July, 1988

Analysis of the convergence history of fluid flow through nozzles with shocks

Mohammad Saleem, *San Jose State University*
A Cheer, *University of California - Davis*
M Hafez, *University of California - Davis*
T Pulliam

88-3795-CP

Analysis of the Convergence History of flow through Nozzles with Shocks

by

A. Y. Cheer*, M. Saleem
Dept. of Mathematics, U.C. Davis,

T. H. Pulliam
Computational Fluids Branch
NASA-Ames Research Center,

and
M. Hafez
Dept. of Mechanical Engineering, U.C. Davis.

Abstract

Convergence of iterative methods for the solution of the steady quasi-one-dimensional nozzle problem with shocks is considered. The finite-difference algorithms obtained from implicit schemes are used to approximate both the Euler and Navier-Stokes Equations. These algorithms are investigated for stability and convergence characteristics. The numerical methods are broken down into their matrix-vector components and then analyzed by examining a subset of the eigensystem using a method based on the Arnoldi process. The eigenvalues obtained by this method are accurate to within 5 digits for the largest ones and to within 2 digits for the ones smaller in magnitude compared to the eigenvalues obtained using the full Jacobian. In this analysis we examine the functional relationship between the numerical parameters and the rate of convergence of the iterative scheme.

Acceleration techniques for iterative methods like Wynn's ε-algorithm are also applied to these systems of difference equations in order to accelerate their convergence. This acceleration translates into savings in the total number of iterations and thus the total amount of computer time required to obtain a converged solution. The rate of convergence of the accelerated system is found to agree with the prediction based on the eigenvalues of the original iteration matrix. The ultimate goal of this study is to extend this eigenvalue analysis to multi-dimensional problems and to quantitatively estimate the effects of different parameters on the rate of convergence.

I. Introduction

Over the past decade a multitude of algorithms and computer codes has evolved in the attempt to model aerodynamic flows. A major expense in these efforts is the large amount of computer time required for the solution of any realistic problems. In particular, large numbers of iterations are required to achieve convergence of the solutions. The effort to minimize these costs has spawned the development of many numerical methods and acceleration techniques whose time iterations are designed to converge rapidly to the steady state solution of the governing equations. The success of these methods has been sporatic and to date there has been little mathematical analysis on what factors determine whether a given numerical method or acceleration technique will result in more rapid convergence particularly when applied to a system of non-linear partial differential equations.

In the present study we utilize acceleration techniques such as Wynn's ε-algorithm and analysis techniques such as eigensystem analysis to numerically study the convergence properties of an iterative scheme applied to the quasi-one-dimensional Euler and Navier-Stokes equations for flow through nozzles with shocks. Using a time-dependent eigensystem analysis we study the convergence and stability properties by analyzing the dependence of convergence of the code on the discretization technique, boundary conditions, time-step, number of grid points, and the physics of the problem.

*Research supported in part by NASA grant #NCA2–172

An outline of the equations and the iterative scheme is presented followed by a description of the method of Arnoldi and Wynn's ε-algorithm.

II. Equations and Iterative Scheme

The system of equations we are considering is:

$$\frac{\partial Q}{\partial t} + \frac{1}{a}\frac{\partial (aF)}{\partial x} = S + \frac{1}{Re}\frac{\partial F_\nu}{\partial x} \qquad (1)$$

where
$$Q = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e + p) \end{bmatrix}$$

$$S = \begin{bmatrix} 0 \\ \frac{p}{a}\frac{\partial a}{\partial x} \\ 0 \end{bmatrix}, \quad F_\nu = \begin{bmatrix} 0 \\ \tau_x \\ f \end{bmatrix}$$

$$p = (\gamma - 1)(e - \tfrac{1}{2}\rho u^2), \quad c^2 = \frac{\gamma p}{\rho}$$

$\tau_x = \frac{4\mu\, u_x}{3}$, a = variable cross-sectional area of the nozzle, where

a(x) = 1-4 * (1 - athroat)* x * (1-x), and

$$f = u\,\tau_x + \frac{\mu}{(\gamma - 1)}(Pr)^{-1}\frac{\partial}{\partial x}(c^2), \; 0 \le x \le 1.$$

Here Pr is the Prandtl number, Re the Reynolds number, p the pressure, c the speed of sound, F the flux vector, ρ the density, u the velocity and e the energy.

These equations are discretized in the following way:

$$\frac{Q^{N+1} - Q^N}{h} + \frac{1}{a}\frac{\partial}{\partial x}(aF)^{N+1} \qquad (2)$$

$$= S^{N+1} + \frac{1}{Re}\frac{\partial F_\nu^{N+1}}{\partial x}$$

where h = Δt, and t = time.

