AIAA-99-3265

UNI-DIRECTIONAL IMPLICIT ACCELERATION TECHNIQUES FOR COMPRESSIBLE NAVIER-STOKES SOLVERS

E. Turkel *

Tel-Aviv University, Israel

V. Vatsa †

NASA Langley Research Center, Hampton, VA

V. Venkatakrishnan ‡

Boeing Commercial Airplanes, Seattle, Washington

Abstract

We solve the steady-state compressible Navier-Stokes equations around complex aerodynamic bodies. A central difference scheme is used augmented by a matrix valued artificial viscosity. To reach a steady state a multi-stage algorithm is used to march the solution in pseudo-time. Multigrid is then applied to further accelerate the convergence to a steady-state. However, because of the high aspect ratio, cells that are needed to resolve the turbulent boundary layers at high Reynolds numbers this scheme still requires thousands of sweeps through the mesh. To further accelerate the convergence a block implicit method is added to the above algorithm. Because of difficulties with A.D.I. methods this implicit operator is added in only one direction, usually normal to the boundary layer. To partially account for the other directions the complete linearized Jacobian is included in the diagonal term of the implicit operator. Hence, a block tridiagonal matrix needs to be inverted at each stage of the multistage scheme. This method can also be adapted to include a low speed preconditioning to accelerate the convergence rates at small Mach numbers. Results are presented for flow about an ONERA wing at several different inflow Mach numbers.

*Professor, Department of Mathematics, Associate Fellow
†Senior Fellow
‡Principal Engineer, Senior Member

Introduction

Solving the compressible steady state Navier-Stokes equations in three dimensional complex geometries requires thousands of explicit sweeps through the mesh in order to reduce the residual by several orders of magnitude even using multigrid to accelerate the rate of convergence. In contrast, when solving the inviscid equations around an isolated three dimensional wing one can frequently get solutions to within engineering accuracy in only one to two hundred sweeps through the mesh (i.e. minimal work units). The main difference between the two cases is the high stretching needed to resolve the boundary layer at high Reynolds numbers. Because of this stretching, the aspect ratio between the directions normal and tangential to the boundary layer can be of the order of thousands or even ten thousands. This high aspect ratio severely deteriorates the convergence rates of explicit schemes. Implicit schemes such as ADI or residual smoothing do not overcome this deficiency.

In this paper we will analyze and use a scheme which combines both an implicit method with a Runge-Kutta time stepping algorithm, [1, 13]. In one direction (usually denoted j in an generalized (i, j, k) coordinate system) the scheme is fully implicit. This results in a block tridiagonal system that needs to be inverted in the j direction. and requires the storage of three 5×5 matrices (in three dimensions) at each grid point. In addition to extra storage, extra computer time required by the block inversion. Since the numerical system is explicit in the other space directions we cannot use a one step method in time. Instead we embed the implicit method within a multi-stage cycle. This multi-stage Runge-Kutta method is designed to be stable for the explicit portion of the algorithm. In order to reduce the computational effort, the LU decomposition of the left hand side is computed only on the first stage of a five stage method. This LU decomposition is then reused at every stage. Further savings in CPU time can be realized by freezing the LU decomposition once the residual has been reduced two orders of magnitude. In addition to the implicit treatment of the j-line we also augment the diagonal sub-matrix by the linearization of the terms that affect the diagonal part of the right hand side.

For low speeds it is well known that the standard algorithms do not converge very well. To overcome this difficulty preconditioning techniques have been devised to alter the wave speeds. These new systems have all the speeds approximately equal as the Mach number approaches zero [6, 7, 8, 10]. It was also shown that the accuracy deteriorates as the flow approaches an incompressible state. To get the correct steady state at low Mach numbers it is necessary to also change the artificial viscosity/upwinding to account for the preconditioning [11]. However, one can now reverse the process and consider the main purpose of low speed preconditioning as altering the steady state through changes to the artificial viscosity. One can then formulate more general preconditioners to accelerate the approach to a steady state that account for the low speed preconditioners but include other considerations [9, 10].

