4

A Generalized Unstructured Finite Volume Solver with Application to Turbomachinery

by

David Car

B.S. The Ohio State University (1993)

SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master of Science
at the
Massachusetts Institute of Technology

February 1995

Signature of Author _____
Department of Aeronautics and Astronautics
January 23, 1995

Certified by _____
Associate Professor Jaime Peraire
Thesis Supervisor, Department of Aeronautics and Astronautics

Accepted by _____
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

A Generalized Unstructured Finite Volume Solver with Application to Turbomachinery

by
David Car

Submitted to the Department of Aeronautics and Astronautics
on January 23, 1995 in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

A numerical algorithm is developed for the solution of two dimensional, compressible Euler equations for applications in unsteady turbomachinery flows on unstructured meshes. Two techniques for generating the mesh are chosen in this work: Delaunay and Steiner. The capability of controlling the resolution and quality is addressed and an example shown for a compressor blade with the inclusion of periodic boundaries.

A finite volume approach is used for the solution of the Euler equations. The technique is based on a linear approximation over the entire domain and the use of generalized control volumes for the evaluation of the fluxes. These control volumes are arbitrarily constructed about a node in the mesh and are contained within the patch of elements connected to that node. The numerical algorithm can be written in terms of edges, giving it a finite difference character. It is shown that the approximation used to spatially integrate the time derivative term in the Euler equations is crucial to the numerical scheme's dispersion characteristics. Numerical dissipation is added to stabilize the scheme and adequately capture shocks. The dissipation is based on Jameson's Symmetric Limited Positive scheme (SLIP) which uses a flux limiting procedure. Methods of reducing the amount of dissipation in smoothly varying areas of the flow are addressed and leads to a class of numerical schemes known as Essentially Local Extremum Diminishing (ELED). Finally, it is shown that the choice of control volumes can add a diffusion-like term into the numerical algorithm which must be taken into account in the application of the numerical dissipation.

Results are computed for a shock tube and a convected entropy perturbation to validate and analyze the numerical algorithm. Finally, the numerical algorithm is used to simulate a transonic axial compressor at two operating speeds, 67 and 85 percent of the design speed with a uniform inlet flow. After an unchoked converged solution is obtained for the 67 percent design speed simulation, a periodic axial total pressure distortion is introduced at the inlet boundary to reproduce a forced response test on the compressor.

Thesis Supervisor: Jaime Peraire
Associate Professor of Aeronautics and Astronautics

Acknowledgements

I would like to thank Jaime (Jaume) Peraire for all his help in writing this thesis, his friendship and his support.

I have to thank my wife, Kym, whose love and support truly makes life a whole lot easier. Her editing and selfless support in the creation of this work does not go unnoticed. With her, life just makes a little more sense.....I love you more than words.....

I'd like to thank my parents for shoving me out the door when I was 17. It was the best thing they could have done for me. Also, their unending support is what made my undergraduate days possible.

I'd like to thank Rick Foutter for his friendship and great sense of humor. Remember Rick, "They just don't get it, do they?" Always question scientists, that's what they're there for.

I'd like to thank God for all the blessings He has bestowed upon me and for His son Jesus Christ. Maybe if I stop to listen once in awhile, things might just work out a lot easier.

Contents

List of Figures

Chapter 1
Introduction

The objective of this work is to develop a numerical algorithm for the solution of the two dimensional, compressible Euler equations for applications in unsteady turbomachinery flows. The importance of this work is the ability to adequately predict flow phenomena within a turbomachine. These predictions can be used to enhance experimental data in regions where data is not available or is inadequate. This can give a better understanding of the observed flow phenomena and help guide further experimental and design research.

The numerical simulations of these flows have primarily used structured grids with either finite volume or finite difference approximations to the governing flow equations. Over the last decade, unstructured flow solvers have gained strong popularity in computational fluid dynamics. A key factor is the arbitrariness of the mesh so that complex domains can be meshed easily. The use of unstructured meshes also gives the added ability of adapting the mesh to flow features for increased resolution. Shock capturing methods for unstructured flow solvers have also advanced recently [11, 12]. For these reasons, an unstructured approach was taken in this work.

The first topic addressed is mesh generation. There are many techniques available [2]. Two techniques have been chosen in this work: Delaunay and Steiner. The capability of controlling the mesh resolution and quality is addressed and examples are shown, including periodic boundaries, to demonstrate the capability developed.

A novel finite volume approach is presented for the solution of the model, scalar, non-linear, convection equation using an unstructured mesh. The technique is based on a linear approximation over the entire domain and the use of generalized control volumes for the evaluation of the fluxes. These control volumes are arbitrarily constructed about a node in the mesh and are contained within the patch of elements connected to that node. It is shown that the numerical algorithm can be written in terms of edges of the mesh, giving it a finite difference character. An issue not often addressed in finite volume techniques is the ability of the scheme to adequately represent unsteady flow phenomena. This is important since axial compressor flows are inherently unsteady. In this work, it is shown that the approximation used to spatially integrate the time derivative term in the Euler equations is crucial to the numerical scheme's dispersion characteristics.

Numerical dissipation is added to stabilize the solution while producing accurate results and adequately capturing shocks. The dissipation is based on the Total Variational Diminishing (TVD) property of the one dimensional convection equation and the Symmetric Limited Positive (SLIP) scheme of Jameson [11, 12]. Methods of reducing the amount of

dissipation in smoothly varying areas of the flow are addressed so that smooth extrema resulting from unsteady phenomena are not significantly attenuated. This leads to a class of numerical schemes known as Essentially Local Extremum Diminishing (ELED). It is also shown that the choice of control volumes can add a diffusion-like term into the numerical algorithm which must be taken into account in the application of the numerical dissipation.

The numerical scheme is extended to the compressible Euler equations and two simulations are performed to validate the algorithm developed. These simulations are a shock tube and a convected density perturbation in a uniform flow. The shock tube is used to test the numerical scheme's ability to capture shocks without numerical oscillations. The density perturbation is used to study the dispersion characteristics and the numerical dissipation. The algorithm is then applied to a transonic axial fan stage at two different operating speeds, 67 percent and 85 percent design speed. Steady state solutions are obtained and the results compared with the Euler turbine equation. An axial total pressure distortion is introduced into the steady state solution of the 67 percent design speed case to simulate a forced response test. Finally, conclusions and further recommendations are drawn from this work.

Chapter 2
Unstructured Mesh Generation

Unstructured meshes of triangles offer great flexibility when representing complex, two dimensional domains. An unstructured mesh is an arbitrary assembly of non-overlapping triangles which completely cover the region of interest. These triangles are often referred to as elements. Unstructured meshes give the added ability of adaptivity without the need for a large increase in the number of mesh points, contrary to structured meshes. This adaptivity also gives the increased ability to capture transient phenomena in the flow. With this in mind, an unstructured mesh was chosen over a structured mesh for a turbomachinery application.

The unstructured mesh's increased flexibility also lends itself to increased complexity. The data structure for storing the mesh should require as little storage as possible and still contain all the information necessary for geometric searches. The discrete boundaries in the mesh need to be adequately represented. Adequately represented means that information concerning the definition of the boundary, such as point location, tangents, vectors and curvature, must be easily attainable. Finally, the ability to perform an efficient geometric search on the mesh must be addressed. The geometric search of interest here consists of identifying the element of the mesh containing a given point.

Two popular techniques, the Delaunay and Steiner triangulations, are adequate for generating arbitrary two dimensional meshes [2, 24, 23, 4]. These techniques were chosen because of their similarity and generality. The algorithms presented in this chapter were implemented in a computer program titled 2D Unstructured Grids (2DUG) which can be used to create meshes for arbitrary domains and specifically for turbomachinery applications.

2.1 Data Structures

The nodes of the mesh are numbered and their actual cartesian (x,y) coordinates are stored in an array. The numbers of the nodes correspond to their position in the array, which is known as the global node list. To describe the connectivity of the nodes, an element data structure is used.

The elements of the mesh are defined by the three nodes which make up that element. Each element has three local nodes which contain pointers to the global node list. These local nodes are numbered so that the element has a counterclockwise orientation. This is

illustrated in figure 2.9.

Also included in this data structure are the three elements adjacent to the element. Each of the adjacent elements corresponds to the local node opposite it. Figure 2.9 shows the correspondence between the local node of an element and its adjacent elements. If an element does not have an adjacent element opposite to one of its local nodes, its value is set to zero signifying no element. This only occurs if the element is next to a boundary of the domain.

2.2 Geometric Curve Representation

The geometric representation of a curve is accomplished using a splining technique to ensure a smoothly varying and continuously defined curve. A technique developed in [5] is effective and efficient in producing smoothly varying splines for a given set of discrete points that define a curve. Each geometric curve is defined as a composite curve made of curve segments. The curve segments are represented as a cubic function of a parametric coordinate, s, the position of the endpoints, $\vec{R}$, and the tangent vector at the endpoints, $\vec{T}$. The coordinate of a point on each curve segment is given by:

$$\vec{R}(s) = \begin{bmatrix} 1 & s & s^2 & s^3 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{R}_1 \\ \vec{R}_2 \\ \vec{T}_1 \\ \vec{T}_2 \end{bmatrix}$$

where

$$\begin{aligned} \vec{R}_1 &= \textit{starting point of curve segment} \\ \vec{R}_2 &= \textit{ending point of curve segment} \\ \vec{T}_1 &= \textit{tangent vector at starting point} \\ \vec{T}_2 &= \textit{tangent vector at ending point} \end{aligned}$$

This is illustrated in figure 2.11. The coordinate, s, varies between zero and one, where zero and one correspond to the starting and ending points of the curve segment respectively.

2.2.1 Interpolation

The objective is to obtain a smoothly varying spline for the composite curve made up of curve segments. Each segment requires the specification of the tangent at its endpoints. These tangents are determined by requiring a continuous second derivative at each of the control points connecting the curve segments [5]. This leads to a tridiagonal system of equations to solve for the tangents at each of the points defining the curve. The only

additional requirement is the specification of the tangents at the two endpoints of the composite curve.

The parametric coordinate defining the curve segments is used to specify the location along the composite curve. Since the parametric coordinate for each curve segment is a local variable, it is possible to define a new global parametric coordinate, s^*, for the composite curve. This is accomplished by adding the curve segment number to the parametric coordinate. The first curve segment that defines the composite curve is numbered zero. For instance, $s^* = 1.5$ would be the local parametric coordinate $s = .5$ associated with the second curve segment. The above equation for $R(s)$, with $s = .5$ and the tangents and locations of the endpoints of curve segment one are used to determine the geometric coordinate of this point along the composite curve. Thus, $s^* = 1.5$ lies midway between control points one and two along the curve. This gives a reliable and efficient means for defining a curve in space. The tangent at any point along the composite curve is given by:

$$\vec{T} = \frac{\partial \vec{R}}{\partial s}$$

2.3 Geometric Searching

A geometric search is used in the mesh generation process to determine which element contains a given point. This can be accomplished in many ways. Some popular techniques use binary tree data structures [2] or digital trees [17, 3]. Another alternative is to use a walking algorithm. This technique is illustrated in figure 2.3 and is based on the use of area coordinates.

There are three area coordinates associated with a given point and an element (see figure 2.4). The coordinates are the ratios of the triangle areas defined by the three edges of the element with the given point and the total area of the element. The area of the element is given by the determinant:

$$Area = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

Since the nodes of the element are oriented in a counterclockwise manner, the sign of the determinant is guaranteed to be positive. The three areas associated with an edge of the element and the point are found in the same manner (see figure 2.4):

$$A_1 = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_p \\ y_2 & y_3 & y_p \end{vmatrix}, \; A_2 = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_3 & x_1 & x_p \\ y_3 & y_1 & y_p \end{vmatrix}, \; A_3 = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_p \\ y_1 & y_2 & y_p \end{vmatrix}$$

The area coordinates of the element, labeled (ξ_1, ξ_2, ξ_3) , are:

$$\xi_1 = \frac{A_1}{Area}, \; \xi_2 = \frac{A_2}{Area}, \; \xi_3 = \frac{A_3}{Area}$$

Each node of the element has a corresponding area coordinate. The first node has coordinate (1,0,0), the second node (0,1,0) and the third (0,0,1).

A point lying inside the element has all three determinants, A_1, A_2, A_3, positive. Also, it is easily verified that the three area coordinates will sum to one since each coordinate represents a fraction of the total area of the element, i.e.:

$$\sum_{k=1}^{3} \xi_k = 1$$

Therefore, it is only necessary to calculate two of the three area coordinates. The third is determined from the above relation. A point lying outside the element will have at least one of its area coordinates less than zero because the nodes will have a clockwise orientation resulting in a negative determinant.

The walking algorithm is implemented by starting from an arbitrary element and evaluating the three area coordinates of the given point with respect to that element. If any of the area coordinates are negative, the element corresponding to the node with a most negative area coordinate is searched next. This is repeated until all three area coordinates of a given element are positive. This is the element containing the point.

The domain in which the search takes place must be simply connected or the search may fail. Therefore, the convex hull of the domain must be entirely triangulated including the interior of geometric objects within the domain.

2.4 Meshing Techniques

2.4.1 Delaunay Triangulation

The Delaunay triangulation of a set of points is the triangulation that maximizes the minimum angle of the triangulation and by doing so minimizes the maximum containment circle of any triangle. The containment circle of an element is the circle passing through the three points that define the element (see figure 2.1). This construction guarantees that no other node of the mesh may lie within the containment circle of any element. This property is known as the circumcircle criteria.

Triangulating a given domain using this technique requires a starting mesh. The starting mesh must completely enclose the domain that is to be triangulated [24]. This mesh can be made up of two elements which are Delaunay, i.e. they satisfy the circumcircle criteria. Once these elements are created, it is then a matter of inserting points into the region to be triangulated and ensuring the newly created elements satisfy the Delaunay criteria. This is known as the insertion algorithm [2] and is summarized as follows:

Insertion Algorithm

1. Create a starting mesh enclosing the domain.
2. Insert point and determine in which element the newly inserted point lies.
3. Find all elements surrounding this element that fail to meet the circumcircle test with the newly inserted point and remove them, forming a convex cavity as shown in figure 2.5.
4. Retriangulate the cavity by joining the new point to the exposed edges from the convex cavity.
5. Return to step two to insert additional points.

Step two of the list corresponds to searching through the domain and finding which element contains the newly inserted node. This is accomplished using the searching algorithm discussed earlier.

The Delaunay technique lacks the ability to guarantee that a given geometric boundary will be part of the final triangulation. Therefore, in applying this technique for the purpose of creating a mesh around a given object, the algorithm needs to be modified to ensure the edges of a given geometric boundary are preserved when the triangulation is complete. This modified algorithm is known as a constrained Delaunay triangulation. In this work the algorithm for the constrained Delaunay triangulation, developed by George [6], has been adopted. The process is implemented after all the points have been inserted and is based on recursively swapping the edges intersecting the edge to be recovered. This process is illustrated in figure 2.8.

2.4.2 Steiner Triangulation

A Steiner triangulation is a triangulation produced by gradually modifying an existing mesh using a local mesh operation such as point insertion and edge swapping. With this type of algorithm, geometric boundaries that might be lost with the Delaunay technique can be preserved. This triangulation still requires the creation of starting elements. Instead of finding all the elements that fail the circumcircle test, three new elements are created from the element containing the new point. The three edges that define the element containing the newly inserted point are considered *suspect* edges and are placed on a stack. This is illustrated in figure 2.6 with the suspect edges marked with *X's*. The element pairs that share these suspect edges are determined. Based on a given mesh quality criteria, a decision is made on whether or not to swap the edge [2]. In this respect, the algorithm described in the previous section for creating Delaunay triangulations can be considered a Steiner algorithm if the circumcircle criteria is used. However, if different

criteria for edge swapping are used, a much wider class of triangulations can be generated. Edge swapping is a local operation only involving the two elements that share the edge. Swapping the edge creates two more suspect edges which are added to the suspect edge stack as shown in figure 2.6. If the element pair sharing a suspect edge does not fail the circumcircle test then the edge is removed from the stack. This process is repeated recursively until no suspect edges are left. The algorithm is summarized as follows:

Steiner Triangulation

1. Create a starting mesh enclosing the domain.
2. Insert new point into triangulation and determine element in which this newly inserted point lies.
3. Subdivide the element containing the new point into three new elements and remove the containment element.
4. Place edges forming the deleted element on a suspect edge stack.
5. Take edge from stack and determine which elements share this edge.
6. Determine if the elements sharing this edge violate the Delaunay criteria. If so, then swap suspect edge and place the four new suspect edges on the suspect edge stack.
7. If suspect edges remain, go to step five.