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial Q}\frac{\partial Q}{\partial x} = A\frac{\partial Q}{\partial x}, \; A = \text{Flux Jacobian.}$$

The nonlinear terms linearized about Q^N are:

$$F^{N+1} = F^N + \frac{dF^N}{dQ^N}(Q^{N+1} - Q^N) + \ldots \qquad (3a)$$

$$S^{N+1} = S^N + B^N (Q^{N+1} - Q^N) + \ldots \qquad (3b)$$

$$F_\nu^{N+1} = F_\nu^N + M^N (Q^{N+1} - Q^N) + \ldots, \qquad (3c)$$

where $M^N = \frac{dF_\nu^N}{dQ^N}$, $B^N = \frac{dS^N}{dQ^N}$

Substituting these equations into equation (1) we obtain:

$$\left[I + \frac{h}{a}\frac{\partial(aA^N)}{\partial x} - \left(B^N + \frac{\partial}{\partial x}\left(\frac{M^N}{Re}\right)\right)\right]\Delta Q^N \qquad (4)$$

$$= h\,S^N + \frac{h}{Re}\frac{\partial F_\nu^N}{\partial x} - \frac{h}{a}\frac{\partial}{\partial x}(a\,F^N)$$

A combination of second and fourth order dissipation terms of the following form is added:

$$-(1/a)\,\Delta t\,\nabla_x (\sigma a\,[d_2 - d_4\Delta_x\nabla_x])\,\Delta_x Q^N \qquad (5)$$

where $\sigma = |u| + c$, $c^2 = (\gamma p)/\rho$,
$d_2 = \varepsilon_2 \Phi$, $\Phi = |\Delta_x \nabla_x p| / |p|$, and
$d_4 = \varepsilon_4 - \min(\varepsilon_4, d_2)$.

The parameter ε_2, which is denoted by EPS2 in the rest of this paper, is chosen to be 1.0 and ε_4 = EPS4 = 0.01. The dissipation operator is nonlinear and enhances the shock capturing accuracy of the algorithm. The terms are added both explicitly and implicitly to equation (4). The difference operators are applied at all interior points. More details can be found in [5,6].

The iteration process condenses to

$$M\,\Delta Q^N = R \qquad (6)$$

where M is the matrix containing boundary conditions, implicit smoothing and fluxes, and R is the right-hand side which incorporates the source vector S, the viscous fluxes and any explicit dissipation terms. Since both M^{-1} and R are functions of Q^N, we can thus write the iteration process as:

$$Q^{N+1} = F(Q^N) \quad (7)$$

where $F(Q^N) = Q^N + M^{-1}R$.

For a more thorough discussion please see [4,5].

III. Arnoldi's Method applied to the Nozzle Code

To obtain a subset of the eigenvalues of the above operator $F(Q^N)$, whose exact form at time t is unknown, we use a method based on the Arnoldi process [1]. Rather than form the matrix of the system analytically by hand, we use the computer to linearize our system by numerically approximating the Frechet derivatives where

$$\frac{dF}{dQ} \cdot v = \frac{F(Q+\varepsilon v)-F(Q-\varepsilon v)}{2\varepsilon} + O(\varepsilon^2) \quad (8)$$

where v is an arbitrary vector and $\varepsilon = 0.001 * (\|Q\|_2/\|v\|_2)$. The method is capable of extracting a subset of the total spectrum and only requires the Frechet derivatives for arbitrary vectors v to accomplish this. The crux of the algorithm can be condensed into the following. Starting with an arbitrary starting vector v_1, we construct m orthonormal vectors spanning a Krylov subspace [2]:

Let $e_1 = (v_1^T v_1)^{-1/2} v_1$,

for k = 1,m:

$$v_{k+1} = A\, e_k - \sum_{i=1}^{k} h_{jk}\, e_j\,;\ h_{jk} = e_j^T A\, e_k$$

$$h_{k+1,k} = (v_{k+1}^T, v_{k+1})^{-1/2}$$

$$e_{k+1} = (h_{k+1,k})^{-1} v_{k+1}$$

next k.

The m eigenvalues of the matrix $H = [h_{jk}]$ are approximations to the eigenvalues of $[A]$, where $[A] = [dF/dQ]$ is the Jacobian matrix in our case. These eigenvalues are a good approximation to the desired subset of eigenvalues we seek. For a more thorough discussion on the method please see [7,1].

IV. Wynn's ε–Algorithm applied to the Nozzle code

To accelerate the convergence of our iterative scheme we applied Wynn's ε-algorithm [9] which we now summarize.