J-line scheme

We consider a general preconditioning scheme for marching the steady state equations in pseudo-time. For simplicity of notation we consider the two dimensional case. Extensions to three dimensions are straightforward.

$$\mathbf{P}^{-1}\frac{\partial w}{\partial t}+\frac{F_{i+1/2,j}-F_{i-1/2,j}}{\Delta\xi}+\frac{G_{i,j+1/2}-G_{i,j-1/2}}{\Delta\eta} = d_{i+1/2,j}-d_{i-1/2,j}+d_{i,j+1/2}-d_{i,j-1/2}, \qquad (1)$$

$$\begin{aligned} d_{i+1/2,j} &= \frac{1}{2}\left(\mathbf{Q}^{-1}|\sigma(\mathbf{Q}A)|\right)_{i+1/2,j}(w_{i+1,j}-w_{i,j}),\\ d_{i,j+1/2} &= \frac{1}{2}\left(\mathbf{Q}^{-1}|\sigma(\mathbf{Q}B)|\right)_{i,j+1/2}(w_{i,j+1}-w_{i,j}) \end{aligned} \qquad (2)$$

where A and B are the Jacobian matrices and σ is a matrix function. We redefine the matrices A and B so that they include the dependence on the mesh, i.e. $\frac{A}{\Delta\xi}\to A$, $\frac{B}{\Delta\eta}\to B$. Here P is the preconditioning matrix. When we do not use low speed preconditioning we have $\mathbf{Q}=I$. For low speed flow we instead choose $\mathbf{Q}=\mathbf{P}_{lowspeed}$ where $\mathbf{P}_{lowspeed}$ is the low speed preconditioner given in [7, 10, 12] (with $\alpha=\delta=0$).

We replace all the spatial derivatives by central differences. Eq. (2) displays only a second difference artificial viscosity. In practice, a sum of second and fourth differences in each direction is used [2]. The second difference has a nonlinear switch so that it is turned on only in the neighborhood of shocks. The purpose of this viscosity is to introduce an entropy condition for the selection of the correct shock profiles. The fourth difference is a linear term which is introduced to prevent the appearance of odd-even oscillations. Such oscillations prevent the scheme from converging to a steady state. For simplicity of notation we have shown in (2) a linear second difference artificial viscosity. In practice this is used for the implicit part of the scheme while the explicit portion contains the full combined second/fourth difference nonlinear artificial viscosity.

To advance the scheme in time we use a Runge-Kutta method given by:

$$\begin{aligned} w^0 &= w^{(n)}\\ w^{k+1} &= w^0+\alpha_k\Delta t\mathbf{P}R \quad k=1\ldots K \\ w^{(n+1)} &= w^K \end{aligned} \qquad (3)$$

where R is the residual given by

$$\begin{aligned} R = &-\left(\frac{F_{i+1/2,j}-F_{i-1/2,j}}{\Delta\xi}+\frac{G_{i,j+1/2}-G_{i,j-1/2}}{\Delta\eta}\right)\\ &+d_{i+1/2,j}-d_{i-1/2,j}+d_{i,j+1/2}-d_{i,j-1/2} \end{aligned}$$

We first consider a Jacobi method for determining $\mathbf{P}$, i.e. $\mathbf{P}$ is derived from the linearized coefficients of w_{ij} of the steady state portion of (1). This comes only from the artificial viscosity terms. Hence,

$$\mathbf{Q}\mathbf{P}_J^{-1}=\sigma(\mathbf{Q}A)+\sigma(\mathbf{Q}B). \qquad (4)$$

This definition has a difficulty in the normalization. To see this, we consider the special case of a scalar viscosity so that $\sigma(A)$ is the spectral radius of A. Then $\mathbf{P}$ reduces to a scalar times the identity matrix. This scalar effectively changes the time step which however is fixed by stability considerations. Hence, we need to modify (4) to

$$\mathbf{QP}_J^{-1} = \frac{\sigma(\mathbf{Q}A) + \sigma(\mathbf{Q}B)}{\rho(\mathbf{Q}A) + \rho(\mathbf{Q}B)} \tag{5}$$

where $\rho(A)$ is the spectral radius of A. For the original scheme $\Delta t = \frac{\mathrm{CFL}}{\rho(\mathbf{Q}A)+\rho(\mathbf{Q}B)}$. Hence, we cancel Δt from both sides of the equation. However, the term on the left hand side comes from the artificial viscosity. Thus, this cancellation depends on the details of the artificial viscosity. For example, there might not be any cancellation if a *CUSP* artificial viscosity [3, 6] is used instead of a matrix viscosity.

