Assessment of Preconditioning Methods

R. Radespiel [†], E. Turkel [††], N. Kroll [†]

[†] DLR, Institute of Design Aerodynamics
Braunschweig, Germany

[††] School of Mathematical Sciences
Sackler Faculty of Exact Sciences
Tel-Aviv University
Tel-Aviv, Israel

Abstract

We consider the steady state equations for a compressible fluid. For low speed flow the system is stiff since the ratio of the convective speed to the speed of sound is very small. To overcome this difficulty we alter the time evolution of the equations while retaining the same steady state analytic solution. In order to achieve high numerical resolution we also alter the artificial viscosity of the numerical scheme. It is convenient to do this by using other sets of variables in addition to the conservative variables. We examine the effects of different preconditioners, artificial dissipation and grid density on accuracy and convergence of the numerical sulutions. The numerical results obtained for inviscid and viscous 2D and 3D flows indicate that efficient multigrid computations of flows with very low Mach numbers are now possible.

Contents

1 Introduction

Over the past years several preconditioning methods have appeared with the aim to solve nearly incompressible flow problems with numerical algorithms designed for the compressible flows. The developments are motivated by several observations. First, there exist flow problems of mixed compressible/incompressible type. That is, part of the flow region is incompressible with locally low Mach numbers, whereas significant compressibility effects occur in other regions of the flow. A typical example in aerodynamics is the flow over a multielement airfoil at high angle of attack. Moreover, low speed flows may be compressible, because of surface heat transfer or volumetric heat addition. Finally, engineers prefer to use existing compressible flow codes over the broadest range of flow conditions possible in order to avoid to deal with multiple flow codes.

The difficulty in solving the compressible equations for low Mach numbers are associated with the large disparity of the acoustic wave speed, $u+a$, and the waves convected at the fluid speed, u. The application of preconditioning changes the eigenvalues of the system of compressible flow equations in order to remove this large disparity of wave speeds. For example, the time derivatives are premultiplied by a matrix which slows the speed of the acoustic waves down towards the fluid speed. A review of preconditioning methods for convection dominated flow problems has been published by Turkel [13]. Successful applications to low Reynolds number viscous flows are reported by Choi and Merkle [2].

There are two basic views of preconditioning methods. The first, which arose in applications to elliptic equations , views preconditioning merely as a way to accelerate the convergence of the iterative method. This is usually done on the matrix level. Thus, once convergence is achieved the solution should be identical to the original solution. The second approach, which we advocate for low speed flow is to view preconditioning as changing the differential equations to be solved. Thus, as a consequence of preconditioning the basic scheme can be changed, e.g. the artificial viscosity or upwinding. Similarly, the boundary conditions satisfied are based on the preconditioned equations rather than the physical equations. The justification for this is presented in [12] is that the "standard" numerical schemes for the compressible equations do not converge to a scheme for the incompressible equations (using a pseudo-compressibility approach) as the Mach number goes to zero. However, a proper preconditioning leads to a numerical scheme that does behave appropriately for low Mach numbers.

In this study we present a generalization of the preconditioners given by Turkel [13] [14] and also Choi-Merkle [2]. Also, the characteristic variables needed to compute flow conditions at inflow or outflow boundaries are derived. The accuracy of numerical solutions of the preconditioned flow equations is investigated next. We discuss nonconservative and conservative artificial dissipation models and the effects of the preconditioning matrix upon accuracy. The analysis is supported by numerical results for almost incompressible turbulent flow over a 2D airfoil. The influence of dissipation coefficients and the choice of preconditioning matrix upon grid convergence of the numerical solutions are investigated. Moreover, the first successful application of preconditioning for 3D viscous flows over a wing is demonstrated.

In conclusion, the present work shows that preconditioning can be combined with well-known convergence acceleration techniques such as residual smoothing and multigrid. Indeed, it is no surprise that the clustering of eigenvalues with preconditioning improves the damping of transient high-frequency modes to an extent, that efficient multigrid computations of flows with very low Mach numbers are now possible.

2 Algorithm I

We consider the preconditioned Euler equation written as

$$P^{-1}w_t + Aw_x + Bw_y = 0 \tag{1}$$

The form of the matrices A, B depend on the choice of the variables w . We shall first consider the variables $w = w_0 = (p, u, v, S)$ with $dS = dp - a^2 d\rho$. We then have

$$A_0 = \begin{pmatrix} u & \rho a^2 & 0 & 0 \\ \frac{1}{\rho} & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} \quad B_0 = \begin{pmatrix} v & 0 & \rho a^2 & 0 \\ 0 & v & 0 & 0 \\ \frac{1}{\rho} & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}$$

The preconditioners we shall consider contain a free parameter β. Unfortunately, the papers that have dealt with this subject all use different definitions of this free variable. Thus, for Choi and Merkle $\beta \sim a^2$ while for Turkel $\beta^2 \sim u^2 + v^2$ (based on Chorin's work on pseudo-compressibility) while for Van Leer $\beta = \sqrt{1 - M^2}$ (based on fluid dynamic notation). In this paper we shall concentrate on the Choi-Merkle and Turkel preconditioners. To avoid confusion we shall use the Choi-Merkle choice for β within this paper. In terms of these variables we consider the following preconditioner which is a generalization of previous preconditioners. For $\delta = 0$ we recover the preconditioner suggested by Turkel [13] [14] while for $\delta = 1, \alpha = 0$ we recover the preconditioner suggested by Choi and Merkle [2]. Note that δ is a new free parameter to generalize the Choi-Merkle and Turkel preconditioners it does not correspond to the δ in the Choi-Merkle paper.

$$P_0^{-1} = \begin{pmatrix} \frac{a^2}{\beta M_r^2} & 0 & 0 & \delta \\ \frac{\alpha u}{\rho \beta M_r^2} & 1 & 0 & 0 \\ \frac{\alpha v}{\rho \beta M_r^2} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P_0 = \begin{pmatrix} \frac{\beta M_r^2}{a^2} & 0 & 0 & -\frac{\beta M_r^2}{a^2}\delta \\ -\frac{\alpha u}{\rho a^2} & 1 & 0 & \frac{\alpha u}{\rho a^2}\delta \\ -\frac{\alpha v}{\rho a^2} & 0 & 1 & \frac{\alpha v}{\rho a^2}\delta \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Choi and Merkle chose

$$M_r^2 \quad = \quad \begin{cases} \epsilon, & M < \epsilon, \\ M^2, & \epsilon \leq M < 1 \\ 1 & M \geq 1; \end{cases} \qquad \beta = a \quad \text{(inviscid)}$$

Here we shall define βM_r^2 directly. The guiding principle is that near stagnation points this should be cutoff off and not allowed to go to zero. We also desire that for supersonic flow that preconditioner be turned off, i.e. for high speed flow we want that $\beta M_r^2 = a^2$, $\alpha = 0$, $\delta = 0$. One choice is $\beta M_r^2 = \min[\max\left(u^2 + v^2, K(u_\infty^2 + v_\infty^2)\right), a^2]$. This returns βM_r^2 to its non-preconditioned value at $M = 1$. One can also force this at a lower Mach number M_0 by choosing $\beta M_r^2 = \min[\max\left((u^2 + v^2)(1 + \frac{(1-M_0^2)}{M_0^4}M^2), K(u_\infty^2 + v_\infty^2)\right), a^2]$. This returns βM_r^2 to its non-preconditioned value at M_0. K seems to depend on the numbers of mesh nodes near the stagnation points. For viscous flow K needs to be chosen larger ($\sim .15$) because of the large number of points in the boundary layer (and also to account for the viscous time step). For duct flow one expects to again choose a very small K even for viscous flows since there usually are no stagnation points. K also depends on α [13] and may depend on the local cell Reynolds number [2]. We note that by using the variables $(\frac{dp}{\rho a}, du, dv, dS)$ the matrices A and B would be symmetric. However, since we do not presently consider symmetric preconditioners this is not of any consequence. Thus,

$$P_0 A_0 = \begin{pmatrix} \frac{\beta M_r^2 u}{a^2} & \rho \beta M_r^2 & 0 & \frac{-\beta M_r^2 u \delta}{a^2} \\ \frac{1}{\rho}\left(1 - \frac{\alpha u^2}{a^2}\right) & (1-\alpha)u & 0 & \frac{\alpha u^2 \delta}{\rho a^2} \\ \frac{-\alpha u v}{\rho a^2} & -\alpha v & u & \frac{\alpha u v \delta}{\rho a^2} \\ 0 & 0 & 0 & u \end{pmatrix} \tag{3}$$