Let $\varepsilon_{-1}^n = 0,\ \varepsilon_0^n = Q^n,\ n \geq 0$

where Q^n is the solution vector at time $t = n\,\Delta t$,

then $$\varepsilon_{k+1}^n = \varepsilon_{k-1}^{n+1} + (\varepsilon_k^{n+1} - \varepsilon_k^n)/\mu^2\,,\ n,k \geq 0 \quad (9)$$

where $\mu^2 = (\varepsilon_k^{n+1} - \varepsilon_k^n)\,(\varepsilon_k^{n+1} - \varepsilon_k^n)$

The value ε_{2m}^n is the same as the m^{th} order Shanks transformation [3]. If a table is constructed consisting of columns $\varepsilon_{-1}{}^n, \varepsilon_0{}^n, \varepsilon_1{}^n, \ldots$, the odd columns can be eliminated leaving only the even columns. The entries in these columns ε_{2m}^n are obtained by the following formula [3]:

$$E = \omega_1 N + \omega_2 S + \theta\,\omega_3 W + \omega_4 C \quad (10)$$

where $\omega_1 = (\delta^2/\alpha^2)$, $\omega_2 = (\delta^2/\beta^2)$,
$\omega_3 = (\delta^2/\gamma^2)$, $\omega_4 = 1-\omega_1-\omega_2 + \theta\,\omega_3$,
$\alpha^2 = \|N - C\|^2$, $\beta^2 = \|S - C\|^2$,
$\delta^2 = \|E - C\|^2$, $\gamma^2 = \|W - C\|^2$,

and where the new value E is a combination of the value to the North (N), the value to the south (S), the value to the west (W) and the value in the center (C). Except for the first column, where $\theta = 0$, the value for $\theta = 1$. Please refer to [3,9] for details.

V. Description of the Problem

We apply the numerical method presented in Section II to calculate flow through a nozzle with shocks in one-dimension under the following flow conditions. The nozzle is of length 1, the area at either end is 1, the area at the throat is 0.8 and the location of the shock is at 0.7. At the entrance of the nozzle, the Mach number is 0.553, the density is 1 and the pressure is 1.0. Although the viscous effects are not important in this 1-D nozzle problem, they are studied because of their importance in multidimensional problems.

VI. Numerical Results

(a) Application of Arnoldi's Method

When applying the algorithm based on the Arnoldi Method outlined in section III to the nozzle code we note that the eigenvalues obtained from this method (compared to the eigenvalues obtained with the full jacobian) are accurate to within 10^{-5} for the largest ones in magnitude, and accurate to within 10^{-2} for the smaller ones in two iterations. For error estimates please see [7].

(b) Eigensystem analysis of the 1-D Nozzle code

In this section we analyze how changes in the numerical parameters of the iterative scheme affect the rate of convergence of the calculations. More precisely, we will discuss how changes in various parameters affect the eigenvalues (and thus the rate of convergence) of the nozzle operator (7) for the problem described in section V. The parameters we examine are (a) the number of iterations, (b) grid size, (c) Dirichlet and linear extrapolated boundary conditions, (d) size of the time-step and (e) addition of 2nd and 4th order dissipation terms. For the viscous case, we also study the effects on the eigenvalues due to changes in the Reynolds number. Results are discussed below and summarized in Table Ia.

Presently, the discussion will be restricted to the case of discretization of the Euler equations with linear boundary conditions, and with the number of grid points jmax = 100. The magnitude of the largest eigenvalue $|\lambda_1|$ for the inviscid equations at iterations n = 50, 75, 100 and 200, is $|\lambda_1|$ = 0.97519, 0.97175, 0.97042, 0.96936 respectively. The magnitude of the largest eigenvalue decreases as n increases and approaches the eigenvalue of the steady configuration. This indicates that the rate of convergence may start out slowly but improves to the asymptotic rate of convergence as the number of iterations n increases. This is indicated in figure 6 where the slope of the residual approaches a constant after about 125 iterations of the code. This behavior is observed for the viscous case as well where $|\lambda_1|_{viscous}$ = 0.9792, 0.9761, 0.9746, and 0.9732 respectively at Re = 1000. The eigenvalues for both of these cases are plotted in figures 1 & 2. Note that in figure 1 there are two eigenvalues with negative real parts denoted by the symbol ⊗ but plotted on the positive real axis.

The number of spatial grid points (jmax) is a measure of the size of the Jacobian dF/dQ. For jmax = 10, 30, 50, 70, 90 and 100, and linear boundary conditions, $|\lambda_1|$ = 0.8850, 0.9689, 0.9694, 0.9693, 0.9693, and 0.9693 respectively. Initially $|\lambda_1|$ increases as the number of grid points increases until jmax = 50. As jmax increases, both the total number of eigenvalues and the number of eigenvalues of large magnitude increase. For jmax = 30 there are 12 eigenvalues greater than 0.5. This number increases to 15 for jmax = 100. This indicates that for this case, the rate of convergence and acceleration methods such as eigenvalue annihiliation will depend on the grid size as well as other numerical parameters.