To check the stability of the scheme we consider only the linearized constant coefficient equation. We then Fourier transform with variables ω_x, ω_y and set $\theta = \omega_x \Delta x$, $\phi = \omega_y \Delta y$ We further assume that all the matrices that appear are symmetric (or can be simultaneously symmetrized) which is true for the Navier-Stokes equations. We assume that each stage of the Runge-Kutta process is identical. We can then express (3) in Fourier space as

$$\hat{w}^{k+1} = \hat{w}^0 + \alpha_k z \hat{w}^k \tag{6}$$

where z is Δt times the Fourier transform of $\mathbf{P}R$. The total Runge-Kutta cycle is then

$$\hat{w}^{n+1} = P_K(z)\hat{w}^0 \tag{7}$$

where $P_K(z)$ is a polynomial in z of order K, the number of stages. The domain of stability is the domain in the complex plane for which $|P_K(z)| \leq 1$. For a scalar equation this multistage scheme is stable for a scheme without any acceleration techniques when the eigenvalues of $i\mathbf{P}(A\sin(\theta) + B\sin(\phi))$ lie within the stability region of the Runge-Kutta scheme. For a system of equations this is a necessary but not sufficient condition. When $\mathbf{P}$ is symmetric and positive definite it is also sufficient. We consider a matrix valued artificial viscosity [5] where $\sigma(A) = |A|$. For the Jacobi method we require

$$\begin{aligned}\Delta t(\rho(QA) + \rho(QB))(|QA| + |QB|)^{-1}\\ (QA\sin(\theta) + QB\sin(\phi)) \leq 1\end{aligned} \tag{8}$$

We now generalize (4) and define the J-line method for the linearized equations by

$$\begin{aligned}&-\frac{|\mathbf{Q}B| + \mathbf{Q}B)}{2}\Delta w_{i,j+1} - \frac{(|\mathbf{Q}B| - \mathbf{Q}B)}{\Delta} w_{i,j-1}\\ &\quad + (|\mathbf{Q}A| + |\mathbf{Q}B|)\Delta w_{i,j}\\ &\quad = \Delta t(\rho(\mathbf{Q}A) + \rho(\mathbf{Q}B))R\end{aligned} \tag{9}$$

with

$$\begin{aligned}R &= \frac{\mathbf{Q}A}{2}(w_{i+1,j} - w_{i-1,j}) + \frac{\mathbf{Q}B}{2}(w_{i,j+1} - w_{i,j-1})\\ &+ \frac{1}{2}|\mathbf{Q}A|(w_{i+1,j} - 2w_{ij} + w_{i+1,j})\\ &+ \frac{1}{2}|\mathbf{Q}B|(w_{i,j+1} - 2w_{ij} + w_{i,j-1})\end{aligned} \tag{10}$$

This represents the linearization of the residual. Of course, in the code this is replaced by the full nonlinear residual so that we get the correct steady state.