The same procedures for geometric searching outlined in the previous section are used for this triangulation.

If one of the suspect edges is a geometric boundary that is to be kept, it can be saved by not swapping it. In this way, an algorithm for a constrained Delaunay triangulation is easily created. An advantage of the Steiner procedure is that different criteria to determine whether a suspect edge should be swapped can be incorporated. Other criteria such as minimizing the maximum angle can be used. In this way, this technique is more general and is usually preferred over the standard Delaunay triangulation.

2.5 Mesh Refinement

The Delaunay and Steiner techniques both assume that the points to be triangulated are known apriori. Generally, this is not the case. Therefore, a method of determining the points to be triangulated based on some type of mesh control is needed. Also, the triangulations may have badly shaped elements. These elements need to be identified and corrected in some way. These issues will be addressed in this section.

2.5.1 Mesh Control Functions

A mesh control function is used in order to specify the characteristics of the mesh to be generated. This function gives the local mesh size (given by the edge length) as a function of the spatial coordinates. The mesh control function must be supplied and is specified by using a background mesh, point sources and line sources.

Background Mesh

The background mesh must completely enclose the domain to be triangulated. The background mesh is comprised of background elements. The nodes of these elements are associated with a specified edge length. By linearly interpolating over each element, a piecewise linear distribution of the edge length is produced over the entire domain. This edge length is used in the refinement process and discussed later.

Point and Line Sources

Point sources are used in the refinement process to cluster elements in the areas where they appear. While similar, line sources are defined by two points and are used in the refinement process to cluster elements around them. The equation that governs the edge length specified by the point sources is:

$$\delta = \begin{cases} \delta_{min} & if\ r < r_{min} \\ \delta_{min}\ \ln(2)\ \exp\left(\frac{r - r_{min}}{r_{max} - r_{min}}\right) & if\ r > r_{min} \end{cases}$$

where r_{min} is the radius up to which the edge size δ_{min} is required ; r_{max} is the radius at which $\delta = 2\delta_{min}$; and r is the distance from a given point to the source. The line sources are governed by the same equation as the point sources with the distance r being the distance to the line source.

2.5.2 Refining Mesh According to the Mesh Control Functions

The starting point for the refinement procedure is an initial triangulation. This triangulation is obtained by triangulating the points lying on the boundary curves of the domain [24, 2]. The process of refining the mesh begins by systematically sweeping through the edges of the current mesh. The midpoint of the edge is determined and the element of the background mesh containing this point is determined. This background element is used to linearly interpolate the edge size from its three nodes. This edge size is labeled δ_e.

The distance of the edge's midpoint from the point and line sources is also determined. The edge length determined from the point and line sources are labeled δ_p and δ_l respectively.

The minimum of these three edge lengths is then used, i.e.:

$$\delta_{required} = minimum(\delta_e, \delta_p, \delta_l)$$

If the edge length is less than a specified percentage of the required edge length, $\delta_{required}$, the edge is removed by collapsing the two end nodes to the midpoint. If the edge length is greater than a specified percentage of the required edge length, a new point is inserted into the triangulation at the midpoint of the edge. The mesh is continually swept over until all edges meet the requirement of the specified control functions. This refinement technique, adopted from [17], does not guarantee that created elements will be well shaped.

2.5.3 Fixing Skew Elements

Skew elements are considered to be elements containing an interior angle greater than some maximum specified angle. If an element contains an angle greater than this maximum, a swapping operation is attempted in an effort to reduce skewness. This is illustrated in figure 2.7. This operation is performed iteratively until no more edges are swapped.

2.5.4 Smoothing the Mesh

The quality of the mesh may be enhanced by moving the nodes to produce a smooth mesh. This is achieved by updating the position of the nodes according to:

$$(x^*, y^*) = \frac{\sum_{k=1}^{n}(x_k, y_k)}{n}$$

where the $*$ superscript denotes the new position of the node and n is the number of nodes connected to it. This is performed in an iterative manner by sweeping over the mesh and updating the position of each node separately. Performing approximately 10 sweeps over the mesh is sufficient for enhancing the quality. The nodes of the boundaries are not included in this procedure since they must remain fixed.

2.6 2D Unstructured Grids

The techniques mentioned earlier were used to develop an unstructured mesh generator for arbitrary two dimensional domains. The flowchart for the operation of the code is:

Flowchart for 2DUG

1. Read in project file

2. Read in boundary curves
3. Read in background mesh and point and line sources
4. Spline boundaries to meet mesh control criteria
5. Create starting elements
6. Insert boundary points and triangulate
7. Recover boundary edges
8. Determine protected elements
9. Refine mesh according to control function
10. Smooth mesh
11. Recover any boundary edges lost in the refinement process
12. Remove unwanted elements from triangulation and correct skew elements
13. Smooth mesh
14. Output database

Several examples are used to test the capability of the mesh generator, 2DUG. The ability to control the mesh generation process is controlled through the control function.

2.7 Examples

2.7.1 Rectangular Region

The first test case is a simple square region. Two different background meshes are used to demonstrate the user's ability to control the mesh generation process. The first is a simple background mesh consisting of two elements. Each node of the background mesh carries the same edge size. This creates a fairly uniform mesh over the entire domain. The second example contains a line source spanning diagonally across the region. This clusters elements along the diagonal of the square region.

Uniform Background Mesh

The result from this mesh generation is shown in figure 2.12. The mesh appears to be fairly uniform and no badly shaped elements are visible. The edge size was specified to

be .05 at the nodes of the background mesh. The maximum edge length over the entire mesh is approximately .12 and the minimum is .03. This is within a reasonable range of the specified edge length. The maximum specified interior angle for a given element was 110°. The maximum interior angle of an element over the entire mesh was 128° which is slightly above the specified value. Overall, the generated mesh was acceptable.

Line Source in Background Mesh

The result from the mesh generator is shown in figure 2.13. The elements are clustered around the diagonal of the square region as expected. This gives the user very good control of the mesh with very little input.

2.7.2 Compressor Blade

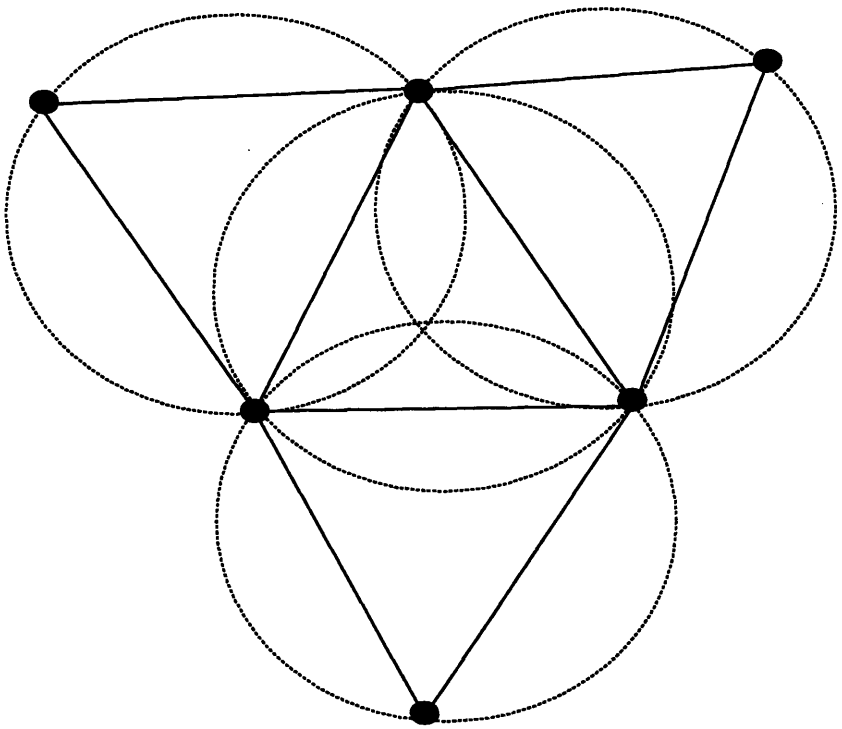
The cross section of a compressor blade is used for this example. This example tested the mesh generator's ability to construct periodic boundaries in the mesh generation process. A periodic boundary is a boundary that has an identically matching counterpart. These boundaries have pairs of coinciding points and are used in flow solvers in situations where the flow may be modeled as periodic in a certain direction. The background mesh contains two elements which specify a uniform edge size over the entire mesh. Also included are two point sources at the leading and trailing edges of the blade, as well as a line source along the blade. The point sources are chosen to help resolve the leading and trailing edges of the blade. The line source is chosen to cluster elements in the blade passage along the blade.

The result is shown in figures 2.14 and 2.15. The mesh generator effectively clusters elements in the blade passage and is able to adequately resolve the leading and trailing edges. The upper and lower boundaries appear to be periodic. This is verified by patching a duplicate of the mesh to the original mesh and indeed showing that the upper and lower boundaries are the same. This is shown in figure 2.16.

2.8 Conclusions

The mesh generator developed with the techniques outlined in this chapter appears to yield good results for the two dimensional cases tested. The mesh generator is able to mesh complex two dimensional domains with very little user input. Mesh control functions provide the added ability to control the mesh depending on the application. The compressor example showed that this mesh generator offers the ability to mesh various two dimensional compressor cascades with periodic boundaries for an application with a

flow solver. Overall, these techniques are flexible, providing meshes with good quality for a wide range of applications.



No other node lies within the circumcircle of
any element

Figure 2.1: Circumcircle Test

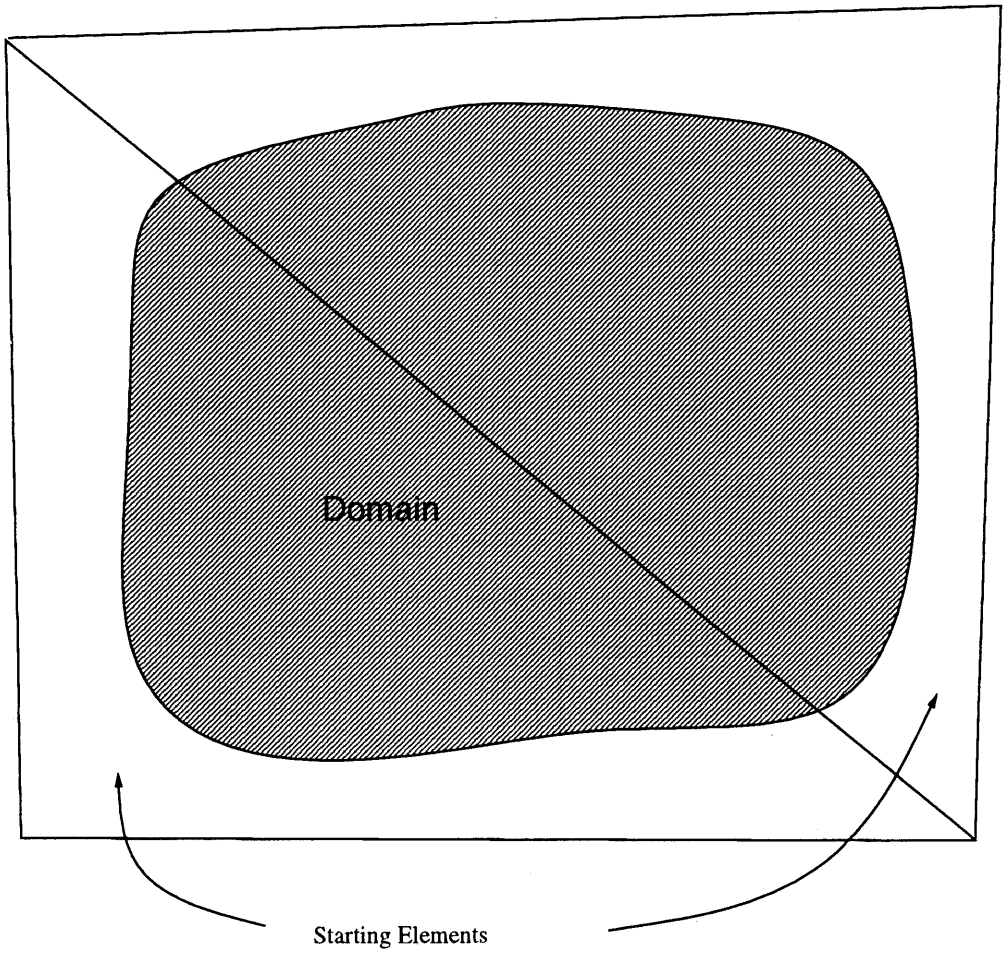


Figure 2.2: Convex Hull

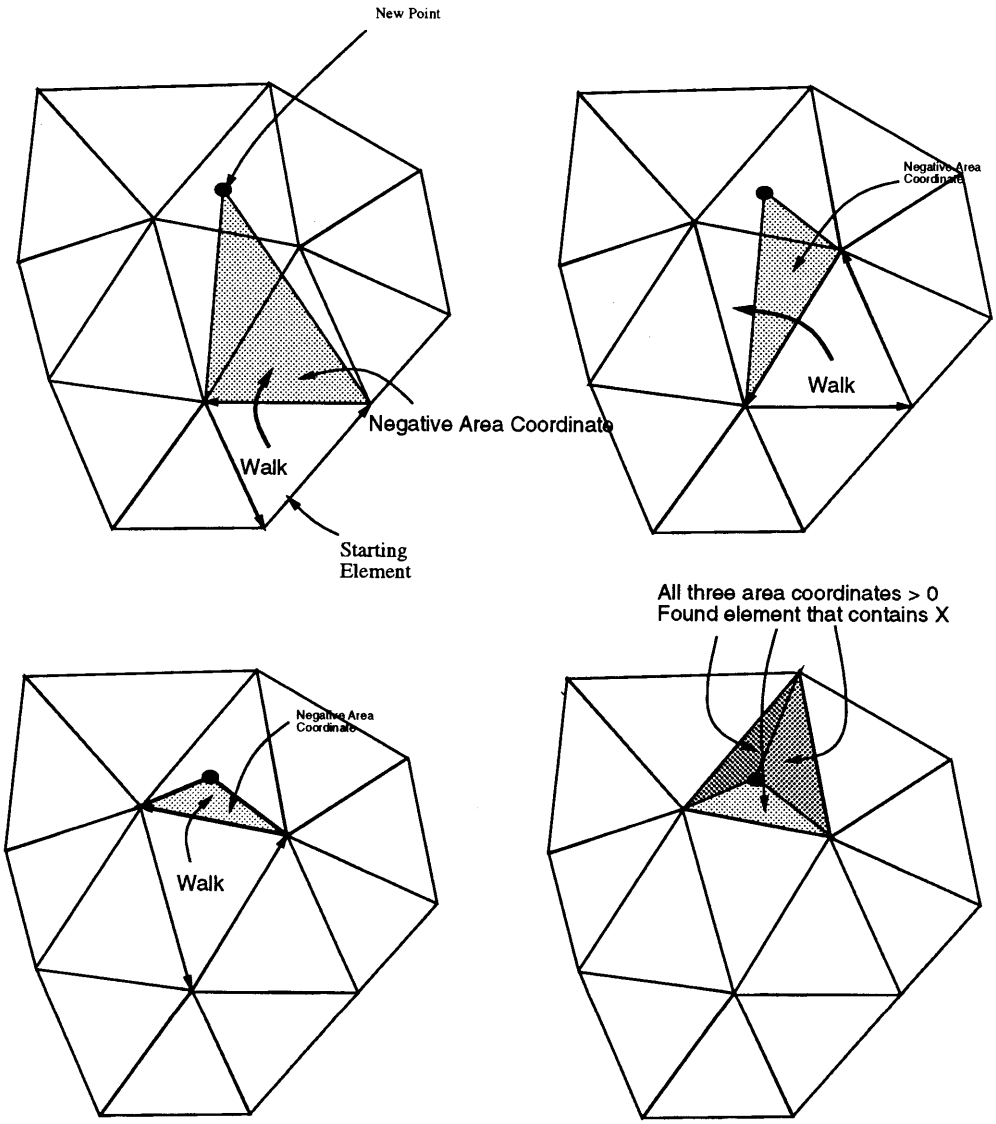
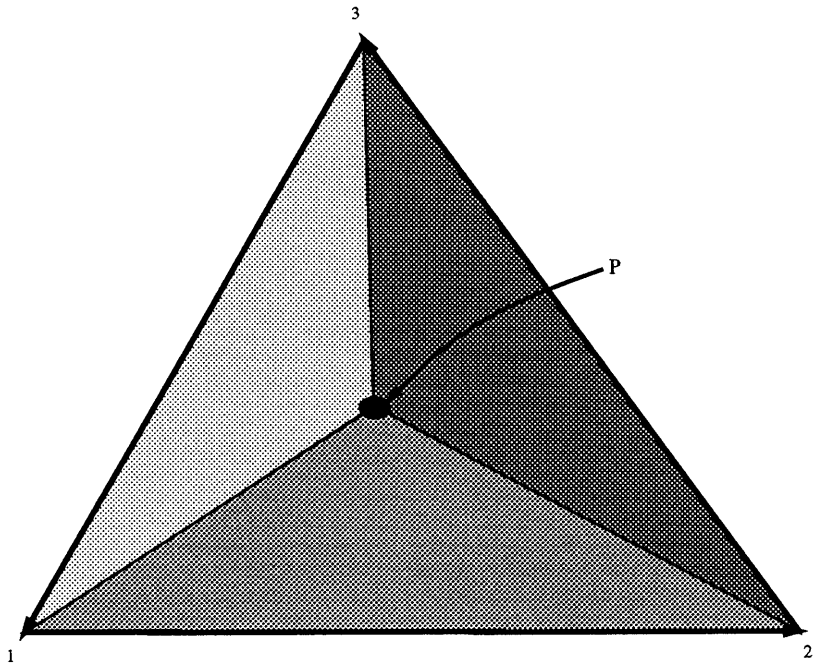