Boundary conditions affect convergence by changing the magnitude of the largest eigenvalue. In this study the results of applying linear extrapolated boundary conditions and Dirichlet boundary conditions are compared. The linear extrapolated characteristic boundary conditions at the entrance are obtained by forming a linear combination of the solution in the previous step. The quantities at the left boundary are:

$$\rho_1 = 2\,\rho_2 - \rho_3 \tag{11a}$$

$$u_1 = (1/2)\,\rho_1(R1 - R2) \tag{11b}$$

$$e_1 = \rho_1/(\gamma-1) + (1/2)\,\rho_1 u_1^2 \tag{11c}$$

where R1 and R2 are the 1-D Riemann invariants of the flow and the subscripts indicate the value of the variable at that grid point. The Dirichlet boundary conditions are taken to be the values obtained from the exact solution. The maximum eigenvalue for the case of linear extrapolated characteristic boundary conditions is 0.96933 whereas it is 0.93951 for the Dirichlet boundary conditions. This translates to roughly one half the number of iterations required to achieve maximum error in the residual of 1.0×10^{-10} using Dirichlet boundary conditions as compared to linear boundary conditions (see table Ia). We note that a difference in the eigenvalues in the third digit translates to roughly 20 iterations of the code with linear extrapolated boundary conditions and 10 iterations of the code with Dirichlet boundary conditions.

For both the Euler and the viscous results, the eigenvalues with large absolute values are well separated. This is expected since the local effects of viscosity on the shock structure are minor. As a result, the nozzle code is well suited for convergence acceleration schemes.

For the viscous case, we calculate flow at Re = 1000, 5000 and 10000 and note that the viscous eigenvalues approach the Euler eigenvalues in the limit from above – that is, the eigenvalues decrease as the Reynolds number increases. In the limit as the Reynolds number goes to infinity, the eigenvalues of the viscous case converge to the eigenvalues of the Euler case.

The effect of changing the time-step, (Δt is varied between 0.15 and 0.30), is observed. At $\Delta t = 0.3$, the residual starts oscillating after time-step 200 and a converged solution is not obtained. This is consistent with the fact that the largest eigenvalue is greater than one in magnitude (see table Ia). For the cases where the chosen Δt gives a convergent solution, we see that as Δt is changed the magnitude of the largest eigenvalue also changes. The relationship between the time step and the maximum eigenvalue is such that there is a given time step that will minimize the maximum eigenvalue. This is in accordance with linear stability analysis of the numerical method [6].

In order to study the effects of 2nd and 4th order dissipation on the eigenvalues, we first run the code 300 time-steps until the residual is less than $4x10^{-7}$. After 300 iterations, the dissipation terms are altered. The eigenvalues obtained are thus the ones associated with this altered system of equations.

In the inviscid case with both 2nd and 4th order dissipation included, there are 15 eigenvalues with modulus greater than 0.5, and the largest ones are well separated with $|\lambda_1| \cong 0.9693319$. There are also two eigenvalues with negative real parts (see figure 1). When 2nd order dissipation is turned off (EPS2 = 0) the effect on the largest eigenvalue is minimal $|\lambda_1|_{EPS2=0} = 0.9693434$. We note that, in contrast to the case when both 2nd and 4th order dissipations are included, there are no eigenvalues with negative real parts for this case (see figure 3). The larger end of the spectrum remains well separated with $|\lambda_1|_{EPS2=0} - |\lambda_2|_{EPS2=0} \cong 0.144434$.

Removing 4th order dissipation but keeping 2nd order dissipation in the calculations, has a more dramatic effect on the spectrum. We note that the distribution of the eigenvalues is quite different in this case than in the previous cases (compare figures 1, 3 and 4). In this case the smallest eigenvalues that are clustered together about zero tend to become more distinct (see figure 4), and the eigenvalues are more spread apart. The number of eigenvalues larger than 0.5 in magnitude increases from 11 to 39. The largest eigenvalues occur in a complex conjugate pair and $|\lambda_1|_{EPS4=0} - |\lambda_3|_{EPS4=0} \cong 0.1154$. This indicates that in order to accelerate the convergence for this case the effects of at least two eigenvalues must be corrected for. Moreover, as the number of eigenvalues of large magnitude increases, one must annihilate the contribution of more eigenvalues to accelerate the rate of convergence.

When both 2nd and 4th order dissipation terms are turned off completely after 300 iterations, our calculations give two eigenvalues with magnitudes greater than 1 indicating instability in the numerical algorithm as expected (see figure 5). Moreover, the eigenvalues are further spread apart in a circle.

(c) Using the eigensystem analysis to predict unstable behavior

As a numerical experiment, we apply the numerical algorithm described in the body of the paper to calculate flow through a nozzle under the same conditions as before with linear boundary conditions except (1) the entrance mach number is now 2.0 (supersonic), and (2) the shock is at $x = 0.3$ which is upstream of the nozzle throat. This problem is known to have an unstable solution [8].