The eigenvalues of P_0A_0 are

$$\lambda_0 = u$$

and

$$\lambda_\pm = 0.5\left(zu \pm \sqrt{z^2u^2 + 4\beta M_r^2(1-\frac{u^2}{a^2})}\right), \qquad z = 1-\alpha+\frac{\beta M_r^2}{a^2} \tag{4}$$

The right eigenvectors are given by the columns of

$$R^{-1} = \begin{pmatrix} \rho\beta M_r^2 & \rho\beta M_r^2 & 0 & 0 \\ \lambda_+ - \frac{\beta M_r^2}{a^2}u & \lambda_- - \frac{\beta M_r^2}{a^2}u & 0 & \frac{\delta u}{\rho a^2} \\ \frac{\alpha v\lambda_+}{u-\lambda_+} & \frac{\alpha v\lambda_-}{u-\lambda_-} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

We note that for $\delta = 0$ and $\alpha = 0$ the two eigenvectors for the convective speeds are orthogonal to each other and to the acoustic eigenvectors. Of course, orthogonality of eigenvectors depends on the basis chosen, i.e. on the variables used in constructing P_0 and A_0. Furthermore, as the Mach number goes to zero the determiant of R^{-1} depends only on βM_r^2. Thus, the condition number of the transformation is well behaved as long as βM_r^2 is bounded away from zero as the Mach number approaches zero. see [4] for the importance of orthogonal eigenvectors.

For generalized coordinates we replace this by

$$U = us_x + vs_y \qquad |S_\xi|^2 = s_x^2 + s_y^2$$

$$\lambda_\pm = 0.5\left(zU \pm \sqrt{z^2U^2 + 4\beta M_r^2(|S_I|^2 - \frac{U^2}{a^2})}\right)$$

where $S_\xi = (s_x, s_y)$ is the cell face vector in ξ-direction. The largest eigenvalue is then used to determine the (inviscid) time step. We note that these eigenvalues are independent of δ. For M ~ 0, $\lambda_\pm \sim 0.5u\left((1-\alpha) \pm \sqrt{(1-\alpha)^2+4}\right)$. So for $\alpha = 0$ the condition number is $\frac{1+\sqrt{5}}{1-\sqrt{5}} \sim 2.6$ while for $\alpha = 1$ the condition number is 1. We shall also express these matrices in terms of the variables $w_4 = (p, u, v, T)$. Let

$$g = 1 + (\gamma-1)\delta \quad = \quad \begin{cases} 1, & \text{if} \quad \delta = 0 \\ \gamma, & \text{if} \quad \delta = 1 \end{cases}$$

Then,

$$A_4 = \begin{pmatrix} u & \rho a^2 & 0 & 0 \\ \frac{1}{\rho} & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & (\gamma-1)T & 0 & u \end{pmatrix} \qquad B_4 = \begin{pmatrix} v & 0 & \rho a^2 & 0 \\ 0 & v & 0 & 0 \\ \frac{1}{\rho} & 0 & v & 0 \\ 0 & 0 & (\gamma-1)T & v \end{pmatrix}$$

Let $m^2 = \frac{\beta M^2}{a^2}$, then

$$P_4^{-1} = \begin{pmatrix} \frac{1}{m^2} - (\gamma-1)\delta & 0 & 0 & \frac{\gamma p}{T}\delta \\ \frac{\alpha u}{\rho\beta M^2} & 1 & 0 & 0 \\ \frac{\alpha v}{\rho\beta M^2} & 0 & 1 & 0 \\ \frac{1}{\rho c_p}\left(\frac{1}{m^2} - g\right) & 0 & 0 & g \end{pmatrix} \qquad P_4 = \begin{pmatrix} m^2 g & 0 & 0 & -m^2\frac{\gamma p}{T}\delta \\ -\frac{\alpha u g}{\rho a^2} & 1 & 0 & \frac{\alpha u}{\rho a^2}\frac{\gamma p}{T}\delta \\ -\frac{\alpha v g}{\rho a^2} & 0 & 1 & \frac{\alpha v}{\rho a^2}\frac{\gamma p}{T}\delta \\ \frac{1}{\rho c_p}(m^2 g - 1) & 0 & 0 & 1-(\gamma-1)m^2\delta \end{pmatrix} \tag{6}$$

The last equation in this system is

$$\frac{(\gamma-1)T}{\gamma p}\left[\frac{a^2}{\beta M^2}-g\right]p_t+gT_t+(\gamma-1)T(u_x+v_{y)}+uT_x+vT_y=0$$

Combining this with the first equation for p_t we get $\rho c_p(T_t+uT_x+vT_y)=p_t+up_x+vp_y$ or equivalently $S_t+uS_x+vS_y=0$. In conservation form we have

$$\begin{gathered}
\left[\tfrac{1}{\beta M^2}-\tfrac{(\delta-1)(\gamma-1)}{a^2}\right]p_t+(\delta-1)\tfrac{\rho}{T}T_t+(\rho u)_x+(\rho v)_y=0\\
\left[\tfrac{\alpha+1}{\beta M^2}-\tfrac{(\delta-1)(\gamma-1)}{a^2}\right]up_t+(\delta-1)\tfrac{\rho u}{T}T_t+\rho u_t+(\rho u^2+p)_x+(\rho uv)_y=0\\
\left[\tfrac{\alpha+1}{\beta M^2}-\tfrac{(\delta-1)(\gamma-1)}{a^2}\right]vp_t+(\delta-1)\tfrac{\rho v}{T}T_t+\rho v_t+(\rho uv)_x+(\rho v^2+p)_y=0\\
\left[\tfrac{H+\alpha(u^2+v^2)}{\beta M^2}-\tfrac{(\delta-1)(\gamma-1)H}{a^2}-1\right]p_t+\left[\rho c_p+(\delta-1)\tfrac{\rho H}{T}\right]T_t+\\
\rho uu_t+\rho vv_t+(\rho uH)_x+(\rho vH)_y=0
\end{gathered}$$

The algorithm of Choi-Merkle is equivalent to $\delta=1$. We now rewrite the characteristic variables in the x direction in terms of T rather than S.

$$\begin{gathered}
dR_1=dv\Leftrightarrow\lambda=u\\
dR_2=-(\gamma-1)dp+\frac{\gamma p}{T}dT\Leftrightarrow\lambda=u\\
dR_\pm=[\lambda_\pm-u(1-\alpha)+\frac{\delta(\gamma-1)\beta M^2u}{a^2}]dp+\rho\beta M^2du-\delta\frac{\gamma p\beta M^2u}{a^2T}dT\Leftrightarrow\lambda=u\pm\lambda_\pm
\end{gathered}$$

3 Algorithm II

In [12], [5] an alternate way of preconditioning the fluid dynamics system was introduced. In this approach we do not update the conservation variables, rather we update

$$W=\begin{pmatrix}p'\\ \rho u\\ \rho v\\ E'\end{pmatrix}$$

$$F=\begin{pmatrix}\rho u\\ \rho u^2+p'\\ \rho uv\\ \rho H'u\end{pmatrix},\quad G=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p'\\ \rho H'v\end{pmatrix},\quad Q=\begin{pmatrix}p'\\ \rho u\\ \rho v\\ H'\end{pmatrix}$$

$$\frac{\partial W}{\partial t}=\mathbf{P}\left[\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}\right]$$

where $p'=p-p_\infty$, $E'=c_p\rho(T-T_\infty)-(p-p_\infty)+\frac{\rho(u^2+v^2)}{2}=E+p_\infty-\frac{\gamma p_\infty}{(\gamma-1)\rho_\infty}\rho$ and $\rho H'=E'+p'$. Then

$$\mathbf{P}=\mathbf{I}\quad+\quad\Delta\cdot\begin{pmatrix}1-\frac{1}{\Delta} & -\frac{u}{G+h_\infty} & -\frac{v}{G+h_\infty} & \frac{1}{G+h_\infty}\\ -B_2 & \frac{uB_2}{G+h_\infty} & \frac{vB_2}{G+h_\infty} & -\frac{B_2}{G+h_\infty}\\ -B_3 & \frac{uB_3}{G+h_\infty} & \frac{vB_3}{G+h_\infty} & -\frac{B_3}{G+h_\infty}\\ -B_4 & \frac{uB_4}{G+h_\infty} & \frac{vB_4}{G+h_\infty} & -\frac{B_4}{G+h_\infty}\end{pmatrix}$$

where β was previously defined and $h = c_pT$, $G = \frac{u^2+v^2}{2}$, $\Delta = \frac{(G+h_\infty)\beta M_r^2}{h}$ and $\alpha = 1$,

$$
\begin{aligned}
B_1 &= \frac{1}{\beta M_r^2 2} - \frac{1}{(\gamma-1)h} \\
B_2 &= (B_1 + \frac{\alpha}{\beta M_r^2})u \\
B_3 &= (B_1 + \frac{\alpha}{\beta M_r^2})v \\
B_4 &= B_1 H' + \frac{\alpha(u^2+v^2)}{\beta M_r^2}
\end{aligned}
$$

