Proceedings of 1992 Conference on Advanced Earth-to-Orbit Propulsion Technology, May 19-21, 1992, NASA Marshall Space Flight Research Center, Huntsville, AL, Editor: S.T. Wu

RELIABILITY ENHANCEMENT OF NAVIER-STOKES CODES THROUGH CONVERGENCE ENHANCEMENT*

Charles L. Merkle, George Dulikravich, Sankaran Venkateswaran, Kwang Choi and Philip E. O. Buelow

ABSTRACT

Three specific approaches for enhancing code reliability and convergence are being investigated as potential means for providing reductions in CPU time and the "real time" required to obtain solutions. The first approach is based on time derivative extrapolation; the second on time-derivative preconditioning; while the third compares the convergence and reliability characteristics of pressure-based and time-based algorithms. The basic idea of the extrapolation technique, the Distributed Minimal Residual (DMR) method, is to seek acceleration factors for each equation that minimize future residuals. The DMR method is also combined with multi-grid strategies for additional acceleration. The preconditioning approach seeks to extend the range of convergence of transonic codes into low Mach number regimes through control of the eigenvalues. Special attention is given to preconditioning in high Reynolds number flows. Similar eigenvalue control considerations are also considered for enhancing reliability in regions of strong grid stretching. Finally, comparison of the robustness and convergence of pressure-based codes with density-based procedures is investigated with the intent of identifying advantageous characteristics that can be transported between these two methods thereby providing improvements in both of them.

INTRODUCTION

Computational techniques and computer capabilities have progressed rapidly in the last decade. Three-dimensional computations are becoming routine. It is possible to solve complex fluid dynamic problems by numerical techniques, but in many instances a small change incurs an unwarranted amount of operator intervention, and/or an unexpectedly large increase in computer time. For example, a minor change in geometry or grid can have a major effect on convergence, while coaxing a new problem through to a successfully converged solution can take days or weeks for even an experienced numerical analyst. Difficulties of this nature frequently dominate the cost of numerical solutions and hinder their effective implementation in design procedures. The present paper is directed towards evaluating convergence acceleration schemes that enhance existing algorithms.

The paper is divided into three parts. The first two present distinct methods for enhancing the convergence and robustness of existing algorithms, while the third compares the characteristics of two popular methods. The two convergence acceleration techniques include time extrapolation methods and time derivative preconditioning. The third section compares density-based methods with pressure-based procedures.

DMR METHOD WITH MULTI-GRID ACCELERATION

The DMR (Distributed Minimal Residual) acceleration method is being applied to the steady, incompressible, Navier-Stokes equations. The basic idea of the DMR method is to seek acceleration factors for each equation in order to minimize the L2 norm of future residuals. By the nature of

*Work supported by NAS8-38661. Computational time was provided by a grant from NAS.

this method, it is only applicable to inherently iterative schemes. The DMR method is applied at a certain interval and combines a number of consecutive iterative solutions to predict the optimal acceleration factors. In order to present the effectiveness of the DMR method, the explicit Runge-Kutta time-stepping method is being used as a basic algorithm. Artificial compressibility is introduced to the incompressible code for local time-stepping.

Multi-grid methods are also being considered for use with the DMR acceleration method. Low frequency errors diminish very slowly on fine grids and multi-grid enables these low frequency errors to be reduced quickly on successively coarser grids. The larger grid spacing also permits the tracking of the

smaller number of grid points. The fine grid calculation is important because it not only improves the accuracy of the solution but also removes high frequency errors effectively so these errors will not appear as low frequency errors on coarse grids.

BOUNDARY TREATMENT

Inlet/Exit A non-reflecting boundary treatment is employed at the exit allowing for recirculating flow there. The non-reflecting boundary condition demands that the amplitude of an incoming wave be constant in time. This condition allows computation to handle the variable pressure at the exit boundary. The outgoing waves depend only on information from within the boundary. Those equations which represent outgoing waves can be solved directly to give two velocity components at the exit. At the inlet, a characteristic boundary treatment is employed. Two velocity components are specified while pressure is obtained by solving the characteristic form of the equation.

Solid wall On a solid boundary, the pressure gradient normal to the wall calculated from the momentum equations. This gives physically correct pressure distribution rather than assuming zero pressure gradient at the wall.

Characteristic and non-reflecting boundary treatment Consider the equations in primitive variables form,

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial h} + \frac{\partial G}{\partial z} + H = 0 \qquad (1)$$

where, Q, E, F, G and H take their normal values. At any boundary, the behavior of outward running waves is defined by the solution within the boundary. The behavior for inward running waves depends on the solution exterior to the boundary and is defined by a boundary condition.

For enforcing the boundary conditions, the governing equation may be converted to an equivalent set of wave equations. These wave equations represent nonlinear waves propagating at characteristic velocities. The characteristic velocities are given as the solutions to an eigenvalue problem. For 1-D problems, the wave propagation direction is well defined. For multidimensional problem, there is no unique direction of propagation. The boundary condition analysis used here only requires that one coordinate direction be diagonalizable at a time.