To analyze the solution to this problem we first obtain the exact starting values for density and pressure at each grid point according to the following equations. The density and pressure are obtained by:

$$\Delta\rho = \rho_{i+1} - \rho_i = \rho\{M^2/(1-M^2)\}\{\Delta a/a\}, \tag{12a}$$

$$\rho u\, a(x) = \text{constant}, \tag{12b}$$

and

$$(e + p)/\rho = \text{constant}. \tag{12c}$$

Here ρ is the density, p the pressure, M is the Mach number, u the velocity and a the area. The Rankine-Hugoniot shock jump relation for the density is,

$$\Delta\rho_{shock} = \rho_2 - \rho_1 = \frac{\rho_1(M_1^2 - 1)}{1 + \frac{M_1^2}{2}(\gamma-1)} \tag{13}$$

Based on this solution, we form the Jacobian using Frechet derivatives and calculate the eigenvalues of the system using both Arnoldi's method and an IMSL eigenvalue routine on the full Jacobian. The spectrum for this system indicates that this system is unstable since it includes six eigenvalues of magnitude greater than 1.0. When the above values are used as initial conditions in our code, the solution blows up in 20 iterations thus verifying the instability of this solution.

This exercise demonstrated that by obtaining the eigenspectrum, we could predict unstable behavior.

(d) Convergence Acceleration Analysis

Based on the results of the eigenvalues analysis of the previous sections, we know that the eigenvalues of largest magnitudes are separated. This indicates that the result of applying acceleration techniques like Wynn's ε-algorithm can produce a dramatic reduction in the error. Moreover, since we know the magnitude of the eigenvalues, we can predict the rate of convergence when the effect of the eigenvalues on the solution are corrected for. In this study, we applied Wynn's ε-algorithm using 3, 5, 7, 9 and 11 terms. Results are summarized in Table 1(b).

Let ΔQ^n = residual after n iterations of the code. The following formulas are used for an estimate of the largest eigenvalue

$$\lambda_1 \cong (\Delta Q^n/\Delta Q^0)^{1/n}, \tag{14}$$

and the rate of convergence

$$R \cong -\log[(\Delta Q^n/\Delta Q^0)^{1/n}]. \tag{15}$$

Using these formulas we obtain $|\lambda_1| \cong 0.9693$ for the Euler case with linear boundary conditions, and $|\lambda_1|_{viscous} \cong 0.9735$ for the viscous case at $Re = 1000$. These values agree with those obtained from the method of Arnoldi to four decimal places. After Wynn's ε-algorithm is applied once with 5 terms, we find that the estimates of the largest eigenvalues using equation (14) is 0.691 for the Euler case (see table Ib, column 4). In this estimate, ΔQ^0 is taken to be the residual at the iteration before the update, and ΔQ^n to be the residual n iterations after the update. This measures the effect of the acceleration step on the residual. When ΔQ^0 is taken to be the first iteration after the update, the estimate of the largest eigenvalue is 0.7813. These values indicate that the error introduced into the solution by the first five eigenvalues are initally corrected for and thus reducing the amplification of the error from $|\lambda_1|^n$ to $|\lambda_6|^n$. The sixth largest eigenvalues of the system is $|\lambda_6| \cong 0.6928$ for the Euler case. As the number of iterations increase, the effect of the acceleration step is diminished. The slope of the residual returns to that of the unaccelerated case after about 60 iterations (see figure 6). When acceleration is applied again at this point, we see that the slope of the residual is even steeper. This is because the estimates of the eigenvalues are more accurate than before since more accurate iterates are used. These results are summarized in table (Ib), and plotted in figure 6. Here we plot the resulting residual after 3, 5, and 9 terms are used in Wynn's ε-algorithm to update the solution at iterations #200 and #300. The rate of convergence, using equation (15) for the case when 5 terms are used in Wynn's ε-algorithm, is 0.0135 before and 0.160 immediately after acceleration. The rate of convergence based on the 6th eigenvalue is 0.159. Figure 7 plots the resulting residual when 3 and 5 terms are used in Wynn's ε-algorithm to accelerate the convergence for the case Re = 1000.