For the J-line method, the Fourier transform is

$$\begin{aligned}\{-i\mathbf{Q}B\sin(\phi) + |\mathbf{Q}B|(1-\cos(\phi)) + |\mathbf{Q}A|\}(z-1) =\\ \Delta t(\rho(\mathbf{Q}A) + \rho(\mathbf{Q}B))R\end{aligned} \tag{11}$$

with

$$\begin{aligned}R = i\mathbf{Q}A\sin(\theta) + i\mathbf{Q}B\sin(\phi)\\ - |\mathbf{Q}A|(1-\cos(\theta)) - |\mathbf{Q}B|(1-\cos(\phi))\end{aligned} \tag{12}$$

To get a feel of this amplification matrix we compute $z(\theta,\phi)$ for some representative cases of θ,ϕ, for a scalar equation. The scheme is stable when $z(\theta,\phi)$ is in the domain of stability of the Runge-Kutta scheme. Define $\Delta t' = \Delta t(\rho(QA) + \rho(QB))$. We then rewrite (11) as

$$\begin{aligned}z(\theta,\phi) = 1 + \Delta t'\Big(iQA\sin(\theta) + iQB\sin(\phi)\\ -|QA|(1-\cos(\theta)) - |QB|(1-\cos(\phi))\Big)\\ \Big(-iQB\sin\phi + |QB|(1-\cos(\phi)) + |QA|\Big)^{-1}\end{aligned} \tag{13}$$

Note that if we double the value of $|A|$ on the diagonal of the implicit operator we replace the denominator of (13) by

$$-iQB\sin\phi + |QB|(1-\cos(\phi)) + 2|QA|$$

In this case we have

$$\begin{aligned}z(\pi,\phi) &= 1 + \Delta t'\Big(iQB\sin(\phi)\\ &\quad - 2|QA| - |QB|(1-\cos(\phi))\Big)\\ &\Big(-iQB\sin\phi + |QB|(1-\cos(\phi)) + 2|QA|\Big)^{-1}\\ &= 1 - \Delta t'\end{aligned}$$

Choosing $\Delta t' = 1$ implies $z = 0$ and so a one step Runge-Kutta method would kill the highest mode in the θ direction. For a multistage scheme there is not necessarily a zero of the RK polynomial on the real axis. Nevertheless, $z(\pi,\phi)$ is a scalar independent of the Jacobian matrices and so we can choose $\Delta t'$ to maximize the convergence rate of this mode within the restrictions of stability.

If we keep the original scheme and do not double the contribution of $|QA|$ to the implicit operator we can still calculate the growth factor for special modes. In particular,

$$z(\pi,\pi) = 1 - \frac{\Delta t'(2|QA| + 2|QB|)}{|QA| \ + 2|QB|}$$

Similarly,

$$z(\frac{\pi}{2},\pi) = 1 - \Delta t' + \frac{iQA\Delta t'}{|QA| + 2|QB|}$$

and

$$z(\pi,\frac{\pi}{2}) = 1 - \Delta t' - \frac{\Delta t'|QA|}{-iQB + |QA| + |QB|}$$

Experience indicates that if we choose $\Delta t'$ so that these values of $z-1$ are inside the stability domain that usually the scheme is stable for all modes.

If $A = 0$ then $z = 1 - \Delta t'$ and we again can have very high convergence rates. In this case each stage is a perfect linearized, implicit method. Thus, a one step Runge-Kutta method is equivalent to backward Euler. In this case a multistage method is less efficient than the one stage method. At the other extreme when $B = 0$ then the scheme reduces to a standard explicit scheme (with the time step multiplied by $\frac{\rho(a)}{|A|}$) and so we need the multistage method to stabilize the scheme. For general A, B the stability limit is a compromise between the two extremes. Thus, a smaller number of stages may be efficient. Nevertheless, for the sake of comparisons with the existing code, a five stage Runge-Kutta has been used for the computations. The J-line method becomes more efficient as B becomes larger relative to A, e.g. for large aspect ratios.