As an example, consider the boundary condition of r the x-direction of Eq. 1. Premultiplying Eq. 1 by the matrix, S^{-1}, where S is the similarity matrix of the Jacobian, $\partial E/\partial Q = A$, gives

$$S^{-1}\frac{\partial Q}{\partial t} + S^{-1}ASS^{-1}\frac{\partial Q}{\partial x} + S^{-1}H' = 0 \qquad (2)$$

Let's define a column vector *L* as follows

$$L=\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = S^{-1}ASS^{-1}\frac{\partial Q}{\partial x} \qquad (3)$$

L's defined as above are used for the equations corresponding to the outgoing waves while for the equations corresponding to incoming waves, the characteristic boundary treatment and non-reflecting boundary treatment have different approaches. In characteristic boundary treatment for incoming waves, flow properties are specified instead of solving the equations. On other hand, the non-reflecting boundary treatment defines L's differently for only the equations corresponding to incoming waves [1,2]

$$L_i = -\left(S^{-1}H'\right)_i \qquad (4)$$

here the subscript i represents the equation corresponding to the incoming wave. This condition constrains the amplitude of the incoming wave constant with time so that the outgoing waves are not allowed to reflect back into the domain.

Multigrid Protocol The schedule for the multigrid strategy (V-cycle) is as follows.

1) Make n_1 iteration sweeps on the finest grid.

2) Transfer this solution to the next coarser grid level by injection and perform n_2 iterations on that coarser grid.

3) Repeat step 2) until the coarsest grid level is reached.

4) On the coarsest grid level, compute the difference between approximated solution and the injected one from the previous finer level.

5) Transfer the above difference back to the finest grid level by a bilinear interpolation.

6) Update the finest grid solution by adding the transferred value to the old finest grid solution.

7) Repeat steps 1) through 6) until the solution converges.

Test cases A two dimensional viscous flow through a diverging channel was solved as the first test case. The channel starts with constant area section followed by a diverging section (having 5.7 degrees diffuser semi-angle) and ends with another constant area section. The exit has twice the area of the inlet and the upper and lower walls are symmetric about the center line. The finest grid level has 96x64 unclustered grid cells while successively coarser grids were obtained by skipping every second grid point in both directions. The Reynolds number Re = 100, CFL = 2.8, s (von Neumann number) = 0.4, b (artificial compressibility) = 3.0, and no artificial dissipation was introduced. Here n_1 (number of iteration sweeps on the finest grid level) = 3, n_2 =10 and n_3 = 1. Figure 1 shows the convergence histories for this case. In the beginning, faster convergence was obtained as the number of grid levels involved in the multigrid calculations is increased. However, the convergence histories of multigrid calculations begin to flatten out after certain convergence is reached (after residual dropped down one or one and half order of magnitude). A close investigation revealed that coarse grid solutions and fine grid solutions are locally heading in different directions, thus corrections transferred from coarse grid levels bring fine grid solution close to their old values obtained one iteration earlier. To avoid this , the convergence history is monitored and once convergence becomes significantly slower, the code switches to the single grid mode. This switching was applied to two grid level calculation (M=2) and the residual continued decreasing.

The second test case was a two dimensional U-shaped channel with constant cross section area. Two grid levels were used having 256x64 cells and 128x32 cells. Other parameters are the same as in the previous case except n_1 = 5 and n_2 = 20. As shown in Fig. 2, the residual drops quickly and flattens out. Again, calculation was switched automatically to a single grid for further convergence enhancement.

ACCELERATION SCHEMES BASED ON PRECONDITIONING

Many propulsion-related flowfields contain large regions of low speed flows where heat addition

or adjacent high speed flow regions dictate that the compressible equations be used. Representative examples include the reacting flow in the combustion chamber, the flow in ducts and seal regions or the upstream section of highly convergent nozzles. The compressible, time-marching algorithms that are frequently used for these flows encounter convergence difficulties because of the eigenvalue stiffness that appears at low Mach numbers. This slowdown can be avoided by introducing a preconditioning matrix to control the eigenvalues. Preconditioning alters the acoustic speeds of the system, making them of the same order as the particle velocity [3,4]. We have demonstrated [4] a preconditioning system for inviscid flows that provides convergence that is independent of Mach number. In the present section, we discuss preconditioning methods for the Navier-Stokes equations.

For our computational testbed, we use a Navier-Stokes code that handles either planar or axisymmetric geometries. Both implicit ADI and explicit multi-stage Runge-Kutta methods have been implemented. The preconditioned codes use second-order, central-differences in the flux discretizations. The effectiveness of the preconditioned Navier-Stokes code is demonstrated by the computation of nozzle flows. Convergence results for different nozzle area ratios, shapes and sizes show that preconditioning that scales both the inviscid eigenvalues and the diffusive time scales improves convergence by one or more orders of magnitude when there are significant regions of low Mach number flow. In addition, nozzle flows with Reynolds number ranging from very high to very low are computed.

The preconditioned Navier-Stokes equations are identical to the form given in Eq. 1 except that the time derivative is multiplied by a preconditioning matrix, Γ. For convenience, we also switch from the variable Q to the variable Q_V, which is defined as $Q_V=(p,u,v,T)^T$. The resulting set of equations then becomes,

$$\Gamma\frac{\partial Q_V}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial h}+\frac{\partial G}{\partial z}+H=0 \qquad (5)$$

The eigenvalues of this new system remain well conditioned at all Mach numbers from very low speeds through the supersonic regime. Careful choice of the parameters in the preconditioning matrix controls convergence as both Reynolds number and Mach number are varied over wide ranges. By the use of these parameters, simultaneous control of both the inviscid time-step (CFL number) and the viscous time-step (the von Neumann number) is obtained. This method has been shown to be effective in obtaining rapid convergence over a wide range of Mach numbers and Reynolds numbers [4].