VII. Summary

A method of extracting a subset of eigenvalues based on the method of Arnoldi is tested and the eigenvalues obtained are satisfactorily accurate as compared to the case when the full Jacobian was used. This is then applied to the iterative scheme of section II to study its convergence and stability properties by solving the nozzle problem with shocks in 1-D. The eigenvalues obtained indicate how the rate of convergence depends on numerical parameters like grid-size, time-step, number of iterations, boundary conditions and artificial dissipation. Knowing the magnitudes of the largest eigenvalues gives information on the stability and rate of convergence of the numerical scheme. Moreover, they give information on whether the numerical method is amenable to acceleration techniques like eigenvalue annihilation and Wynn's ε-algorithm. For our case, the above mentioned acceleration techniques are successful when applied since the eigenvalues of largest magnitudes are well separated. The prediction of the rate of convergence based on the eigenvalue analysis agrees with the values obtained numerically with and without acceleration. This analysis technique is being extended to study inviscid and viscous flows in both 2 and 3-dimensions where the viscous effects have a much greater influence on the solution.

References

[1] Arnoldi, W. E., (1951) "The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem:, Quart. Appl. Math. Vol. 9, No. 1, pp. 17-29.

[2] Eriksson, L. E. & Rizzi, (1985) A computer aided analysis of the convergence to steady state of discrete approximations to the Euler equations. J. Comp. Physics. Vol. 57, No. 1, pp. 90-128.

[3] Hafez, M., Palaniswamy, S., Kuruvilla, G., and Salas, M.D. (1987) Applications of Wynn's ε-algorithm to Transonic Flow-Calculations. AIAA 87-1143.

[4] Jesperson, D. C. and Pulliam, T. H. (1983) Approximate Newton Methods and Flux Vector Splitting, AIAA -83-1899, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA. July 13-15, 1983.

[5] Pulliam, T. H. (1986) Implicit solution methods in computational fluid dynamics: Applied Numerical Math., Vol. 2, No. 6, pp. 441-474.

[6] Pulliam, T. H. (1985) Efficient Solution Methods for Navier-Stokes Equations, Lecture notes for the Von Karman Institute for Fluid Dynamics. Lecture series: Numerical Techniques for viscous flow Computations in Turbomachnery Bladings, Von Karman Institute, Rhode-St-Genese, Belgium.

[7] Saad, Y. (1980) Variations on Arnoldi's method for computing the eigenelements of large unsymmetric matrices. Linear Algebra & Applications, 34:269-295.

[8] Shapiro, A. "The Dynamics and Thermodynamics of Comp. Fluid Flow" The Ronald Press Co. New York, 1954.

[9] Wynn, P., (1966) On the Convergence and Stability of the Epsilon Algorithm, J. SIAM Num. Anal. Vol. 3, No. 1, pp. 91-122.

TABLE I (a) Effects of different time steps and boundary conditions

Iterations	**Euler -- dt=0.20** Linear Char. BCs	Dirichlet BCs	**Viscous -- dt=0.20 ** Re # 1000 Linear BCs	Re # 1000 Dirichlet	**** V A R I A B L E T I M E S T E P **** [INVISCID FLOW, DIRICHLET BOUNDARY CONDITIONS] dt=0.15	dt=0.175	dt=0.25	dt=0.275	dt=0.3
1	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}
100	1.81×10^{-4}	3.3×10^{-5}	6.34×10^{-5}	2.78×10^{-5}	1.28×10^{-4}	6.45×10^{-5}	1.24×10^{-5}	2.98×10^{-5}	2.42×10^{-4}
200	8.56×10^{-6}	6.54×10^{-8}	4.78×10^{-6}	4.17×10^{-7}	1.17×10^{-6}	2.76×10^{-7}	1.05×10^{-8}	3.24×10^{-6}	7.19×10^{-4}
300	3.80×10^{-7}	1.28×10^{-10}	3.23×10^{-7}	5.45×10^{-9}	1.05×10^{-8}	1.15×10^{-9}	1.39×10^{-11}	3.51×10^{-7}	1.80×10^{-3}
400	1.69×10^{-8}	2.34×10^{-13}	2.13×10^{-8}	7.10×10^{-11}	9.43×10^{-11}	4.80×10^{-12}	1.44×10^{-14}	3.81×10^{-8}	4.04×10^{-3}
500	7.49×10^{-10}	2.12×10^{-15}	$1.4 \ 1 \times 10^{-9}$	9.25×10^{-13}	8.44×10^{-13}	2.02×10^{-14}	2.0×10^{-15}	4.13×10^{-9}	4.03×10^{-3}
Max Eigenvalue	0.9693	0.9395	0.9732	0.9575	.9539	.9467	.9255	-0.97802	-1.01978 (does not converge)

