AN EFFICIENT PRECONDITIONING METHOD AND ITS APPLICATION TO NUMERICAL SIMULATION OF STEADY AND UNSTEADY VISCOUS FLOWS

Han Zhong-hua*, Qiao Zhi-de*, Song Wenping*, Xiong Juao-tao*
*** National Key Laboratory of Aerodynamic Design and Research, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, P.R.China**

Keywords: *Navier-Stokes Equations, preconditioning method, multigrid method, fully implicit scheme, LU-SGS Method*

Abstract

A time-accurate, fully implicit preconditioning method is developed and applied to solve a variety of steady and unsteady viscous flow problems. The space discretizaition is performed by employing a finite-volume cell-centred scheme and using a central difference with nearly 2nd-order accuracy. The time marching is based on a dual-time stepping method proposed by Jameson. LU-SGS (Lower-Upper Symmetric-Gauss-Seidel) implicit scheme within the framework of FAS (Full Approximation Scheme) multigrid method is applied to implement sub-iterations. A low-speed matrix preconditioning is utilized to improve the efficiency and accuracy of simulating incompressible flows or low-speed flow regions in compressible flows. In order to obtain time-accurate solutions, only the pseudo-time derivatives are preconditioned. Moreover, the original LU-SGS scheme and multigrid method are revised according to the preconditioned governing equations. The numerical examples show that the fully implicit preconditioning method is very efficient and robust for solving both 2-D and 3-D, steady and unsteady flow problems, especially for incompressible flow problems.

1 Introduction

Over the past two decades, time-marching algorithms have been widely used for compressible flow simulations. However, it is proved by practice that these "standard" numerical schemes for the compressible equations are not efficient for incompressible flows, and do not converge to the solution of the incompressible equations as the Mach number approaches zero [1]. The difficult in solving compressible equations for low Mach number flows are closely related to the stiffness of systems caused by large disparity of the different eigenvalues. To overcome the above difficult, several methods have been developed for solving nearly incompressible flow problems. Among them, pseudo-compressibility [2] method and preconditioning method [1, 3-11] may be the most popularly used approaches. Due to the assumption of incompressibility, pseudo-compressibility can only be used for low-speed flow simulation, whereas preconditioning method is appropriate for both compressible and incompressible flows. The development of preconditioning method is motivated by two main considerations. First, the actual flow can contain both compressible and incompressible flows simultaneously. Second, it is preferable to develop such a method that is suit for flows at all flow regimes.

Several inviscid and viscous preconditioning methods have been available in the past two decades. Early studies of preconditioning for low Mach-number flows were reported by Briley et al. [3], and D.Choi and Merkle [4], and also by Merkle and Y.-H. Choi [5]. Turkel [6] published a review of preconditioning methods for convection-dominated flow problems and developed a two-parameter preconditioner. Van Leer and Lee [7] proposed a matrix preconditioning for an explicit scheme to

remove stiffness of Euler Equations at all flow regimes. More recently, Y. -H, Choi and Merkle [8] proposed a matrix preconditioning method for viscous flows and successfully applied to 2-D low Reynolds number flows. Turkel [9, 10, 11] developed a generalization of Turkel's [6] and Choi-Merkle's preconditioners for compressible equations. In Ref. [10], Turkel's preconditioner was successfully applied to inviscid and viscous flows over 2-D airfoils and 3-D wing using explicit Runge-Kutta scheme. More recently, Muradogly and Caughey [12] proposed a three-parameter preconditioner to accelerate the convergence to steady-state solutions of Euler equations at all speed. Another kind of preconditioning method, namely Block-Jacobian preconditioning method, was reported by Allmaras [13], Pierce and Giles [14] et al. it was used to damp the high-frequency error and accelerate the calculation speed of Navier-Stokes equations. However, Block-Jacobian preconditioning method does nothing to remove the stiffness of the systems for low Mach numbers. Among the available preconditioning methods mentioned above, Choi-Merkle's and Turkel's preconditioning methods seem to be appropriate for low Mach number, viscous flows. Hence, Both Choi-Merkle and Turkel's preconditioners are considered in this paper.

Nowadays, one of the challenges for developing preconditioning method is associated with the calculation of time-dependent flows. In Ref [15, 16], The Turkel's preconditioner, incorporated with dual-time stepping method, was extended to solve unsteady flows with explicit Runge-Kutta scheme. In present work, Turkel's and Choi-Merkle's preconditioning methods are combined with LU-SGS [17] (Lower-Upper Symmetric-Gauss-Seidel) implicit scheme and dual-time method, resulting in a fully implicit preconditioning method for steady and unsteady flow problems. A FAS (Full Approximation Scheme) multigrid method is also utilized to accelerate the convergence. A variety of 2-D and 3-D steady and unsteady flows are simulated with Mach number from 10^{-5} though 0.84. The effect of local preconditioning on convergence rate and accuracy is investigated. The present method is expected to use as an efficient and robust tool for aerodynamic optimization design of airfoil and wing.

2 Computational Methods

2.1 Governing Equations

After introduced pseudo-time derivative and its matrix preconditioning, the non-dimensional form of the three-dimensional compressible Navier-Stokes equations can be written as

$$\boldsymbol{P}\frac{\partial \boldsymbol{W}}{\partial \tau}+\alpha\frac{\partial \boldsymbol{W}}{\partial t}+\frac{\partial(\boldsymbol{E}-\boldsymbol{E}_v)}{\partial x}+\frac{\partial(\boldsymbol{F}-\boldsymbol{F}_v)}{\partial y}+\frac{\partial(\boldsymbol{G}-\boldsymbol{G}_v)}{\partial z}=\boldsymbol{0} \tag{1}$$

where, τ denotes pseudo time; for steady flows, α is set to be 0, while for unsteady flows α is taken as 1; $\boldsymbol{P}$ is preconditioning Matrix(or preconditioner) and will take on various forms depending on the different choices. When $\boldsymbol{P}$ is identity matrix, Eq. (1) recovers to the non-preconditioned form. The additional vectors in Eq. (1) are:

$$\begin{aligned}
\boldsymbol{W}&=\{\rho,\rho u,\rho v,\rho w,\rho E\}^{\mathrm{T}}\\
\boldsymbol{E}&=\{\rho u,\rho u^2+p,\rho uv,\rho uw,(\rho E+p)u\}^{\mathrm{T}}\\
\boldsymbol{F}&=\{\rho v,\rho uv,\rho v^2+p,\rho vw,(\rho E+p)v\}^{\mathrm{T}}\\
\boldsymbol{G}&=\{\rho w,\rho wu,\rho wv,\rho w^2+p,(\rho E+p)w\}^{\mathrm{T}}\\
\boldsymbol{E}_v&=\{0,\tau_{xx},\tau_{xy},\tau_{xz},\beta_x\}^{\mathrm{T}}\\
\boldsymbol{F}_v&=\{0,\tau_{yx},\tau_{yy},\tau_{yz},\beta_y\}^{\mathrm{T}}\\
\boldsymbol{G}_v&=\{0,\tau_{zx},\tau_{zy},\tau_{zz},\beta_z\}^{\mathrm{T}}
\end{aligned} \tag{2}$$