PRECONDITIONING RESULTS

Convergence Acceleration Figure 3 illustrates the convergence rates with and without preconditioning for an axisymmetric nozzle of area ratio 0.01 (Area of throat/Area of inlet). In the absence of preconditioning, the initial convergence is reasonable, but it slows down dramatically after a few hundred iterations. The same figure shows the convergence of the same problem with preconditioning converges to machine accuracy in less than 500 steps. Consequently, preconditioning accelerates convergence by at least an order of magnitude for this problem.

The reason for the non-preconditioned convergence behavior in Fig. 3 can be seen from the flowfield solution. Figure 4 shows the converged Mach number contours corresponding to the above test case. The Mach numbers at the upstream end of the nozzle are around 10^{-3}. In the absence of preconditioning, the eigenvalue stiffness in the low Mach number region causes the solution to converge extremely slowly there. In the transonic region near the throat, the eigenvalues are well-conditioned and the convergence is quite good. This transonic region explains the initial rapid convergence of the non-preconditioned solution on Fig. 3. The errors in the transonic portion of the nozzle are rapidly eliminated until the slow convergence of the upstream region starts to dominate. Examination of the flowfield after 600 steps confirms the flowfield near the throat appears well-converged while that near the inlet is far from converged. When preconditioning is used, the eigenvalues are well-conditioned throughout the whole flowfield and the solution converges uniformly until machine accuracy is reached.

Viscous Nozzle Computations Figure 5 shows the convergence and converged Mach number contours for a viscous nozzle calculation (Area ratio=0.04, Inlet Re $=10^5$). The grid is suitably stretched in

the near-wall region. For the viscous case shows that machine accuracy is reached in about 1000 steps. Figure 5a also shows the convergence for an inviscid case computed on the stretched grid and on a uniform grid of the same size. These curves show grid-stretching causes both the viscous and inviscid solutions to converge slower than the uniform grid case. This is an Important observation since practical geometries often involve regions where strong grid stretching is employed. More detailed results show these differences can be partially removed by a proper definition of CFL.

Implicit ADI Vs. Explicit Runge-Kutta Equation 5 was solved using both an implicit ADI algorithm and an explicit four-stage Runge-Kutta algorithm. Figure 6 shows the convergence rate of both schemes for two different nozzle area ratios---0.8 and 0.04. The ADI scheme is seen to converge at approximately the same rate for the two cases while the Runge-Kutta scheme performance deteriorates as the nozzle contraction becomes stronger. This latter result is related to the fact that the 0.8 case was run with a CFL=2 while the 0.04 case was run with a CFL=1 (in both cases, these values correspond approximately to the CFL limit).

Low Reynolds Number Nozzles As a final example, the flow through a low Reynolds number conical nozzle used space-based applications is shown on Fig. 7a. The Mach number contours for a nozzle Reynolds number, Re = 25 are shown on Fig. 7a, while Fig. 7b shows the convergence rate. Again the preconditioning matrix provides reasonable convergence for this very low Reynolds number flowfield.

THE RELATIONSHIP BETWEEN PRESSURE-BASED AND DENSITY-BASED ALGORITHMS

As a third focus, we compare the characteristics of pressure-based algorithms with those of density-based algorithms. Pressure-based methods, include the Spalding-Patankar method [5], the MAC method [6] and derivatives. Here, we focus on the PISO method [7-8]. In general, pressure-based procedures use an uncoupled, sequential solution procedure, and rely on diagonal dominance. The density-based methods include the time marching schemes that were developed for transonic aerodynamics [9-10]. They do not require diagonal dominance, and allow the use of central differences in the inviscid limit. Recent trends have emphasized upwind differencing [11]. Density-based methods employ both implicit and explicit algorithms. The implicit formulations, upon which we concentrate here, are solved in a coupled fashion. In the present section, we write the compressible version of the PISO pressure-based scheme in a common format with an Euler implicit, density-based scheme and compare the algorithms. An important part of the comparison, is the expression of both schemes in a common vector form.

VECTOR FORMULATION OF THE PRESSURE-BASED METHODS

The equations of motion were expressed in vector form above. To simplify the notation, we deal here with the two-dimensional equations. For convenience, we also define a vector $Q_V=(p,u,v,T)^T$. The vector, Q_V, contains the primary dependent variables that appear in the PISO scheme. This change of variables is the first step in switching the equations from density-based to pressure-based relations.

Vector Equations for the PISO Algorithm The PISO algorithm is a time-dependent pressure-based scheme that is composed of a predictor step followed by two corrector steps [7]. In the PISO scheme, the inviscid flux vectors are split into two parts, a 'linear' and a 'nonlinear' part so their Jacobians can be written in a (lower or upper) triangular form that enables sequential solution. The terms in the linear part are centrally differenced, while the ones in the nonlinear vectors are generally upwind differenced. Some approximation is required before triangular matrices are obtained. The ultimate benefit of the uncoupling procedure depends upon the impact the approximations have on convergence.