We can evaluate this efficiently by defining $S = \Delta \cdot \left(d\rho - \frac{ud(\rho u)+vd(\rho v)-dE'}{G+h_\infty}\right) = \frac{\beta M_r^2 dp}{(\gamma-1)h}$. Then

$$
\begin{aligned}
dp_{new} &= S \\
d(\rho u)_{new} &= d(\rho u)_{orig} + B_2 S \\
d(\rho v)_{new} &= d(\rho v)_{orig} + B_3 S \\
d(E')_{new} &= d(E')_{orig} + B_4 S
\end{aligned}
$$

We note, from the appendix, that the transformation of w_2 with respect to w_0 is singular at stagnation points while the transformation of w_3 is not. Hence, it is important to use E' and not E.

With this approach all operations (e.g. residual smoothing, multigrid etc.) are done on the new variables and not the conservative variables. The rationale is that since the pressure is more appropriate than density at lower speeds we should interpolate the pressure between grids rather than the density. Similarly, since the density is almost constant it is more beneficial to smooth the residual of the pressure rather than the residual of the density.

4 Artificial Viscosity

For a central difference scheme it is necessary to add artificial dissipative terms to the central difference approximation of the spatial derivatives. One usually adds a nonlinear second difference to control oscillations near shocks and a linear fourth difference to damp high frequency oscillations. We are interested in the functional form of these differences. Hence, our examples will be with a second-difference artificial dissipation. Extensions to fourth differences and nonlinearities are straightforward. Similarly, for one sided schemes the central difference plus artificial dissipation is replaced by a Roe matrix formulation.

A typical preconditioned finite difference schemes can be expressed as

$$
\Delta Q_c = \Delta t P \left[\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}\right] = \Delta t P \left[A\frac{\partial Q_c}{\partial x} + B\frac{\partial Q_c}{\partial y}\right] \tag{7}
$$

where Q_c represents the conservative variables and the partial spatial derivatives are replaced by central difference approximations, either cell centered or at nodal points. We shall consider both conservative and non-conservative ways of adding artificial dissipative terms. The dissipation need not be expressed in terms of the conservative variables Q_c. Using another set of variables Q_V we have $\Delta Q_c = \frac{\Delta Q_c}{\Delta Q_V}\Delta Q_V$. Using different variables we can introduce the appropriate preconditioner in two ways. Either we can completely change variables in all places and the the matrices A and B in (7) change as well as the preconditioner P. The new precondioner P_i is then a function of P_0 (2) given by $P_i = \frac{\partial Q_i}{\partial Q_0} P_0 \frac{\partial Q_0}{\partial Q_i}$. Various choices for this are presented in the appendix. An alternative is to leave the spatial derivatives in conservation form and to transform only the time derivatives. Almost following Choi and Merkle's notation we denote this preconditioner by Γ (our Γ is the inverse of that used by Choi and Merkle). Thus, in conservative variables we have $\Gamma_c = P_c$. We choose as our "basic" form that given in (p, u, v, S) coordinates (i.e. P_0). Let Q_c be the conservative variables then for any other set of variables we have $\Gamma_i = \frac{\partial Q_i}{\partial Q_c} P_c = P_i \frac{\partial Q_i}{\partial Q_c} = \frac{\partial Q_i}{\partial Q_0} P_0 \frac{\partial Q_0}{\partial Q_c}$.

We therefore replace (7) by

$$\Delta Q_V = \Delta t \left\{ \Gamma_V \left[\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} \right] + \left(\mid \sigma(P_V A_V) \mid \frac{\partial Q_V}{\partial x} \right)_x + \left(\mid \sigma(P_V B_V) \mid \frac{\partial Q_V}{\partial y} \right)_y \right\} \tag{8}$$

A_V, B_V are the Jacobian matrices of F and G with respect to the V variables and σ is matrix function. For example, if σ is proportional to the spectral radius then we have a scalar artificial viscosity, in Q_V variables, while if $\sigma(A) \sim A$ we have a matrix-valued artificial viscosity. Once we have found ΔQ_V we can either update this to find Q_V or else we can convert back to find ΔQ_c. Thus, for example, residual smoothing and multigrid can be done on either the Q_c variables or the Q_V variables.

The formula (8) is easy to implement . For a scalar viscosity (i.e. σ is a scalar function) the viscosity remains scalar after preconditioning. The main difference is that now differences of the Q_V variables are added to each of the conservation variables rather than differences of the conserved variables. We note that in the original paper of Jameson, Schmidt and Turkel[7] differences of ρH rather than E were added to the energy equation. However, the formulation (8) is nonconservative. Since we shall also be interested in transonic flows with shocks we present the addition of the artificial dissipation in a conservative formulation. Furthermore, Merkle (private communication) indicates that for internal flow the conservative form is important even for subsonic flows without shocks in order to preserve global quantities, e.g. total mass flow. Dropping the subscript V :

$$\Delta Q = \Delta t \Gamma \left\{ \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \left(\Gamma^{-1} \mid \sigma(PA) \mid \frac{\partial Q}{\partial x} \right)_x + \left(\Gamma^{-1} \mid \sigma(PB) \mid \frac{\partial Q}{\partial y} \right)_y \right\} \tag{9}$$

$A = \frac{\partial F}{\partial Q}$, see (6). Even if σ is a scalar function we now have to evaluate matrix-vector product and so the numerical effort corresponds to a matrix-valued viscosity. An alternate way to derive a suitable artificial viscosity model is to replace the time derivative of the conservative quantities by time derivatives of different quantities in the partial differential equation. We therefore replace (1) by

$$\frac{\partial \psi}{\partial t} = AQ_x + BQ_y \quad \text{or} \quad \frac{\partial \psi}{\partial t} = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}$$

If we assume that $d\psi = R^{-1} dQ$ and we add an artificial viscosity we can replace this by

$$R^{-1} \frac{\partial Q}{\partial t} = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \left(V_x R^{-1} \frac{\partial Q}{\partial x} \right)_x + \left(V_y R^{-1} \frac{\partial Q}{\partial y} \right)_y \tag{10}$$

where $V_x = \mid \sigma(\frac{\partial F}{\partial \psi}) \mid = \mid \sigma(\frac{\partial F}{\partial Q} \frac{\partial Q}{\partial \psi}) \mid = \mid \sigma(AR) \mid$. Comparing (9) with (10) the only difference is in the order of the matrices in the artificial viscosity. In our original formulation the preconditioning premultiplies A while in the second formulation the preconditioning postmultiplies A (and similarly in the y direction).

In the present work we present numerical results which are obtained for $\sigma(PA)$ and $\sigma(PB)$ defined as scalar functions. Following previous work on multigrid schemes for the Navier-Stokes equations these functions take into account the ratio of the spectral radii in the different coordinate directions, that is

$$\sigma(PA) = k^{(4)} \tilde{\sigma}(PA) \left(1 + \left(\frac{\tilde{\sigma}(PB)}{\tilde{\sigma}(PA)} \right)^{\zeta} \right) \tag{11}$$

and

$$\sigma(PB) = k^{(4)}\tilde{\sigma}(PB)\left(1 + \left(\frac{\tilde{\sigma}(PA)}{\tilde{\sigma}(PB)}\right)^{\zeta}\right)$$

where $\tilde{\sigma}$ denotes the spectral radius and $k^{(4)}$ is the artificial viscosity coefficient corresponding to fourth-difference terms. These scaling functions yield sufficient artificial dissipation for general curvilinear grids with cells of high aspect ratio.