where, ρ,u,v,w,E denote density, components of velocity vector, total energy per unit mass, respectively. Pressure and temperature are given by the equation of state for perfect gas:

$$\begin{aligned}
p&=\rho(\gamma-1)\left[E-0.5(u^2+v^2)\right]\\
T&=p/\rho
\end{aligned} \tag{3}$$

where γ is ratio of specific heat and is taken as 1.4 for air. The viscous shear stresses and the heat fluxes are of the form

$$
\begin{aligned}
\tau_{xx} &= 2\mu u_x + \lambda(u_x + v_y + w_z) \\
\tau_{yy} &= 2\mu v_y + \lambda(u_x + v_y + w_z) \\
\tau_{zz} &= 2\mu w_z + \lambda(u_x + v_y + w_z) \\
\tau_{xy} &= \tau_{yx} = \mu(u_y + v_x) \\
\tau_{yz} &= \tau_{zy} = \mu(v_z + w_y) \\
\tau_{xz} &= \tau_{zx} = \mu(u_z + w_x) \\
\beta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + kT_x \\
\beta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + kT_y \\
\beta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + kT_z
\end{aligned} \tag{4}
$$

where k is the coefficient of thermal conductivity and is determined by using the assumption of constant Prandtl number. The bulk viscosity λ is taken to be $-2\mu/3$ according to Stokes's hypothesis. For turbulent flows, the total viscosity μ calculated as

$$\mu = \mu_l + u_t \tag{5}$$

where μ_l is molecular viscosity calculated by Sutherland law, and eddy viscosity μ_t is determined by turbulence model. Then, Eq. (1) can be called preconditioned Reynolds averaged Navier-Stokes equations. Spalart-Allmaras one-equation turbulence model [18] is used for all the calculations of the present work.

For the unsteady calculation using dual-time method, a sub-iteration procedure for pseudo-time is performed at each physical time step. At the convergence of sub-iterations ($\frac{\partial \boldsymbol{W}}{\partial \tau}$ approachs zero), the influence of preconditioning will be removed. Then the time-accurate solution can be correctly obtained.

Rather than using conservative variables $\boldsymbol{W} = \{\rho, \rho u, \rho v, \rho w, \rho E\}^{\mathrm{T}}$, primitive variables $\boldsymbol{Q} = \{p, u, v, w, T\}^{\mathrm{T}}$ are frequently used for viscous flows, especially for low-Mach number flows. Then Eq.(1) can be written as

$$\boldsymbol{\Gamma}\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial(\boldsymbol{E} - \boldsymbol{E}_v)}{\partial x} + \frac{\partial(\boldsymbol{F} - \boldsymbol{F}_v)}{\partial y} + \frac{\partial(\boldsymbol{G} - \boldsymbol{G}_v)}{\partial z} = 0 \tag{6}$$

where, $\boldsymbol{\Gamma} = \boldsymbol{P}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{Q}}$ represents preconditioning matrix for primitive variables. Note that, In Eq.(6), the flux terms keep the form of conservation law, then the shock wave occurred in transonic flow can be correctly captured.

2.2 Preconditioning Matrix

The proconditioner which is a generlization of preconditioners given by Turkel[6] and by Choi and Merkle[8] is considered here. It corresponds to the dependent variables $\boldsymbol{W}_0 = \{p, u, v, w, S\}^{\mathrm{T}}$ with $dS = dp - c^2 d\rho$. This general preconditioner reported by Turkel[9] takes the form

$$\boldsymbol{P}_0 = \begin{bmatrix} \frac{c^2}{\beta^2 Ma_r^2} & 0 & 0 & 0 & \delta \\ \frac{\varepsilon u}{\rho\beta^2 Ma_r^2} & 1 & 0 & 0 & 0 \\ \frac{\varepsilon v}{\rho\beta^2 Ma_r^2} & 0 & 1 & 0 & 0 \\ \frac{\varepsilon w}{\rho\beta^2 Ma_r^2} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where δ and ε are free parameters; c is local sound speed; βMa_r is associated with local Mach number. If $\delta = 0$, Eq.(7) corresponds to Turkel's preconditioner ; if $\delta = 1, \varepsilon = 0$, Eq.(7) becomes the Choi-Merkle' preconditioner . For the preconditioner indicated by Eq.(7), the correspondent form of $\boldsymbol{\Gamma}$ in Eq.(6) is

$$\boldsymbol{\Gamma} = \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{W}_0}\boldsymbol{P}_0\frac{\partial \boldsymbol{W}_0}{\partial \boldsymbol{Q}} \tag{8}$$

According to present authors' experience, when local Mach number is larger than 0.6, the effect of preconditioning can be gradually removed. For transonic flows, the local Mach number of the most flow region is larger than 0.6, whereas a small flow region (such as the flow in the boundary layer or near the stagnation points) is incompressible. It is preferable to switch on preconditioning in these low-speed regions and to smoothly reduce the effect of preconditioning outside these regions. On the other hand, the preconditioner has singularity as local Mach number approach zero near the stagnation point and in the boundary layer. Based on the above considerations, a control technique upon $\beta^2 Ma_r^2$ proposed by Turkel[9] is used as

$$\beta^2 Ma_r^2 = \min\{\max[K_1 c^2 Ma^2 (1 + \frac{1 - Ma_0^2}{Ma_0^4} Ma^2), K_2^2 V_\infty^2], c^2\} \tag{9}$$

where V_∞ is the velocity of free stream; K_1, K_2 are free parameters. K_1 is typically 1.0-1.1, and K_2 is typically 0.5-1.0. From Eq.(9), it is shown

that $\beta^2 Ma_r^2$ returns to the non-preconditioned value "1" when the local mach number Ma is larger than Ma_0. In this study, $K_1 = K_2 = 1.0$ and $Ma_0 = 0.6$ are used for all the computations.

To demonstrate the effect of preconditioning on the characteristic systems of governing equations in curvilinear coordinates, the Condition Number [1] (CN) is used to quantitively denote the eigenvalue stiffness. It is defined as

$$CN = \frac{Max|\lambda_i|}{Min|\lambda_i|}, i = 1 \sim 5 \qquad (10)$$

where λ_i is the eigenvalue of Jacobian matrix.

