

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI[®]

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Eigensystem Analysis of a Numerical Method for Fluid Dynamics

by

Isabelle Hemmings

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto

© Copyright by Isabelle Hemmings 1998



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-40897-3

Canada

Abstract

Eigensystem Analysis of a Numerical Method for Fluid Dynamics

Isabelle Hemmings

Master of Applied Science

Graduate Department of Aerospace Science and Engineering

University of Toronto

1998

Arnoldi's method was implemented to approximate the semi- and fully-discrete eigenvalues of ARCID, a quasi-one-dimensional Euler solver. Three test cases were run to see the effects of different CFL numbers on the convergence rate and the approximations of the largest fully-discrete eigenvalue. Verification of the eigenvalue approximations was performed and an appropriate subspace size for Arnoldi's method was determined. A subspace size of about 40% of the original matrix size yields good results for the largest eigenvalues. The asymptotic convergence rate of the solver was found to agree closely with the largest eigenvalue of the fully-discrete operator matrix, confirming the linear behaviour of the operator. Similarly, a good correlation between the eigenvectors of the largest fully-discrete eigenvalues and the remaining error at a right-hand-side residual value of 10^{-10} was found. Overall, these results indicate that, while Arnoldi's method is a very useful tool for obtaining the maximum eigenvalues of large matrices, it can require large subspace sizes which could be prohibitive in two and three dimensions.

Acknowledgements

Over the course of this thesis, there are a number of people who encouraged me and supported me through the good and the bad. I would like to thank them here:

- David Zingg for his support and suggestions throughout
- My parents, Bob and Micheline, and my brothers, Rich and Rob, for listening to me and for being my cheerleaders
- My fiancé, Bailey, and his parents for their support and free food
- My friend Gail who worked out with me to relieve some stress
- All of my friends who cheered with me when it went well and listened to me when I was stuck

Thank you all. I could not have finished this without you guys.

Contents

List of Figures	v
1 Introduction	1
2 Background Theory	4
2.1 ARC1D	4
2.2 Eigenanalysis	8
2.2.1 Semi-Discrete Eigenvalues	8
2.2.2 Fully-Discrete Eigenvalues	8
2.2.3 Arnoldi's Method	10
2.2.4 Frechet Derivatives	11
2.2.5 Implementation	11
3 Results	13
3.1 Semi-Discrete Results	13
3.2 Convergence Rate	15
4 Conclusions	23
References	24

List of Figures

3.1	Exact λ (+) vs Arnoldi with 20 (x), 40 (o) and 60 (squares) Iterations . . .	14
3.2	Exact λ (+) vs Arnoldi with 144 Iterations (o)	14
3.3	Convergence History with CFL=5	15
3.4	Exact σ (+) vs Arnoldi with 20 (x), 40 (o) and 60 (squares) Iterations, CFL=5	16
3.5	Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=5	17
3.6	Real Part of First Eigenvector, CFL=5	17
3.7	Remaining Error, CFL=5	18
3.8	Real Parts of First (x) and Second (o) Eigenvectors, CFL=5	18
3.9	Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=2.5	19
3.10	Real Part of First Eigenvectors, CFL=2.5	19
3.11	Remaining Error, CFL=2.5	20
3.12	Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=10	21
3.13	Real Part of First Eigenvector, CFL=10	21
3.14	Remaining Error, CFL=10	22

Chapter 1

Introduction

In computational fluid dynamics, there is a search for faster, more efficient Navier-Stokes equation solvers. To this end, the study of stability and convergence rates of a solver and its time-marching and spatial discretization methods is necessary. One of the tools to study these methods is eigensystem analysis of the linearized semi- and fully-discrete operator matrices. Since the matrices involved are so large, approximations are used. In 1951, Arnoldi [2] created a method that was based on Lanczos's method of minimized iterations. This method approximates eigenvalues of large matrices and is guaranteed to finish in a finite number of iterations. Using a Krylov subspace, this method creates a Hessenberg matrix whose eigenvalues approximate those of the original matrix. Several researchers have implemented Arnoldi's method into their stability analysis and as a tool to study new techniques added to the solvers over the years.

Eriksson and Rizzi [5] use Arnoldi's method combined with Frechet derivatives to approximate the eigenvalues of their solution method. From the eigenvalues, they could determine a method for improving their convergence rates. They realized the benefits of eigensystem analysis, which include the following:

- the effects of the boundary conditions and artificial dissipation can be seen in detail by their eigensystem decomposition
- the eigensystems may be able to show the strengths and weaknesses of computational meshes
- the eigensystem analysis is general so it can analyze any method

Several other authors have used this same technique to analyze their problems and offer refinements of the solutions techniques. The techniques have included Wynn's ϵ -algorithm, eigenvalue annihilation and shifting operators in one and two dimensions [3], [4], [6], [13].

Allmaras uses Arnoldi's method in order to better analyze his multigrid developments [1]. He finds that the traditional Fourier analysis is insufficient for "real problems" and uses algebraic smoothing analysis for a more accurate picture of developments in the multigrid codes. Arnoldi's method is used to solve the eigenvalue problem generated in this method.

Cheer and Saleem [3] use Arnoldi's method in order to create a spectrum shift that allows for faster convergence of their solver. The shift depends on the size of the eigenvalue spectrum of the problem at hand. This is dealt with by using Arnoldi's method to approximate the spectrum. Since it captures the boundaries of the spectrum well, it can be used in the computation of the shifting operator without much error.

Cheer, et al. [4] use Arnoldi's method to approximate the convergence rate of the solver and to determine if acceleration techniques are required. Saleem, Pulliam and Cheer also examine the convergence rate and applications of convergence-acceleration techniques to ARC2D, a two-dimensional, approximately factored Navier-Stokes solver developed at NASA Ames Research Centre [13]. They also use Arnoldi's method to approximate the eigenvalues of the Jacobian matrix.

Mahajan, et al. [9] use Arnoldi's method to examine the effects of artificial dissipation on the stability of their time marching method. They conclude that it is a useful and efficient means of looking at stability without having to compute the very large Jacobian matrices involved.