In the previous section we introduced a preconditioner that depended on parameters α and δ. δ does not affect the eigenvalues of PA and so does not affect a scalar artificial viscosity. $\alpha = 1$ reduces the largest eigenvalue compared with $\alpha = 0$. This improves the condition number but also decreases the artificial viscosity. We thus expect $\alpha = 1$ to converge slower than $\alpha = 0$ in spite of the fact that we have reduced the condition number. However, we expect $\alpha = 1$ to be more accurate. For a matrix viscosity (or a Roe matrix) we expect similar but a less pronounced dependence on α.

5 Boundary Conditions

In many codes the boundary conditions in the far field are based on characteristic variables, even for viscous flow. Thus at inflow the incoming variables corresponding to positive eigenvalues are specified while the outgoing variables corresponding to negative eigenvalues are extrapolated. Once we change the time dependent equations we also change the characteristics of the system (though not the signs of the eigenvalues). Hence, it is necessary to also modify the boundary conditions for the preconditioned system.

Since the eigenvalues and eigenvectors are invariant under a similarity transform we shall use the (p, u, v, S) set of variables to determine the new eigenvalues and characteristic variables. We shall only consider the flow normal to the boundary which we take to be a line $x = const$. Thus, we need to use the characteristic variables we found above corresponding to the left eigenvectors of P_0A_0 (3). Flow variables at a boundary are then obtained by taking the values for R_1, R_2 and R_+ from upstream flow whereas downstream values are used for R_-. When we diagonalize $P_0A_0 = RDR^{-1}$ the rows of R are the left eigenvectors of P_0A_0 while the columns of R^{-1} are the right eigenvectors of P_0A_0. Explicitly we have:

$$R = \begin{pmatrix} \lambda_+ - (1-\alpha)u & \rho\beta M^2 & 0 & -\frac{\delta\beta M^2 u}{a^2} \\ \lambda_- - (1-\alpha)u & \rho\beta M^2 & 0 & -\frac{\delta\beta M^2 u}{a^2} \\ \frac{\alpha v}{\rho(\beta M^2 - \alpha u^2)} & \frac{\alpha u v}{\beta M^2 - \alpha u^2} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

$$R^{-1} = \begin{pmatrix} K_+(\lambda_+ - u) & K_-(\lambda_- - u) & 0 & 0 \\ \frac{K_+(\beta M^2 - \alpha\lambda_+ u)}{\rho\beta M^2} & \frac{K_-(\beta M^2 - \alpha\lambda_- u)}{\rho\beta M^2} & 0 & \frac{\delta u}{\rho a^2} \\ -\frac{K_+\lambda_+\alpha v}{\rho\beta M^2} & -\frac{K_-\lambda_-\alpha v}{\rho\beta M^2} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\lambda_+ - \lambda_-} & \frac{-1}{\lambda_+ - \lambda_-} & 0 & 0 \\ -\frac{\lambda_- - (1-\alpha)u}{\rho\beta M_r^2(\lambda_+ - \lambda_-)} & \frac{\lambda_+ - (1-\alpha)u}{\rho\beta M_r^2(\lambda_+ - \lambda_-)} & 0 & \frac{\delta u}{\rho a^2} \\ -\frac{\lambda_+\alpha v}{\rho\beta M_r^2(\lambda_+ - u)(\lambda_+ - \lambda_-)} & \frac{\lambda_-\alpha v}{\rho\beta M_r^2(\lambda_+ - u)(\lambda_+ - \lambda_-)} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $K_{\pm} = \frac{\pm 1}{(\lambda_+ - \lambda_-)(\lambda_{\pm} - u)}$, (see also (5)) . The corresponding characteristic variables given in terms of differentials are:

$$dR_1 = \frac{\alpha v}{\beta M^2 - \alpha u^2}\left[\frac{dp}{\rho} + u du\right] + dv \Leftrightarrow \lambda = u$$

$$dR_2 = dS \Leftrightarrow \lambda = u$$

$$dR_{\pm} = [\lambda_{\pm} - u(1-\alpha)]dp + \rho\beta M^2 du - \frac{\delta\beta M^2 u}{a^2} dS \Leftrightarrow \lambda = u \pm \lambda_{\pm}$$

We note that the terms involving δ also contain M^2 terms. Hence, for low Mach numbers these terms are negligible even for the Choi-Merkle preconditioner.

Let variables with a subscript "b" denote the variables to be computed at the boundary. Variables with a subscript "a" denote the flow state approaching the boundary, i.e. specified at inflow, whereas variables with a subscript "l" are used for the flow leaving the boundary, e.g. extrapolated at inflow. Note, that this notation can be consistently used for inflow and outflow boundaries. The characteristic boundary conditions are expressed in (p, u, v, T) variables as:

$$p_b = \frac{\lambda_+ p_a - \lambda_- p_l}{\lambda_+ - \lambda_-} - \frac{u(1 - \alpha - \delta\frac{(\gamma-1)\beta M_r^2}{a^2})(p_a - p_l) - \rho\beta M_r^2(u_a - u_l) + \frac{\delta\rho\beta M_r^2 u}{T}(T_a - T_l)}{\lambda_+ - \lambda_-}$$

$$p_b = p_l + \frac{\lambda_+(p_a - p_l)}{\lambda_+ - \lambda_-} - \frac{u(1 - \alpha - \delta\frac{(\gamma-1)\beta M_r^2}{a^2})(p_a - p_l) - \rho\beta M_r^2(u_a - u_l) + \frac{\delta\rho\beta M_r^2 u}{T}(T_a - T_l)}{\lambda_+ - \lambda_-}$$

$$u_b = u_a - \frac{\lambda_+ - u(1-\alpha)}{\rho\beta M^2}(p_b - p_a) \tag{13}$$

$$T_b = T_a + \frac{(\gamma - 1)T}{\gamma p}(p_b - p_a)$$

$$v_b = v_a - \frac{\alpha v}{\beta M^2 - \alpha u^2}\left[\frac{p_b - p_a}{\rho} + u(u_b - u_a)\right]$$

In generalized coordinates u and v are replaced by the normal and tangential components of the velocity relative to the boundary. During our computations which are reported below we have observed that certain sets of simplified boundary conditions yield almost identical results in terms of convergence rates and steady state solutions. One of these is:

$$v_b = v_a, \quad p_b = p_l, \quad u_b = u_a, \qquad \textit{either} \quad T_b = T_a \quad \textit{or} \quad \rho_b = \rho_a + \frac{p_b - p_a}{a^2} \tag{14}$$

The simplified boundary condition has been used in all of the computations below if not otherwise mentioned. One should be cautious, however, to generalize this experience with the boundary conditions because in all of our computations we used boundaries which were located far away from the aerodynamic surface.

6 Computational Results

The preconditioning approach described above is studied numerically for inviscid and viscous flows over airfoils and wings with freestream conditions corresponding to incompressible and compressible flow conditions.

6.1 Flows over 2D Airfoils

Two codes were used to simulate the airfoil flows in the present work. Both are central-difference finite volume codes with an addition of artificial dissipative terms. A multi-stage Runge-Kutta scheme is used to advance the solution in time. Local time steps, residual smoothing and multigrid are used to accelerate the convergence to a steady state. In one code the unknown conservative variables are stored at the nodes of the stencil [9]. In this code a scalar artificial dissipation is used based on fourth differences of (p,u,v,T) in the four conservation equations: continuity, momentum and energy respectively. In the second code the variables are located at the cell centers. This is the basic code used in [11]. For the preconditioning the variables are changed from the conservative variables to $W_p = (p', \rho u, \rho v, E')^t$. These are used for all operations including implicit residual smoothing and multigrid transfers between grids.

Both codes use an artificial viscosity based on a blend of second and fourth differences. Since we are not interested in shocks we set the coefficient of the second difference, $k^{(2)}$, to zero. The fourth difference is still needed to prevent the occurrence of odd-even oscillations. Its coefficient, $k^{(4)}$, is given for each case. We denote the first code with variables at the nodes and updating (p, u, v, T) as Code1 and the second code with variables at the center of the cells and using $(p', \rho u, \rho v, E')$ variables as Code2. Both codes used the same five stage Runge-Kutta formula coupled with an implicit residual smoothing. Code1 used CFL=7 (based on the preconditioned time step). The residual smoothing needs the CFL of the unsmoothed scheme and this was taken as 3.5. These are the same values typically chosen for the non-preconditioned scheme. Code2 required a lower CFL and a significantly lower value for use in the residual smoothing. It is unclear why there is a need for additional residual smoothing. Both codes use a multigrid acceleration with a fixed sawtooth and W cycle with one smoothing on the finest mesh and two smoothings on all coarser meshes.