For Choi-Merkle preconditioner with $\delta = 1, \varepsilon = 0$, the eigenvalues of the preconditioned systems in ξ direction are

$$\lambda_{1,2,3} = q_\xi, \lambda_{4,5} = \frac{q_\xi(1+\beta^2 Ma_r^2/c^2) \pm c'}{2}$$
$$c'^2 = q_\xi^2(1+\frac{\beta^2 Ma_r^2}{c^2}) + 4\beta^2 Ma_r^2(\xi_x^2+\xi_y^2+\xi_z^2-\frac{q_\xi^2}{c^2}) \quad (11)$$
$$q_\xi = q_\xi = \xi_x u + \xi_y v + \xi_z w$$

Note that condition numbers is proved to be nearly 2.6 as Ma approaches zero

For Turkel preconditioner with $\delta = 0$, the eigenvalues of the preconditioned systems in ξ direction are

$$\lambda_{1,2,3} = q_\xi, \lambda_{4,5} = zq_\xi \pm \sqrt{z^2 q_\xi^2 + \beta^2 Ma_r^2(\xi_x^2+\xi_y^2+\xi_z^2-\frac{q_\xi^2}{c^2})}$$
$$z = 0.5(1-\varepsilon+\frac{\beta^2 Ma_r^2}{c^2}), q_\xi = \xi_x u + \xi_y v + \xi_z w \qquad (12)$$

When $Ma \to 0$, $\lambda_{4,5} \approx 0.5 q_\xi[(1-\varepsilon) \pm \sqrt{(1-\varepsilon)^2+4}]$, the condition numbers for $\varepsilon = 0$ and $\varepsilon = 1$ are proved to be nearly 2.6 and 1 respectively. The computing experience suggests that $\varepsilon = 0$ is relatively more efficient in actual application. Hence, $\varepsilon = 0$ is used in this paper. Then the condition number of using Turkel preconditioner is the same as that of using Choi-Merkle preconditioner .

For non-preconditioned systems, condition number is infinite if $M_a \to 0$. From the above discussion, it is clearly shown that both the Choi-Merkle and Turkel preconditioner can be used to the effectively eliminate eigenvalue stiffness of governing equations as $M_a \to 0$.

2.3 Spatial Discretization

A finite volume cell- centred scheme on structured grids and a central difference with nearly 2nd-order accuracy are used for the space discretization.

The governing equations are applied to an arbitrary control volume denoted by V. After utilizing Gauss's law, Eq.(1) becomes

$$\oint_V \boldsymbol{\Gamma}\frac{\partial \boldsymbol{Q}}{\partial \tau}dV + \alpha\oint_V \frac{\partial \boldsymbol{W}}{\partial t}dV + \oint_S(\overline{\overline{\boldsymbol{H}}} - \overline{\overline{\boldsymbol{H}_v}})\cdot \boldsymbol{n}dS = \boldsymbol{0} \quad (13)$$

where $\overline{\overline{\boldsymbol{H}}}$ and $\overline{\overline{\boldsymbol{H}_v}}$ denote inviscid and viscous flux vector, respectively. Let $\boldsymbol{q}_b$ denote the velocity vector of the boundary of control volume. The following equations is valid

$$\frac{d}{dt}\oint_V \boldsymbol{W}dV = \oint_V \frac{\partial \boldsymbol{W}}{\partial t}dV + \oint_S(\boldsymbol{W}\otimes \boldsymbol{q}_b)\cdot \boldsymbol{n}dS \qquad (14)$$

Substituting Eq.(14) into Eq.(13) yields

$$\oint_V \boldsymbol{\Gamma}\frac{\partial \boldsymbol{Q}}{\partial \tau}dV + \alpha\frac{d}{dt}\oint_V \boldsymbol{W}dV + \oint_S(\overline{\overline{\boldsymbol{H}}} - \alpha\boldsymbol{W}\otimes\boldsymbol{q}_b - \overline{\overline{\boldsymbol{H}_v}})\cdot \boldsymbol{n}dS = \boldsymbol{0} \qquad (15)$$

The flow variables are defined at center of the cell denoted by (i,j,k), and then the semi-discrete form of governing equations can be written as

$$V_{i,j,k}\boldsymbol{\Gamma}_{i,j,k}\frac{\partial \boldsymbol{Q}_{i,j,k}}{\partial \tau} + \alpha\frac{d}{dt}(V_{i,j,k}\boldsymbol{W}_{i,j,k}) + \boldsymbol{G}_{i,j,k} - \boldsymbol{G}^v_{i,j,k} = \boldsymbol{0} \qquad (16)$$

here, $\boldsymbol{G}_{i,j,k}$ and $\boldsymbol{G}^v_{i,j,k}$ are, respectively, the net convective and viscous flux out of the cell. A artificial dispassion term are introduced into Eq.(16) which becomes

$$V_{i,j,k}\boldsymbol{\Gamma}_{i,j,k}\frac{\partial \boldsymbol{Q}_{i,j,k}}{\partial \tau} + \alpha\frac{d}{dt}(V_{i,j,k}\boldsymbol{W}_{i,j,k}) + \boldsymbol{G}_{i,j,k} - \boldsymbol{G}^v_{i,j,k} - \boldsymbol{D}_{i,j,k} = \boldsymbol{0} \qquad (17)$$

The details of $\boldsymbol{D}_{ijk}$ can be found in Ref [10] and Ref. [19].

2.4 Fully Implicit Dual Time Stepping with LU-SGS Sub-iteration

For unsteady flow calculation using dual-time stepping method, an explicit scheme such as Runge-Kutta scheme is usually used to perform

the sub-iterations. Then the pseudo-time step $\Delta\tau$ will be restricted because of stability. In addition, $\Delta\tau$ is also restricted by physical time step Δt. In this paper, a LU-SGS implicit sub-iteration is implemented, which results in very efficient method due to the large pseudo-time step $\Delta\tau$ out of the restriction of stability and physical time step Δt. The dual-time method with LU-SGS sub-iteration is also modified in this paper according to the introducing of preconditioning method. Although an explicit treatment is applied to viscous terms as well as preconditioning matrix in this paper, the present scheme is fully implicit, which will be discussed later.

The physical time derivative in Eq.(17) is replaced by a backward difference formula(BDF) with 2nd-order accuracy. After some rearrangement, Eq.(17) becomes

$$V_{i,j,k}^{n+1}\frac{\partial \boldsymbol{Q}_{i,j,k}^{n+1}}{\partial \tau}+\alpha\boldsymbol{\Gamma}_{i,j,k}^{-1\ n+1}\frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{n+1}-4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n}+V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t}$$
$$+\boldsymbol{\Gamma}_{i,j,k}^{-1\ n+1}(\boldsymbol{G}_{i,j,k}^{n+1}-\boldsymbol{G}_{i,j,k}^{v\ n+1}-\boldsymbol{D}_{i,j,k}^{n+1})=\boldsymbol{0} \tag{18}$$