The vector notation of the predictor and first corrector for the PISO algorithm for the compressible Navier-Stokes equations is:

$$\left\{I_{ce}+I_m\left[\Gamma+\Delta t\left(\frac{\partial}{\partial x}A_N{}^n+\frac{\partial}{\partial y}B_N{}^n-L_v\right)\right]\right\}\Delta Q^* = -\Delta t\, I_m\left[\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}-L_v(Q_V)\right]^n \qquad (12) \qquad (6)$$

$$\left\{\Gamma+\Delta t\left[\frac{\partial}{\partial x}A_L^{*}+\frac{\partial}{\partial y}B_L^{*}\right]\right\}\Delta Q^{**}+I_e\left(\frac{\partial}{\partial x}A_N^{*}+\frac{\partial}{\partial y}B_N^{*}-L_v\right)\Delta Q^{**}=-\Delta t\left\{\Gamma^{*}\Delta Q^{*}+\left(\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}\right)^{*}\right\} \quad (13)$$

with the second corrector similar. In these expressions, the variables ΔQ^{*}, ΔQ^{**}, ΔQ^{***}, are defined as $\Delta Q^{*}=Q_v^{*}-Q_v^{n}$, $\Delta Q^{**}=Q_v^{**}-Q_v^{*}$, and $\Delta Q^{***}=Q_v^{n+1}-Q_v^{**}$. The selection matrices, I_m=diag(0,1,1,0) and I_{ce}=diag(1,0,0,1), select the momentum equations and the continuity and energy equations respectively for the predictor step, while I_e=diag(0,0,0,1) selects the energy equation in the corrector step. These equations as written remain coupled. The method used to uncouple them is given below.

Uncoupling the PISO Algorithm The compressible PISO method relies primarily upon uncoupling the energy equation from the continuity and momentum equations by lagging the equation of state, and modifying the time derivative terms. This corresponds to making the density a function of only the pressure for the update of the continuity and momentum equations. The coefficient matrix of the time derivative on the left hand side of Eq.5 is replaced by an approximate matrix in which only the pressure derivative is retained in the continuity equation. The temperature derivatives in the momentum equations are also dropped. After the energy equation has been uncoupled , an expression for the pressure is obtained by combining the continuity and momentum equations to form the pressure Poisson relation.

The benefits of the sequential solution cannot be realized unless the approximations have an negligible or moderate effect on convergence. As a means of assessing this, we compare the vector stability characteristics of both the pressure- and density-based schemes.

Formulation of the Density-Based Methods The density-based methods take many forms [9-11]. Here we limit to a representative, implicit procedure;

$$\left[\Gamma+\Delta t\left\{\frac{\partial}{\partial x}A+\frac{\partial}{\partial y}B-L_v\right\}\right]\Delta Q=-\Delta t\left[\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}-L_v Q_v)\right]^{n} \quad 9), \quad (8)$$

where Γ is the identity matrix, and A and B are the Jacobians of the flux vectors. Here all four equations are mutually coupled , and the matrices A and B must be inverted at every grid point to obtain a solution.

STABILITY RESULTS

Euler Implicit Method The stability characteristics of a system of equations are determined by the eigenvalues of the amplification matrix which must generally be evaluated numerically. The four equations of motion generate four stability eigenvalues at every wave number. Here, we present the maximum of these four eigenvalues. Because the pressure- and density-based methods differ only in the treatment of the convective terms, we concentrate on the inviscid equations.

Stability results for the Euler implicit time-marching procedure are shown in Fig. 8. This figure shows the stability results at CFL=1, for flow Mach numbers of 0.3 and 0.03. At Mach 0.3, the amplification factors are reasonable, with a minimum value of about 0.94 at mid-wave-numbers. At Mach 0.03, the qualitative shape of the maximum eigenvalue surface is unchanged, but the stiffness of the eigenvalues raises the minimum amplification factor at the middle wave numbers to 0.999. Amplification factors this close to unity indicate very slow convergence, a result that is verified by actual computations. These amplification factors are lowered proportionally when the CFL is increased.

PISO Method Although the PISO algorithm uncouples the equations, the variables remain coupled and a vector stability analysis is still needed. With the equations written in vector form, Eqs. 6 and 7, the stability analysis follows by standard methods. The PISO results are for first-order upwind differencing of the convective terms.

Figure 9 shows the maximum eigenvalue at a Courant number of one for the compressible PISO algorithm at the same two Mach numbers shown above, 0.3 and 0.03. As can be seen, the PISO scheme is quite stable at this Courant number except in the low wave number region. The amplification factors exhibit a minimum in the mid-wave-number region much as they did for the Euler implicit case, but the magnitude of this minimum decreases as the Mach number is reduced, whereas the Euler-implicit method became less stable at lower Mach numbers. This is consistent with the commonly observed behavior of pressure-based schemes. The amplification factors at low wave numbers indicate a weak instability that is essentially independent of Mach number and persists in the incompressible limit. Numerical experiments demonstrate this instability can lead to divergence in computational solutions. Surprisingly, this instability disappears when central differencing is used for the convective terms. At higher values of the Courant number, the approximations used to uncouple the energy equation cause the PISO scheme to exhibit high wave number instabilities.

SUMMARY AND CONCLUSIONS

Two representative CFD algorithms, one pressure-based and one density-based, have been compared to determine their similarities and differences. The PISO algorithm has been taken as representative of pressure-based methods, while the Euler implicit algorithm has been chosen as representative of density-based methods. The PISO algorithm is a time-dependent method that uses a multi-step procedure composed of a predictor followed by two correctors. A key feature of the PISO algorithm is that it update the variables in an uncoupled, sequential manner. Density-based methods generally use a coupled solution procedure.