Figure 2.3: Geometric Search



$$\xi_1 = \frac{\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_p \\ y_1 & y_2 & y_p \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}} \qquad \xi_2 = \frac{\begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_p \\ y_2 & y_3 & y_p \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}}$$

$$\xi_3 = 1 - \xi_1 - \xi_2$$

Figure 2.4: Area Coordinates

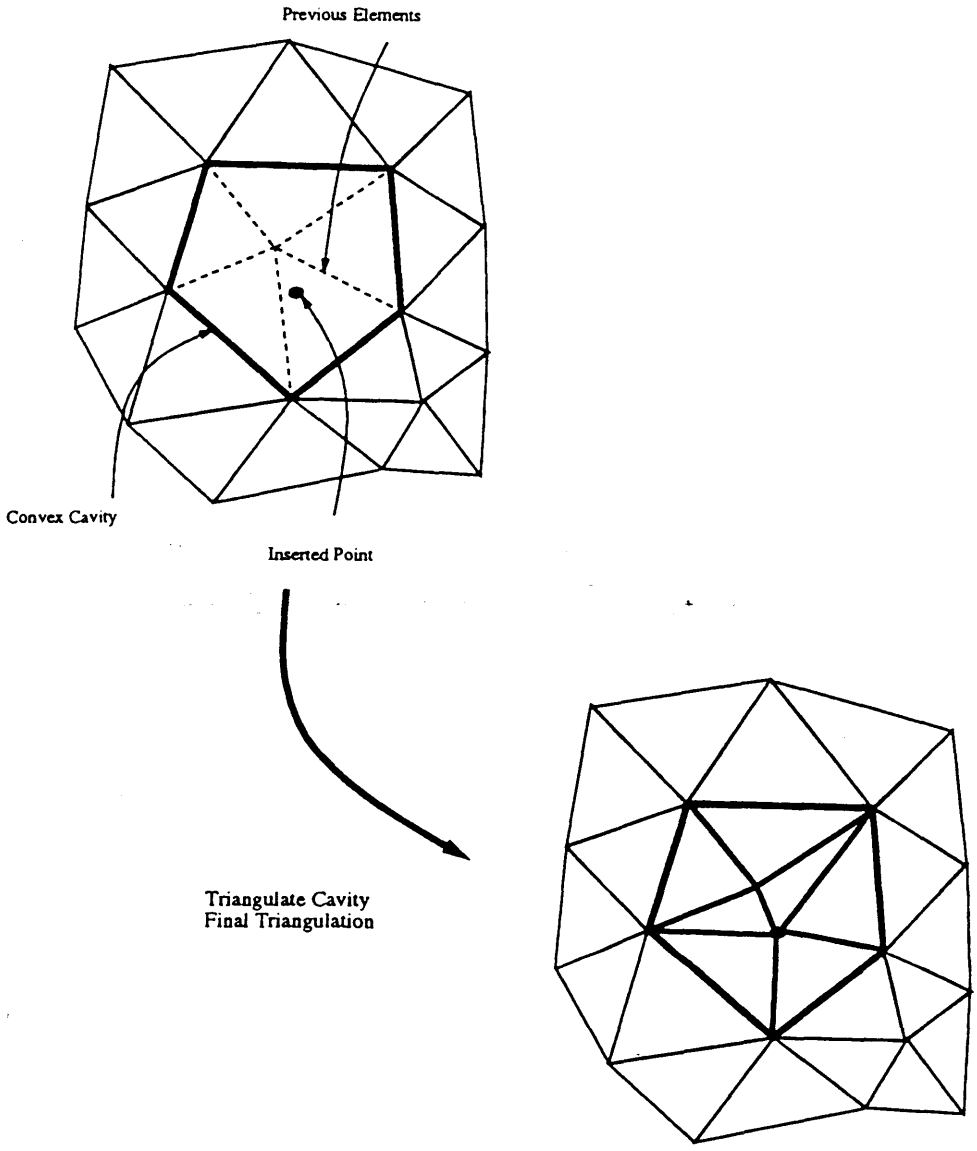


Figure 2.5: Convex Hull

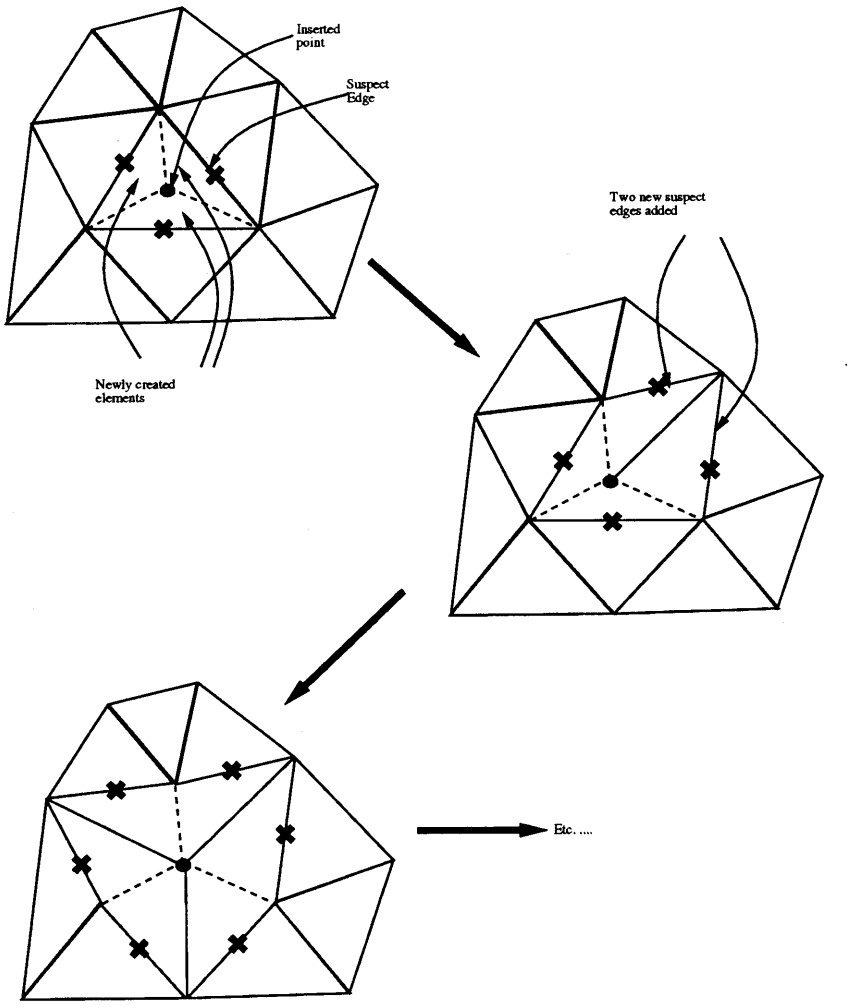


Figure 2.6: Steiner Algorithm

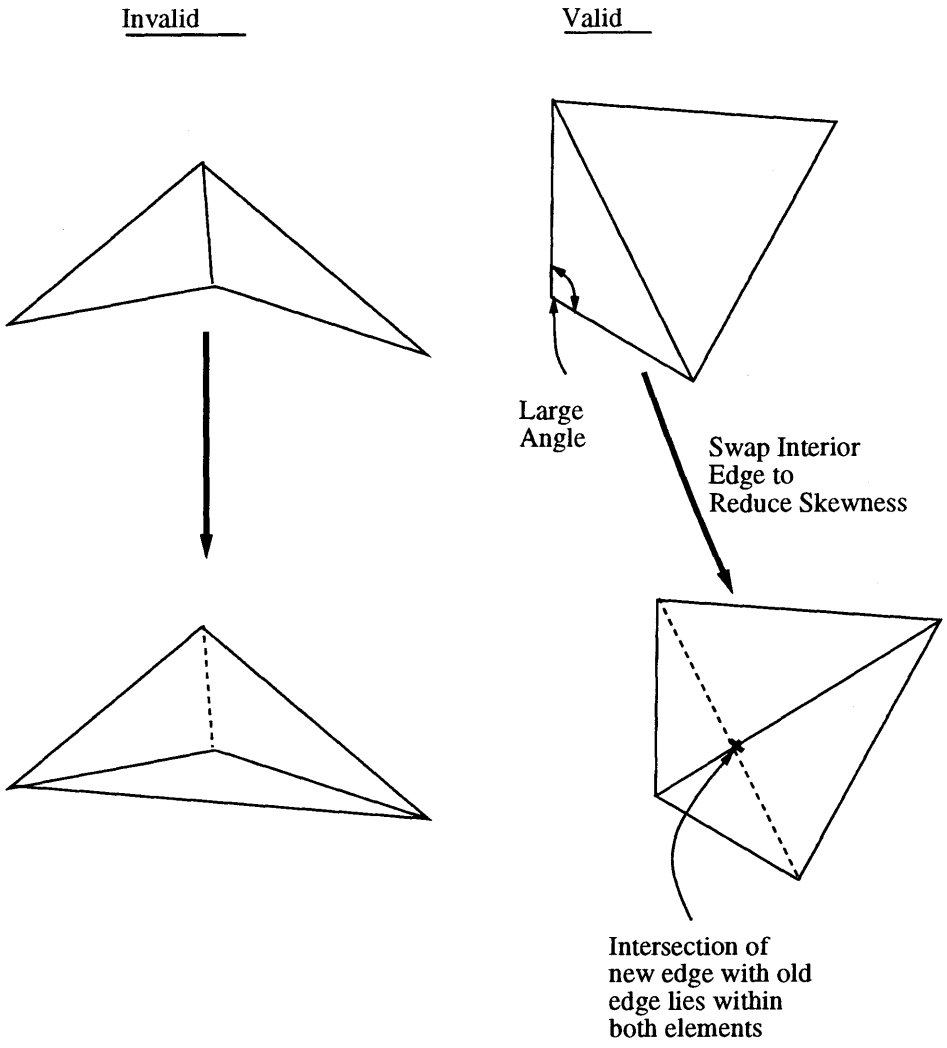


Figure 2.7: Edge Swapping

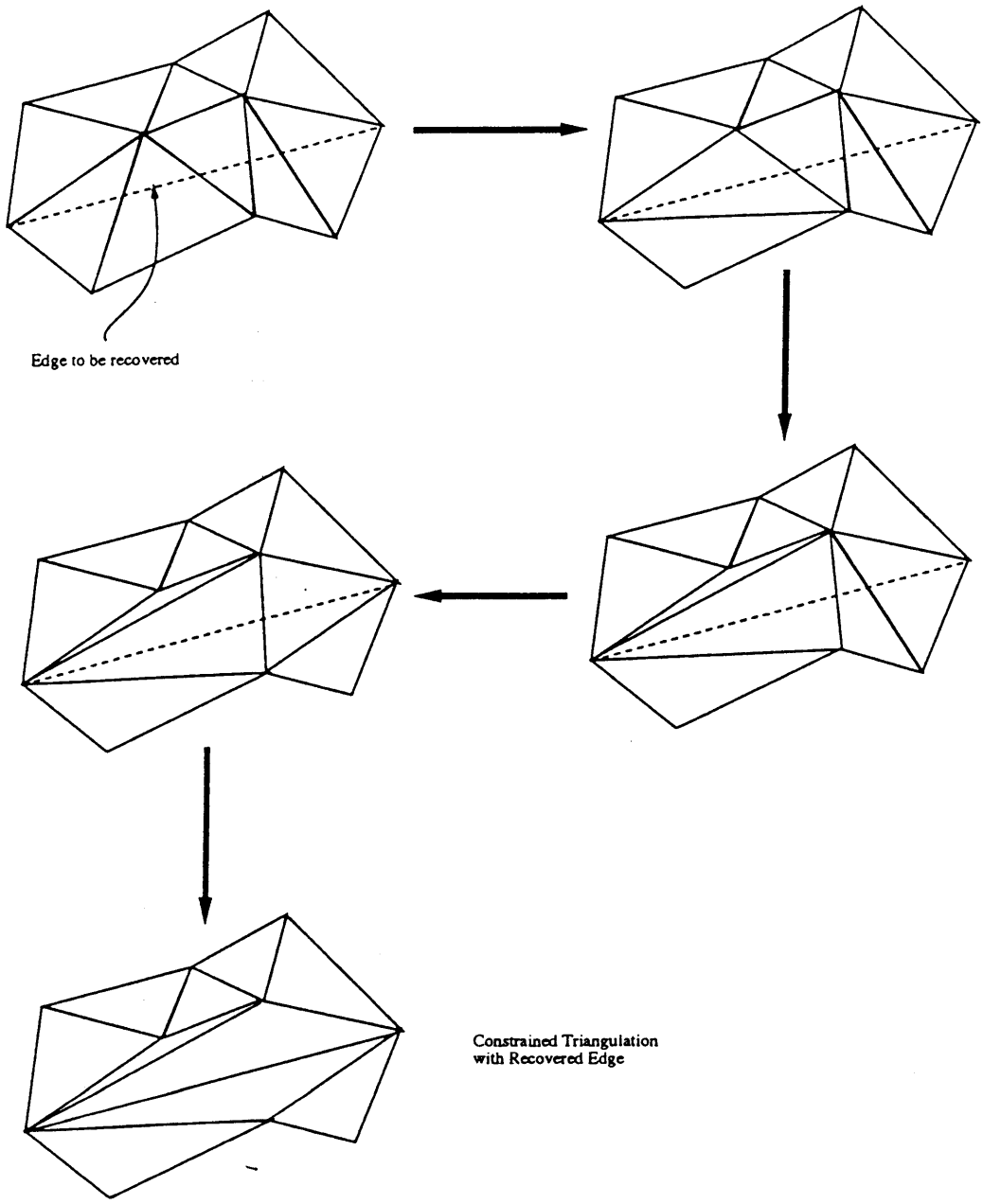
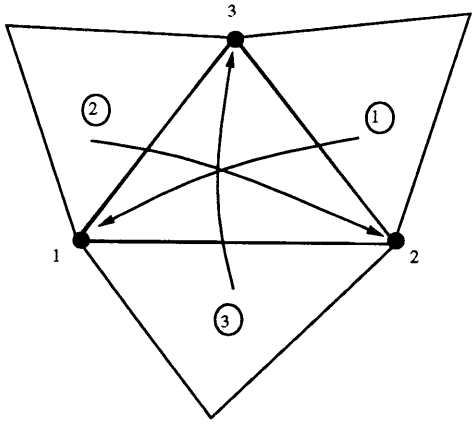