The first numerical test case is inviscid flow over NACA 0012 airfoil. In figure 1 we present the grid used for the calculations. The grid has cells clustered near the leading and trailing edges in order to allow accurate drag computations. The Mach number contours for $M_\infty = 0.01, \alpha = 2^\circ$ are displayed in figure 2. Grid convergence of the total force coefficients is displayed in figure 3 for Code1 and in figure 4 for Code2. The results are plotted versus $1/N$ where N is the total number of grid cells. As a reference we have included the results for subsonic compressible flow at $M_\infty = 0.63$ which was computed without preconditioning. Grid convergence for the drag, which should be zero for inviscid flow without shocks, is quite sensitive to the details of the implementation of the wall slip boundary condition, and also to the amount of artificial viscosity used. For example, the pressure drag on the coarse 56x10 grid may even change sign when changing the input value of $k^{(4)}$ from 1/32 to 1/64. A comparison of figures 3 and 4 highlights the sensitivities observed during our numerical computations.

The corresponding convergence behavior of the inviscid flow computations towards the steady state is displayed in figures 5 and 6. A 4-level multigrid was used. Convergence is monitored by plotting the L_2-norm of the residual of the continuity equation computed on the finest mesh. It is seen that both codes converge at about the same convergence rate for both the incompressible and compressible flow cases.

We next consider viscous flow about a RAE 2822 airfoil. The flow conditions chosen are $M_\infty = 0.01, \alpha = 1.89^\circ$, Re $= 5.7$ million. We use a Baldwin-Lomax turbulence model with transition at 11% of the chord. In figure 7 we present the grid used for the calculations. This grid was constructed for subsonic problems and clusters the cells near the leading and trailing edges. The farfield boundary is located about 20 chord lengths away from the airfoil. The numerical flow solution is presented in figure 8 for Code1 and in figure 9 for Code2. We observe very good agreement between both codes in the representation of the flow details. The pressure distribution is typical for subsonic flow over a supercritical airfoil. The skin friction displays the effect of fixed transition at $x/c = 0.11$. Note, that pressure oscillations near the trailing edge, for which compressible flow solvers are notorious, are absent in the numerical result due to the correct scaling of artificial viscosity.

The numerical accuracy of the solution is investigated next by changing the grid density. Figures 10-14 show the variation of total force coefficients under grid refinement. Figure 10 demonstrates that the pressure drag is the most sensitive to numerical errors. Both Code1 and Code2 yield comparable grid convergence. The discrepancies in pressure drag between Code1 and Code2 are most likely due to the numerical treatment of the farfield boundary. Differences can occur because the farfield conditions, given in section 5, are prescribed at the grid points of the computational boundary with Code1 whereas Code2 prescibes farfield conditions at the ghost points located outside of the domain. Some differences in pressure drag are also observed when comparing the results for the characteristic condition (13) and the simplification (14) as shown in figure 11. We note that the freestream state in all of the computations is defined using a point vortex to represent the aerodynamic effect of the airfoil. Figure 11 displays significant discrepancies when the solution is obtained without the vortex representation. The effects of dissipation coefficients, ζ (see 11) and $k^{(4)}$, are shown in figure 12. As expected, the numerical resolution is improved with smaller dissipation coefficients, $k^{(4)}$. The effect of ζ is less pronounced, however, $\zeta = 0$ is more accurate. Figure 13-14 show the influence of the preconditioning parameter, α, for Code1 and Code2. The choice $\alpha = 1$ yields a smaller artificial viscosity for low Mach numbers and hence, it is more accurate. Comparisons for nonconservative and conservative dissipation formulations yield very good agreement for the incompressible flow conditions (not shown here).

The residual convergence is presented in Figures 15-18. Now a 5-level multigrid was used because of the finer mesh. Figure 15 displays a comparison for the two flow solvers used in the present study where Code2 converges somewhat faster. The effect of increased dissipation levels is to improve convergence rate, according to figure 16. Combining the results of figures 11 and 14 one finds that $k^{(4)} = 1/32, \zeta = 2/3$ is a good compromise between accuracy and convergence. Figure 17 shows that the convergence of Code1 is worse for $\alpha = 1$ compared with $\alpha = 0$. The coefficient of the artificial viscosity is proportional to the largest eigenvalue. Hence, although $\alpha = 1$ is optimal in terms of equalizing the wave speeds (and so in terms of the time step) it reduces the amount of dissipation. Thus, the code with $\alpha = 1$ converges more slowly than with $\alpha = 0$. Also, the Courant number used for $\alpha = 1$ had to be reduced. Similar results are obtained with Code2, according to figure 18. Again, Code2 converged faster than Code1. Recall that the algorithmic differences between Code1 and Code2 concern the flow variables for which residuals are computed and the artificial dissipation model. Moreover, numerical implementations of boundary conditions and the multigrid transfer operartors differ because Code1 stores the flow variables at the grid nodes whereas Code2 is cell centered. Also Code2 transfers the variables W_P between the grids rather than the conservative variables used by Code1.

We next consider the same airfoil but with $M_\infty = 0.30, \alpha = 2,79°$, Re = 6.5 million. Hence, we are in the lower subsonic range but not at very extremely low Mach numbers. The conergence plot for this is given in figure 19 comparing the preconditioning of code2 to the nonpreconditioned case. We see that even in this Mach number range the preconditioning helps considerably. In figure 20 we present the convergence of the lift and drag which show the improved accuracy of the preconditioned code even at this Mach number.

Finally, we present some numerical results for a well-known viscous transonic flow case in figures 21 and 22. The flow features a strong shock on the upper surface with local flow separations at the foot of the shock and near the trailing edge. Based on the grid convergence study presented in figure 22 and similar published results [10] we find that the accuracy of the preconditioned Code1 with conservative dissipation formulation surpasses the results given without preconditioning. On the other hand, the preconditioned code with the nonconservative dissipation formulation yields a shift in the shock position and therefore, false values of lift and pressure drag are obtained.

6.2 Flow over 3D Wing

The three-dimensional calculations have been carried out with a cell-vertex method [8] which corresponds closely to Code1 used for the 2D flows. The method employs central differencing with artificial viscosity, multi-stage time stepping, residual smoothing and multigrid. The present cases were run with a five stage scheme, a Courant number of 7.5 and 3-level full multigrid with W-cycling. Preconditioning is obtained with the Choi-Merkle matrix, that is $\delta = 1, \alpha = 0$ and the dissipation is formulated according to eq.(9).

We consider the incompressible viscous flow over the ONERA-M6 wing at an angle of attack of $\alpha = 2°$, Re=11million. The effect of turbulence is modelled according to the Baldwin and Lomax model. The grid used for the calculations has C-H topology and consists of 224x48x32 cells, these are 224 cells in streamwise, 48 cells in normal and 32 cells in spanwise directions. Figure 23 shows the grid in the plane of symmetry. The residual convergence for computations at $M_\infty = 0.10, 0.01, 0.001$ are presented in figure 24. It is shown that the rate of convergence is independent of the free stream Mach number. Without preconditioning, on the other hand, no convergence could be obtained for Mach numbers less than 0.1. Figure 25 compares the surface pressure distribution at spanwise section $\eta = 0.8$. There is no effect of the freestream Mach number visible. Mach number independence is further confirmed by figure 26 where the total lift and pressure drag coefficients are plotted versus the Mach number. These values are constant if the Mach number approaches zero. The numerical accuracy of the incompressible solution is illustrated in figure 27. Here, the variation of total force coefficients under grid refinement is shown. As for the 2D cases pressure drag is more sensitive than lift. Note, that the convergence rate towards the infinitely dense mesh is very similar when comparing with the results in two dimensions.

7 Concluding Remarks

The present report presents analysis and comparisons of preconditioning methods for low speed flows from an application-type point of view. Recent preconditioners given by Turkel and Choi-Merkle are covered and generalized into a single formulation. The report describes all technical details which are neccessary in order to extend existing compressible flow codes for the low speed range.