Then, the pseudo time derivative in above equation is replaced by a backward first-order difference as

$$V_{i,j,k}^{n+1}\frac{\boldsymbol{Q}_{i,j,k}^{m+1}-\boldsymbol{Q}_{i,j,k}^{m}}{\Delta\tau}$$
$$+\alpha\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}\frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{m+1}-4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n}+V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t} \tag{19}$$
$$+\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}(\boldsymbol{G}_{i,j,k}^{m+1}-\boldsymbol{G}_{i,j,k}^{v\ m}-\boldsymbol{D}_{i,j,k}^{m})=\boldsymbol{0}$$

where the subscript "n". and "m".denote the physical time level and pseudo-time level, respectively. Note that $\boldsymbol{G}_{i,j,k}^{v}$, $\boldsymbol{D}_{i,j,k}$ and $\boldsymbol{\Gamma}_{i,j,k}^{-1}$ are treated explicitly in Eq.(19). Let $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ be the Jacobian matrices of the convective normal components at the cell interfaces along the $i-$, $j-$ and $k-$ directions, respectively. The convective flux $\boldsymbol{G}_{i,j,k}^{m+1}$ can then be linearized about the time level "m". After dropping terms of the second and higher order, this yields

$$\begin{aligned}\boldsymbol{G}_{i,j,k}^{m+1}\approx\boldsymbol{G}_{i,j,k}^{m}&+(\boldsymbol{A}\Delta\boldsymbol{Q})_{i+1/2,j,k}^{m}-(\boldsymbol{A}\Delta\boldsymbol{Q})_{i-1/2,j,k}^{m}\\&+(\boldsymbol{B}\Delta\boldsymbol{Q})_{i,j+1/2,k}^{m}-(\boldsymbol{B}\Delta\boldsymbol{Q})_{i,j-1/2,k}^{m}\\&+(C\Delta\boldsymbol{Q})_{i,j,k+1/2}^{m}-(C\Delta\boldsymbol{Q})_{i,j,k-1/2}^{m}\end{aligned} \tag{20}$$

Substitute Eq.(20) into Eq.(19), one obtains

$$\left[(\frac{V_{i,j,k}^{n+1}}{\Delta\tau}\boldsymbol{I}+\frac{3\alpha}{2}\frac{V_{i,j,k}^{n+1}}{\Delta t}\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}+\delta_\xi\boldsymbol{A}+\delta_\eta\boldsymbol{B}+\delta_\zeta\boldsymbol{C}\right]\Delta\boldsymbol{Q}^{m}=$$
$$-\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}\left[\alpha\frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{m}-4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n}+V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t}+R_{i,j,k}^{m})\right] \tag{21}$$

where, $\Delta\boldsymbol{Q}^{m}=\Delta\boldsymbol{Q}_{i,j,k}^{m+1}-\Delta\boldsymbol{Q}_{i,j,k}^{m}$; $\boldsymbol{I}$ is identity matrix; $\delta_\xi,\delta_\eta,\delta_\zeta$ denote the spatial operators along the $i-$, $j-$ and k directions, respectively; $R_{i,j,k}^{m}=\boldsymbol{G}_{i,j,k}^{m}-\boldsymbol{G}_{i,j,k}^{v\ m}-\boldsymbol{D}_{i,j,k}^{m}$ is the flux residual vector for grid cell (i,j,k) .Let $\Delta\tau\to\infty$, the Eq.(20) becomes

$$\left[(\frac{3\alpha}{2}\frac{V_{i,j,k}^{n+1}}{\Delta t}\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}+\delta_\xi\boldsymbol{A}+\delta_\eta\boldsymbol{B}+\delta_\zeta\boldsymbol{C}\right]\Delta\boldsymbol{Q}^{m}=-R_{i,j,k}^{*\ m} \tag{21}$$

where

$$R_{i,j,k}^{*\ m}=\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}(\alpha\frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{m}-4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n}+V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t}+R_{i,j,k}^{m}) \tag{22}$$

is defined as "preconditioned unsteady residual".

Referencing the Jameson and Yoon's derivation [17] for LU operator and Jacobian splitting, the dual-time-stepping scheme may be written as follows:

$$(\boldsymbol{L}+\boldsymbol{D})\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{U})\Delta\boldsymbol{Q}^{m}=-R_{i,j,k}^{*\ m} \tag{22}$$

where

$$\begin{aligned}\boldsymbol{L}&=-\boldsymbol{\Gamma}_{i,j,k}^{-1}(\boldsymbol{A}_{i-1,j,k}^{+}+\boldsymbol{B}_{i,j-1,k}^{+}+C_{i,j,k-1}^{+})\\\boldsymbol{D}&=\frac{3\alpha}{2}\frac{V_{i,j,k}^{n+1}}{\Delta t}\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}+(r_{\tilde{A}}+r_{\tilde{B}}+r_{\tilde{C}})\boldsymbol{I}\\\boldsymbol{U}&=\boldsymbol{\Gamma}_{i,j,k}^{-1}(\boldsymbol{A}_{i+1,j,k}^{-}+\boldsymbol{B}_{i,j+1,k}^{-}+C_{i,j,k+1}^{-})\end{aligned} \tag{23}$$

The Jacobian matrix in i-direction is obtained by following splitting:

$$\boldsymbol{A}^{\pm}=\frac{\boldsymbol{A}\pm r_{\tilde{A}}}{2},\quad r_{\tilde{A}}\ge\max(|\lambda_{\tilde{A}}|),\ \tilde{\boldsymbol{A}}=\boldsymbol{\Gamma}^{-1}\boldsymbol{A} \tag{24}$$

where $r_{\tilde{A}}$ is the eigenvalue of $\tilde{\boldsymbol{A}}$. Similar procedurse are applied for Jacobian matrices $\boldsymbol{B}$ and $\boldsymbol{C}$.

The initial values for sub-iteration are taken as $\boldsymbol{Q}_{i,j,k}^{m}=\boldsymbol{Q}_{i,j,k}^{n}$. Starting with $\boldsymbol{Q}_{i,j,k}^{1}=\boldsymbol{Q}_{i,j,k}^{n}$, the sequence of iterations $\boldsymbol{Q}_{i,j,k}^{m},m=1,2,3...$ converges to $\boldsymbol{Q}_{i,j,k}^{n+1}$, as the "preconditioned unsteady residual" $R_{i,j,k}^{*\ m}$ equals zero.

In the case of convergence of sub-iterations, the following equation is valid as $\boldsymbol{Q}_{i,j,k}^{m+1}\to\boldsymbol{Q}_{i,j,k}^{n+1}$, $\boldsymbol{R}_{i,j,k}^{m}\to\boldsymbol{R}_{i,j,k}^{n+1}$ and $\boldsymbol{\Gamma}_{i,j,k}^{-1\ m}\to\boldsymbol{\Gamma}_{i,j,k}^{-1\ n+1}\ne\boldsymbol{0}$:

$$\alpha \frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{n+1} - 4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n} + V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t} + \boldsymbol{R}_{i,j,k}^{n+1} = \boldsymbol{0} \quad (25)$$

Substituting $R_{i,j,k}^{n+1} = \boldsymbol{G}_{i,j,k}^{n+1} - \boldsymbol{G}_{i,j,k}^{v\ n+1}$ into the above equations and setting α equal to 1 leads to a fully second-order implicit scheme in time for the governing equations,

$$\frac{3V_{i,j,k}^{n+1}\boldsymbol{W}_{i,j,k}^{n+1} - 4V_{i,j,k}^{n}\boldsymbol{W}_{i,j,k}^{n} + V_{i,j,k}^{n-1}\boldsymbol{W}_{i,j,k}^{n-1}}{2\Delta t} + (\boldsymbol{G}_{i,j,k}^{n+1} - \boldsymbol{G}_{i,j,k}^{v\ n+1}) = \boldsymbol{0} \quad (26)$$

The following points should be highlighted here for the above scheme: 1) the preconditioning procedure does not affect the time-accurate solution of governing equations; 2) the viscous terms are no longer time lagged as indicated by Eq. (26); 3) though the sub-iteration in pseudo-time, the linearization and factorization errors go to zero, and second-order time-accurate solution can be obtained.