The preconditioner, $\mathbf{P}$, changes the eigenvalues of the system. This is true whether or not we use low speed preconditioning. In particular the eigenvalues of $|\mathbf{P}A|$ now depend on u, v, c and not just u, c. There are no simple formulas for these eigenvalues for the two dimensional Euler matrices. Hence, we do not necessarily have $\Delta t = (\rho(QA) + \rho(QB))^{-1}$. Thus, the stability requirements for this multistage algorithm are different than for the original scheme. In particular, we note that one can construct a multistage scheme even in one dimension based on an implicit operator which is no longer unconditionally stable. Thus, the multistage procedure can reduce the stability properties of a one-step algorithm. We have presented the J-line method (10) for the Euler equations. For the Navier-Stokes equations we add the appropriate viscous terms to the right hand side of (10) and its linearization in the j direction and the diagonal terms to the left hand side of (10). Since we frequently use only the thin layer equations in the j direction the viscous terms are completely accounted for in the linearization. However, experimentation shows that in many cases the viscous terms are not needed in the implicit operator. When low speed preconditioning is used then the equations are frequently expressed in (p, u, v, T) variables (see [7, 8, 10, 12]). The use of these variables for the J-line method simplifies many of the terms in the implicit operator. Also, when multigrid is used to accelerate the convergence only a constant artificial viscosity is used on the coarse meshes. This coarse grid artificial viscosity can be either scalar or a matrix valued even though the artificial viscosity on the finest grid is matrix valued.

Previous work has indicated that one cannot use residual smoothing in conjunction with either the Jacobi or j-line smoothings. However, this was based on the assumption that the parameters of the residual smoother would remain the same as in the standard algorithm. However, these parameters depend on the eigenvalues of the (modified) system and so need to be changed. We thus perform residual smoothing in all directions except for the j direction. The parameter of the residual smoothing should be appropriately modified. In the tests performed for this study a constant residual smoothing was used.

Results

In order to improve the convergence rate of the standard code one needs to augment the dissipation depending on the aspect ratio. A typical addition is to multiply the artificial dissipation in the i direction [4] by

$$d_i \leftarrow d_i * \left(1 + (\frac{\lambda_j}{\lambda_i})^{\zeta} + \frac{\lambda_k}{\lambda_1})^{\zeta}\right) \tag{14}$$

The choice $\zeta = 1$ leads to an isotropic artificial viscosity which is useful for inviscid flows but not for highly stretched meshes. Choosing $\zeta = 0$ gives an individual eigenvalue scaling in each direction. A typical compromise between accuracy and convergence rate is $\zeta = \frac{1}{2}$. The linearization of the artificial viscosity for the left hand side of (10) is always based on $\zeta = 0$ independent of the ζ that is used on the right hand side.

We consider turbulent flow about the ONERA M6 wing. The Reynolds number is 21.666 million and the angle of attack is 3.06°. We first consider transonic flow with $M_\infty = .84$. The *CFL* of the standard scheme is 6.5. However, since the J-line method did not use any residual smoothing we can only use a *CFL* of 2.0. The convergence rate in L_2 of the density residual is displayed in figure 1. We see that, indeed, $\zeta = 0$ slows the convergence for the standard code. However, for the J-line algorithm choosing $\zeta = 0$ improves the convergence rate. This could be caused by the improved linearization with $\zeta = 0$. Hence, eigenvalue scaling improves both the accuracy and the convergence rate. Furthermore, $\zeta = 0$ decreases the total running time since one no longer needs to compute exponentials or square roots in (14). In figure 2 we display the convergence rate for the total drag. We see that the steady state drag differs depending on whether $\zeta = 0$ or $\zeta = \frac{1}{2}$. We expect $\zeta = 0$ to give the more accurate results. The rate of convergence is similar to that of the residual with the J-line method with $\zeta = 0$ being the fastest.

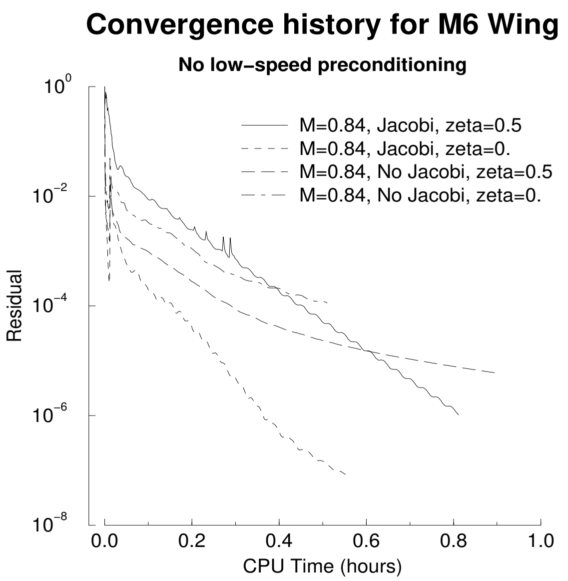

Figure 1: Residual convergence

The second case has $M_\infty = .1$. We now combine the J-line acceleration with the low speed preconditioning [9]. For this case we use $CFL = 1.0$ and also