The characteristics of the two methods are compared by expressing them both in vector form. When written in this fashion, the flux vectors in the PISO scheme are triangular in nature, while those of the density-based method contain no natural character to facilitate solution. The triangular matrices in the PISO scheme are obtained by splitting the convective fluxes into two parts and using an approximate linearization to remove the rest of the coupling between the variables.

Comparison of the stability characteristics of pressure-based and density-based methods indicates there are substantial differences. The amplification factors of the pressure-based methods provide stability at small CFL's, but become unstable at large time steps. This conditional instability of the overall algorithm arises despite the fact that the amplification factors of the individual steps are less than unity. This is a clear result of the fact that even in the uncoupled solution procedure, the individual variables remain coupled, and propagate errors as a system. Stability analyses demonstrate that the difficulties arise from the approximate linearization of the time derivative that is used in the PISO algorithm.

A transformation of the density-based procedure from the conservative ($\rho, \rho u, \rho v, e$) to the 'viscous' (p,u,v,T) variables that are used in the PISO method along with a preconditioning matrix provides an Euler implicit method with amplification factors similar to the PISO method and Mach and Reynolds number independent convergence. The resulting expression behaves in the same manner at low speeds as does the PISO scheme, while it returns back to the traditional time-marching method at transonic and supersonic Mach numbers, thereby providing excellent convergence at all Mach numbers.

REFERENCES

1. Thomson, Kevin W. , "Time-dependent Boundary Conditions for Hyperbolic Systems, II," J. of Computational Physics 89, pp439-461, 1990.

2. Thomson, Kevin W. "Time-dependent Boundary Conditions for Hyperbolic Systems, II," J. of Computational Physics 68, pp1-24, 1987.

3. Merkle, C. L. and Choi, Y.-H., ``Computation of Low Speed Flows with Time-Marching Procedures," International Journal for Numerical Methods in Engineering, Vol. 25, 1985, pp. 293-311.

4. Choi, Y.-H. and Merkle, C. L., ``Time-Derivative Preconditioning in Viscous Flows," AIAA-91-1652, 22nd Fluid Dynamics, Plasmadynamics and Lasers Conference, Honolulu, Hawaii, June 24-27, 1991.

5. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Series in Computational Methods in Mechanics and Thermal Sciences, McGraw-Hill Book Company, 1980.

6. Harlow, F. H. and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surfaces", Physics of Fluids, Vol. 8, No. 12, Dec. 1965, pp. 2182-2185.

7. Issa, R. I., "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting", Journal of Computational Physics, Vol. 62, 1986, pp. 40-65.

8. Issa, R. I., Gosman, A. D. and Watkins, A. P., "The Computation of Compressible and Incompressible Recirculating Flows by a Non-Iterative Implicit Scheme", Journal of Computational Physics, Vol. 62, 1986, pp. 66-82.

9. Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form", Journal of Computational Physics, Vol. 22, 1976, pp. 87-110.

10. Briley, W. R. and McDonald, H., "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method", J. Computational Physics, Vol. 24, 1977, pp. 372-397.

11. Roe, P. L., "Characteristic-Based Schemes for the Euler Equations", Annual Review of Fluid Mechanics, Vol. 18, 1986, pp. 337-365.

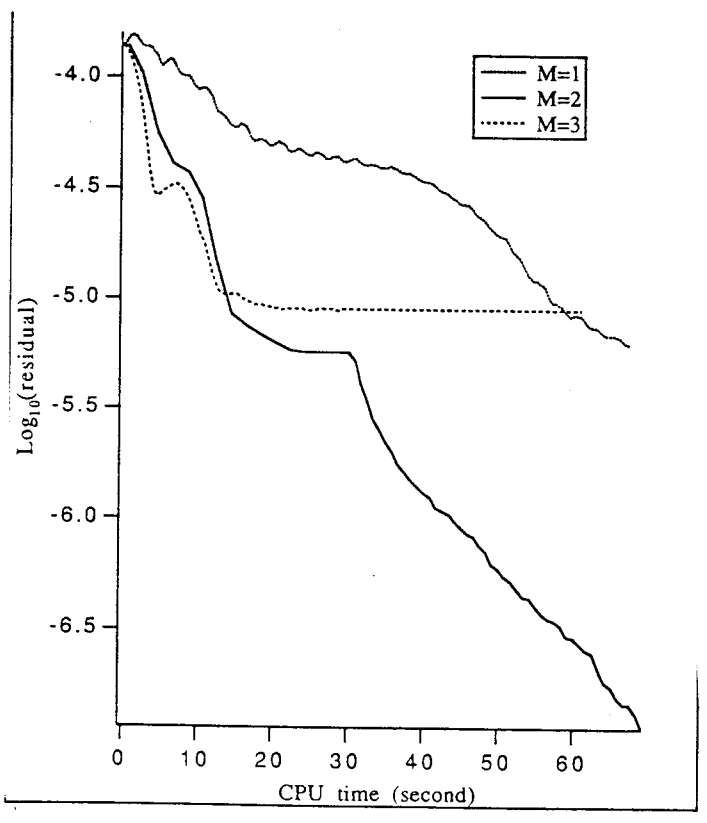


Figure 1. Convergence histories for a diverging channel (Re=100): M =1, single grid level; M=2, two grid level with automatic switching; M=3, three grid level without automatic switching