While Arnoldi's method is used for analyzing algorithms, it has never been fully explored as a research tool. This thesis examines the effectiveness of the Arnoldi method in approximating the eigenvalues arising in a flow solver. Also, it will apply that method of approximation to study the stability and effects of various parameters on a quasi-one-dimensional Euler equation solver, ARC1D. This will provide a better understanding of the effectiveness and limitations of Arnoldi's method in analyzing algorithms for fluid flow.

The purpose of this thesis is to implement and evaluate the effectiveness of Arnoldi's method as a tool to analyze solution methods. The main components of the thesis are:

1. Create a program to run Arnoldi's method
2. Verify approximated eigenvalues against exact eigenvalues
3. Examine the semi-discrete and fully-discrete eigenvalues and their associated eigenvectors
4. Gain a better understanding of the convergence behaviour of ARC1D
5. Show agreement with the $\lambda - \sigma$ relationship of ARC1D

Chapter 2

Background Theory

To verify Arnoldi's method, the eigenvalues associated with ARC1D were found and analyzed. In order to verify the accuracy of the eigenvalues in Arnoldi's method, a linearized version of ARC1D was used. ARC1D was used as the testing ground for the Arnoldi process because it is feasible to calculate exact eigenvalues in one dimension. Calculating the approximate eigenvalues can be done in little time as well, making verification of the accuracy of Arnoldi's method practical. A Matlab routine was written to take the matrices from ARC1D and calculate the eigenvalues and compare them with the ones computed by the Arnoldi approximations.

2.1 ARC1D

ARC1D is a quasi-one-dimensional solver for the Euler equations based on ARC2D, a two-dimensional approximate factorization method developed at NASA Ames Research Centre. The information in this section is found in Pulliam's notes on ARC2D [10] but modified for one dimension. Adjustments for quasi-one dimensional flows are discussed later, but for simplicity, the one-dimensional Euler equations are used to explain the method.

The one-dimensional Euler equations have the form:

$$\frac{\partial Q}{\partial t} = -\frac{\partial E}{\partial x} \quad (2.1)$$

where

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

Applying the implicit Euler timestep, we get:

$$Q^{n+1} = Q^n + h \left(\frac{\partial Q}{\partial t} \right)^{n+1} \quad (2.2)$$

Using 2.1, we have:

$$Q^{n+1} = Q^n + h \left(-\frac{\partial E}{\partial x} \right)^{n+1} \quad (2.3)$$

Subtract Q^n from both sides:

$$Q^{n+1} - Q^n = -h \left(\frac{\partial E}{\partial x} \right)^{n+1} \quad (2.4)$$

Let $\Delta Q^n = Q^{n+1} - Q^n$. Linearize E with respect to Q to find an approximation for the flux Jacobian $\frac{\partial E}{\partial x}$:

$$E^{n+1} = E^n + \frac{\partial E^n}{\partial Q} \Delta Q^n + \dots \quad (2.5)$$

Let $A^n = \frac{\partial E^n}{\partial Q}$. Substitute this into 2.4 and combining like terms, the equations become:

$$\left(I + h \frac{\partial A^n}{\partial x} \right) \Delta Q^n = -h \frac{\partial E^n}{\partial x} \quad (2.6)$$

Apply the centred difference approximation to the first partial derivative with respect to x :

$$\left(I + \frac{h}{2\Delta x} (A_{j+1}^n - A_{j-1}^n) \right) \Delta Q_j^n = -\frac{h}{2\Delta x} (E_{j+1}^n - E_{j-1}^n) \quad (2.7)$$

This is only good for the interior nodes; the boundaries are dealt with using the Riemann invariant boundary conditions.

$$R_1 = \frac{p}{\rho^\gamma} \quad (2.8)$$

$$R_2 = u + \frac{2c}{\gamma - 1} \quad (2.9)$$

$$R_3 = u - \frac{2c}{\gamma - 1} \quad (2.10)$$

where c is the speed of sound. For the inflow boundary, R_1 and R_2 are determined from the initial conditions and R_3 is extrapolated from the interior nodes. For the outflow boundary, R_3 are determined from the initial conditions and R_1 and R_2 are extrapolated from the interior nodes. From these invariants, the solution variables can be found.

The interior scheme needs artificial dissipation to allow for better convergence. Also a switch is needed to allow for good shock capturing. This switch combines fourth and second difference artificial dissipation. The pressure switch is:

$$\Upsilon_j = \frac{|p_{j+1} - 2p_j + p_{j-1}|}{|p_{j+1} + 2p_j + p_{j-1}|} \quad (2.11)$$

This pressure switch leads to the determination of the two artificial dissipation variables $\epsilon^{(2)}$ and $\epsilon^{(4)}$:

$$\epsilon^{(2)} = \kappa_2 \Delta t \max(\Upsilon_{j+1}, \Upsilon_j, \Upsilon_{j-1}) \quad (2.12)$$

$$\epsilon^{(4)} = \max(0, \kappa_4 \Delta t - \epsilon_j^{(2)}) \quad (2.13)$$

where κ_2 and κ_4 are user defined parameters. The fourth order dissipation is turned off near the shock regions by the pressure switch. The fourth difference artificial dissipation stencil then becomes:

$$\nabla_x \left(\sigma_{j+1} \epsilon_{j+1}^{(4)} + \sigma_j \epsilon_j^{(4)} \right) \Delta_x \nabla_x \Delta_x Q_j \quad (2.14)$$

where σ is spectral radius of the flux Jacobian, and ∇_x indicates a backward difference while Δ_x indicates a forward difference. The second difference artificial dissipation stencil then becomes:

$$\nabla_x \left(\sigma_{j+1} \epsilon_{j+1}^{(2)} + \sigma_j \epsilon_j^{(2)} \right) \Delta_x Q_j \quad (2.15)$$

These stencils are written for the right-hand side. The left-hand side's matrix is filled in a similar manner. Each entry in the $[1, -4, 6, -4, 1]$ stencil is added to the correct block of the matrix. The first and last interior nodes are modified with an upwind and downwind biased scheme, respectively. This makes the matrix a block pentadiagonal one, which is easy to invert.