Figure 2.8: Edge Recovery

Element Based Data Structure



Edge Based Data Structure

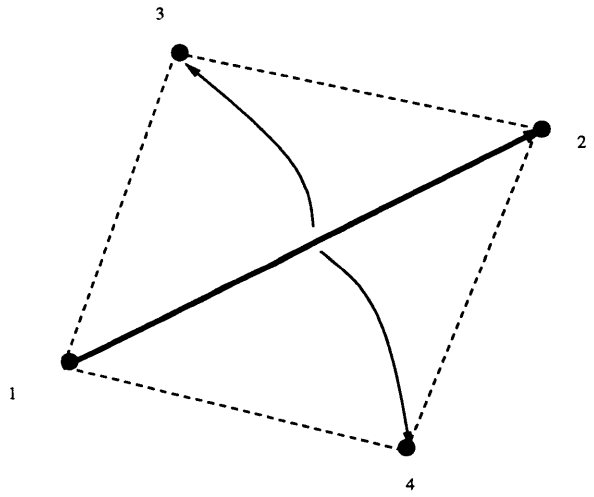


Figure 2.9: Data Structures

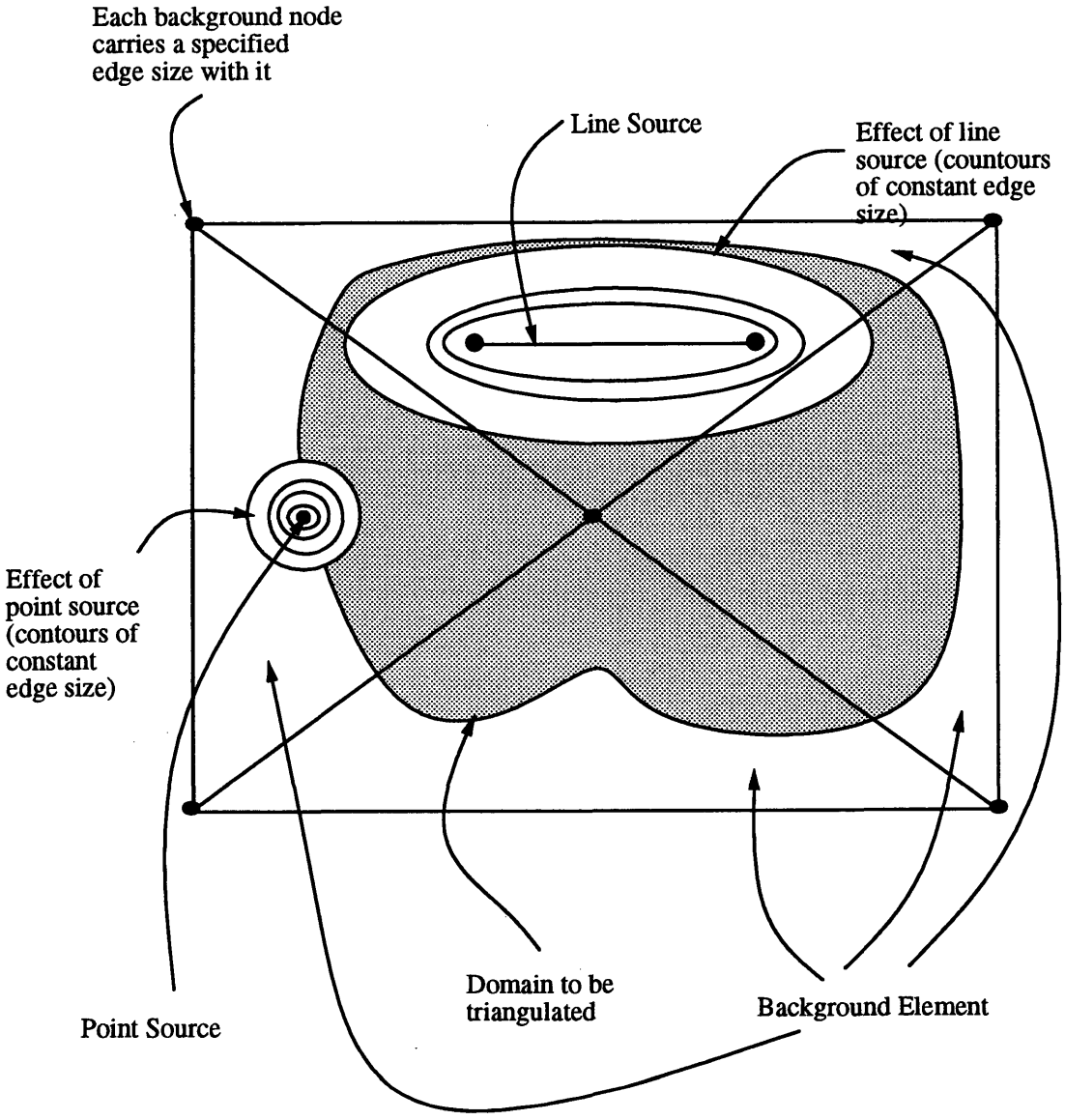


Figure 2.10: Graphic of Background Mesh

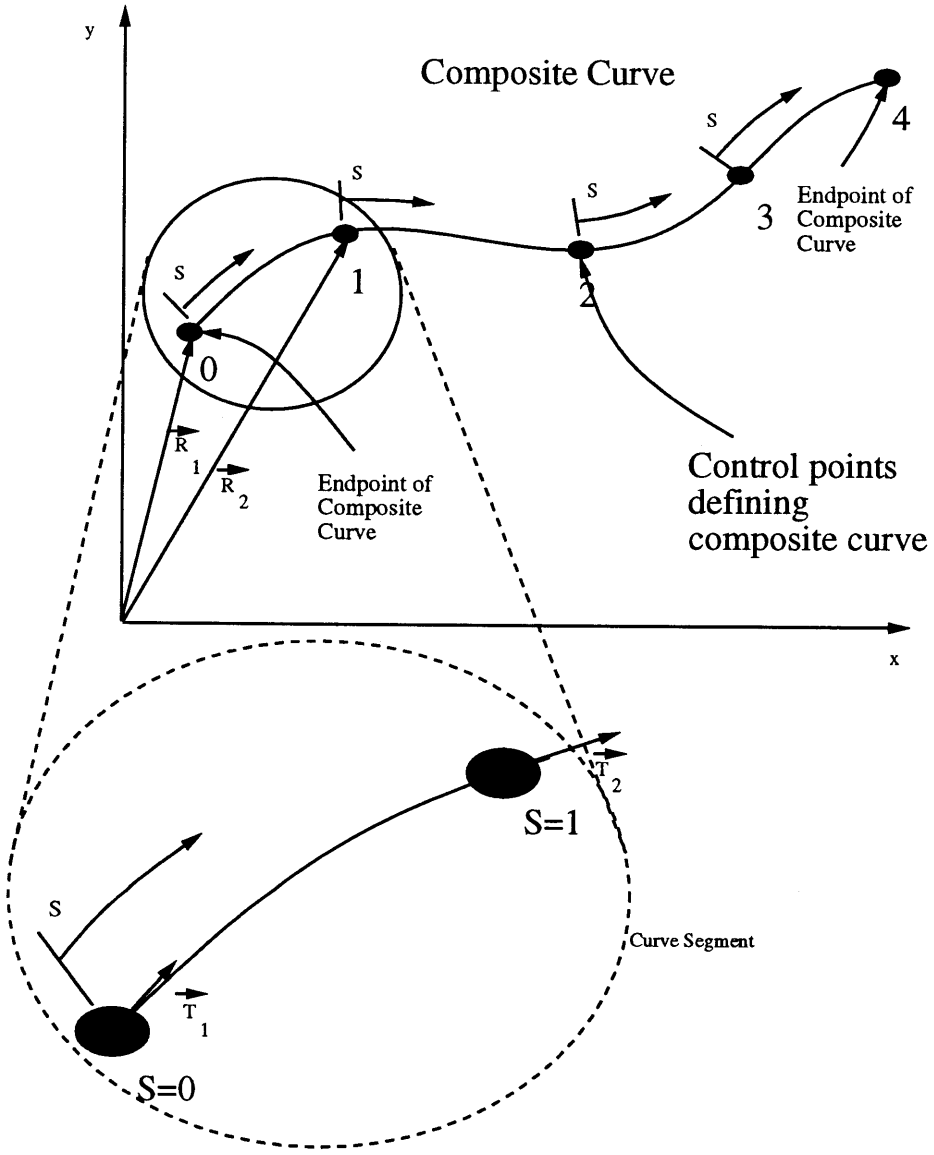


Figure 2.11: Geometric Curve Definition

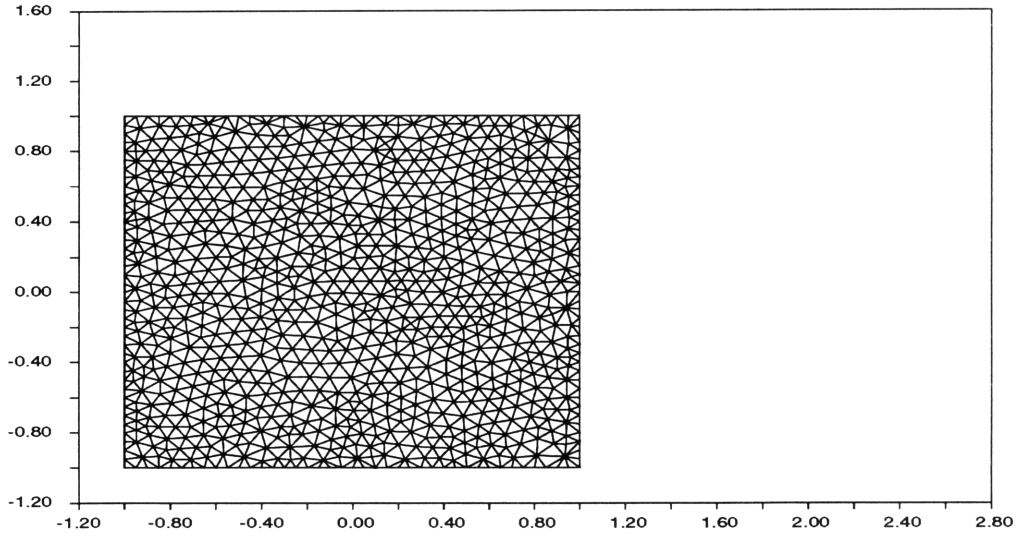


Figure 2.12: Square Mesh Test Case

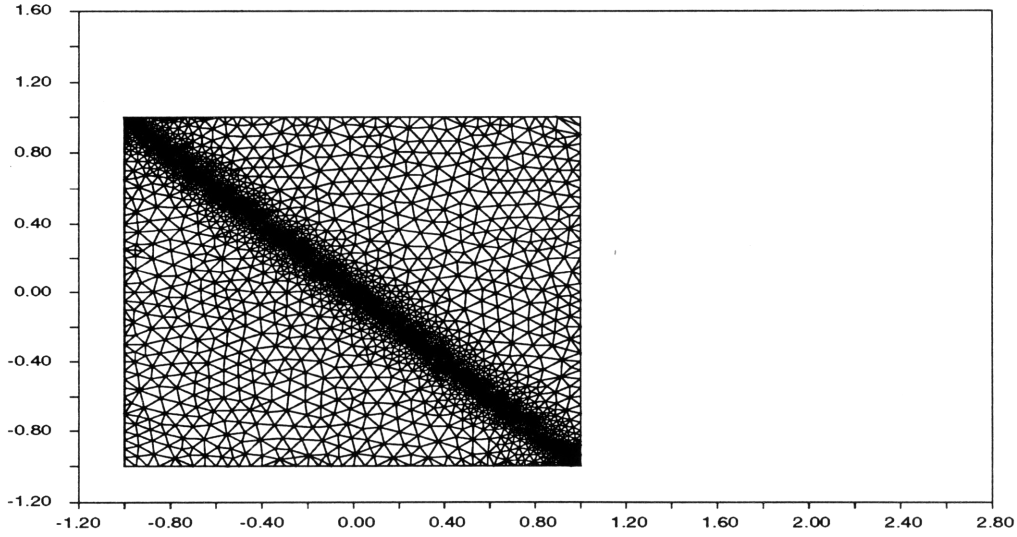


Figure 2.13: Square Mesh Test Case with Line Source

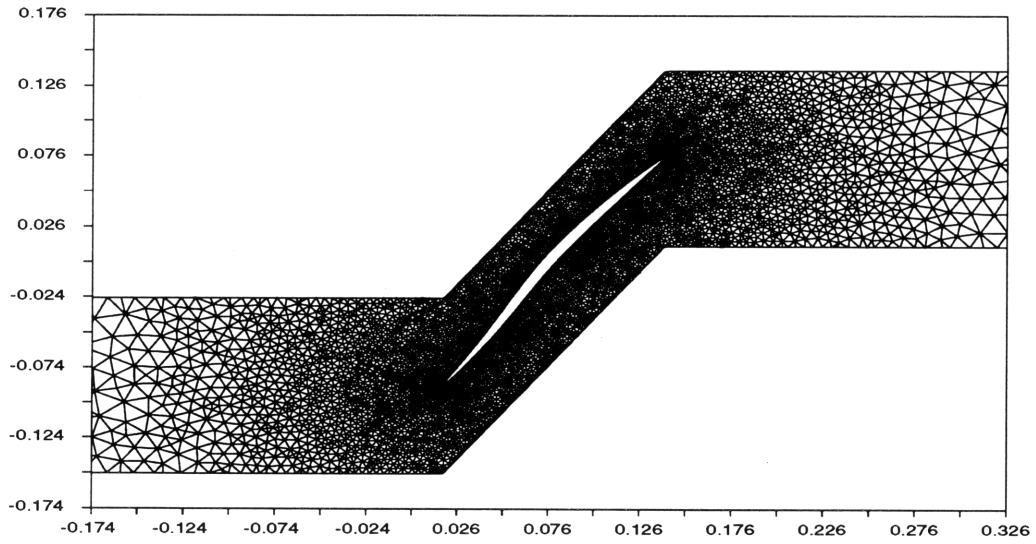


Figure 2.14: Compressor Blade Test Case

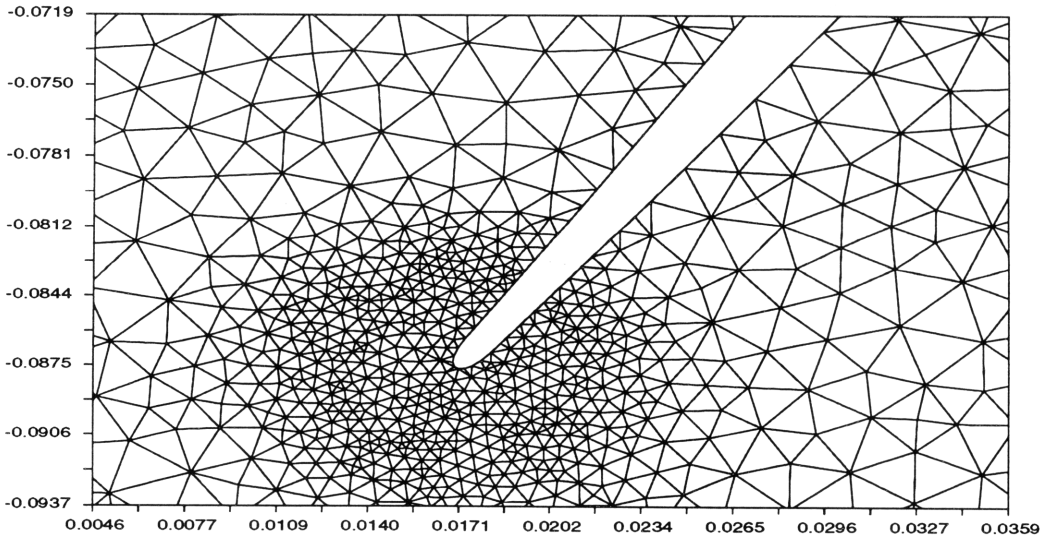


Figure 2.15: Compressor Blade Test Case Leading Edge

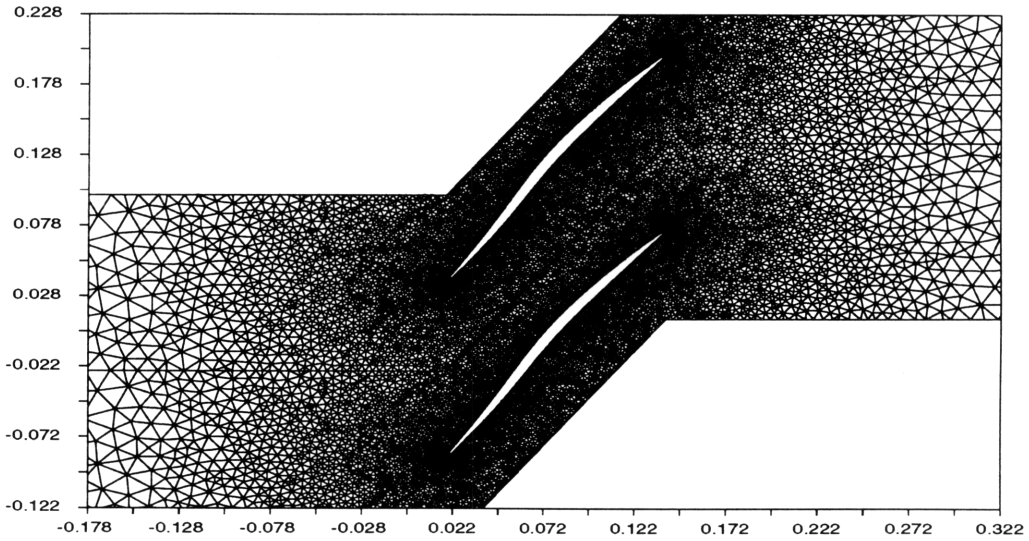


Figure 2.16: Patched Meshes for Compressor Blade Test Case

Chapter 3
Formulation of a 2D Generalized Unstructured Finite Volume Scheme

Axial turbomachines contain flows which are three dimensional, compressible and viscous. However, some regions along the blade span of the turbomachine may be modeled using a two dimensional approximation of the flow, assuming the radial deviation of the flow from the leading to the trailing edge of the blade is small. Ignoring viscous effects, the flow in these regions is governed by the two dimensional, compressible, inviscid Euler equations.

One method for numerically solving the Euler equations is to use the differential form of these equations and obtain approximations to the derivatives. This approach is known as the finite difference technique, it is a popular method of solving the Euler equations. This method requires the use of a structured mesh and does not readily apply when using an unstructured mesh. Another method, and the one adopted in this chapter, is to represent the domain as a collection of non-overlapping volumes. The fluid flow within each of these volumes, or subvolumes, is governed by the Euler equations written in integral form. Therefore, the problem over the whole domain is broken into smaller problems over each subdomain or subvolume. This approach is known as the finite volume approximation to the Euler equations.

An unstructured mesh provides the means of partitioning a given domain into subdomains. The integral form of the Euler equations is numerically approximated over each subdomain. In this chapter a generalized, unstructured, finite volume scheme is formulated. Generalized means that given an unstructured mesh, the control volumes associated with a node can be chosen arbitrarily provided they are contained within the elements connected to the node. The flow variables are defined at the nodes of the mesh and are approximated over the whole domain using a piecewise linear representation.

Since turbomachinery flows are unsteady, it is important to analyze a numerical scheme's ability to capture unsteady phenomena. A Fourier analysis is used to investigate the scheme's dispersion characteristics. It will be shown that the unstructured finite volume scheme developed in this chapter is capable of adequately representing unsteady phenomena.

The addition of artificial dissipation to the scheme developed will be addressed. This dissipation is based on the work of Jameson [11, 12] and incorporates the concept of Total Variational Diminishing (TVD) schemes. The scheme developed is shown to possess good shock capturing properties. Finally, the time integration for the scheme and the time step

restriction is discussed.

3.1 Formulation

3.1.1 Piecewise Linear Approximation (Shape Functions)

Given an unstructured mesh of triangles, a piecewise linear continuous approximation of a function can be written as [25]:

$$\phi(x,y) \approx \sum_{i=1}^{n} \phi_i N_i(x,y) \tag{3.1}$$

where n is the number of nodes in the mesh, and ϕ_i are the values of $\phi(x,y)$ at the nodes of the mesh. The function $N_i(x,y)$ is termed the hat function associated to node i and is defined as:

$$N_i(x,y) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \quad j = 1, \ldots, n \tag{3.2}$$

and varies linearly within each element, as illustrated in figure 3.1. Note that within an element these hat functions are identical to the area coordinates described in Chapter 2.

3.1.2 General Control Volumes

A generalized control volume is a volume which can be arbitrarily drawn around a given node and lies within the patch of elements connected to that node. The control volumes must completely fill the space between the nodes without overlapping. In the present work, the generalized control volumes associated to a node are determined after arbitrarily specifying a point on each edge and a point in the interior of each element. This is illustrated in figure 3.3.

3.1.3 Finite Volume Scheme Using the Model Scalar Equation

The model scalar convection equation in two dimensions is used as the starting step in the development of a numerical scheme for the compressible Euler equations. The model equation is:

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0$$

Since this equation is satisfied everywhere within a domain, then it must also hold that the integral of this equation over the entire domain must also equal zero, i.e:

$$\int\int_{\Omega} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} \right) \partial\Omega = 0 \tag{3.3}$$

A more useful form of this equation for finite volume schemes is to use the divergence theorem to rewrite the flux term in terms of a contour integral around the domain:

$$\int\int_{\Omega} \frac{\partial u}{\partial t} \partial\Omega = -\oint_S (f \cdot n_x + g \cdot n_y) dS \tag{3.4}$$

where S is the boundary enclosing the domain and n_x, n_y are the outward pointing normals of this boundary. (Note, this equation is identically satisfied for any volume within the domain as well.)

Fluxes

The right side of equation 3.4 can be written as:

$$\oint_S (f \cdot \hat{n}_x + g \cdot \hat{n}_y) dS = \sum_{j=1}^{nv} \oint_{S_j} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) dS_j \tag{3.5}$$

where nv is the number of non-overlapping subvolumes in Ω and S_j is the boundary of each subvolume as illustrated in figure 3.2. Note that portions of a boundary shared by any two subvolumes will contribute quantities which are identical in magnitude, but different in sign to the summation. Therefore, the net contribution from the interior boundaries is zero. Any subvolume containing a surface lying on the outer boundary of the domain will have a net contribution to the summation.

The integrals in equation 3.5 can be approximated as follows:

$$\sum_{j=1}^{nv} \oint_{S_j} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) \, dS_j \approx \sum_{j=1}^{nv} \oint_{S_j} \left(\sum_{i=1}^{np} f_i N_i(x,y) \cdot \hat{n}_x + g_i N_i(x,y) \cdot \hat{n}_y \right) dS_j \tag{3.6}$$

where $N_i(x,y)$ are the hat functions described earlier. Figure 3.4 shows a restricted generalized control volume constructed about a node labeled j. The points where the segments defining the boundary of the control volume intersect are numbered in a counterclockwise manner with the actual mesh overlayed on top. The subvolume is labeled as subvolume j for the node that is contained within.