2.5 Multigrid Method

LU-SGS Method has good damping characteristics for high-wave-number error. Effective removal of low-wave-number error is accomplished by using a FAS multigrid scheme developed by Jameson [20]. A coarser grid is created by removing every other grid line from the finer grid, essentially doubling the grid spacing in each direction. The multigrid cycle begins by iterating a fixed number of LU-SGS sweeping. Then the solution is restricted to coarse grid using a restrict operator. The residual $R_{i,j,k}^{*\ m}$ of fine grid is also restricted for calculating a forcing function that drives the solution on coarse grid. The forcing function is defined as the difference between the restricted fine-grid residuals and coarse-grid residuals of the first LU-SGS sweeping. The procedure is then repeated recursively. Once the coarsest grid is reached, an interpolation operator accomplishes the corrections procedure. The restriction and interpolation operators of the multigrid cycle are both based on volume-weighted averaging and trilinear(bilinear in 2D) interpolation, respectively. Computational experience suggests that the performance of this multigrid algorithm highly dependents on the number of iterations on each level and the type of multigrid cycle (such as “V” cycle or “W” cycle etc.). In present work, a 3-level V-cycle multigrid method is implemented as a simple but effective tool for accelerating the convergence of sub-iteration at each physical time step. At each multigrid cycle, the number of LU-SGS sweeping for each level is 5, 10 and 15, respectively. It is shown by computing practice that “5-10-15 V” cycle provides a relatively efficient and robust way for a wide variety of flow simulations.

2.6 Turbulence model

SA one-equation turbulence model is used to calculate the eddy viscosity μ_t. The turbulence equation is solved with such a numerical procedure that is similar to that of solving the governing equations. Note that the turbulence equation is only solved on the finest grid, and the eddy viscosity μ_t on the coarse grid is “fixed”.

2.7 Boundary Conditions

On solid surfaces, the no-slip condition is imposed by setting the velocity components u, v and w to zero. The normal momentum relation is used to specify the pressure gradient in the first cell off the wall; for high Reynolds number flows, the normal pressure gradient may be approximately set to zero. The far-field boundary conditions are treated by simplified boundary conditions proposed in Ref. [10]. Computing experience suggests that it is nearly identically efficient compared with the characteristic boundary condition based on the preconditioned systems.

3 Numerical Examples

A variety of 2-D and 3-D steady and unsteady flows are simulated with Mach number from 10^{-5} to 0.84. The effect of local preconditioning on convergence rate and accuracy is investigated. The computing practice in this paper shows that little difference exits between turkel’s[6] and Choi-Merkle’s [8] preconditioners.

Therefore, they will not be distinguished in the following investigation.

For the simulation of all the steady-state flows, Δt is set to be infinite as the time-marching scheme used in this study is unconditional stable. For all the unsteady computation, about 72 physical steps are used in one period （$\Delta t = period/72$ ）. Starting from the corresponding steady-state solution, the unsteady calculation usually takes 2-3 periods to reach fully periodic solution.

All the structured grids are generated with algebraic method based on transfinite interpolation [21] and elliptical smoothing technique. The grids have the first near-surface grid point below y+ = 0.7 to ensure the sublayer of the turbulent shear flow is sufficiently resolved. The first grid point is specified based on the relationship between y+, Reynolds number, and the skin-friction for a flat plate boundary layer. For turbulent flow, this relationship is $y^+ = \Delta y \cdot \sqrt{c_f/2}\,\mathrm{Re}$,where $c_f = 0.455/\ln^2(0.06\,\mathrm{Re}_x)$ and Δy is the distance for the first grid point away from the airfoil surface.

3.1 Examples of Steady-State Flows

3.1.1 Flows over NACA0012 Airfoil

The simulation of flows over NACA0012 airfoil is taken as the first example to demonstrate the effect of preconditioning, with Mach number ranging from ultra low speed to transonic flow regimes. The chord Reynolds number is 1.85 million, and the angle of attack is 3.59^0. A C Type grid (See Fig.1) consists of 265×65 points is used for these computations.

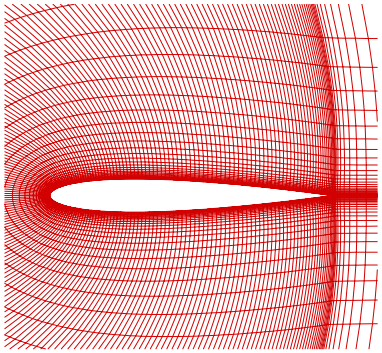

Fig.1 Room in view of C-type grid of NACA0012 airfoil

Fig.2 shows the comparison of convergence rates for low-Mach number flows with and without preconditioning. It is shown that the convergence rates are markedly improved by using preconditioning method. For the calculation without preconditioning, the convergence remarkably slows down as Mach number decreases. The residual stalls after dropping about 3.0 orders in magnitude for $Ma \le 0.01$. With preconditioning the convergence rate is nearly independent with Mach number for low-Mach number flows. The convergence histories of continuity equation for ultra-low-Mach-number flows with preconditioning are shown in Fig.3. It is shown that the residual is reduced until the limit of machine error is arrived.

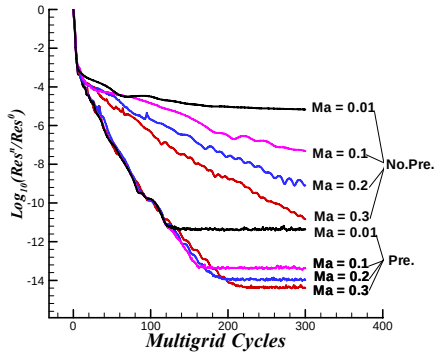

Fig.2 Effect of preconditioning on convergence rate for Mach number from 0.01 though 0.3

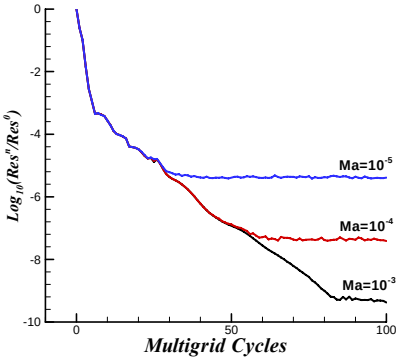

Fig.3 Convergence history of calculation with preconditioning for ultra low Mach number flows

The effect of preconditioning on convergence rate for subsonic and transonic flows is shown in Fig.4. $Ma = 0.5$ represents subsonic flow regime and $Ma = 0.7$ denotes transonic flow regime with moderate shock wave in the flow field. As illustrated in Fig.4, the convengence rate is significantly improved when the preconditioning is imposed.