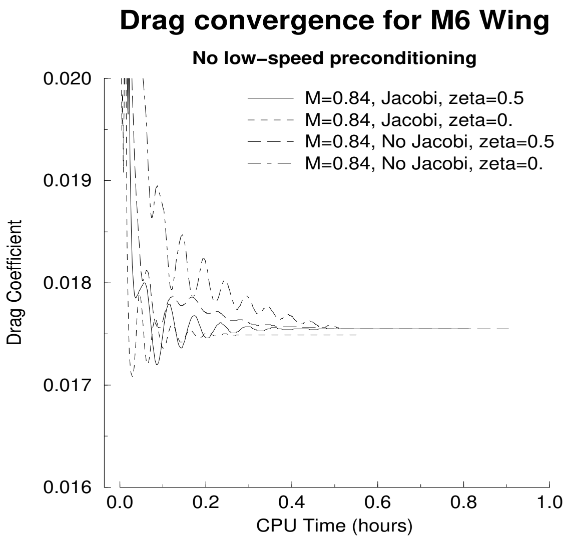

Figure 2: Drag convergence

need to increase the cutoff of β relative to the usual low speed preconditioning (see [8, 10, 12]). The convergence of the residuals are shown in figure 3. In figure 4 we display the convergence rate for the total drag. We again see that the steady state drag differs depending on whether $\zeta = 0$ or $\zeta = \frac{1}{2}$.

The rate of convergence of the drag is similar to that of the residual. The combined J-line and low speed preconditioning method, with $\zeta = 0$ having a similar convergence rate to the original low speed preconditioning but with $\zeta = \frac{1}{2}$.

We see from these figures that even without residual smoothing that the J-line method with $\zeta = 0$ is more efficient and more accurate than the standard multistage scheme combined with residual smoothing. Using residual smoothing in the i and k directions improves the robustness of the calculation but did not significantly improve the convergence rate.

Conclusion

The J-line method requires an additional 75 locations per mesh point to store the three 5×5 matrices for three dimensional problems. It is also more expensive per sweep through the mesh than the original algorithm because of the need to invert the block tridiagonal matrices. However, the L and U factorization can be stored without further increasing the storage. This factorization considerably reduces the