The line integral in equation 3.6 is determined by evaluating the integral on each segment, or interval, and summing all the contributions. The integral over each segment is evaluated using Gauss Quadrature [25]. Since the approximating functions, $N_i(x,y)$, are linear, this function can be integrated exactly over an interval by sampling the function at the midpoint of the interval and multiplying by the interval's length.

For example, the contribution to the line integral for subvolume j from interval (1,2) is:

$$\oint_{S_{j_{(1,2)}}} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) dS_{j_{(1,2)}} \approx f_{M_{(1,2)}} \Delta y^{(1,2)} - g_{M_{(1,2)}} \Delta x^{(1,2)} \tag{3.7}$$

where the subscript M denotes the midpoint of interval (1,2) and $\Delta x^{(1,2)}$, $\Delta y^{(1,2)}$ are given by:

$$\begin{aligned} n_{x_{(1,2)}} dS_{j(1,2)} &= \Delta y^{(1,2)} \\ n_{y_{(1,2)}} dS_{j(1,2)} &= -\Delta x^{(1,2)} \end{aligned}$$

The values f_M and g_M are determined from the approximation in equation 3.1. This gives:

$$\begin{aligned} f_{M_{(1,2)}} &= f_j N_j(x_m, y_m) + f_1 N_1(x_m, y_m) + f_2 N_2(x_m, y_m) \\ g_{M_{(1,2)}} &= g_j N_j(x_m, y_m) + g_1 N_1(x_m, y_m) + g_2 N_2(x_m, y_m) \end{aligned}$$

The m subscript denotes the midpoint of the interval (1,2). (Note that the subscripts on f, g and $N(x,y)$ refer to the nodes of the unstructured mesh, not the points of the subvolume. The points of the subvolume are enclosed in parenthesis.) The values of $N(x,y)$ can be thought of as weights that specify the amount of influence a node has on the integration. Substituting into equation 3.7 yields:

$$\begin{aligned} \oint_{S_{j(1,2)}} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) \approx\ & (f_j N_j + f_1 N_1 + f_2 N_2)\, \Delta y^{(1,2)} - \\ & (g_j N_j + g_1 N_1 + g_2 N_2)\, \Delta x^{(1,2)} \end{aligned}$$

where (x_m, y_m) has been dropped with the understanding that $N(x,y)$ is evaluated at the midpoint of the interval of interest. Within a given element the sum of the three hat functions, $N(x,y)$, associated with the nodes of the element must equal one. Therefore, for a given element:

$$\sum_{i=1}^{3} N_i = 1 \Rightarrow N_j = 1 - N_1 - N_2$$

Using the above expression in the previous equation gives:

$$\begin{aligned} \oint_{S_{j(1,2)}} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) \approx\ & [f_j + (f_1 - f_j)N_1 + (f_2 - f_j)N_2]\, \Delta y^{(1,2)} - \\ & [g_j + (g_1 - g_j)N_1 + (g_2 - g_j)N_2]\, \Delta x^{(1,2)} \end{aligned}$$

An analogous integral can be written for each segment belonging to subvolume j.

A general form for this integral that can be used in the implementation of a computer algorithm is needed. Considering all the elements sharing node j yields:

$$\begin{aligned} \oint_{S_j} (f \cdot \hat{n}_x + g \cdot \hat{n}_y)\, \partial\Omega_j \approx\ & \sum_{e=1}^{ne} \sum_{i=1}^{2} \left[f_j^{(e)} + \left[\sum_{k=1}^{2} \Delta f_{jk}^{(e)} N_k^{(i,i+1)} \Delta y^{(i,i+1)} \right] - \right. \\ & \left. g_j^{(e)} + \left[\sum_{k=1}^{2} \Delta g_{jk}^{(e)} N_k^{(i,i+1)} \Delta x^{(i,i+1)} \right] \right] \end{aligned}$$

where the first summation extends over the elements attached to node j, the second is the *local* node number for the element and the third summation is over the segments for

subvolume j contained within the element. Each element is assumed to have node j as its third local node making the innermost summation valid. The second summation over the subvolume faces are local numbers as well. For each element sharing node j, two segments must be considered. This is illustrated in figure 3.5. The increments Δf_{jk} are:

$$\Delta f_{jk} = f_k - f_j$$

and likewise for Δg_{jk}. The above expression may further be simplified by noting that:

$$\sum_{e=1}^{ne}\sum_{i=1}^{2} f_j \Delta y^{(i,i+1)} = 0 \quad , \quad \sum_{e=1}^{ne}\sum_{i=1}^{2} g_j \Delta x^{(i,i+1)} = 0$$

since f_j and g_j are constant for the volume and may be taken out of the summation. The above expression then simplifies to:

$$\oint_{S_j} (f \cdot \hat{n}_x + g \cdot \hat{n}_y)\, \partial\Omega_j \approx \sum_{e=1}^{ne}\sum_{i=1}^{2}[\sum_{k=1}^{2}(\Delta f_{jk}^{(e)} N_k^{(i,i+1)} \Delta y^{(i,i+1)} - \Delta g_{jk}^{(e)} N_k^{(i,i+1)} \Delta x^{(i,i+1)})]$$

Since a given edge is shared among two elements, the above equation is further simplified by combining expressions that multiply the same differences Δf_{jk} and Δg_{jk}. An expression formulated in terms of the edges connected to node j is:

$$\oint_{S_j} (f \cdot \hat{n}_x + g \cdot \hat{n}_y) \partial\Omega_j = \sum_{k=1}^{nedges} (c_{jk}\Delta f_{jk} + l_{jk}\Delta g_{jk}) \tag{3.8}$$

where:

$$\begin{aligned} c_{jk} &= N_j^{R1}\Delta y^{R1} + N_j^{R2}\Delta y^{R2} + N_j^{L1}\Delta y^{L1} + N_j^{L2}\Delta y^{L2} \\ l_{jk} &= N_j^{R1}\Delta x^{R1} + N_j^{R2}\Delta x^{R2} + N_j^{L1}\Delta x^{L1} + N_j^{L2}\Delta x^{L2} \\ \Delta f_{k,j} &= f_k - f_j \\ \Delta g_{k,j} &= g_k - g_j \\ nedges &= \textit{number of edges connected to node } j \end{aligned}$$

(See figure 3.6.) The scheme now has the appearance of a finite difference scheme with differences taken along the edges. The terms c_{jk} and l_{jk} are known as edge coefficients throughout this work. At a boundary face the only consideration is to use a different sampling point for *R2* to close the subvolume as shown in figure 3.7 and ignore face *R1*.

Each edge of the mesh is associated with two subvolumes, one for each node as shown in figure 3.6. Therefore, four edge coefficients are associated with an edge, a pair for each volume. These coefficients are used to define the flux contributions to the nodes. The four coefficients for an edge are labeled c_{jk}, l_{jk}, c_{kj} and l_{kj}. (Note the subscripts correspond to the differences used for the flux, i.e. Δf_{kj} or Δf_{jk}.) For the edge shown in figure 3.6, the flux contribution to nodes j and k are:

$$\begin{aligned} \textit{node } j: &\quad c_{jk}\Delta f_{jk} - l_{jk}\Delta g_{jk} \\ \textit{node } k: &\quad c_{kj}\Delta f_{kj} - l_{kj}\Delta g_{kj} \end{aligned}$$

Once the edge coefficients are known, it is possible to loop over all the edges of the mesh and use the above expressions to calculate the integrals corresponding to all the subvolumes.

In general, $c_{jk} \neq -c_{kj}$ and $l_{jk} \neq -l_{kj}$ because these depend on the shape of the volumes which is arbitrary. But with a particular choice of the control volumes, it is possible to obtain $c_{jk} = -c_{kj}$ and $l_{jk} = -l_{kj}$ for all interior edges. (Note c_{jk} is compared to the negative of c_{kj} since the orientation of the segments for the subvolume surrounding node k is opposite to that for the subvolume of node j.) The symmetry properties of the edge coefficients will be discussed further when dealing with artificial dissipation.

3.1.4 Evaluating $\int \int_{\Omega_j} \frac{\partial u}{\partial t} \partial \Omega_j$

The numerical approximation now has the form:

$$\sum_{j=1}^{nv} \left[\int \int_{\Omega_j} \frac{\partial u}{\partial t} \partial \Omega_j = \sum_{k=1}^{nedges} (c_{jk} \Delta f_{jk} + l_{jk} \Delta g_{jk}) \right] \tag{3.9}$$

This equation states that the convection equation is identically satisfied within each subvolume of the domain. To evaluate the left side of the equation several options are possible.