The quasi one-dimensional nozzle case only needs minor modifications. Since the area varies through the nozzle, the solution and flux vectors now include the area in their variables and a source term is added to the momentum equation:

$$\frac{\partial(\rho u S)}{\partial t} + \frac{\partial(\rho u^2 + p)S}{\partial x} = p \frac{dS}{dx} \quad (2.16)$$

Applying the implicit Euler scheme and the linearization process as before, the modified equations become:

$$\left[I + \Delta t \frac{\partial A^n}{\partial x} - \Delta t B^n \right] \Delta Q^n = -\Delta t \frac{\partial E^n}{\partial x} + \Delta t P^n \quad (2.17)$$

where

$$P^n = \begin{bmatrix} 0 \\ p \frac{dS}{dx} \\ 0 \end{bmatrix} \quad (2.18)$$

$$B^n = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{S} \frac{dS}{dx} \left(\frac{\gamma-1}{2} u^2 \right) & \frac{1}{S} \frac{dS}{dx} (-(\gamma-1)u) & \frac{1}{S} \frac{dS}{dx} (\gamma-1) \\ 0 & 0 & 0 \end{bmatrix} \quad (2.19)$$

and the other variables have the same meaning as before.

The boundary conditions described above are explicit boundary conditions. This changes the first and last nodes of the matrix that is inverted in each timestep. Since the boundaries are explicit, there is no need to fill the matrix entries that are associated with them. In order to have a matrix that is consistent throughout, we changed the boundary conditions to fixed boundaries. This removes the influence of the boundaries from any other node in the stencils, thus we can observe the behaviour of the interior scheme first.

Artificial dissipation in ARC1D is a mixed non-linear dissipation with a pressure switch to improve shock capturing. This is modified to linearize the system. A fourth difference, constant coefficient dissipation is used throughout with no pressure switch in both the right- and left-hand side calculations. On the right hand side, this dissipation has the form:

$$-\epsilon_e \Delta t [\nabla_x \Delta_x]^2 Q^n \quad (2.20)$$

where σ is spectral radius of the flux Jacobian, ϵ_e is a user defined coefficient [10], δt is the time step and ∇_x and Δ_x are the forward and backward difference, respectively. On both sides of the equations, the first and last interior nodes are modified by using an upwind or downwind biased scheme for the dissipation, respectively.

2.2 Eigenanalysis

Two sets of eigenvalues may be examined: the semi-discrete or λ eigenvalues and the fully discrete or σ eigenvalues.

2.2.1 Semi-Discrete Eigenvalues

The semi-discrete formulation of the original PDE is devised by discretizing the spatial derivatives. This has the following form:

$$\frac{dQ}{dt} = R(Q) \quad (2.21)$$

where Q and R are vectors of length $3N$, where N is the number of nodes. Linearizing the right hand side, we get:

$$R(Q) = R(Q^*) + \frac{\partial R(Q^*)}{\partial Q} \Delta Q \quad (2.22)$$

where Q^* is the steady state solution, and $\frac{\partial R}{\partial Q}$ is the Jacobian matrix of dimension $3N \times 3N$. Therefore, $R(Q^*) = 0$, and we only need be concerned with the second term on the right hand side. The eigenvalues of the Jacobian matrix $\frac{\partial R}{\partial Q}$, defined as λ , are of interest since they describe how quickly the transient will die down and the solution will become the steady state solution. For the transient solution to tend to zero, the λ eigenvalues are required to be in the left half of the complex plane.

The matrix associated with these eigenvalues is not immediately available from ARC1D. In order to remove this, you have to take the left hand side:

$$I - hA \quad (2.23)$$

and be left with A only.

2.2.2 Fully-Discrete Eigenvalues

The fully-discrete form of the original PDE is created when the time marching method and the spatial discretization have both been implemented. This has the following form:

$$Q^{n+1} = f(Q^n) \quad (2.24)$$

Linearizing f , we get:

$$f(Q) = f(Q^*) + \frac{\partial f(Q^*)}{\partial Q} (Q^* - Q) \quad (2.25)$$

As in the semi-discrete form, Q^* is the steady state solution; therefore $f(Q^*) = Q^*$. $\frac{\partial f}{\partial Q}$ is once again a Jacobian matrix and its eigenvalues σ are of interest. Stability of the time marching steps combined with spatial discretizations can be evaluated by examining the σ eigenvalues. If they all lie within the unit circle on the imaginary plane, then the solution method is stable. Also, the asymptotic convergence rate of the solver can be estimated to be the magnitude of the largest σ . Therefore, the σ calculations could be used to determine convergence rates without having to record a convergence history.

We have to remember that the σ are really the eigenvalues of the matrix associated with the fully discretized form. In order to get the correct eigenvalue problem for an implicit method as used in ARC1D, we need to consider the following equation:

$$M(Q^n)\Delta Q^n = P(Q^n) \quad (2.26)$$

where $P(Q^n) = hR(Q^n)$. Let $Q^{n+1} = f(Q^n)$. Substituting into the above, we get:

$$f(Q^n) = Q^n + M^{-1}(Q^n)P(Q^n) \quad (2.27)$$

We'd like to linearize f at this point. This means we must expand using the Taylor theorem about Q^* . In particular, we need to find $\frac{\partial f}{\partial Q}(Q^*)$. Differentiating 2.27 with respect to Q yields:

$$\frac{\partial f}{\partial Q}(Q^*) = I + M^{-1}(Q^*)\frac{\partial P(Q^*)}{\partial Q} + \frac{\partial M^{-1}(Q^*)}{\partial Q}P(Q^*) \quad (2.28)$$

Of course, at steady state, $P(Q^*)$ is zero and hence the last term of the equation is zero.

Using Taylor's Theorem to linearize 2.27 about Q^* with the Jacobian defined by 2.28 and making the appropriate substitution, we get:

$$f(Q^n) = f(Q^*) + \left[I + M^{-1}(Q^*)\frac{\partial P(Q^*)}{\partial Q} \right] (Q^* - Q^n) \quad (2.29)$$

This is in the form $u_{n+1} = Cu_n + b$, where $C = I + M^{-1}\frac{\partial P}{\partial Q}(Q^*)$, so we have the eigenvalue problem:

$$\left(I + M^{-1}(Q^*)\frac{\partial P(Q^*)}{\partial Q} \right) x = \sigma x \quad (2.30)$$

Multiplying on the left by $M(Q^*)$, we get the following generalized eigenvalue problem:

$$\left(M(Q^*) + \frac{\partial P}{\partial Q}(Q^*) \right) x = \sigma M(Q^*) x \quad (2.31)$$

In ARCID, $M(Q^*)$ is the left hand side matrix at the converged solution and the Jacobian of P is hA . If we use the interior scheme only and ignore the boundaries for the moment. the result is a generalized eigenvalue problem whose eigenvalues are σ . In the linearized form of ARCID, $M = I - hA$, so the generalized eigenvalue problem becomes a regular eigenvalue problem of $(I - hA)^{-1}$.