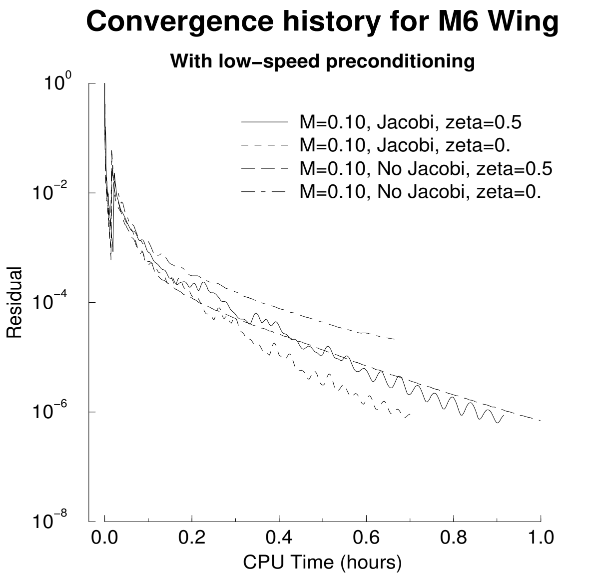

Figure 3: Residual convergence with low speed preconditioning

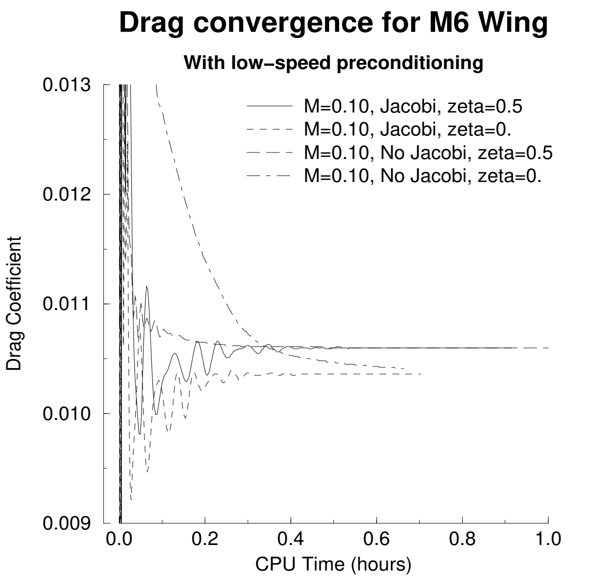

Figure 4: Drag convergence with low speed preconditioning

CPU time since the factorization is frozen during a Runge-Kutta cycle. Furthermore, once the residual is reduced by some two orders of magnitude the *LU* factorization can be completely frozen. In addition, for a multiblock code one needs to use the J-line scheme only in blocks that contain cells with a high aspect ratio, generally in the boundary layer or wakes. Thus, blocks in the far-field may not need the J-line scheme reducing both computer storage and time.

The computations demonstrate that even given the high CPU cost per sweep the J-line method nevertheless converges faster than the standard JST algorithm [2, 5] both in terms of cycles and total computer time. This occurred for both transonic flows and for low speed cases. Furthermore, this technique allows an eigenvalue scaling for the artificial viscosity which is more accurate than the compromise suggested by Martinelli [4].

Acknowledgements

This work was supported under the NASA/industry Advanced Subsonic Technology program. The authors would like to thank the Numerical Aerodynamics Simulation Division at NASA Ames Research Center for the use of the Cray C90.

References

[1] Mavriplis, D.J. *Directional Agglomeration Multigrid Techniques for High reynolds Number Viscous Flow Solver*, AIAA paper 98-0612, 1998.

[2] 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.

[3] Jameson, A. *Artificial Diffusion, Upwind Biasing, Limiters and their effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flow*, AIAA Paper 93-3559, 1993.

[4] Martinelli, L. and Jameson, A., *Validation of a Multigrid Method for the Reynolds Averaged Equations* AIAA Paper 88-0414, 988.

[5] Swanson, RC, Turkel E, 1987. On Central Difference and Upwind Schemes, *J. Comput. Phys.* 101:292-306.

[6] Swanson, RC, Radespiel R, Turkel E, 1997. On Some Numerical Dissipation Schemes *13th AIAA Computational Fluid Dynamics Conference, AIAA pap* 97-1945.

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

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

[9] Turkel E, *Preconditioning-Squared Methods for Multidimensional Aerodynamics*, 13th AIAA Computational Fluid Dynamics Conference, AIAA pap 97-2025, 1997.

[10] Turkel, E. *Preconditioning Techniques in Computational Fluid Dynamics*, Annual Review of Fluid Mechanics, Vol. 31, 385-416, 1999.

[11] Turkel E, Fiterman A, van Leer B, *Preconditioning and the Limit to the Incompressible Flow Equations*, **Computing the Future: Frontiers of Computational Fluid Dynamics 1994** ed. Caughey DA, Hafez MM, Wiley Publishing, 215-234, 1994.

[12] Turkel E, Radespiel R, Kroll N, *Assessment of Two Preconditioning Methods for Aerodynamic Problems*, Computers and Fluids, 613-634, 1997.

[13] Venkatakrishnan, V, *Improved Convergence of Compressible Navier-Stokes Solvers*, AIAA paper 98-2967, 1998.