TABLE I (b) Results of applying Wynn's ε - Algorithm

Iterations	no acceleration	**** INVISCID WYNN'S ε-ALGORITHM *** Linear Boundary Conditions -- $\Delta t = 0.20$ 3-terms	5-terms	5-terms	9-terms	11-terms	(Inviscid) Largest Eigenvalues in abs.value	Viscous -- Re = 1000 Linear B.C. -- $\Delta t = 0.20$ 5-terms	3-terms
1	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	4.38×10^{-3}	$4{:}38 \times 10^{-3}$		4.38×10^{-3}	4.38×10^{-3}
200	8.56×10^{-6}	8.56×10^{-6}	8.56×10^{-6}	8.56×10^{-6}	8.56×10^{-6}	8.56×10^{-6}	0.9693	4.78×10^{-6}	4.78×10^{-6}
		acc. applied	acc applied	acc. applied	acc. applied	acc. applied	0.8393	acc. applied	acc. applied
		iter=200	iter=222	iter=200	iter=200	iter=252	0.8252	iter=200	iter=200
300	3.80×10^{-7}	2.78×10^{-11}	5.51×10^{-13}	1.99×10^{-12}	4.52×10^{-13}	9.49×10^{-14}	0.7276	5.14×10^{-12}	2.86×10^{-10}
		acc. applied	----------	acc. applied	acc. applied	----------	0.7276	acc. applied	acc. applied
310	2.79×10^{-7}	1.48×10^{-11}	4.02×10^{-13}	7.81×10^{-14}	machine zero	6.77×10^{-14}	0.6928	4.51×10^{-15}	3.03×10^{-14}
400	1.69×10^{-8}	machine zero	2.87×10^{-14}	machine zero	----------	machine zero	0.6928	machine zero	machine zero
							0.6205		
#iterations	800	340	355	310	301	299	0.6204	@ 302	@ 302
to reach 10^{-14}							0.5937		
$^*A_{200,235}$	0.9693	0.8386	#0.8915	0.7358	0.6911	##0.5873	0.5937	0.7118	0.7967
$^*A_{201,236}$	0.9693	0.8464	$A_{223,255}$ = 0.7813	0.7961	0.7835	$A_{254,285}$ = 0.8047	0.5365	0.7943	0.8403
							0.5365		