2.2.3 Arnoldi's Method

Finding the eigenvalues of the full matrices in one dimension is feasible on a computer, but moving to two dimensions causes some problems with memory. Arnoldi's method [2] helps with this. It approximates the eigenvalues of the original matrix with those of a smaller Hessenberg matrix. A considerably smaller subspace is often sufficient for very accurate answers, at least for the largest eigenvalues [12].

The algorithm is as follows:

- 1) select a vector v_1 , which is normalized.
- 2) for $k=1..n$

$$h_{i,k} = \langle v_i, Av_k \rangle, i = 1..k$$

$$w_k = Av_k - \left\{ \sum_{i=1}^k h_{i,k} v_i \right\}$$

$$h_{k+1,k} = \|w_k\|_2, \text{ if } h_{k+1,k} = 0 \text{ then stop}$$

$$v_{k+1} = \frac{w_k}{h_{k+1,k}}$$

- 3) calculate the eigenvalues of $H = [h_{i,j}]$.

In other words, a new vector w_k is calculated as a new basis vector for the subspace at each iteration. If $h_{k+1,k} = 0$ for any iteration, that means that there are no more

orthogonal vectors to be chosen in the subspace and the procedure must stop, as it has created the largest basis possible. If $h_{k+1,k} \neq 0$ before the n^{th} step, then the first n vectors form an orthonormal basis for the subspace $\mathcal{K}_n = \text{span}\{v_1, Av_1, \dots, A^{n-1}v_1\}$ [11].

A question arises as to the choice of the first vector v_1 . Saleem proves that the best choice for the starting vector in Arnoldi's process for the Euler equations and the ARC1D solution technique is the solution vector Q^* [14]. His results have been accurate to five significant figures [14].

2.2.4 Frechet Derivatives

An obvious problem arises in Arnoldi's method – how to save memory in computing Av_i , where A is a Jacobian of either f or R . These are large matrices in one dimension (e.g., 300x300 on a grid with 100 nodes) and enormous matrices in two dimensions (e.g., 40,000x40,000 on a grid with 10,000 nodes). In order to save memory, Frechet derivatives are used to calculate the necessary matrix-vector multiplication. The idea behind this derivative is very similar to a finite difference method of a simple derivative. The second-order version of the Frechet derivative [15] is:

$$\frac{\partial f(Q)}{\partial Q} v = \frac{f(Q + \epsilon v) - f(Q - \epsilon v)}{2\epsilon} \quad (2.32)$$

This is very similar to the definition of a derivative. ϵ is an arbitrarily small number. In this thesis, ϵ is chosen to be $0.001 * \frac{Q^*}{v_i}$ as suggested by Eriksson and Rizzi, as well as Saleem [5], [14].

These derivatives are used to approximate the Av multiplication in Arnoldi's method. Instead of using the Jacobian matrix, only the solution Q^* is saved and used in the Arnoldi subroutine, which saves memory.

2.2.5 Implementation

Arnoldi's method was used to calculate both λ and σ eigenvalues. Therefore, the Frechet derivative subroutine must be able to approximate the semi- and fully-discrete matrices. In order to accomplish this, the Frechet subroutine calls the ARC1D subroutines to either perform one time step or to do one right hand side calculation for each perturbation of the vector. The non-dimensionalized results from ARC1D are used as input for the derivatives.

A Matlab routine was written in order to calculate the exact eigenvalues of the matrices created in ARC1D. These results were then compared to the Arnoldi's method results. The λ eigenvalues were assessed by taking the A matrix from ARC1D, while the σ eigenvalues were computed from $(I - hA)^{-1}$.

Chapter 3

Results

Three cases were chosen to study. All were uniform flow with a Mach number of 0.189 with 48 interior nodes, a constant dissipation coefficient of value $\epsilon_e = 1.25$ and Dirichlet boundary conditions. For the initial condition, the solution at the middle node was set to $0.1Q_{steady}$, where Q_{steady} is the steady state solution. Three different CFL numbers are chosen to see the effect on the eigenvalues and the eigenvectors. Case 1 is run with a CFL number of 5. Case 2 is run with a CFL of 2.5. Case 3 is the higher CFL number run with a CFL of 10. All cases are run

3.1 Semi-Discrete Results

The λ eigenvalues were examined for only one CFL number, as the change in CFL number will not affect the results of the semi-discrete form.

In terms of capturing the eigenspectrum, good results were obtained from Arnoldi's method using 40 iterations and even better results were obtained using 60 iterations. The largest exact eigenvalue's magnitude is 19.937. With 40 iterations, Arnoldi's method gives a largest magnitude of 19.723, and after 60 iterations, Arnoldi's method gives 19.935. Figure 3.1 shows the effects of using 20, 40 and 60 iterations in Arnoldi's method. When using the maximum subspace size, Arnoldi's method captures the largest magnitude eigenvalues quite well (see Figure 3.2), but does not capture the eigenspectrum exactly, which is because of round-off error. Because of this, subspaces larger than 60 would be inefficient to use.