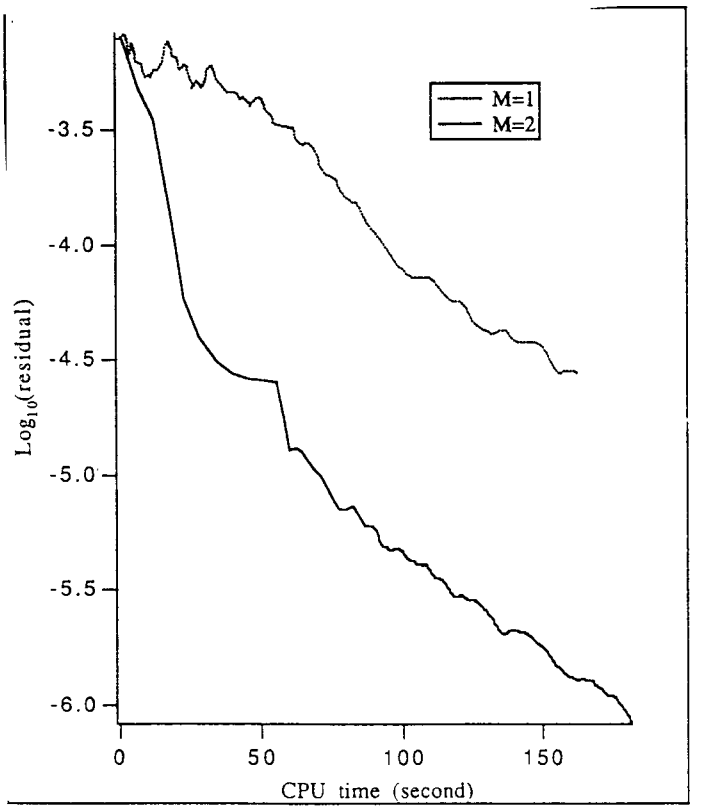


Figure 2. Convergence histories for a U-shaped channel (Re=100): M=1, single grid level; M=2: two grid level with automatic switching

17. Merkle, C. L. and Abdullah, S. "In preparation.

18. Withington, J. P., Shuen, J. S., and Yang, V., "A Time Accurate, Implicit Method for Chemically Reacting Flows at All Mach Numbers", AIAA Paper 91-0581, AIAA 29th Aerospace Sciences Meeting, Jan. 1991, Reno, NV.

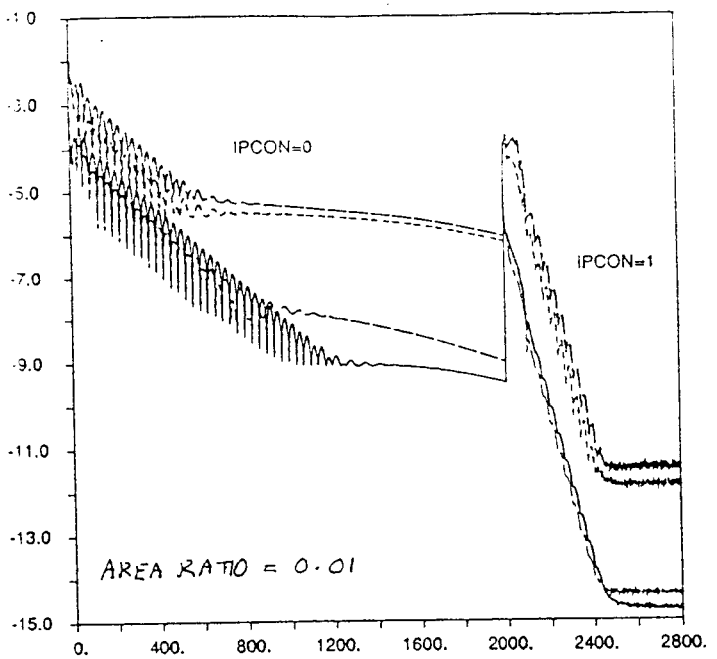
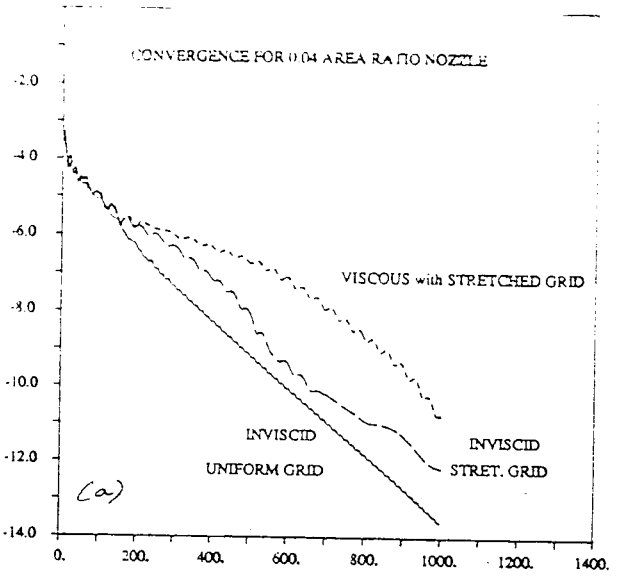


Fig. 3 - Convergence with and without preconditioning for a 0.01 Area Ratio Nozzle



