIAA 88-0414, 1988.

[27] Martinelli, L., Jameson, A., and Grasso, F., *A Multigrid Method for the Navier-Stokes Equations*, AIAA 86-0208, Jan. 1986.

[28] Jameson, A., *Transonic Flow Calculations*, Princeton University Report MAE 1651, March 1984, in Numerical Methods in Fluid Dynamics, edited by F. Brezzi, Lecture Notes in Mathematics, Vol. 1127, Springer-Verlag, pp: 156-242, 1985.

[29] Jameson, A., *Multigrid Algorithms for Compressible Flow Calculations*, Mechanical and Aerospace Engineering Report 1743, Princeton Univ. NJ, Proceedings of the Second European Conference on Multigrid Methods, Cologne, October 1985.

[30] Soni, B.K., *Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics*, AIAA Paper 85-1526, 1985.

[31] Jameson, A., and Baker, T.J., *Solution of the Euler Equation for Complex Configurations*, AIAA Paper 83-1929, Proceedings of 6th AIAA Computational Fluid Dynamics Conference, Danvers, July 1983.

[32] Jameson, A., *Solution of the Euler Equations fro Two-dimensional, Transonic Flow by a Multigrid Method*, Applied Mathematics and Computation, Vol. 13, pp: 327-356, 1983.

[33] Jameson, A., Schmidt, W., and Turkel, E., *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*, AIAA Paper 81-1259, AIAA 14th Fluid and Plasma Dynamic Conference, Palo Alto, June 1981.

[34] Thomas, P.D., and Lombard, C.K., *Geometric Conservation Law and its Application to Flow Computations on Moving Grids*, AIAA Journal, Vol. 17, No. 1, pp: 1030-1037, 1979.

[35] Baldwin, B.S., and Lomax, H., *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*, AIAA Paper 78-257, 16th Aerospace Sciences Meeting, Huntsville, Alabama, January 1978.

[36] Hirt, C.W., Amsden, A.A., and Cook, J.L., *An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds*, Journal of Computational Physics, Vol. 14, No 3, pp: 227-253, March 1974.

[37] Gordon, W.N., and Hall, C.A., *Construction of Curvilinear Coordinate Systems and Application to Mesh Generation*, International Journal of Numerical Methods in Engineering, Vol. 7, pp: 461-477, 1973.