The effects of the preconditioning matrices, artificial dissipation and grid density on accuracy and convergence are thoroughly investigated by using 2D airfoil flows as test cases. Preconditioning for 3D viscous flows is also applied with success. This demonstrates that preconditioning has now matured to the point where real applications are possible.

There are several interesting questions that have not been touched by the present work, though. Among these is our observation that the lower cut-off value for the term βM_r^2 in the preconditioning matrices seems to depend on the freestream velocity rather than on the local flow or on absolute velocity values. (see section 2). This problem is related to the question in how far preconditioning can be used to cluster eigenvalues in the stagnation region of hypersonic flows. In this case the freestream flow speed is very large but there is a local region with low Mach numbers embedded in the flow.

Finally, future work is neccessary in order to extend the preconditioning approach towards the use of multiequation turbulence models and chemically reacting flows. In these cases additional stiffness due to the presence of strong source terms must be taken into account in an appropriate way.

Acknowledgement

The present work was supported by the German-Israeli Foundation for Scientific Research and Development, GIF, under research grant no. I178-008.06/91 .

A Appendix I

We define the sets of variables

$$w_0 = (p, u, v, S)$$

$$w_c = w_1 = (\rho, \rho u, \rho v, E)$$

$$w_2 = (p, \rho u, \rho v, E)$$

$$w_3 = (p, \rho u, \rho v, E - h_\infty \rho)$$

$$w_4 = (p, u, v, T)$$

$$w_5 = (p, \rho u, \rho v, T)$$

Let a be the speed of sound, $q^2 = u^2 + v^2$ and $Q^2 = \frac{\rho q^2}{2}$. Given the preconditioning P_0 in w_0 variables we can compute the preconditioner P_i in w_i coordinates by

$$P_i = \frac{\partial w_i}{\partial w_0} P_0 \frac{\partial w_0}{\partial w_i}, \qquad P_0 \text{ given by } (2)$$

We have the following transformation matrices connecting these variables:

$$\frac{\partial w_0}{\partial w_1} = \begin{pmatrix} \frac{\gamma-1}{2}q^2 & (1-\gamma)u & (1-\gamma)v & \gamma-1 \\ \frac{-u}{\rho} & \frac{1}{\rho} & 0 & 0 \\ \frac{-v}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{\gamma-1}{2}q^2 - a^2 & (1-\gamma)u & (1-\gamma)v & \gamma-1 \end{pmatrix} \frac{\partial w_1}{\partial w_0} = \begin{pmatrix} \frac{1}{a^2} & 0 & 0 & \frac{-1}{a^2} \\ \frac{u}{a^2} & \rho & 0 & \frac{-u}{a^2} \\ \frac{v}{a^2} & 0 & \rho & \frac{-v}{a^2} \\ \frac{1}{\gamma-1} + \frac{M^2}{2} & \rho u & \rho v & \frac{-M^2}{2} \end{pmatrix}$$

$$\frac{\partial w_0}{\partial w_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{-u}{(\gamma-1)Q^2} & \frac{v^2-u^2}{2Q^2} & \frac{-uv}{Q^2} & \frac{u}{Q^2} \\ \frac{-v}{(\gamma-1)Q^2} & \frac{-uv}{Q^2} & \frac{u^2-v^2}{2Q^2} & \frac{v}{Q^2} \\ 1 - \frac{2}{(\gamma-1)M^2} & \frac{-2u}{M^2} & \frac{-2v}{M^2} & \frac{2}{M^2} \end{pmatrix} \frac{\partial w_2}{\partial w_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{u}{a^2} & \rho & 0 & \frac{-u}{a^2} \\ \frac{v}{a^2} & 0 & \rho & \frac{-v}{a^2} \\ \frac{1}{\gamma-1} + \frac{M^2}{2} & \rho u & \rho v & \frac{-M^2}{2} \end{pmatrix}$$

Let $G = \frac{u^2+v^2}{2} + h_\infty$, $L = \frac{1}{\gamma-1} + \frac{M^2}{2} - \frac{h_\infty}{a^2}$ and $E' = E - h_\infty$.

$$\frac{\partial w_0}{\partial w_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{-u}{(\gamma-1)\rho G} & \frac{G-u^2}{\rho G} & \frac{-uv}{\rho G} & \frac{u}{\rho G} \\ \frac{-v}{(\gamma-1)\rho G} & \frac{-uv}{\rho G} & \frac{G-v^2}{\rho G} & \frac{v}{\rho G} \\ 1 - \frac{a^2}{(\gamma-1)G} & \frac{-ua^2}{G} & \frac{-va^2}{G} & \frac{a^2}{G} \end{pmatrix} \quad \frac{\partial w_3}{\partial w_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{u}{a^2} & \rho & 0 & \frac{-u}{a^2} \\ \frac{v}{a^2} & 0 & \rho & \frac{-v}{a^2} \\ L & \rho u & \rho v & \frac{-M^2}{2} + \frac{h_\infty}{a^2} \end{pmatrix}$$

$$\frac{\partial w_0}{\partial w_4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1-\gamma & 0 & 0 & \frac{\gamma p}{T} \end{pmatrix} \qquad \frac{\partial w_4}{\partial w_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{(\gamma-1)T}{\gamma p} & 0 & 0 & \frac{T}{\gamma p} \end{pmatrix}$$

To transform the preconditioned residual in (p, u, v, T) to/from conservative variables we need the following Jacobians which are a direct result of those given above.

$$\frac{\partial w_1}{\partial w_4} = \begin{pmatrix} \frac{\rho}{p} & 0 & 0 & \frac{-\rho}{T} \\ \frac{\rho u}{p} & \rho & 0 & \frac{-\rho u}{T} \\ \frac{\rho v}{p} & 0 & \rho & \frac{-\rho v}{T} \\ \frac{E}{p} & \rho u & \rho v & \frac{-\rho q^2}{2T} \end{pmatrix} \quad \frac{\partial w_4}{\partial w_1} = \begin{pmatrix} \frac{(\gamma-1)q^2}{2} & (1-\gamma)u & (1-\gamma)v & \gamma-1 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{1}{\rho}(\frac{(\gamma-1)q^2}{2} - T) & \frac{(1-\gamma)u}{\rho} & \frac{(1-\gamma)v}{\rho} & \frac{(\gamma-1)}{\rho} \end{pmatrix}$$

$$\frac{\partial w_1}{\partial w_5} = \begin{pmatrix} \frac{1}{T} & 0 & 0 & \frac{-\rho}{T} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\gamma-1} - \frac{q^2}{2T} & u & v & \frac{\rho q^2}{2T} \end{pmatrix} \quad \frac{\partial w_5}{\partial w_1} = \begin{pmatrix} \frac{(\gamma-1)q^2}{2} & (1-\gamma)u & (1-\gamma)v & \gamma-1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\rho}(\frac{(\gamma-1)q^2}{2} - T) & \frac{(1-\gamma)u}{\rho} & \frac{(1-\gamma)v}{\rho} & \frac{\gamma-1}{\rho} \end{pmatrix}$$

We also have $\Gamma_i = \frac{\partial w_i}{\partial w_0} P_0 \frac{\partial w_0}{\partial w_c}$. So $P_i = \Gamma_i \frac{\partial w_c}{\partial w_i}$. In particular $P_c = \frac{\partial w_1}{\partial w_4}\Gamma_4$ and $P_4 = \Gamma_4 \frac{\partial w_1}{\partial w_4}$. For the (p, u, v, T) variables we explicitly have

$$\Gamma_4 = \begin{pmatrix} \beta M_r^2 \delta & 0 & 0 & 0 \\ \frac{-u}{\rho}(1+\alpha\delta) & \frac{1}{\rho} & 0 & 0 \\ \frac{-v}{\rho}(1+\alpha\delta) & 0 & \frac{1}{\rho} & 0 \\ \frac{(\gamma-1)T}{\gamma p}\left[\beta M_r^2 + \frac{q^2}{2} - \frac{a^2}{\gamma-1}\right] & \frac{(1-\gamma)Tu}{\gamma p} & \frac{(1-\gamma)Tv}{\gamma p} & \frac{(\gamma-1)T}{\gamma p} \end{pmatrix}$$