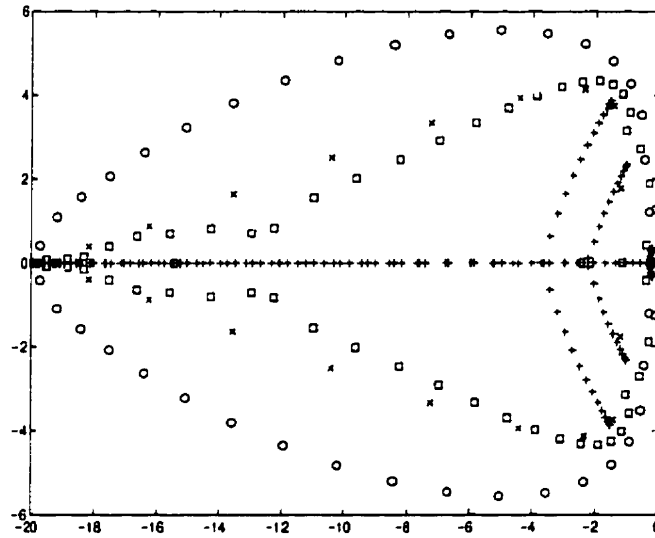


Figure 3.1: Exact λ (+) vs Arnoldi with 20 (x), 40 (o) and 60 (squares) Iterations

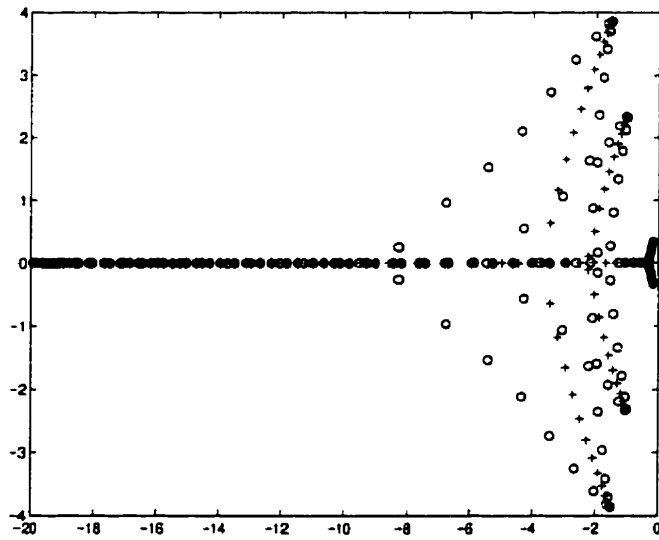


Figure 3.2: Exact λ (+) vs Arnoldi with 144 Iterations (o)

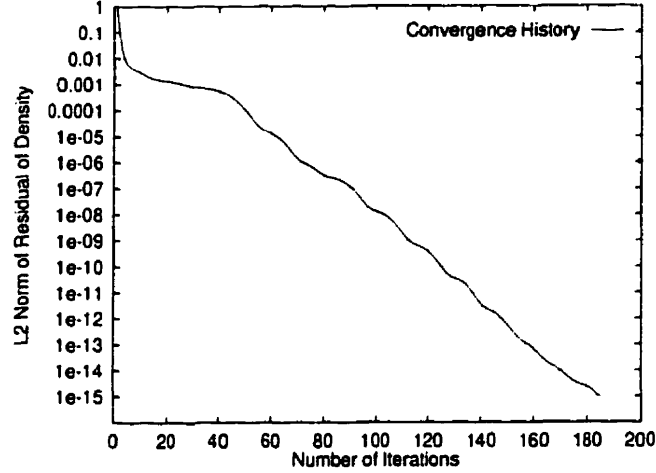


Figure 3.3: Convergence History with CFL=5

3.2 Convergence Rate

The asymptotic convergence rate can be approximated by the largest magnitude of σ . To demonstrate this, the convergence rate of ARC1D was examined by three methods: direct calculation based on the residual, the largest magnitude of the exact eigenvalues and the largest magnitude of the approximate eigenvalues.

Figure 3.3 shows a linear convergence approaching the end of the run. In this area, the convergence rate may be determined by:

$$\log |\sigma_{\max}| = \frac{\log(Res_{n+p}) - \log(Res_n)}{p} \quad (3.1)$$

$$|\sigma_{\max}| = \left(\frac{Res_{n+p}}{Res_n} \right)^{1/p} \quad (3.2)$$

where Res_n is the norm of the residual of density at step n . In Case 1, we estimate the convergence rate to be 0.838. Using Matlab, the largest exact eigenvalue for the generalized eigenvalue problem is found to be $|\sigma_{\max}| = 0.839$. This difference is 0.217%, which is very good agreement.

To verify that Arnoldi's method works, runs were made with 20, 40 and 60 iterations of Arnoldi's method. The results of the eigenvalue approximations are verified versus the exact eigenvalues in Matlab. Figure 3.4 shows that as the number of iterations increase, the eigenvalue spectrum is better captured, although when using more than 60 iterations, the method failed to capture the eigenspectrum as well as it did with 60 iterations. 60

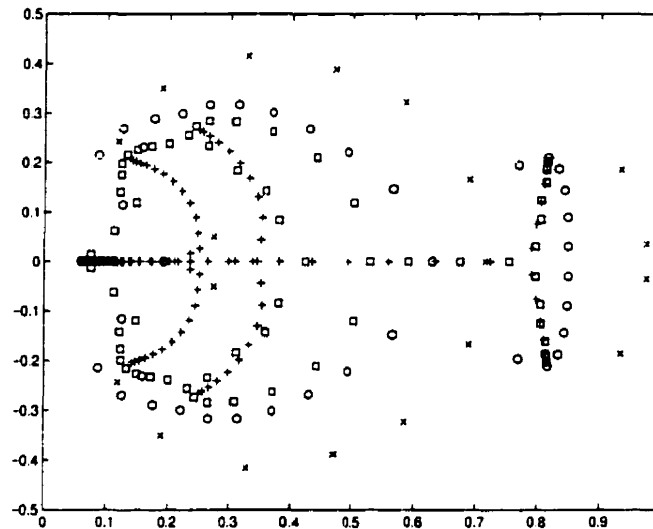


Figure 3.4: Exact σ (+) vs Arnoldi with 20 (x), 40 (o) and 60 (squares) Iterations. CFL=5

iterations were used in the remaining calculations when using Arnoldi's method as seen in Figure 3.5. Compared to other literature, this seems to be a very large number of iterations to use. While most authors choose 5 to 10 iterations of Arnoldi's method to approximate the largest σ eigenvalue, with 5 iterations, the calculations obtained were $|\sigma_{\max}| = 0.8676$ compared to the exact value of 0.8394.