$^*A_{n,m} = \left| (\Delta Q_m) / (\Delta Q_n) \right|^{1/(m-n)}$

$^{\#}A_{222,255}$

$^{\#\#}A_{252,285}$

@ It took 900 iterations for the residual to drop to 10^{-14} for this case

Figure 1

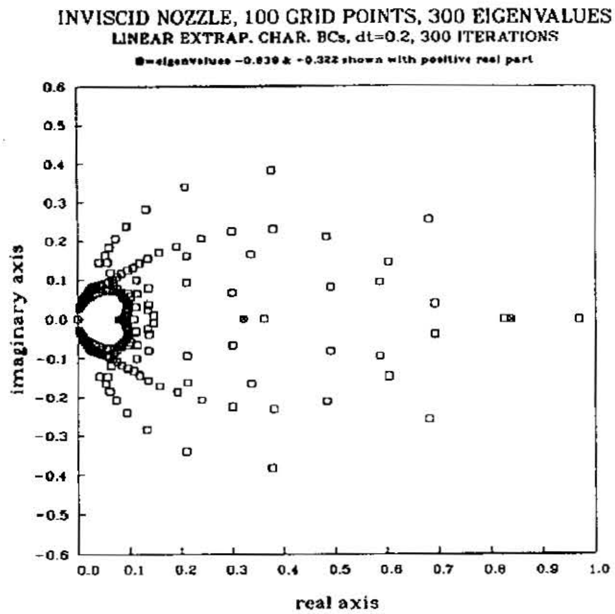

Figure 2

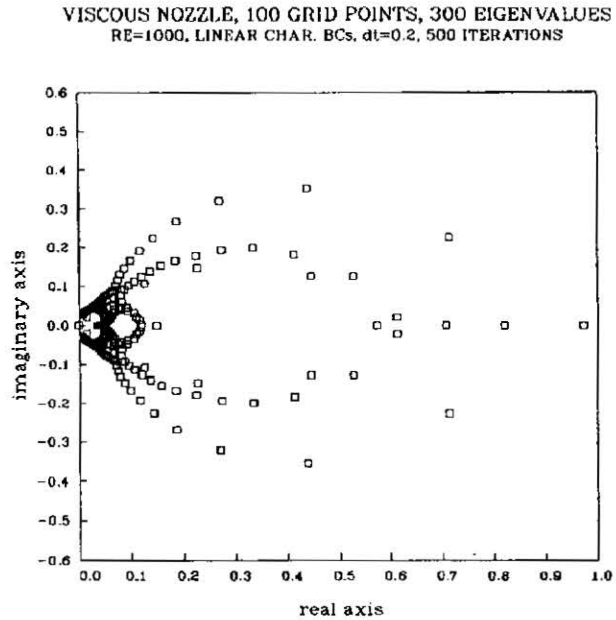

Figure 3

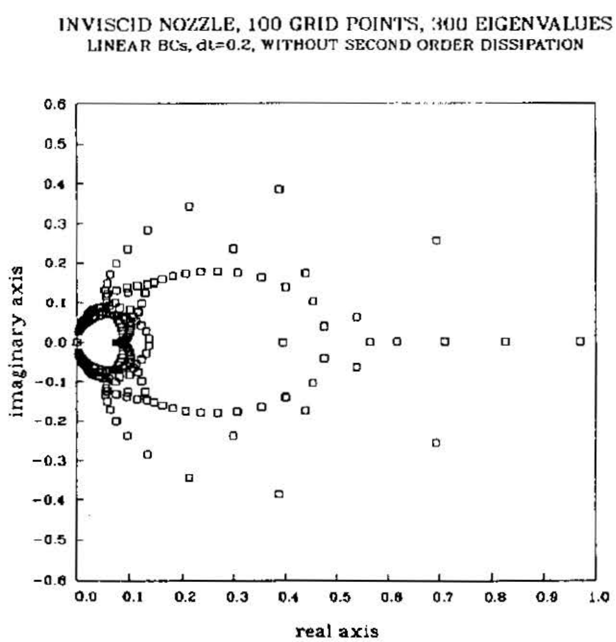

Figure 4

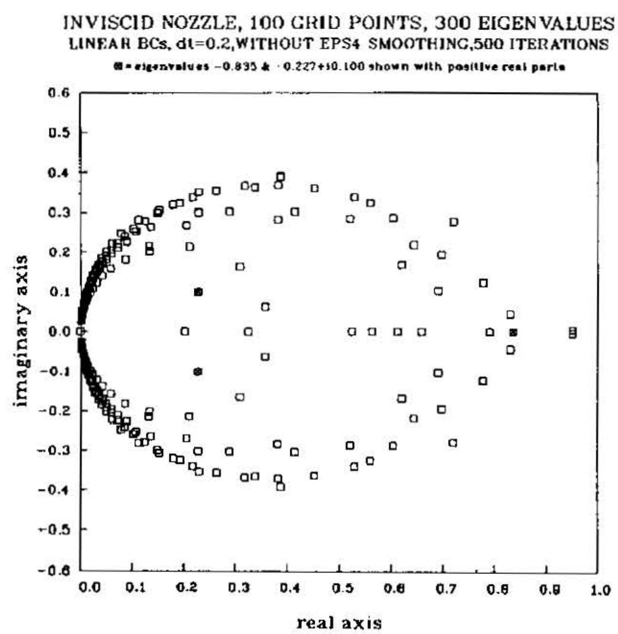

Figure 5

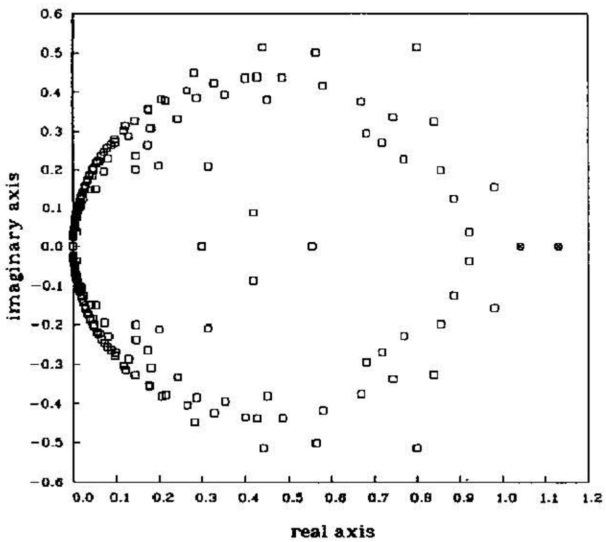

Figure 6

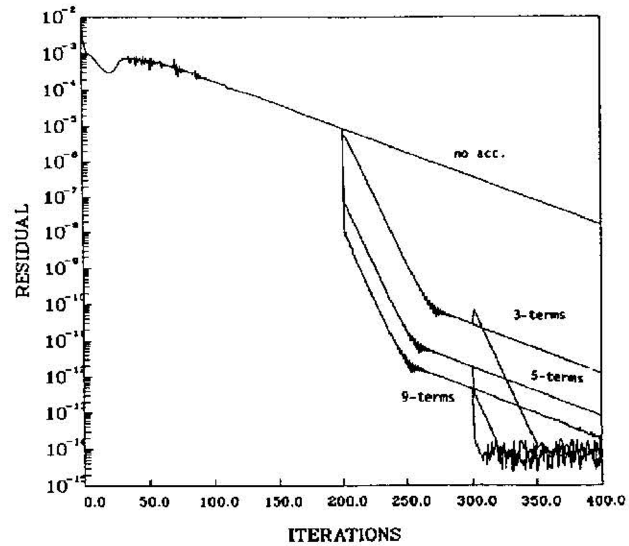

Figure 7