$$-(\gamma-1)(\delta-1)\begin{pmatrix} \frac{\beta M_r^4}{2} & -\frac{\beta M_r^2 u}{a^2} & -\frac{\beta M_r^2 v}{a^2} & \frac{\beta M_r^2}{a^2} \\ -\frac{\alpha u M_r^2}{2\rho} & \frac{\alpha u^2}{\rho a^2} & \frac{\alpha uv}{\rho a^2} & -\frac{\alpha u}{\rho a^2} \\ -\frac{\alpha v M_r^2}{2\rho} & \frac{\alpha uv}{\rho a^2} & \frac{\alpha v^2}{\rho a^2} & -\frac{\alpha v}{\rho a^2} \\ \frac{\beta M_r^2 T}{\gamma p}(\frac{(\gamma-1)M_r^2}{2} - 1) & \frac{(1-\gamma)T\beta M_r^2 u}{\gamma p a^2} & \frac{(1-\gamma)T\beta M_r^2 v}{\gamma p a^2} & \frac{(\gamma-1)T\beta M_r^2}{\gamma p a^2} \end{pmatrix}$$

and

$$\Gamma_4^{-1} = \begin{pmatrix} \frac{1}{\beta M_r^2} & 0 & 0 & 0 \\ \frac{(1+\alpha)u}{\beta M_r^2} & \rho & 0 & 0 \\ \frac{(1+\alpha)v}{\beta M_r^2} & 0 & \rho & 0 \\ \frac{1}{\beta M_r^2}(\frac{E+p}{\rho} + \alpha q^2) - 1 & \rho u & \rho v & \frac{\gamma p}{(\gamma-1)T} \end{pmatrix}$$

$$+(\delta-1)\begin{pmatrix} \frac{1-\gamma}{a^2} & 0 & 0 & \frac{\gamma p}{a^2 T} \\ \frac{(1-\gamma)u}{a^2} & 0 & 0 & \frac{\gamma p u}{a^2 T} \\ \frac{(1-\gamma)v}{a^2} & 0 & 0 & \frac{\gamma p v}{a^2 T} \\ \frac{(1-\gamma)M_r^2}{2} - 1 & 0 & 0 & \frac{\gamma p}{T}(\frac{1}{\gamma-1} + \frac{M_r^2}{2}) \end{pmatrix}$$

Given in this form, $\delta = 1$ simplifies these expressions and does not change the eigenvalues (but does change the eigenvectors) of PA. However, for P_4 (given in (6)) $\delta = 0$ is simpler.

Hence, if $\delta = 0$ it is more efficient to evaluate $\Gamma_4 \vec{x} = P_4(\frac{\partial w_4}{\partial w_1}\vec{x})$ then to use the above formula. Alternatively, one can use P_4 directly rather than Γ_4 (see Appendix II). If we examine the form of Γ_4^{-1} in more detail we see that in the first matrix the first column has terms that go to infinity like $\frac{1}{\beta M^2} \sim \frac{1}{q^2}$ while in the matrix that is multiplied by $\delta - 1$ the terms are all bounded as the Mach number goes to zero. Hence, the terms involving δ are lower order perturbations in Γ_4^{-1}. Hence, we expect that the choice of δ is not crucial.

In [12] it is shown that using the variables $(\frac{dp}{\rho a}, u, v, dp - a^2 d\rho)$ a necessary condition for the convergence of the numerical scheme to a scheme consistent with the incompressible equations is that (for isentropic flow)

$$\mathbf{P}^{-1} \sim \begin{pmatrix} O(1/M^2) & O(1/M) & O(1/M) \\ O(1/M) & O(1) & O(1) \\ O(1/M) & O(1) & O(1) \end{pmatrix} .$$

Translating this into requirements on Γ^{-1} we find that the scheme can convergence to a solution of the incompressible equations as the Mach number goes to zero only if

$$\Gamma_{\mathbf{i}}^{-1} \sim \begin{pmatrix} O(1/q^2) & O(1/q) & O(1/q) \\ O(1/q) + O(u/q^2) & \frac{u^2}{q^2} & \frac{uv}{q^2} \\ O(1/q) + O(v/q^2) & \frac{uv}{q^2} & \frac{v^2}{q^2} \end{pmatrix} \frac{\partial w_0}{\partial w_i} .$$

In practice one does not want Γ to depend on the ratio of u to q (i.e. the angle of the flow) as this creates difficulties near stagnation points. Hence

$$\Gamma_{\mathbf{i}}^{-1} \sim \begin{pmatrix} O(1/q^2) & O(1/q) & O(1/q) & O(1) \\ O(u/q^2) & O(1) & O(1) & O(1) \\ O(v/q^2) & O(1) & O(1) & O(1) \\ O(1/q^2) & O(1) & O(1) & O(1) \end{pmatrix} \frac{\partial w_0}{\partial w_i} .$$

We note that the preconditioner presented by Jorgenson and Pletcher [6] does not satisfy this necessary condition for accuracy at low Mach numbers. The generalized preconditioner presented above does satisfy this condition for all α and δ.

At this stage it is not clear if changes if eigenvectors but not eigenvalues has any effect on the convergence rate to a steady state and on the accuracy.

B Appendix II

Here we present the steps, in detail, necessary to introduce preconditioning into a previously existing code based on Algorithm I (section 2). For simplicity we shall assume an explicit iteration scheme (e.g. Runge-Kutta) possibly augmented by an implicit residual smoothing and multigrid.

1. Precondition the residual
   - Select a reasonable set of variables for the problem e.g. (p, u, v, T). We stress that other sets of variables are possible. This set is used for illustration only.
   - For nonconservative scalar artificial viscosity
     - multiply physical residual (inviscid and Navier-Stokes portions and physical forcing functions (if applicable)) by $\Gamma = P_T \frac{\partial w_4}{\partial w_1}$.

- add scalar artificial viscosity in these new variables to the residual.

- For conservative scalar viscosity
  - scalar viscosity includes Γ^{-1} (see (9))
  - add inviscid, viscous and artificial viscosity fluxes to obtain the total residual
  - multiply this total residual by Γ
- Having calculated the total preconditioned residual in (p, u, v, T) we multiply these by Δt. We then have two ways of updating the solution to the next Runge-Kutta stage
  - One: multiply the above total residual by $\frac{\partial w_1}{\partial w_4}$ to get the residual in conservative variables. Note that for the conservative artificial viscosity we can combine the two steps into one where $\frac{\partial w_1}{\partial w_4}\Gamma_4 = P_c$. This is especially useful for $\delta = 0$. We then add the new residual to the old conservative variables to get new conservative variables.
  - Two: We add the residuals in (p, u, v, T) variables to the old (p,u,v,T) variables to get (p, u, v, T) at the next Runge-Kutta stage. We then calculate the conservative variables as nonlinear functions of (p, u, v, T).

2. Assuming that we have residuals in both conservative variables and (p, u, v, T) variables the residual smoothing can be done in either set of variables. It is not clear which way is preferable.

3. For a multigrid acceleration we need to restrict the residuals from a finer grid to a coarser grid (for FAS the variables are also restricted). Forcing functions appear in the coarse grid residuals from two sources. They arise from the nonlinear-multigrid process which requires forcing functions so that the steady state solution remains staionary on all grids. Forcing functions can also appear from physical sources, e.g. chemical reactions, gravity, Coriolis forces etc. These forces originate on the finest grid and are subsequently brought to coarser grids. Hence, on the third level and all coarser grids two forcings functions have been restricted from a finer mesh, the multigrid forcing function and the physical forcing function. Preconditioning may require these two sets of forcing functions to be treated differently.

   We again have two choices as to which changes of variables are brought. from a finer grid to the next coarser grid. When prolonging variables from the coarse mesh to a fine mesh we always use conservative variables.

   - Conservative variables are used. One now has to keep track of how the various forcing functions have been stored on each grid. If the forcing function is stored after it has been preconditioned then one should not precondition it again after it has been restricted to the next coarser mesh. If a conservative form has been stored then one needs to precondition the forcing function after the restriction. Thus, physical and multigrid forcing terms may be treated differently on the coarse grids.
   - Transfers to coarser meshes are based on differences in the (p, u, v, T) variables. Hence, both residual smoothing and restriction operations are done in the new variables. The conservative variables are used to compute the fluxes. The restriction of the variables (not differences) for the FAS version of the multigrid is also performed on the conservative variables. This is coupled with the second possibility for updating i.e. we update (p, u, v, T) and the calculate the conservative variables as nonlinear functions of (p, u, v, T). Hence, this version of Algorithm I is closer to that of Algorithm II.