Case 1 results for eigenvalues are discussed in the section above. Good agreement with the convergence rate was found for both the exact and the approximate eigenvalues. The eigenvectors of the exact matrix were compared to the difference between the solutions when the residual was at 10^{-10} and 10^{-14} , which gives the remaining error in the solution when the residual is 10^{-10} . The shape of the eigenvector and the remaining error is similar. This indicates that the remaining error is at the outflow boundary, as seen in Figures 3.6 and 3.7. Also, the two eigenvectors associated with the two largest eigenvalues are similar indicating that there could be multiple low frequency modes still present. The magnitude of the two largest eigenvalues in this case are 0.8394 and 0.8346. Their eigenvectors are shown in Figure 3.8.

Case 2 shows a slower convergence rate of 0.9159 in rough calculations. This compares well to the exact matrix's largest eigenvalue of 0.9195 and Arnoldi's method approximation of 0.9203 as seen in Figure 3.9. The remaining errors and the largest eigenvector still compare favourably as in case 1 (Figures 3.10 and 3.11).

Case 3 shows a much faster convergence rate of 0.6974. The exact and approximate

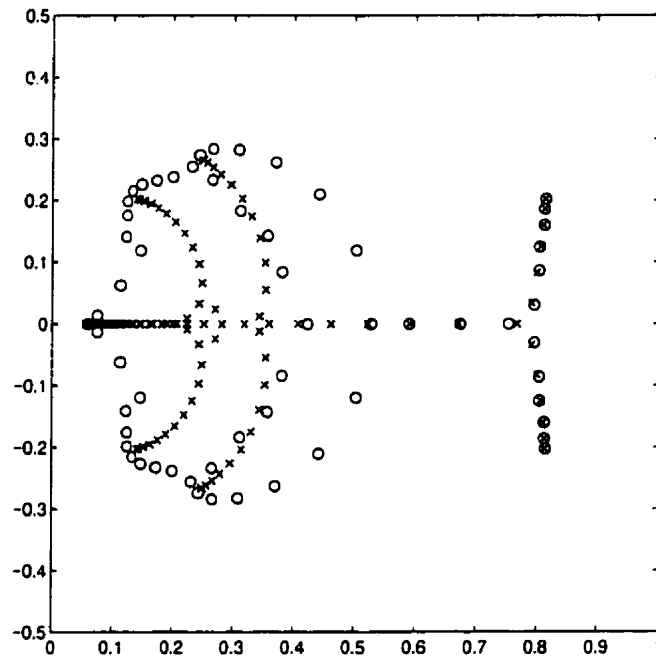


Figure 3.5: Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=5

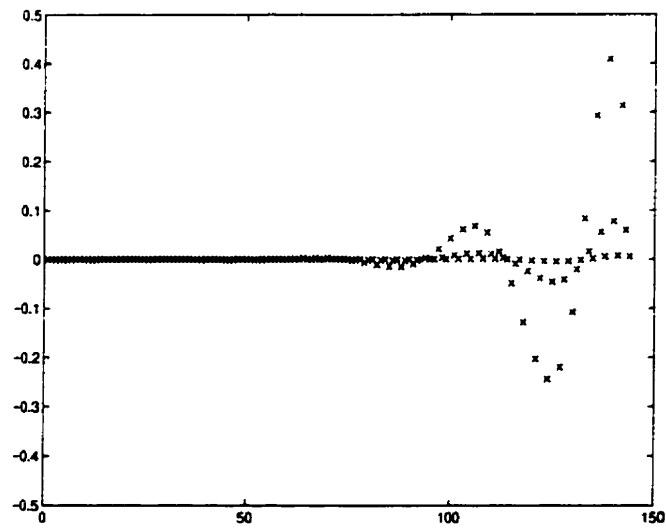


Figure 3.6: Real Part of First Eigenvector, CFL=5

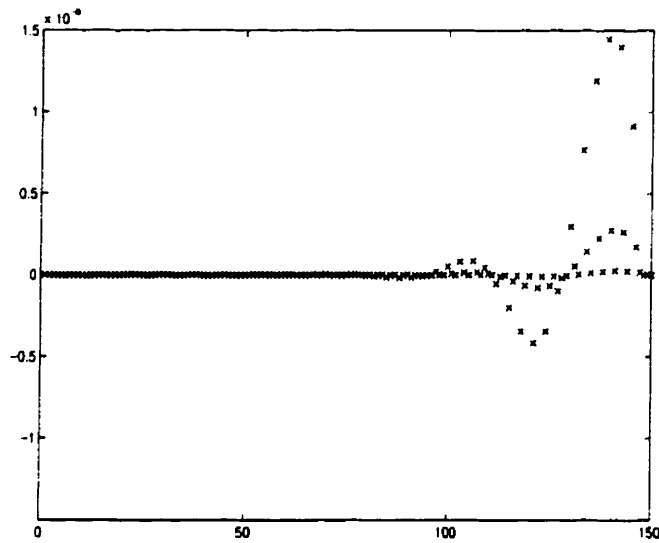