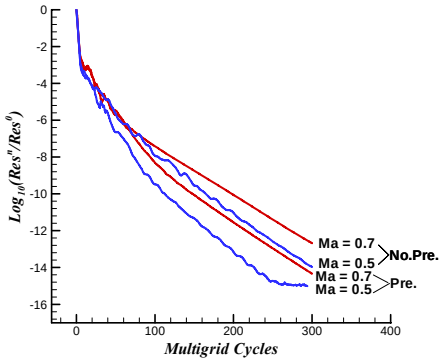

Fig.4 Effect of preconditioning on convergence rate for subsonic and transonic flows

Fig.5 and Fig.6 show the computed pressure distribution with and without preconditioning, respectively. The results are also compared with experimental data and the agreement is very well. One can also find out that the accuracy of solution for $Ma \le 0.1$ seems to be improved by preconditioning.

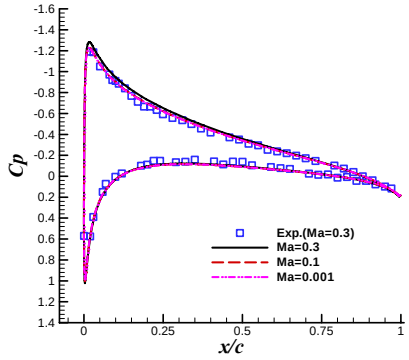

Fig.5 Comparison of computed pressure distribution and experimental data (with preconditioning)

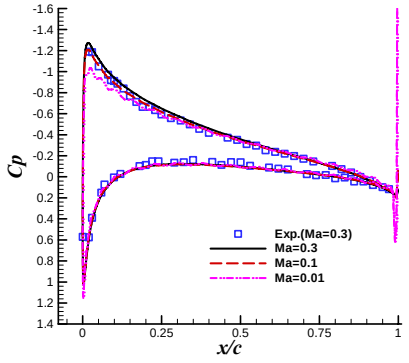

Fig.6 Comparison of computed pressure distribution and experimental data (without preconditioning)

The first example, corresponding to steady-state flows over NACA0012, shows that the present fully implicit preconditioning method can be used to effectively accelerate the convergence and improve the accuracy of simulating low speed, subsonic and transonic flows.

3.1.2 Flows over ONERA M6 Wing

Incompressible and transonic viscous flows over the ONERA M6 wing are employed as the second test case. The Reynolds number based on mean aerodynamic chord is 11.7 million, and the angle of attack is 3.06^0. A C-H Type grid (See Fig.7 and Fig.8) consists of $209\times49\times49$ points is used for these computations.

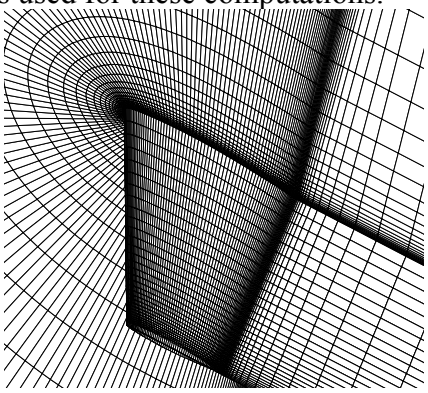

Fig.7 Schematics of C-H type grid for viscous flow over M6 wing

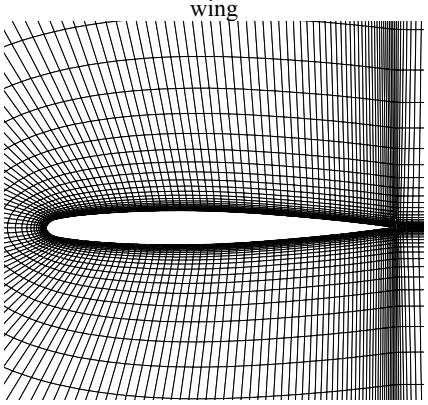

Fig.8 Schematics of C-type grid for root section of M6 wing

The effect of preconditioning for incompressible flows is indicated in Fig.9. It is shown that the convergence is dramatically improved. Fig.10 gives the comparison of pressure distributions with and without preconditioning at 80% span location. One can see that the pressure distribution near the trailing edge is more smooth, with preconditioning switching on .

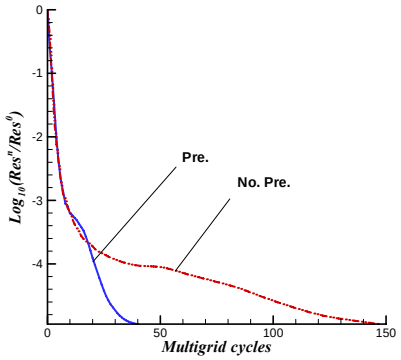

Fig.9 Effect of preconditioning on convergence rate

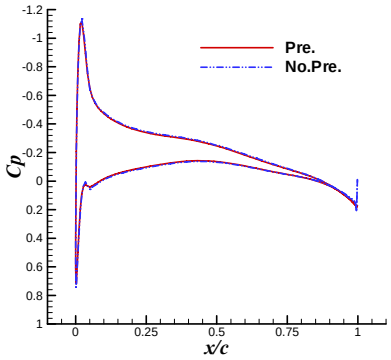

Fig.10 Comparison of pressure distributions for M6 wing with Ma=0.1 at 80% span location

Fig.11-13 shows the case for transonic flow simulation ($Ma = 0.84$). This is a well-known benchmark for transonic flow over wing. Fig.11 indicates the comparison of the convergence histories with and without preconditioning. The convergence rate is significantly improved by using the preconditioning method. Fig.12-13 demonstrates the comparison of pressure distribution at two typical span locations. The results computed by preconditioning are in good agreement with experimental data and that of the non-preconditioning, which validate the developed method for transonic flows.