     We note that the differences between these options may involve only a small change in coding, mainly at which point in the code is the preconditioning matrix applied.

4. Choose a new time step based on the preconditioning (4). This is usually done only for the inviscid portion. The viscous contribution to the time step is then incorporated as usual. Choi and Merkle [2] suggest modifying βM_r^2 based on the local cell Reynolds number.

5. Modify the far field boundaries based on the new characteristics. If characteristic data is also used at physical boundaries (e.g. inflow to a nozzle) these should also be modified (see section [5]).

For Algorithm II (section 3) all operations including residual smoothing and interpolations between grids are done in terms of the $(p', \rho u, \rho v, E')$ variables. This scheme has the advantage that, in theory, setting $M_\infty = 0$ would yield identical results to an incompressible code based on pseudo-compressibility. Algorithm I does not have this property since the multigrid procedures (and possibly the residual smoothing) is still based on compressible conservative variables which are inappropriate in the low Mach number limit. However, successful runs with Algorithm I have been done at freestream Mach numbers as low as 0.0001. Hence, this property does not seem to have much importance in practical applications.

References

[1] Buelow, P.E.O., Venkateswaran, S., Merkle, C.L., *The Effect of Grid Aspect Ratio on Convergence*, AIAA Computational Fluid Dynamics Conference, AIAA paper 93-3367-CP, 1993.

[2] Choi, Y.-H. and Merkle, C.L., *The Application of Preconditioning to Viscous Flows*, Journal of Computational Physics, 105(1993), 207-223.

[3] Chorin, A.J., *A Numerical Method for Solving Incompressible Viscous Flow Problems*, Journal of Computational Physics, 2(1967), 12-26.

[4] Darmofal, D.L. and Schmid,P.J. *The importance of eigenvectors for local preconditioners of the Euler equations,* 12th AIAA Computational Fluid Dynamics Conference, 102-117, 1995.

[5] Fiterman, A., Turkel, E. and Vatsa, V., *Pressure Updating Methods for the Steady-State Fluid Equations*, Proceedings 12th AIAA Computational Fluid Dynamics Conference, AIAA paper 95-1652, 1995.

[6] Jorgenson, P.C.E. and Pletcher R.H., *An Implicit Numerical Scheme for the Simulation of Internal Viscous Flows on Unstructured Grids* AIAA paper 94-0306, 1994.

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

[8] Kroll, N., Radespiel, R., Rossow, C.-C., *Accurate and Efficient Flow Solvers for 3D Applications on Structured Meshes*, VKI Lecture Series, LS-04, 1994.

[9] Radespiel, R., Rossow, C.-C. and Swanson, R.C. *An Efficient Cell-Vertex Multigrid Scheme for the Three Dimensional Navier-Stokes equations*, AIAA J., 28(1990), 1464-1472.

[10] Swanson, R.C., Radespiel,R. *Cell Centered and Cell Vertex Multigrid Schemes for the Navier-Stokes Equations,* AIAA J., 29(1991), 697-703.

[11] Swanson, R.C., Turkel, E., *Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations* , AIAA Computational Fluid Dynamics Conference, AIAA 87-1107-CP, 1987.

[12] Turkel, E., Fiterman, A. and van Leer, B. *Preconditioning and the Limit to the Incompressible Flow equations*, Computing the Future: Frontiers of Computational Fluid Dynamics 1994, editors D.A.Caughey and M.M. Hafez, Wiley Publishing, 215-234, 1994.

[13] Turkel, E., *Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations* , Journal of Computational Physics, 72(1987), 277-298.

[14] Turkel, E., *A Review of Preconditioning Methods for Fluid Dynamics*, Applied Numerical Mathematics, 12(1993), 257-284.

[15] Volpe, G., *Performance of Compressible Flow Codes at Low Mach Numbers*, AIAA J., 31(1993), 49-56.

Figure 1: Mesh lines near NACA 0012 airfoil

Figure 2: Numerical solution for incompressible inviscid flow

Figure 3: Grid convergence of Code1 for inviscid flow, $k^{(4)} = 1/32, \zeta = 2/3$

Figure 4: Grid convergence of Code2 for inviscid flow, $k^{(4)} = 1/32, \zeta = 2/3$

Figure 5: Residual convergence of Code1 for inviscid flow, $\delta = 1, k^{(4)} = 1/32, \zeta = 2/3$

Figure 6: Residual convergence of Code2 for inviscid flow, $k^{(4)} = 1/32, \zeta = 2/3$

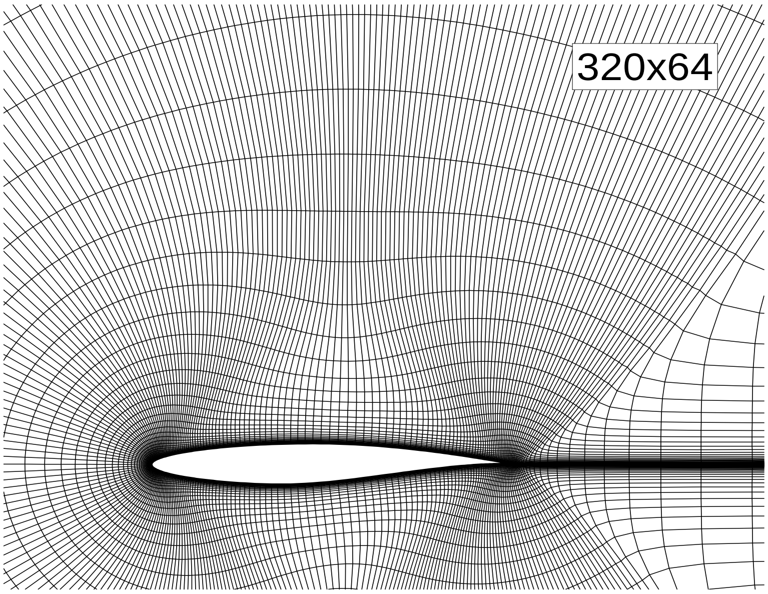

Figure 7: Mesh lines near RAE 2822 airfoil

Figure 8: Numerical solution of Code1 for incompressible viscous flow,$\alpha = 0, \delta = 1, k^{(4)} = 1/32, \zeta = 2/3$

Figure 9: Numerical solution of Code2 for incompressible viscous flow,$\alpha = 1, \delta = 0, k^{(4)} = 1/32, \zeta = 2/3$

Figure 10: Grid convergence of Code1 and Code2 for $k^{(4)} = 1/32, \zeta = 2/3$

Figure 11: Effect of farfield boundary condition on lift and drag of Code2, $\alpha = 0, k^{(4)} = 1/32, \zeta = 2/3$

Figure 12: Effect of artificial viscosity on grid convergence of Code1, $\alpha = 0, \delta = 1$

Figure 13: Effect of preconditioning matrix on grid convergence for Code1,$k^{(4)} = 1/32, \zeta = 2/3$

Figure 14: Effect of preconditioning matrix on grid convergence for Code2,$k^{(4)} = 1/32, \zeta = 2/3$

Figure 15: Residual convergence of Code1 and Code2 for $k^{(4)} = 1/32, \zeta = 2/3$

Figure 16: Effect of artificial viscosity on residual convergence of Code1, $\alpha = 0$

Figure 17: Effect of preconditioning matrix on residual convergence for Code1,$k^{(4)} = 1/32, \zeta = 2/3$

Figure 18: Effect of preconditioning matrix on residual convergence for Code2,$k^{(4)} = 1/32, \zeta = 2/3$

Figure 19: Effect of preconditioning matrix on residual convergence for Code2,$M_\infty = 0.30$

Figure 20: Effect of preconditioning matrix on lift and drag for Code2,$M_\infty = 0.30$

Figure 21: Numerical solution of Code1 for transonic flow ,$k^{((4)} = 1/64, \zeta = 0$

Figure 22: Effect of preconditioning on grid convergence of Code1 for transonic flow ,$k^{((4)} = 1/64, \zeta = 0$

Figure 23: Wing surface and grid in symmetry plane

Figure 24: Effect of freestream Mach number on residual convergence of 3D wing

Figure 25: Effect of freestream Mach number on pressure distribution of 3D wing

Figure 26: Effect of freestream Mach number on force coefficients of 3D wing

Figure 27: Effect of grid density on force coefficients of 3D wing