Figure 3.7: Remaining Error, CFL=5

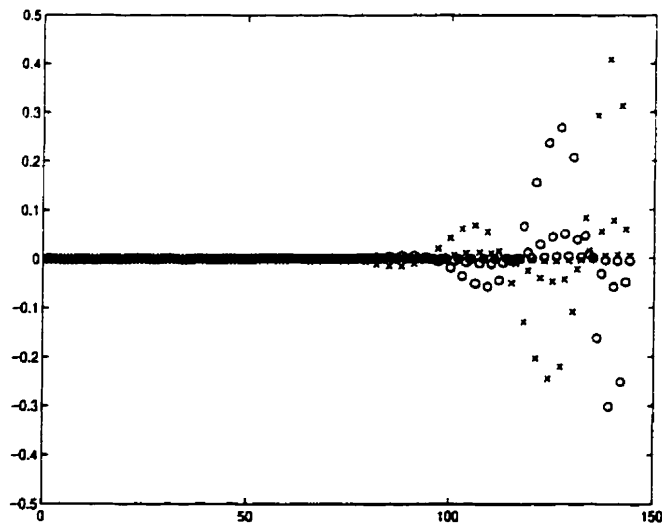


Figure 3.8: Real Parts of First (x) and Second (o) Eigenvectors, CFL=5

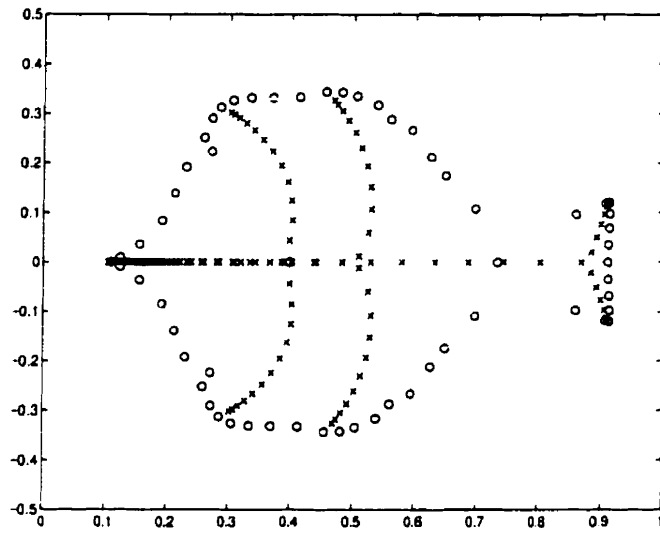


Figure 3.9: Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=2.5

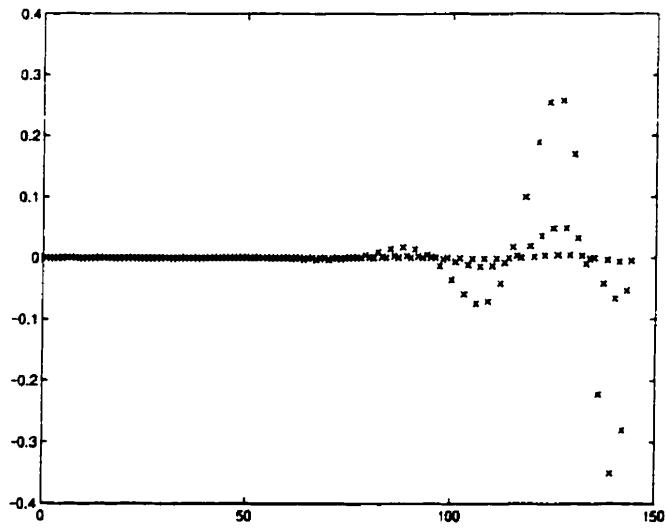


Figure 3.10: Real Part of First Eigenvectors, CFL=2.5

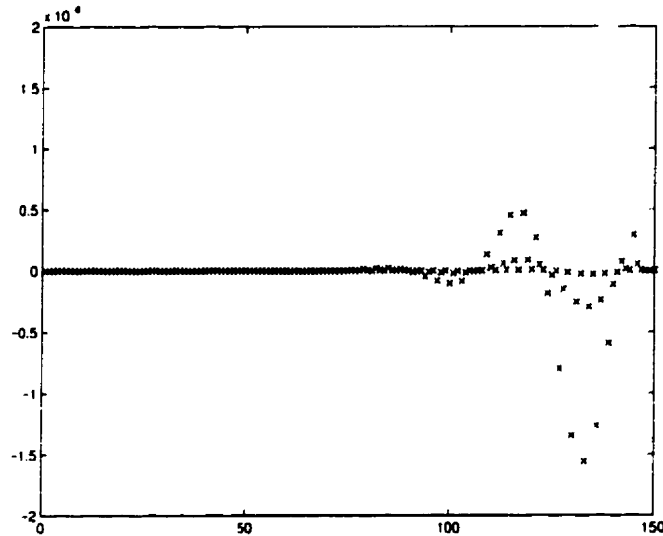


Figure 3.11: Remaining Error, CFL=2.5

largest eigenvalues are 0.6981 and 0.6985, respectively. These are good approximations for the convergence rate (see Figure 3.12). The eigenvector and the remaining error are still in good agreement as well, indicating that the outflow boundary is still the problem area (see Figures 3.13 and 3.14).

Finally, the $\lambda - \sigma$ relationship can be verified using the approximations from Arnoldi's method. For ARC1D, the relationship is:

$$\sigma = \frac{1}{1 - h\lambda} \quad (3.3)$$

For Case 1, the largest approximate σ is $0.8146 + 0.2025i$. The λ associated with that is $-0.1819 + 0.3349i$, which is a very close agreement with the corresponding λ found from the exact calculations, $0.1819 + 0.3348i$. Although this is a good correlation, it is not a feasible means of obtaining the σ_{\max} values because of problems with capturing the small magnitude λ values with Arnoldi's method. Note that the maximum σ in the other cases can be obtained from this λ simply by changing h appropriately.