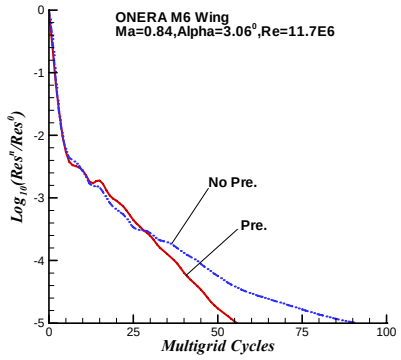

Fig.11 Effect of preconditioning on convergence rate for M6 wing at $Ma = 0.84$

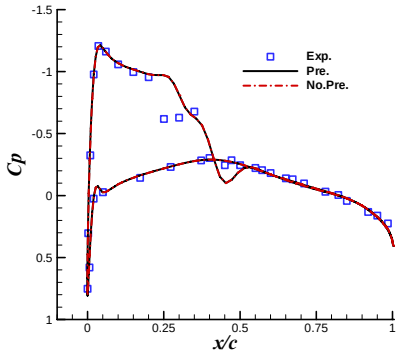

Fig.12 Comparison of pressure distributions at 80% span location for M6 wing at $Ma = 0.84$

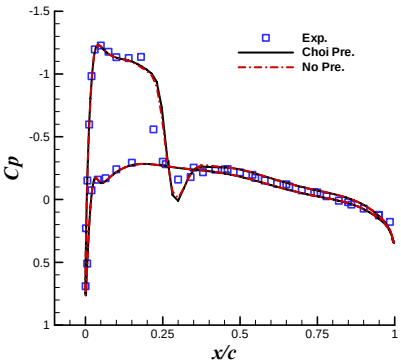

Fig.13 Comparison of pressure distributions at 95% span location for M6 wing at $Ma = 0.84$

The example, corresponds to flows over ONERA M6 wing, indicate that the present fully implicit method can be used to accelerate the convergence for simulating 3D the low-Mach number and transonic viscous flows.

3.2 Example of Unsteady-State Flows

The unsteady flows over a pitching NACA0012 airfoil are considered as the test case. A C-type grid (See Fig.1) consists of 265×65 points is used for these computations. The chord Reynolds number is taken as 5.5 million. The reduce frequency is difined as $k=\omega c/2V_\infty$(ω is angular frequency, c is the chord length of airfoil, and V_∞ is the speed of free stream)

The results for a low-Mach number case are illustrated in figures from 14 to 18. The airfoil is in oscillating with $\alpha=0.016^0+2.51^0\sin(\omega t)$ and $Ma = 0.01$. The reduce frequency equals to 0.0814. A fixed 5 sub-iterations is performed. The whole calculation is terminated at the end of 3 periods of physical time. The effect of preconditioning, shown in Fig.14, indicates that the convergence

rate of sub-iteration is dramatically improved. Comparison of convergence histories of lift, moment and drag coefficient with and without preconditioning is respectively shown in Fig.15，Fig16 and Fig.17. Dramatic difference between the results with and without preconditioning is indicated. With preconditioning, the full periodic solutions for lift, moment and drag coefficient are obtained within 3 periods In contrast, the solutions without preconditioning seem to be not periodic at the end of 3 periods. Furthermore, when taking insight into Fig.18, one can find out that the pressure distribution with preconditioning is qualitatively more reasonable and more correct than that without preconditioning. One can conclude, from the above discussion, that preconditioning can be used to improve both the efficiency and accuracy for low-Mach-number, unsteady flows

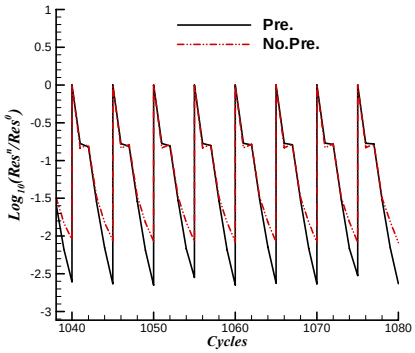

Fig.14 Effect of preconditioning on convergence rate of sub-iteration

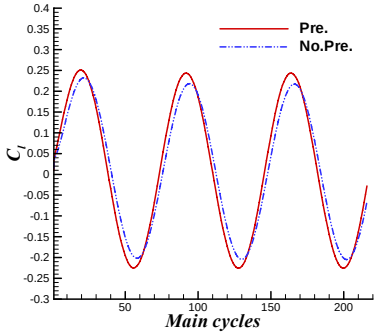

Fig.15 Comparison of convergence histories of lift coefficient with and without preconditioning

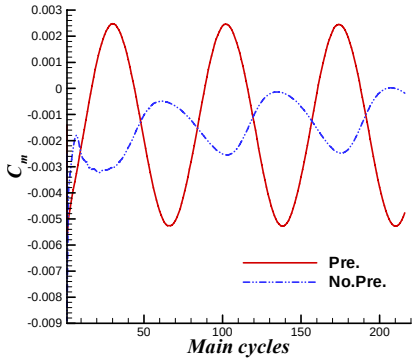

Fig.16 Comparison of convergence histories of lift coefficient with and without preconditioning

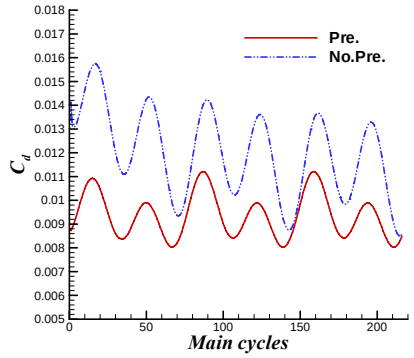

Fig.17 Comparison of convergence histories of drag coefficient with and without preconditioning

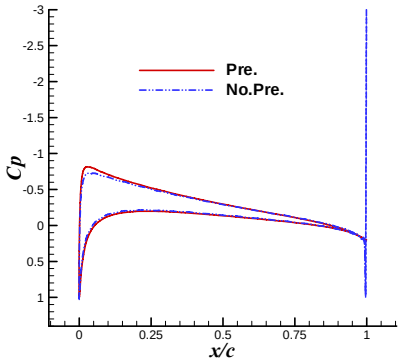

Fig.18 Comparison of pressure distribution with and without preconditioning ($\alpha = 2.526^0$)

The results for a transonic number case are illustrated in figures from 19 to 23. The airfoil is in oscillating with α=0.016^0+2.51^{0}sin(ωt), $Ma = 0.755$ and the reduce frequency equals to 0.0814. The convergence histories for lift, moment, and drag coefficient are receptively shown in Fig.19, Fig.20 and Fig.21. One can conclude that 2 sub-iterations are sufficient to obtain fully periodic solution within 3 periods. The convergence histories of lift ring and moment ring using 2 sub-iterations are illustrated in Fig.22 and Fig.23, respectively. The computed results using present method are

compared with experimental date and good agreement is achieved. The computation for 3 periods takes about 11 minutes of CPU time on a Pentium IV 2.4G computer, which indicates that the present method is very efficient.