Piecewise Constant Approximation

In classical finite volume formulations, this term is approximated as:

$$\int \int_{\Omega_j} \frac{\partial u}{\partial t} \partial \Omega_j \approx A_j \frac{du_j}{dt}$$

This is equivalent to evaluating the integral assuming the distribution of u is piecewise constant over each subvolume, Ω_j. The resulting scheme is:

$$A_j \frac{du_j}{dt} = - \sum_{k=1}^{nedges} (c_{jk} \Delta f_{jk} - l_{jk} \Delta g_{jk}) \tag{3.10}$$

for each volume j. This is an acceptable approach for steady state applications, i.e. driving the right side of equation 3.10 to zero. This form of the scheme will be referred to as the lumped mass matrix form, for reasons that will become apparent later.

It is important to look at the discrete approximation's ability to accurately represent unsteady flows. A Fourier analysis on the semi-discrete equation is used to evaluate the accuracy and dispersion characteristics of the spatial approximation. This analysis assumes that the equation is integrated exactly in time. The dispersion measures the dependency of the numerical wave speed (i.e. the wave speed calculated by the numerical

approximation) or frequency on the wave number. Therefore, it is important that a numerical scheme has little numerical dispersion for unsteady applications since unsteady waves must be propagated at the correct speed.

The details of the analysis will not be discussed here, but are presented in Appendix B. The linearized, scalar model convection equation is used to perform the analysis. This equation is obtained by setting $f = au$ and $g = bu$ in equation 3.1.3 where a and b are constant. This gives:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0 \tag{3.11}$$

A mesh composed of equilateral elements shown in figure B.1 is used for the analysis. The exact solution of the equation is:

$$\hat{u}_e = \hat{u}_e(l, m, 0)\exp(\underbrace{-i\frac{1}{\Delta}(a\theta_x + \frac{2}{3}\sqrt{3}b\theta_y)}_{frequency\ \omega_e}t)$$

where:

$$\begin{aligned}
\Delta &= edge\ length\ of\ elements \\
\theta_x &= l\,\Delta x = l\Delta = phase\ angle\ in\ x-direction \\
\theta_y &= m\,\Delta y = \frac{\sqrt{3}}{2}m\Delta = phase\ angle\ in\ y-direction
\end{aligned}$$

The solution of equation 3.11 using the finite volume scheme of equation 3.10 is:

$$\hat{u}_e = \hat{u}_e(l, m, 0)\exp(\underbrace{-i\frac{2}{3\Delta}\omega(\theta_x, \theta_y)}_{frequency\ \omega_n}t)$$

where:

$$\omega(\theta_x, \theta_y) = a(\sin(\theta_x) + \sin(\frac{\theta_x}{2})\cos(\theta_y)) + \sqrt{3}b\sin(\theta_y)\cos(\frac{\theta_x}{2})$$

The flux faces of the volumes are defined using the centroids of the elements and the midpoints of the edges. (Note, *volumes* and *subvolumes* will be used interchangeably in the rest of this work.) The above equations for the exact and numerical solutions show a difference between the exact and numerical frequencies. A comparison of the numerical scheme's frequency versus the exact frequency for varying phase angles, θ_x and θ_y, is presented in figures 3.8 and 3.9. The solid lines correspond to the exact frequency and the dotted lines to the numerical. Figure 3.8 corresponds to θ_x varying from 0 to π and $\theta_y = 0$. Figure 3.9 corresponds to $\theta_x = 0$ and θ_y varying from 0 to π. Figure 3.10 shows the difference, $\omega_e - \omega_n$, versus both phase angles.

The scheme poorly matches the frequency for waves where $\theta_x > 0.5$ and $\theta_y > 0.3$. This corresponds to a wave number of $l < \frac{1}{2\Delta}$ and $m < \frac{.6}{\sqrt{3}\Delta}$ respectively. The desired range

of phase angles is at least $0 < \theta_x, \theta_y < \frac{\pi}{2}$. Anything above this range is considered a high frequency wave that is attenuated by the addition of artificial dissipation in the numerical scheme.

This is a limitation for time accurate results since the scheme does not correctly replicate wave speeds in the desired range. The resolvable wave number on a fixed mesh of spacing, Δ, can never exceed $l_{max} = \frac{\pi}{\Delta}$ [9]. The value $\theta_x = .5$ corresponds to a wave number $l = \frac{1}{2\Delta}$. Comparing this wave number to the maximum resolvable wave number for a fixed mesh shows:

$$\frac{l}{l_{max}} = \frac{1}{2\pi} \approx \frac{1}{6}$$

This means that only waves with a wave length six times larger than the maximum resolvable wave length are propagated with the desired wave speed. The evaluation is even worse in the y-direction. To improve the dispersion characteristics of the scheme, the left side of equation 3.9 is re-evaluated by using a piecewise linear approximation instead of a piecewise constant.

Piecewise Linear Approximation

The integral on the left side of equation 3.9 can be evaluated using a piecewise linear approximation. This is the same approximation used for the evaluation of the fluxes f and g, giving:

$$\int\int_{\Omega} \frac{\partial u}{\partial t} \partial\Omega = \sum_{j=1}^{nv} \int\int_{\Omega_j} (\sum_{i=1}^{np} \frac{du_i}{dt} N_i) \partial\Omega_j \tag{3.12}$$

where N_i are the hat functions defined earlier. This yields:

$$\sum_{j=1}^{nv} \int\int_{\Omega_j} \sum_{i=1}^{np} \frac{du_i}{dt} N_i \partial\Omega_j = \bar{M} \frac{d\vec{u}}{dt}$$

where

$$\vec{u} = [u_1, u_2, \ldots, u_{np}]^T$$

Using this formulation, the equations corresponding to neighboring volumes are coupled through the matrix $\bar{M}$ (see Appendix A for the form of the matrix $\bar{M}$) which will be termed the *finite volume mass matrix*. The lumped form of this matrix (i.e. summing the entries on each row, placing them on the diagonal and zeroing all off-diagonal terms) recovers the piecewise constant approximation described earlier. The numerical approximation becomes:

$$\bar{M}_{ij} \frac{du_j}{dt} = - \sum_{k=1}^{nedges} (c_{jk} \Delta f_{jk} - l_{jk} \Delta g_{jk}) \qquad \begin{array}{l} i = 1, \ldots, np \\ j = 1, \ldots, nv \end{array}$$

(Note $np = nv$ since each volume contains a node.) The Fourier analysis for this approximation is found in Appendix B. The right side of the equation is evaluated as before. The numerical frequency, ω_n, is:

$$\omega_n = \frac{24[\sin(\theta_x) + \sin(\frac{1}{2}\theta_x)\cos(\theta_y) + \sqrt{3}\sin(\theta_y)\cos(\frac{1}{2}\theta_x)]}{\alpha_1 + \alpha_2(2\cos(\theta_x) + 4cos(\frac{1}{2}\theta_x)\cos(\theta_y))} \tag{3.13}$$

where:

$$\alpha_1 = 22\ ,\ \alpha_2 = 6 - \frac{\alpha_1}{6}$$

The comparisons with the exact frequency are shown in figures 3.11, 3.12 and 3.13. This formulation exhibits much better dispersion characteristics than the original piecewise constant integration. Waves corresponding to phase angles of $\theta_x < 1.$ and $\theta_y < 0.6$ are accurately propagated. This is twice the range of frequencies over the piecewise constant formulation so that one half the number of cells of the mesh are needed to achieve the same dispersion characteristics. This is a good improvement, but still does not achieve the goal of $\theta_x < 1.5$. A way of further improving the dispersion characteristics is to perform the integral using a combination of piecewise constant and piecewise linear functions to achieve the finite element mass matrix used in finite element methods [25].

Finite Element Mass Matrix

Using a combination of piecewise constant and piecewise linear functions to integrate equation 3.12 the function N_i becomes:

$$N(x,y)_i = (1-\beta)\,\eta(x,y)_i^c + \beta\eta(x,y)_i^l$$

where:

$$\begin{aligned} \eta(x,y)_i^c &\equiv \textit{piecewise constant function} \\ \eta(x,y)_i^l &\equiv \textit{piecewise linear function} \\ \beta &\equiv \textit{constant} \end{aligned}$$

If $\beta = 0$, the approximation is piecewise constant and if $\beta = 1$, the approximation is piecewise linear. By placing the points defining the subvolumes at the centroids of the elements and the midpoints of the edges, it is shown in Appendix A that by choosing $\beta = \frac{9}{7}$, the matrix $\bar{M}$ becomes identical to the finite element mass matrix [25]. Performing a Fourier analysis using this matrix yields equation 3.13 for the numerical frequency with $\alpha_1 = 18$. The results are shown in figures 3.14, 3.15 and 3.16. Note the marked improvement in the dispersion characteristics of the scheme as compared with the linear and lumped schemes. Wave numbers in the range $0 < \theta_x < 1.5$ and $0 < \theta_y < 1.25$ are captured with the correct wave speed using this integration. This is in the range desired and is an acceptable scheme for reproducing unsteady, inviscid, fluid flows. In Chapter 4 a test demonstrates the behavior of the three different schemes presented and comparisons are drawn to this analysis.

3.2 Artificial Dissipation

Artificial dissipation is added to stabilize the numerical scheme in the vicinity of discontinuous flow structures (e.g. shocks) and for high frequency sawtooth waves. The dissipation is based on the one dimensional TVD property of the convection equation [10]. This concept is extended to multi-dimensions based on the work of Jameson [11, 12].

3.2.1 Background

The total variation for the one dimensional, scalar convection equation

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0$$

is:

$$TV(u) = \int_{-\infty}^{\infty} \left| \frac{\partial u}{\partial x} \right| dx \tag{3.14}$$

The total variation can only decrease with time [14].

A numerical scheme for the one dimensional convection equation written as:

$$\frac{du_j}{dt} = c_{j+\frac{1}{2}} (u_{j+1} - u_j) + c_{j-\frac{1}{2}} (u_{j-1} - u_j) \tag{3.15}$$

is TVD if $c_{j+\frac{1}{2}},\ c_{j-\frac{1}{2}} \geq 0$ [11]. This is the *positivity condition* and ensures that a local maximum at u_j does not increase with time since $\frac{du_j}{dt} \leq 0$ and a local minimum does not decrease with time since $\frac{du_j}{dt} \geq 0$ if this condition is met. Since most schemes do not satisfy this condition, the fluxes on the right side of equation 3.15 are modified by the addition of an artificial dissipation which is determined to satisfy this requirement. The numerical scheme in equation 3.15 can be rewritten as [11]:

$$\frac{du_j}{dt} + f^*_{j+\frac{1}{2}} - f^*_{j-\frac{1}{2}} = 0 \tag{3.16}$$

where:

$$\begin{aligned}
f^*_{j+\frac{1}{2}} &= f_j - \left(\alpha_{j+\frac{1}{2}} - \frac{1}{2} a_{j+\frac{1}{2}} \right) (u_{j+1} - u_j) \\
f^*_{j-\frac{1}{2}} &= f_j - \left(\alpha_{j-\frac{1}{2}} + \frac{1}{2} a_{j-\frac{1}{2}} \right) (u_j - u_{j-1}) \\
a_{j+\frac{1}{2}} &= \begin{cases} \frac{f_{j+1} - f_j}{u_{j+1} - u_j} & if\ u_{j+1} \neq u_j \\ \frac{\partial f}{\partial u} & if\ u_{j+1} = u_j \end{cases} \\
a_{j-\frac{1}{2}} &= \begin{cases} \frac{f_j - f_{j-1}}{u_j - u_{j-1}} & if\ u_{j-1} \neq u_j \\ \frac{\partial f}{\partial u} & if\ u_{j-1} = u_j \end{cases}
\end{aligned}$$

The term $a_{j+\frac{1}{2}}$ is the approximation to the convection speed $\frac{\partial f}{\partial u}$; $\alpha_{j-\frac{1}{2}}$ and $\alpha_{j+\frac{1}{2}}$ are the added numerical dissipation coefficients used to assure the positivity condition is met. These coefficients must satisfy:

$$\alpha_{j+\frac{1}{2}} \geq \frac{1}{2}\left|a_{j+\frac{1}{2}}\right|, \; \alpha_{j-\frac{1}{2}} \geq \frac{1}{2}\left|a_{j-\frac{1}{2}}\right| \tag{3.17}$$

Therefore, the modified flux $f^*_{j+\frac{1}{2}}$ becomes:

$$f^*_{j+\frac{1}{2}} = f_j + \frac{1}{2}\left(a_{j+\frac{1}{2}} - \left|a_{j+\frac{1}{2}}\right|\right)\Delta u_{j+\frac{1}{2}}$$

and similarly for $f^*_{j-\frac{1}{2}}$. This flux is equivalent to a first order upwind approximation [10, pp441]. Combining both fluxes in equation 3.16 yields:

$$\frac{du_j}{dt} + \underbrace{\frac{1}{2}\left(a_{j+\frac{1}{2}} - \left|a_{j+\frac{1}{2}}\right|\right)(u_{j+1} - u_j) + \frac{1}{2}\left(a_{j-\frac{1}{2}} + \left|a_{j-\frac{1}{2}}\right|\right)(u_j - u_{j-1})}_{\frac{\partial f}{\partial x}} = 0$$

From the above formulation:

$$\frac{\partial f}{\partial x} = \begin{cases} a_{j+\frac{1}{2}}(u_{j+1} - u_j) & if\ a_{j+\frac{1}{2}} < 0 \\ a_{j-\frac{1}{2}}(u_{j-1} - u_j) & if\ a_{j+\frac{1}{2}} > 0 \end{cases}$$

This shows the numerical approximation of $\frac{\partial f}{\partial x}$ is indeed a first order upwind difference stencil depending on the direction of the propagation speed $a_{j+\frac{1}{2}}$.

This formulation is too diffusive and is modified by using a flux limiting procedure [11, 12]. This modifies the added diffusive flux to a less diffusive fourth difference stencil in smooth regions of the flow and to the first order upwind stencil at extremum. The limited diffusive flux is:

$$d_{j+\frac{1}{2}} = \alpha_{j+\frac{1}{2}}\left[\Delta u_{j+\frac{1}{2}} - L\left(\Delta u_{j+\frac{3}{2}}, \Delta u_{j-\frac{1}{2}}\right)\right]$$

where $\alpha_{j+\frac{1}{2}}$ is the same as equation 3.17 and the limiter $L(u,v)$ is a limited average satisfying the following properties:

$$\begin{aligned}
&\textit{Property } 1: \quad L(u,u) = u \\
&\textit{Property } 2: \quad L(\beta u, \beta v) = \beta L(u,v) \\
&\textit{Property } 3: \quad L(u,v) = L(v,u) \\
&\textit{Property } 4: \quad L(u,v) = \begin{cases} 0 & if\ sign(u) \neq sign(v) \\ sign(u)\,|L(u,v)| & if\ sign(u) = sign(v) \end{cases}
\end{aligned}$$

If the limiter is a strict average, i.e. $L(u,v) = \frac{1}{2}(u+v)$ when sign (u) = sign (v), substituting into equation 3.16 gives:

$$\begin{aligned}
d_{j+\frac{1}{2}} - d_{j-\frac{1}{2}} &= -\frac{1}{2}\alpha\underbrace{(u_{j+2} - 4u_{j+1} + 6u_j - 4u_{j-1} + u_{j-2})}_{\frac{d^4u}{dx^4}\Delta x^4} \\
&= -\frac{1}{2}\alpha
\end{aligned}$$

assuming $\alpha_{j+\frac{1}{2}} = \alpha_{j-\frac{1}{2}} = \alpha$.