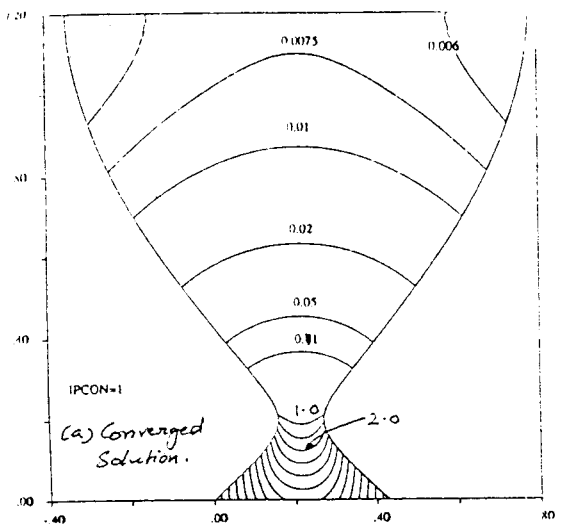


Fig. 4 - Mach Number Contours for 0.01 Area Ratio Nozzole (Invisud Solution)

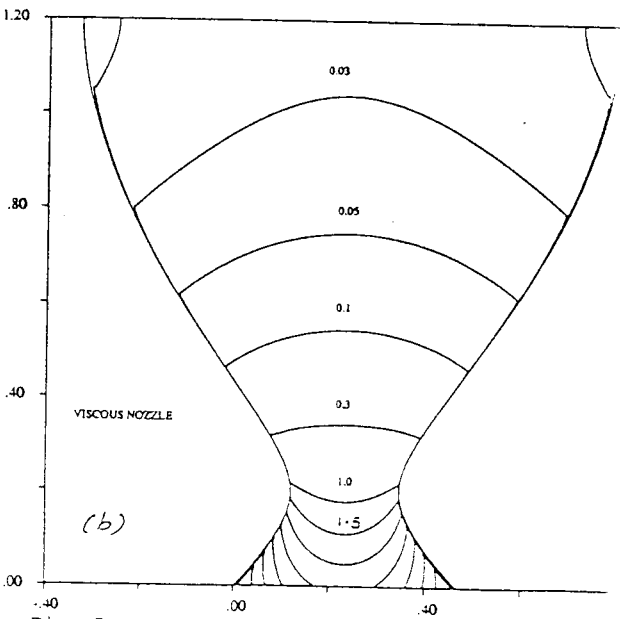


Fig. 5 - Convergence and converged Mach Number Contours for a 0.04 Area Ratio Nozzel. Viscous calculation Re = 1×10^5

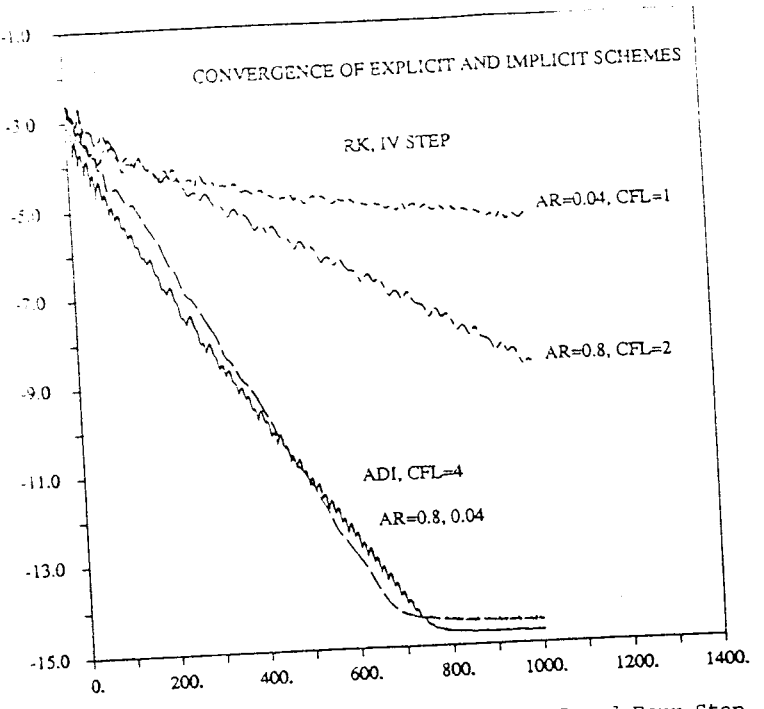


Fig 6 - Convergence using Implicit ADI and Four Step Range kulta for two nozzle - a.t = 0.8 and A.R. = 0.04