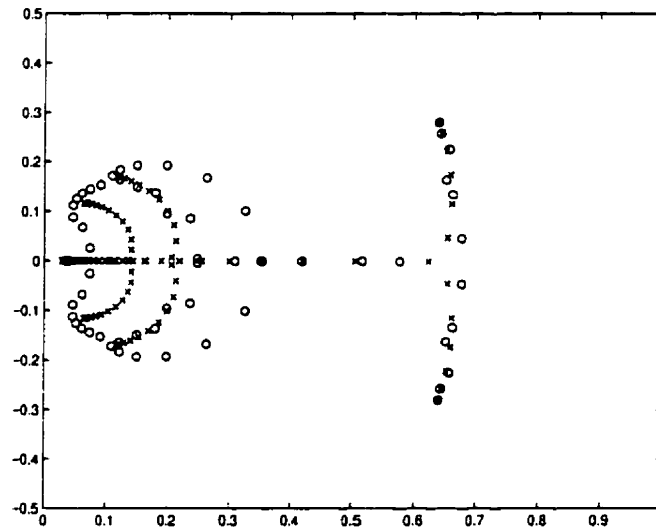


Figure 3.12: Exact (x) vs Arnoldi (o) σ Eigenvalues, CFL=10

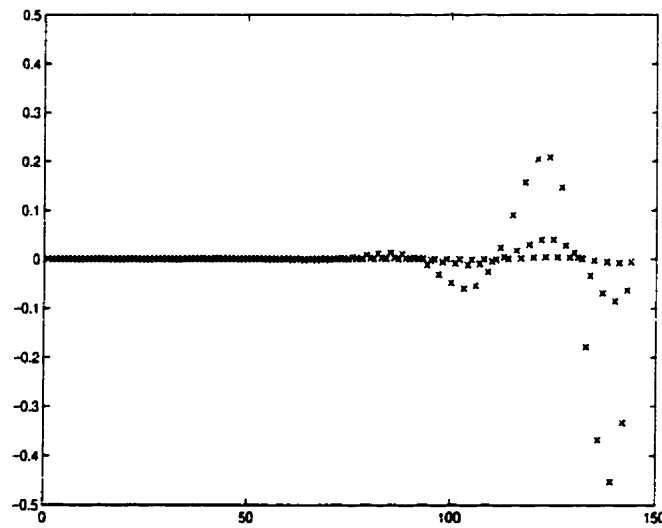


Figure 3.13: Real Part of First Eigenvector, CFL=10

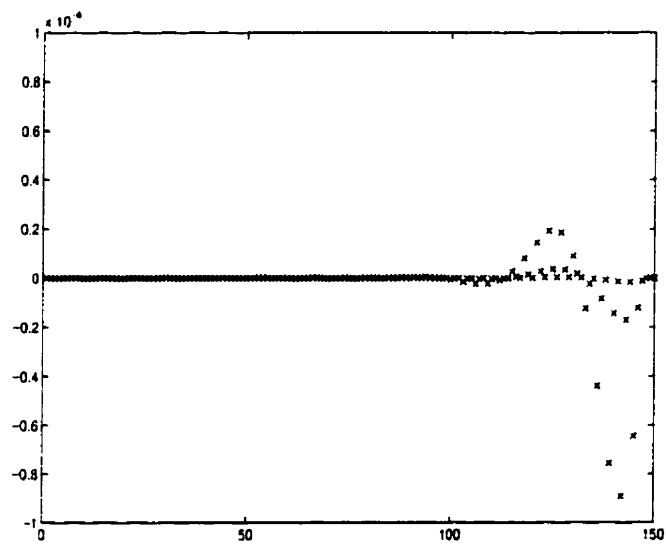


Figure 3.14: Remaining Error, CFL=10

Chapter 4

Conclusions

A program was written to use Arnoldi's method to approximate the λ and σ eigenvalues of ARC1D. Runs were made to verify that these approximations are accurate and to determine appropriate subspace sizes for Arnoldi's method. Three test cases were run to see the effects of changes in CFL number on the convergence rate and the approximations of the largest σ eigenvalue. A number of runs with Arnoldi's method were done to find a good approximation of the eigenspectrum.

The following specific conclusions can be drawn from the results:

- The maximum λ and σ eigenvalues are approximated very well using Arnoldi's method with 60 iterations. Given that the full matrices are 144×144 , this is a surprisingly large requirement.
- The σ eigenvalues correspond well with the convergence rate of the three test cases.
- The eigenvectors of the largest few eigenvalues correspond well to the remaining error.

Overall, these results indicate that, while Arnoldi's method is a very useful tool for obtaining the maximum eigenvalues of large matrices, it can require large subspace sizes which could be prohibitive in two and three dimensions.

References

- [1] Allmaras, S.R., 'Algebraic Smoothing Analysis of Multigrid Methods for the 2-D Compressible Navier-Stokes Equations', AIAA-97-1954, 1997.
- [2] Arnoldi, W.E., 'The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem', *Quarterly of Applied Mathematics*, **9**, 17-29 (1951).
- [3] Cheer, A., and M. Saleem, 'Acceleration of Convergence by Shifting the Spectrum of Implicit Finite Difference Operators Associated with the Equations of Gas Dynamics', *International Journal for Numerical Methods in Fluids*, **12**,443-462 (1991).
- [4] Cheer, A.Y., M. Saleem, T.H. Pulliam and M. Hafez, 'Analysis of the Convergence History of Flow Through Nozzles with Shocks', *AIAA-SIAM First National Fluid Dynamics Congress*, AIAA-88-3795-CP, Cincinnati OH, 1988.
- [5] Eriksson, L. E., and A. Rizzi, 'Computer-Aided Analysis of the Convergence to Steady State of Discrete Approximations to the Euler Equations', *Journal of Computational Physics*, **57**, 90-128 (1985).
- [6] Jespersen, D. C., and P. G. Buning, 'Accelerating an Iterative Process by Explicit Annihilation', *SIAM Journal of Sci. Stat. Comput.*, Vol. 6, No. 3, 639-651, 1985.
- [7] Jespersen, D. C., and T. H. Pulliam, 'Flux Vector Splitting and Approximate Newton Methods', AIAA-83-1899, 1983.
- [8] Lomax, Harvard, T.H. Pulliam, and David Zingg, *Fundamentals of Computational Fluid Dynamics*, <http://goldfinger.utias.utoronto.ca/dwz/>.
- [9] Mahajan, A. J., E. H. Dowell, and D. B. Bliss, 'On the Role of Artificial Viscosity in Navier-Stokes Solvers', AIAA-89-1947-CP, 1989.

- [10] Pulliam, T., 'Efficient Solution Methods for the Navier-Stokes Equations', Lecture Notes for the Von Karman Institute for Fluid Dynamics Lecture Series, 1986.
- [11] Saad, Y.. *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.
- [12] Saad, Y., 'Variations on Arnoldi's Method for Computing Eigenlements of Large Unsymmetric Matrices', *Linear Algebra and Its Applications*, **34**, 269-295 (1980).
- [13] Saleem, M., T.H. Pulliam, and A.Y. Cheer, 'Acceleration of Convergence and Spectrum Transformation of Implicit Finite Difference Operators Associated with Navier-Stokes Equations', *Journal of Computational Physics*, **104**, 1-13 (1993).
- [14] Saleem, M., *Spectrum Analysis and Convergence Acceleration Techniques Applied to Implicit Finite Difference Approximations for the Euler and Navier Stokes Equations*, Ph.D. Thesis, University of California Davis, 1988.
- [15] Sneddon, I.N., *Encyclopaedic Dictionary of Mathematics for Engineers and Applied Scientists*, Pergamon Press, Toronto, 1976.