Figure 3.17 shows three possible configurations for the differences, $\Delta u_{j-\frac{1}{2}}$, $\Delta u_{j+\frac{1}{2}}$ and $\Delta u_{j+\frac{3}{2}}$. The first is an extremum at node j. The consecutive gradients show that $sign\left(\Delta u_{j+\frac{3}{2}}\right) \neq sign\left(\Delta u_{j-\frac{1}{2}}\right)$. Therefore, the limiter switches to the first order upwind scheme since $L(u,v) = 0$. The second shows a smoothly varying region of the flow where $sign\left(\Delta u_{j+\frac{3}{2}}\right) = sign\left(\Delta u_{j-\frac{1}{2}}\right)$. Therefore, the diffusive flux is limited in this region. The third shows a sawtooth wave with $sign\left(\Delta u_{j+\frac{3}{2}}\right) = sign\left(\Delta u_{j-\frac{1}{2}}\right)$. In this case, the diffusive flux is limited. It is important to note that in this case the added limited diffusive flux reinforces the positivity condition. This is shown in [11] given the above properties for the limiter. Various alternative forms of limiters satisfying the above properties will be discussed later.

3.2.2 Extension to the Generalized Finite Volume Scheme

Jameson [11, 12] outlines the extension of this technique to the solution of multi-dimensional, scalar convection equations. The generalized, lumped mass matrix, finite volume approximation to the two dimensional, model convection equation developed earlier is:

$$A_j \frac{du_j}{dt} + \sum_k (c_{jk}\Delta f_{jk} - l_{jk}\Delta g_{jk}) = 0$$

where the summation k is over the edges attached to node j and A_j the area of the control volume. The above equation is rewritten as:

$$A_j \frac{du_j}{dt} + \sum_k a_{jk}\Delta u_{jk} = 0$$

where:

$$a_{jk} = \begin{cases} \frac{c_{jk}\Delta f_{jk} - l_{jk}\Delta g_{jk}}{\Delta u_{jk}} & if\ u_j \neq u_k \\ c_{jk}\frac{\partial f}{\partial u} - l_{jk}\frac{\partial g}{\partial u} & if\ u_j = u_k \end{cases}$$

A scheme that satisfies the positivity condition is formulated by adding a dissipation term to the right side of the above equation of the form:

$$dissipation = \sum_k \alpha_{jk}\Delta u_{jk}$$

The scheme becomes:

$$A_j \frac{du_j}{dt} + \sum_k a_{jk}\Delta u_{jk} = \sum_k \alpha_{jk}\Delta u_{jk} \tag{3.18}$$

The positivity condition is satisfied if [11]:

$$\alpha_{jk} \geq |a_{jk}|$$

This formulation guarantees the maximum principle, i.e. $\frac{du_j}{dt} < 0$ if u_j is a maximum and $\frac{du_j}{dt} > 0$ if u_j is a minimum [11].

Again, this dissipation term is too diffusive in smooth regions of the flow. Therefore, an antidiffusive flux is constructed and applied to each edge in regions where the flow is smoothly varying. This flux is applied by using the limiting procedure described earlier. The total diffusive flux is [11]:

$$dissipation = \sum_k |a_{jk}| \left(\Delta u_{jk} - L\left(\Delta u^+_{jk}, \Delta u^-_{jk} \right) \right) \tag{3.19}$$

The differences, Δu^+_{jk} and Δu^-_{jk}, associated with edge jk are now evaluated based on the solution at the neighboring elements. This is shown in figure 3.18 with edge jk and the associated elements connected to the nodes defining the edge. The elements that edge jk points into and out of are shaded in the first figure. These gradients are constructed by extrapolating edge jk forward and backward and using the nodal values of the elements into and out of which this edge points. This constructs a locally one dimensional gradient at nodes j and k in the direction from node j to k.

The positive increment Δ_{jk} points into the element defined by nodes k, 1 and 2. The values of u at these nodes are used to construct Δu^+_{jk}. A second order Taylor series approximation about node k gives:

$$\begin{aligned} u_1 &= u_k + \nabla u_k \cdot \vec{\Delta_{k1}} \\ u_2 &= u_k + \nabla u_k \cdot \vec{\Delta_{k2}} \end{aligned}$$

where ∇u_k is the gradient at node k; $\vec{\Delta_{k1}}$ and $\vec{\Delta_{k2}}$ are the vectors from node k to nodes 1 and 2 respectively. These are:

$$\begin{aligned} \nabla u_k &= \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} \\ \Delta_{k1} &= (x_k - x_1,\ y_k - y_1) \\ \Delta_{k2} &= (x_k - x_2,\ y_k - y_2) \end{aligned}$$

This gives a system of two equations to solve for the gradient components:

$$\begin{bmatrix} u_1 - u_k \\ u_2 - u_k \end{bmatrix} = \underbrace{\begin{bmatrix} x_k - x_1 & y_k - y_1 \\ x_k - x_2 & y_k - y_2 \end{bmatrix}}_{\bar{A}} \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix}$$

The matrix $\bar{A}$ is inverted to determine the gradients, giving:

$$\begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} = \frac{1}{|\bar{A}|} \begin{bmatrix} y_k - x_2 & -y_k + y_1 \\ -x_k + x_2 & x_k - y_1 \end{bmatrix} \begin{bmatrix} u_1 - u_k \\ u_2 - u_k \end{bmatrix}$$

Note the determinant $|\bar{A}|$ is twice the negative area of the triangle k, 1, 2. The increment Δu_{jk}^{+} is:

$$\Delta u_{jk}^{+} = \vec{\Delta_{kx}} \cdot \nabla u_k$$

This gives:

$$\begin{aligned}
\Delta u_{jk}^{+} &= \frac{1}{|\bar{A}|} \underbrace{[x_x - x_k,\ y_x - y_k] \begin{bmatrix} y_k - x_2 & y_1 - y_k \\ x_2 - x_k & x_k - y_1 \end{bmatrix}} \begin{bmatrix} u_1 - u_k \\ u_2 - u_k \end{bmatrix} \\
&= \frac{1}{|\bar{A}|} \begin{bmatrix} -2(area\ of\ kx1) & -2(area\ of\ kx2) \end{bmatrix} \begin{bmatrix} u_1 - u_k \\ u_2 - u_k \end{bmatrix}
\end{aligned}$$

Therefore, Δu_{jk}^{+} is written as:

$$\Delta u_{jk}^{+} = \epsilon_{k1} (u_1 - u_k) + \epsilon_{k2} (u_2 - u_k) \tag{3.20}$$

where:

$$\epsilon_{k1} = \frac{area\ of\ element\ kx1}{area\ of\ element\ k12}\ ,\ \epsilon_{k2} = \frac{area\ of\ element\ kx2}{area\ of\ element\ k12}$$

The coefficients ϵ_{k1} and ϵ_{k2} are both positive. The same procedure is applied to determine Δu_{jk}^{-}, giving:

$$\Delta u_{jk}^{+} = \epsilon_{k3} (u_j - u_3) + \epsilon_{k4} (u_j - u_4) \tag{3.21}$$

where:

$$\epsilon_{k3} = \frac{area\ of\ element\ kx3}{area\ of\ element\ k34}\ ,\ \epsilon_{k4} = \frac{area\ of\ element\ kx4}{area\ of\ element\ k34}$$

Since the coefficients ϵ_{k1}, ϵ_{k2}, ϵ_{k3} and ϵ_{k4} are non-negative, the limited diffusive flux in equation 3.19 reinforces the positivity condition for each edge attached to node j [11]. This scheme is known as the SLIP (Symmetric Limited Positive) scheme [11, 12].

3.2.3 Limiters and Soft Limiting

Many types of limiters satisfy the four properties described earlier. The Minmod and Van Leer limiters are two types and are defined as:

$$\begin{aligned}
Minmod: \quad L(u,v) &= S(u,v)\, minimum\,(|u|, |v|) \\
Van\ Leer: \quad L(u,v) &= S(u,v)\, \frac{2\,|u|\,|v|}{|u| + |v|}
\end{aligned}$$

where the function $S(u,v)$ is:

$$S(u,v) = \frac{1}{2}\,(sign(u) + sign(v))$$

Therefore,

$$S(u,v) = \begin{cases} 1 & if\ sign(u) = sign(v) = 1 \\ 0 & if\ sign(u) \neq sign(v) \\ -1 & if\ sign(u) = sign(v) = -1 \end{cases}$$

which guarantees the limiters satisfy Property 4.

A problem with these limiters is they do not distinguish between a smooth or a sharp extrema. They become zero at smooth extrema in the flow, decreasing the accuracy of the solution in these regions by adding too much dissipation. Therefore, a limiter not active at smooth extrema is constructed to maintain a low amount of dissipation in these regions. (Not active means $L(u,v) \neq 0$.) These limiters are termed soft, or ELED (Essentially Local Extremum Diminishing), limiters [12].

The Minmod and Van Leer limiters are rewritten as [12]:

$$L(u,v) = \frac{1}{2} D(u,v)(u+v)$$

where:

$$D(u,v) = 1 - \left| \frac{u-v}{|u|+|v|} \right|^q$$

Setting $q = 1$ gives the Minmod limiter and $q = 2$ produces the Van Leer limiter [12]. The function $D(u,v)$ is modified to allow smooth extrema (i.e $D(u,v) \neq 0$ even if $sign(u) \neq sign(v)$). This is accomplished by the modification:

$$D^*(u,v) = 1 - \left| \frac{u-v}{max\left((|u|+|v|), \epsilon \Delta x^r\right)} \right|^q$$

which sets a lower bound on the sum of the gradients, i.e. $|u| + |v| > \epsilon \Delta x^r$. If this inequality is satisfied, the modified $D^*(u,v)$ is the original function, $D(u,v)$.

For the one dimensional, scalar convection equation, the gradients $\Delta u_{j+\frac{1}{2}}$, $\Delta u_{j+\frac{3}{2}}$ and $\Delta u_{j-\frac{1}{2}}$ are all of the order Δx^2 for a locally smooth extrema. This is seen from a Taylor series expansion about node j and $\frac{du_j}{dx} = 0$ if node j is a local extremum:

$$\begin{aligned}
\Delta u_{j+\frac{1}{2}} &= \Delta x \frac{du_j}{dx} + \frac{\Delta x^2}{2!} \frac{d^2 u_j}{dx^2} + \dots \\
\Delta u_{j+\frac{3}{2}} &= 2\Delta x \frac{du_j}{dx} + \frac{(2\Delta x)^2}{2!} \frac{d^2 u_j}{dx^2} + \dots \\
\Delta u_{j-\frac{1}{2}} &= -\Delta x \frac{du_j}{dx} + \frac{\Delta x^2}{2!} \frac{d^2 u_j}{dx^2} + \dots
\end{aligned}$$

Choosing $r = \frac{3}{2}$ and $q > 2$ ensures that the limiter provides an approximation which is of the order Δx^2 [12]. If $|u| + |v| < \epsilon \Delta x^r$, the term in the absolute value of $D^*(u,v)$ is of the order Δx^{2-r}. With $r = \frac{3}{2}$, the term in the absolute value is of order $\Delta x^{\frac{1}{2}}$. The function $D^*(u,v)$ is rewritten as:

$$D^*(u,v) = 1 - \beta(u,v)$$

where $\beta(u,v)$ is of order $\Delta x^{\frac{q}{2}}$. Substituting the expression for $L(u,v)$ using $D^*(u,v)$ into equation 3.19 gives:

$$d_{j+\frac{1}{2}} = \alpha_{j+\frac{1}{2}} \left(\Delta u_{j+\frac{1}{2}} - \frac{1}{2}(1 - \beta(\Delta u_{j+\frac{3}{2}}, \Delta u_{j-\frac{1}{2}})) \left(\Delta u_{j+\frac{3}{2}} + \Delta u_{j-\frac{1}{2}} \right) \right)$$

The limited dissipation contains a term of order Δx^3 and a term which is of order $\Delta x^{1+\frac{q}{2}}$ [11]. If $q \geq 2$ the dissipation is guaranteed to be of order Δx^2.

If u_j is a maximum, the following holds for the scalar convection equation [12]:

$$\frac{du_j}{dt} \leq \underbrace{\frac{1}{2}\left(\alpha_{j+\frac{1}{2}} + \alpha_{j-\frac{1}{2}}\right)\epsilon \Delta x^r}_{\Gamma}$$

This establishes an upper bound for $\frac{du_j}{dt}$. Likewise, if u_j is a local minimum, $\frac{du_j}{dt} \geq -\Gamma$. Therefore, the scheme is ELED since local extremum are allowed to increase or decrease. Note as $\Delta x \rightarrow 0$ then $\Gamma \rightarrow 0$ if $r > 0$.

Choice of ϵ

The choice of ϵ in the soft limiter is somehow arbitrary. It is appropriate to make this term an upper bound of a second derivative since the differences at a smooth extrema are proportional to a second derivative. A second derivative at an extrema is approximated by:

$$\frac{d^2u}{dx^2} \approx \frac{1}{2}\frac{\Delta u_{j+\frac{3}{2}} - \Delta u_{j+\frac{1}{2}}}{\Delta x^2} + O(\Delta x^2)$$

Specifying the numerator gives an upper bound for each edge of a fixed mesh. This gives:

$$\epsilon = \frac{1}{2}\frac{\Psi}{\Delta x^2} = \left.\frac{d^2u}{dx^2}\right|_{upper\ bound}$$

where Ψ is a specified quantity. Substituting this into the expression for $D^*(u,v)$ results in a term which is of order Δx^{r-2}.

Reducing the Limited Dissipation

The coefficient multiplying the limited dissipation, i.e $|a_{jk}|$, is too large for smooth regions of the flow when the limiter is not active. This formulation provides more diffusion than necessary. Therefore, the soft limiting procedure is modified as follows:

$$dissipation = (1 - Vc_4 \cdot D^*(\Delta u^+_{jk}, \Delta u^-_{jk}))|a_{jk}|\left(\Delta u_{jk} - \frac{1}{2}D^*(\Delta u^+_{jk}, \Delta u^-_{jk})(\Delta u^-_{jk} + \Delta u^-_{jk})\right)$$

where $0 < Vc_4 < 1$. Choosing Vc_4 reduces the coefficient $|a_{jk}|$ in smooth regions of the flow where $D^*(u,v) = 1$. Therefore, the dissipation is:

$$dissipation = (1 - Vc_4)|a_{jk}|\left(\Delta u_{jk} - \frac{1}{2}\left(\Delta u^+_{jk} + \Delta u^-_{jk}\right)\right)$$

The term $|a_{jk}|$ is multiplied by $(1 - Vc_4)$ which is less than or equal to one. If j is an extremum, $D^*(u,v) = 0$ and the dissipation becomes the lower order dissipation.

3.2.4 Unsymmetric Edge Coefficients

The finite volume scheme developed in this chapter is conservative in the sense that when summing up the equation for all the control volumes, the contributions corresponding to interior boundaries cancel out identically. This is illustrated in figure 3.2. The addition of artificial dissipation to stabilize the scheme must guarantee that the scheme remains conservative to ensure proper shock capturing properties [10]. The dissipation is based on edges of the mesh and the corresponding edge coefficients, c_{jk} and l_{jk}. To ensure the added dissipation is conservative, the dissipation added to the nodes of an edge must be equal in magnitude and opposite in sign. This guarantees the cancellation of the total dissipation for all edges, therefore ensuring the scheme remains conservative.

The edge coefficients depend on the control volumes used for the nodes defining an edge. There are a pair of coefficients for each node of the edge. If an edge contains two pairs of coefficients not equal in magnitude and opposite in sign, the dissipation calculated for the nodes of the edge will not cancel and the scheme will not be conservative.