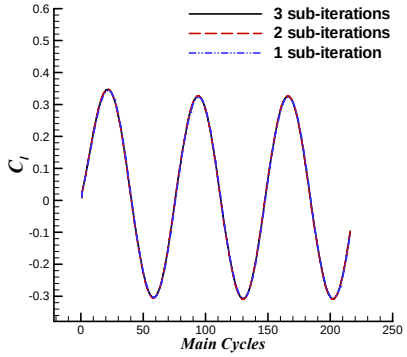

Fig.19 Convergence history of lift coefficient for different number of sub-iterations

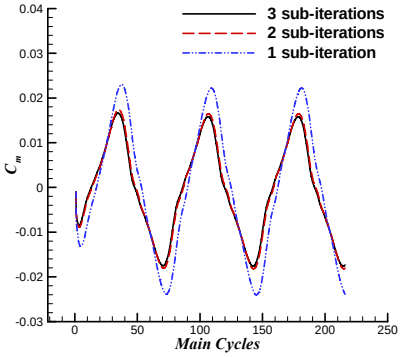

Fig.20 Convergence history of moment coefficient for different number of sub-iterations

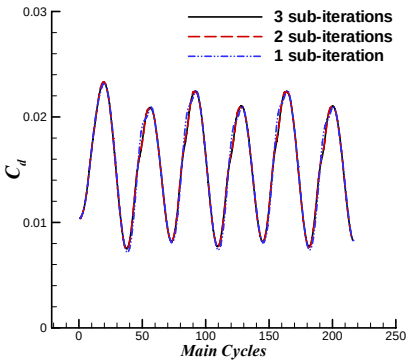

Fig.21 Convergence history of drag coefficient for different number of sub-iterations

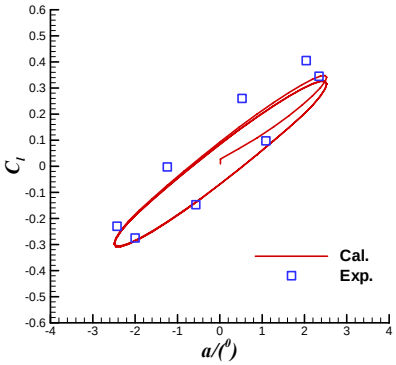

Fig.22. Convergence history of lift ring and comparison with experimental data

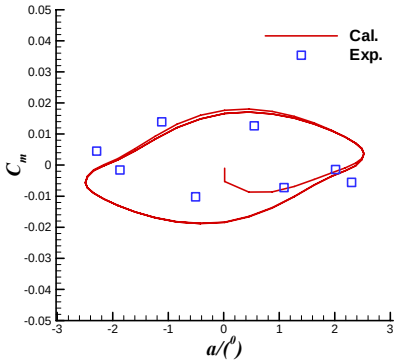

Fig.23. Convergence history of moment-ring and comparison with experimental data

The example, corresponding to flows over a pitching NACA0012, suggests that the present fully implicit preconditioning method is efficient and robust for simulation of unsteady flows with Mach number from low-speed to transonic regimes.

4 Conclusions

A time-accurate, fully implicit preconditioning method is developed for solving compressible Navier-Stokes equations. A Dual-time method with LU-SGS implicit sub-iteration is used for time-dependent flow problems. Low speed preconditioning is imposed on the pseudo-time derivatives of governing equations. The original LU-SGS method for non-preconditioning systems is revised according to preconditioned governing equations. At the convergence of sub-iterations, the effect of preconditioning is removed, and the second-order time accuracy is obtained. The present method is successfully applied to a variety of 2D and 3D, steady and

unsteady viscous flows. It is can be concluded that: 1) present method can be used to improve the convergence rate for steady-state flows from ultra low mach number to transonic regimes;2) The present method is validated to be efficient and robust for time-dependent simulations of low-speed and transonic unsteady-state flows

References

[1]Turkel E. Robust Low Speed Precondition for Viscous High Lift Flows.AIAA Paper 2002-0926, 2002.

[2] Rogers S E, Kwak D. An Upwind Differencing Scheme for the Incompressible Navier-Stokes Equations. Applied Numerical Mathematics, 1991,8:43-46

[3] Briley, W.R., McDonald, H. and Shamroth, S.J. "A Low Mach Number Euler Formulation and Application to Time-Iterative LBI Schemes",AIAA J., Vol.21, No.10,pp.1467-1469, 1983

[4] Choi， D. and Merkle, C.L."Application of Time-Iterative Schemes to Incompressible Flow", AIAA,J., Vol.23,pp.1516-1524,1985

[5] Merkle,C.L. and Choi Y. –H. "Computation of Low-Speed Flow with Heat Addition,"AIAA J. Vol.25,pp.831-839,1987

[6]Turkel, E., "Preconditioned Methods for solving the incompressible and Low Speed Compressible Equations, "Journal of Computational Physics, Vol.72, pp.277-298, 1987

[7]van Leer,B.,Lee, W.T., Roe, P.L. Characteristic Time-Stepping or Local Preconditioning of the Euler Equations.AIAA Paper 91-1552,1991

[8] Choi Y -H, Merkle C L. The Application of Precondition in Viscous Flows. Journal of Computational Phsics, 1993,105:207-223.

[9]Turkel, E. "A Review of Preconditioning Methods for Fluid Dynamics," Applied Numerical Mathematics, Vol.12, pp.257-284, 2993

[10]Turkel， E， Vatsa, V.N. and Radespiel, R. "Preconditioning Methods for Low Speed flow,"14th AIAA Applied Aerodynamics Conference, AIAA paper 96-2460, 1996

[11] Turkel， E， Preconditioning Techniques in Computation Fluid Dynamics, Annual Review of Fluid Mechanics,31, 385-416,1999

[12] Muradoglu， M. and Caughey, D.A. "Implicit Multigrid Solution of the Precondtitioned Muli-Dimensional Euler Equations," AIAA Paper 98-0114, 1998

[13] Allmaras, S. Analysis of a Local Matrix Preconditioner for 2-D Navier-Stokes Equantions, AIAA 11the Computational Fluid Dynamics Conference, 1995.

[14] Pierce,N.A, and Giles, M.B, Preconditioning Compressible Flow Calculations on Stretched Grid, AIAA Paper 1996.

[15] Vatsa, V.N. and Turkel E. "Choice of Variables and Preconditioning for Time Dependent Problems,"AIAA 16th Computational Fluid Dynamics Conference, Orlando, AIAA paper 2003-3692 CP, 2003.

[16] Vatsa, V.N. and Turkel E.Assessment of Local Preconditioners for Steady State and Time Dependent Flows 34th AIAA Fluid Dynamics Conference, Portland, OR AIAA paper 2004-2134, 2004

[17]Jameson, A.， Yoon, S. Multigrid Solutions of the Euler Equations using Implicit Schemes. AIAA Journal Vol. 24, pp. 1737–1743, 1986.

[18]Spalart P.R. Allmaras S.R. A One Equation Turbulence Model for Aerodynamic Flows. AIAA 92-0439, 1992

[19] Jameson A, Schmidt W, Turkel E. Numerical Solutions of the Euler Equations by a Finite Volume Method using Runge-Kutta Time Stepping Schemes. AIAA Paper 81-1259, 1981.

[20] Jameson, A.， Yoon, S. Multigrid Solutions of the Euler Equations using Implicit Schemes. AIAA Journal Vol. 24, pp. 1737–1743, 1986.

[21]Rizzi, A., Eriksson, L.E., Transfinite Mesh Generation and Damped Euler Equation Algorithm for transonic Flow Around Wing-Body Configuration.. AIAA Paper 81-0999, 1981.