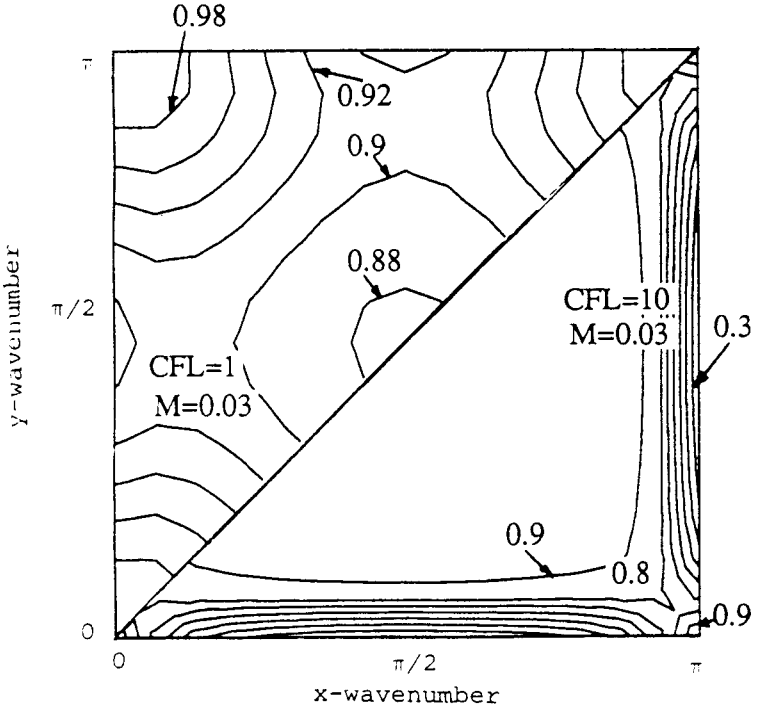
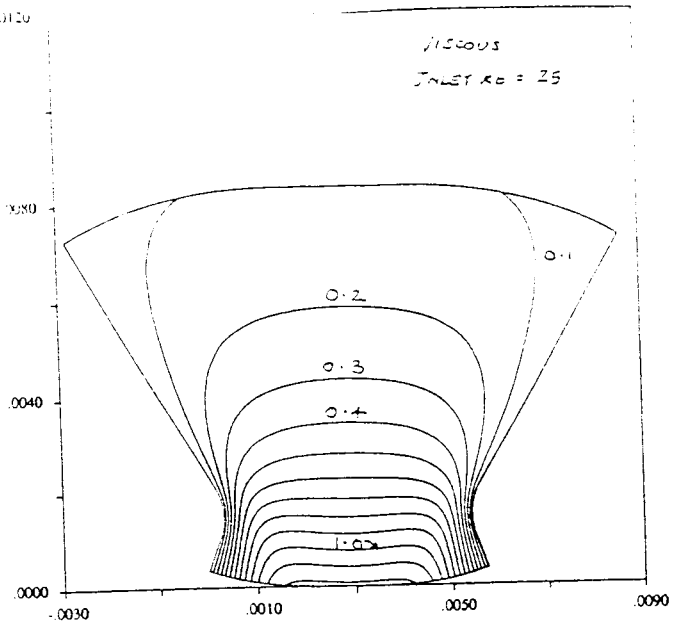


Fig. 8 Amplification factors for approximately-factored, pre-conditioned system showing effects of CFL. Upper triangle, CFL=1, lower triangle, CFL=10, M=0.03.



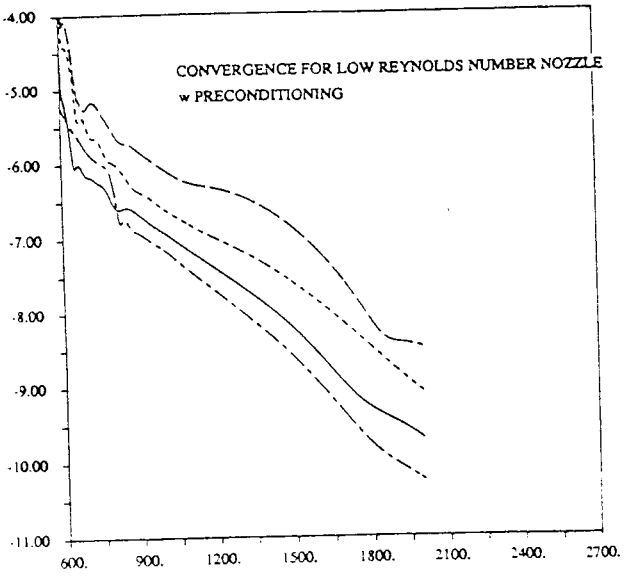


Fig. 7 - Mach Number Contours and Convergence Ratio for Low Reynolds Number Nozzle Re = 25

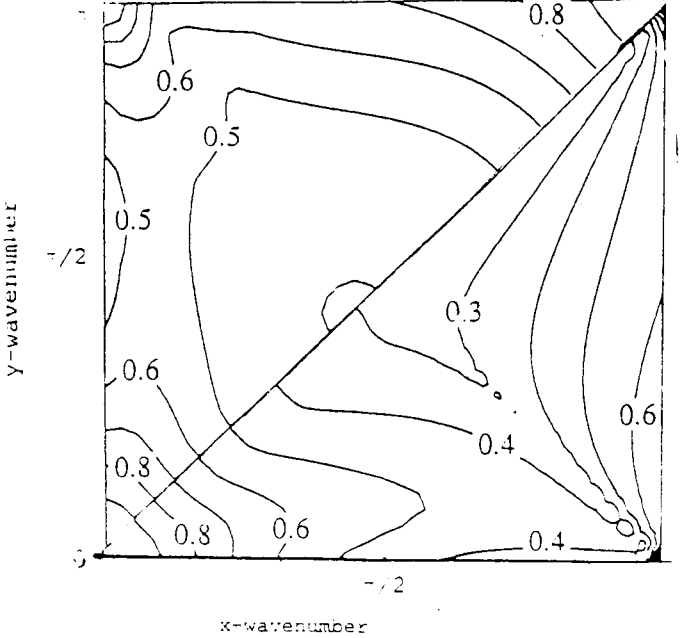


Fig. 9 Amplification factors for PISO algorithm. CFL=1. Upper triangle, M=0.3; lower triangle,