It is possible to define a symmetric pair of coefficients, q_{jk} and r_{jk}, and an antisymmetric pair of coefficients, q'_{jk} and r'_{jk}, for an edge. These coefficients satisfy:

$$q_{jk} = -q_{kj} \ , \ r_{jk} = -r_{kj} \ , \ q'_{jk} = q'_{kj} \ , \ r'_{jk} = r'_{kj}$$

They are defined as:

$$\begin{aligned}
q_{jk} &= \frac{1}{2}(c_{jk} - c_{kj}) \\
r_{jk} &= \frac{1}{2}(l_{jk} - l_{kj}) \\
q'_{jk} &= \frac{1}{2}(c_{jk} + c_{kj}) \\
r'_{jk} &= \frac{1}{2}(l_{jk} + l_{kj}) \\
q_{kj} &= \frac{1}{2}(c_{kj} - c_{jk}) \\
r_{kj} &= \frac{1}{2}(l_{kj} - l_{jk})
\end{aligned}$$

The fluxes for nodes j and k of the edge are rewritten using these new coefficients, yielding:

$$\begin{aligned}
\textit{node } j: \quad c_{jk}\Delta f_{jk} - l_{jk}\Delta g_{jk} &= (q_{jk}\Delta f_{jk} - r_{jk}\Delta g_{jk}) + \left(q'_{jk}\Delta f_{jk} - r'_{jk}\Delta g_{jk}\right) \\
\textit{node } k: \quad c_{kj}\Delta f_{kj} - l_{kj}\Delta g_{kj} &= (q_{kj}\Delta f_{kj} - r_{kj}\Delta g_{kj}) + \left(q'_{kj}\Delta f_{kj} - r'_{kj}\Delta g_{kj}\right)
\end{aligned}$$

The first set of terms in parenthesis for nodes j and k are symmetric, satisfying:

$$(q_{jk}\Delta f_{jk} - r_{jk}\Delta g_{jk}) = (q_{kj}\Delta f_{kj} - r_{kj}\Delta g_{kj})$$

(Recall that $\Delta f_{jk} = -\Delta f_{kj}$ and likewise $\Delta g_{jk} = -\Delta g_{kj}$.) The second set of terms in parenthesis are antisymmetric, satisfying:

$$\left(q'_{jk}\Delta f_{jk} - r'_{jk}\Delta g_{jk}\right) = -\left(q'_{kj}\Delta f_{kj} - r'_{kj}\Delta g_{kj}\right)$$

This antisymmetric term is a diffusion-like term resulting from the choice of the control volumes used for nodes j and k. Therefore, this term must be considered in the positivity condition.

The combined flux and dissipation contributions to the nodes of the edge are:

$$\begin{aligned} \textit{node } j: &\quad (a_{jk} + \beta_{jk} - \alpha_{jk})\,\Delta u_{jk} \\ \textit{node } k: &\quad (-a_{jk} + \beta_{jk} - \alpha_{jk})\,\Delta u_{kj} \end{aligned}$$

where:

$$a_{jk} = \begin{cases} \frac{q_{jk}\Delta f_{jk} - r_{jk}\Delta g_{jk}}{\Delta u_{jk}} & if\ u_j \neq u_k \\ q_{jk}\frac{\partial f}{\partial u} - r_{jk}\frac{\partial g}{\partial u} & if\ u_j = u_k \end{cases}$$

$$\beta_{jk} = \begin{cases} \frac{q'_{jk}\Delta f_{jk} - r'_{jk}\Delta g_{jk}}{\Delta u_{jk}} & if\ u_j \neq u_k \\ q'_{jk}\frac{\partial f}{\partial u} - r'_{jk}\frac{\partial g}{\partial u} & if\ u_j = u_k \end{cases}$$

and α_{jk} is the added dissipation in equation 3.18. If node j is a maximum, the positivity condition requires:

$$a_{jk} + \beta_{jk} - \alpha_{jk} \leq 0$$

This is satisfied if:

$$\alpha_{jk} \geq |a_{jk}| + \beta_{jk}$$

Likewise, for a maximum at node k, the positivity condition requires:

$$-a_{jk} + \beta_{jk} - \alpha_{jk} \leq 0$$

This is satisfied if:

$$\alpha_{jk} \geq |-a_{jk}| + \beta_{jk}$$

Therefore, the artificial dissipation must include the antisymmetric β_{jk} term to ensure the positivity condition is satisfied. Equation 3.19 for this edge becomes:

$$d_{jk} = (|a_{jk}| + \beta_{jk})\left(\Delta u_{jk} - L(\Delta u^+_{jk}, \Delta u^-_{jk})\right)$$

It is noted that if the control volumes are defined by the centroids of the elements and the midpoints of the edges, the edge coefficients will be symmetric in the interior of the domain and $\beta_{jk} = 0$. The only antisymmetric terms would lie on the boundaries of the domain.

3.2.5 Comments

The formulation of the artificial dissipation is based on the lumped mass matrix version of the scheme. Therefore, using this dissipation for the matrix versions does not guarantee the scheme will satisfy the positivity condition for each edge. In practice, however, the use of the consistent mass matrix yields oscillation-free results and has the added advantage of much improved dispersion characteristics.

3.3 Extension to the Euler Equations

Extending the above derivation to the compressible Euler equations is straightforward. The Euler equations governing two dimensional, inviscid, compressible flow are stated in integral, conservative, vector form as:

$$\int\int_{\Omega} \frac{\partial \vec{U}}{\partial t} \partial\Omega + \oint_S (\vec{F} \cdot n_x + \vec{G} \cdot n_y) dS = 0 \tag{3.22}$$

where $\vec{U}, \vec{F}$ and $\vec{G}$ are:

$$\vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad \vec{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{bmatrix} \quad \vec{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{bmatrix}$$

ρ=density, p=pressure, [u,v]= cartesian velocity components, E=total specific energy, H=$E + \frac{p}{\rho}$ [1, 15]. The equation of state is needed to close the system. It is written as:

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2))$$

The finite volume scheme developed in the previous section applies to this equation set by replacing f, g and u with vector form $\vec{F}$, $\vec{G}$ and $\vec{U}$ respectively. This gives:

$$\bar{M}_{ij} \frac{d\vec{U}_j}{dt} = - \sum_{k=1}^{nedges} \left(c_{jk} \Delta \vec{F}_{jk} - l_{jk} \Delta \vec{G}_{jk} \right) \tag{3.23}$$

The lumped finite volume scheme extension to the two dimensional Euler equation is:

$$A_j \frac{d\vec{U}_j}{dt} + \sum_k \left(c_{jk} \Delta \vec{F}_{jk} - l_{jk} \Delta \vec{G}_{jk} \right) = 0$$

The term in the summation can be viewed as the dot product of two vectors:

$$c_{jk} \Delta \vec{F}_{jk} - l_{jk} \Delta \vec{G}_{jk} = [c_{jk}, -l_{jk}] \cdot \begin{bmatrix} \Delta \vec{F}_{jk} \\ \Delta \vec{G}_{jk} \end{bmatrix}$$

A SLIP scheme for the Euler equation is constructed by defining a matrix, $\bar{A}$, that is a function of the state variables, $\vec{U}_j$ and $\vec{U}_k$, for an edge. This matrix satisfies the following properties:

$$\begin{aligned} \bar{A}\left(\vec{U}_j, \vec{U}_k\right)\left(\vec{U}_k - \vec{U}_j\right) &= c_{jk}\Delta\vec{F}_{jk} - l_{jk}\Delta\vec{G}_{jk} \\ \bar{A}\left(\vec{U}, \vec{U}\right) &= \bar{A}\left(\vec{U}\right) \end{aligned}$$

Roe's construction [19] satisfies these conditions and is used to formulate the artificial dissipation [11, 12]. The dissipation has the form:

$$dissipation = \sum_k \left|\bar{A}\left(\vec{U}_j, \vec{U}_k\right)\right|\left(\vec{U}_k - \vec{U}_j\right)$$

where the absolute value of the matrix is determined by decomposing the matrix $\bar{A}$ using the matrix of eigenvectors, $\bar{T}$, and diagonal matrix of eigenvalues, $\bar{\Lambda}$:

$$\bar{A} = \bar{T}\bar{\Lambda}\bar{T}^{-1}$$

The absolute value of $\bar{A}$ is determined using the absolute value of the eigenvalue matrix:

$$|\bar{A}| = \bar{T}\,|\bar{\Lambda}|\,\bar{T}^{-1}$$

This scheme has the form:

$$A_j\frac{d\vec{U}}{dt} + \sum_k \left(c_{jk}\Delta\vec{F}_{jk} - l_{jk}\Delta\vec{G}_{jk} - |\bar{A}|\left(\vec{U}_j, \vec{U}_k\right)\left(\vec{U}_k - \vec{U}_j\right)\right) = 0$$

This is again too diffusive and the use of a limiter is required. The introduction of the limiter yields:

$$dissipation = \sum_k \bar{T}|\bar{\Lambda}| \underbrace{\bar{T}^{-1}\left(\Delta\vec{U}_{jk} - L\left(\Delta\vec{U}^+_{jk}, \Delta\vec{U}^-_{jk}\right)\right)}_{characteristic\ variables}$$

If the edge coefficients are unsymmetric, the extension from the scalar convection equation gives two Roe matrices:

$$\begin{aligned} A\left(\vec{U}_j, \vec{U}_k\right) &= q_{jk}\Delta\vec{F}_{jk} - r_{jk}\Delta\vec{G}_{jk} \\ A'\left(\vec{U}_j, \vec{U}_k\right) &= q'_{jk}\Delta\vec{F}_{jk} - r'_{jk}\Delta\vec{G}_{jk} \end{aligned}$$

The dissipation becomes:

$$dissipation = \sum_k \left(|\bar{A}| + A'\right)\left(\Delta\vec{U}_{jk} - L\left(\Delta\vec{U}^+_{jk}, \Delta\vec{U}^-_{jk}\right)\right)$$

The soft limiter is used along with the procedure that reduces the amount of higher order diffusive flux. Also, the Roe matrix is never calculated identically. Instead a procedure developed by Turkel [22] is used which reduces the matrix multiplication $|\bar{A}|\Delta\vec{U}$ into a process of three vector multiplications and three additions.

Harten's Correction

The dissipation previously developed can introduce non-entropy satisfying solutions. These solutions contain regions in which the entropy discontinuously decreases, violating the second law of thermodynamics. Such regions are known as "expansion" shocks and result from the numerical viscosity $\alpha_{jk} \rightarrow 0$. This occurs when an eigenvalue of the Roe matrix becomes zero. A simple correction developed in [8] is used to ensure that no eigenvalue of the Roe matrix will become zero. This correction is:

$$|\lambda| = \begin{cases} \frac{1}{2}\left(\frac{\lambda^2}{\epsilon} + |\lambda|\right) & if\ |\lambda| < \epsilon \\ |\lambda| & if |\lambda| \geq \epsilon \end{cases}$$

where ϵ is some specified minimum for the absolute value of any of the eigenvalues, λ. (Note that this correction is only applied to the first Roe matrix, $|\bar{A}|$, in the dissipation for the unsymmetric coefficient case.)

3.4 Time Integration

The time integration technique is an explicit two-stage predictor/corrector method. The method is known as the Modified Euler Method and guarantees the scheme remains monotone, i.e. satisfies the positivity condition [20]. It is:

$$\begin{aligned} &predictor: && u_i^* = u_i^n - \tfrac{\Delta t_i}{A_i} Flux_i \\ &corrector: && u_i^{n+1} = \tfrac{1}{2}\left(u_i^n + u_i^* - \tfrac{\Delta t_i}{A_i} Flux_i^*\right) \end{aligned}$$

This integration is second order accurate in time. For the mass matrix version of the scheme, this becomes:

$$\begin{aligned} &predictor: && (u_i^* - u_i^n) = -\Delta t_i \bar{M}_{ij}^{-1} Flux_i \\ &corrector: && (u_i^{n+1} - \tfrac{1}{2}(u_i^n + u_i^*)) = \Delta t_i \bar{M}_{ij}^{-1} Flux_i^* \end{aligned}$$

An inexpensive iterative procedure is used to invert the mass matrix [18, pp49].

3.5 Stability

The time step limitation for the predictor/corrector technique is based on an energy stability analysis performed by Giles [7]. The maximum allowable time step for the lumped finite volume scheme is:

$$\Delta t_i = \frac{\alpha A_i}{\sum_{k=1}^{nedges} \left| [q_{jk} u_i - r_{jk} v_i] + \sqrt{q_{jk}^2 + r_{jk}^2}\, a_i \right|}$$

where a is the speed of sound; u and v are the cartesian velocity components; A is the area of the control volume; and α is a safety factor. The q_{jk} and r_{jk} coefficients are used instead of c_{jk} and l_{jk} since the analysis is based on defining a symmetric approximation to the fluxes, $\frac{\partial \vec{F}}{\partial x}$ and $\frac{\partial \vec{G}}{\partial y}$. The safety factor $\alpha = .8$ is used for the lumped scheme. The finite element and finite volume mass matrix versions use $\alpha = .25$. This proves to be a sufficient time step limitation for the problems presented.

Time accurate calculations are advanced at the minimum local time step over the entire mesh. Steady state applications use local time stepping, i.e. each node is advanced at its own allowable time step to accelerate convergence.

3.6 Conclusions

In this chapter a finite volume scheme for the two dimensional model convection equation is developed. This scheme can be implemented on general, unstructured meshes and uses linear functions from the finite element method to spatially approximate the unknowns. The volumes used for integration are arbitrarily defined by a point within each element and a point that lies on each edge. It is shown that the spatial discretization of the scheme developed may be written in terms of differences along edges of the mesh giving it the appearance of a finite difference scheme.

A Fourier analysis is performed to investigate the scheme's dispersion characteristics. These characteristics are shown to strongly depend on the method of integration used to evaluate $\int\int_\Omega \frac{\partial u}{\partial t}$. A piecewise constant, piecewise linear and a combination of piecewise constant and piecewise linear approximations are chosen as the methods of integration. The piecewise constant method shows the least favorable dispersion characteristics of the three. A piecewise linear integration yields a matrix that couples the nodes connected to a given node. This shows an improvement over the piecewise constant integration. The matrix formed from the piecewise linear approximation is slightly different than the mass matrix used in the finite element method. By restricting the volumes of integration and choosing a combination of piecewise constant and piecewise linear functions to perform the integration, a matrix identical to the finite element mass matrix is determined. This form of integration yields a large improvement over the piecewise constant integration method. Therefore, for unsteady flow calculations this type of evaluation of $\int\int_\Omega \frac{\partial u}{\partial t}$ should be used.

The addition of artifical dissipation to stabilize the scheme is discussed. This dissipation is based on a flux limiting procedure. It is noticed that depending on the control volumes chosen, a certain amount of dissipation is already in the numerical scheme. A method of including this dissipation into the artificial dissipation results in the definition of symmetric and antisymmetric flux terms for each edge. Also, the concept of an ELED scheme is incorporated into the limiting procedure to reduce the amount of dissipation applied at smooth extrema. Finally, the finite volume scheme with artifical dissipation is extended to the compressible Euler equations.

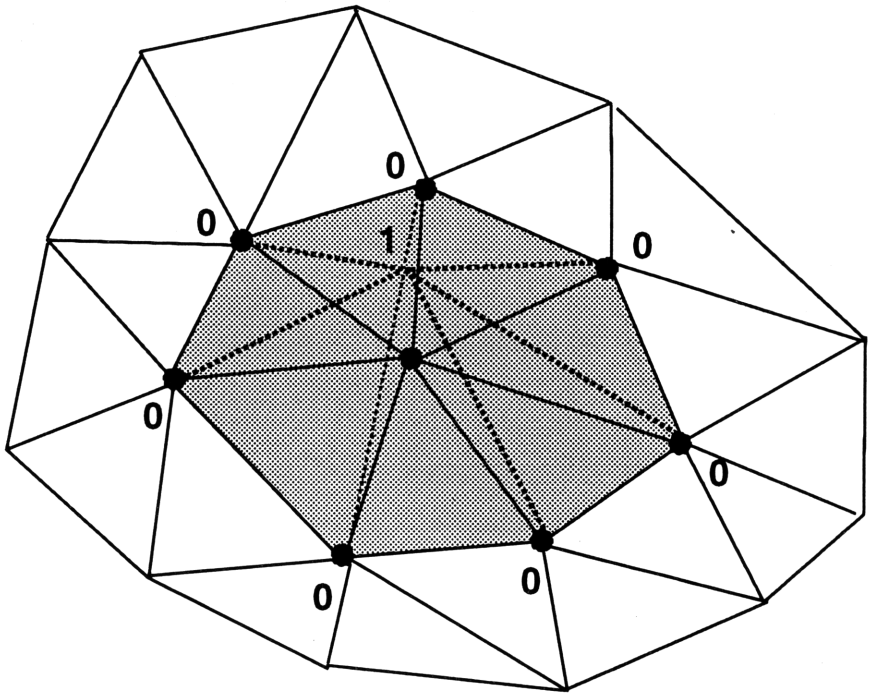


Figure 3.1: Mesh Patch

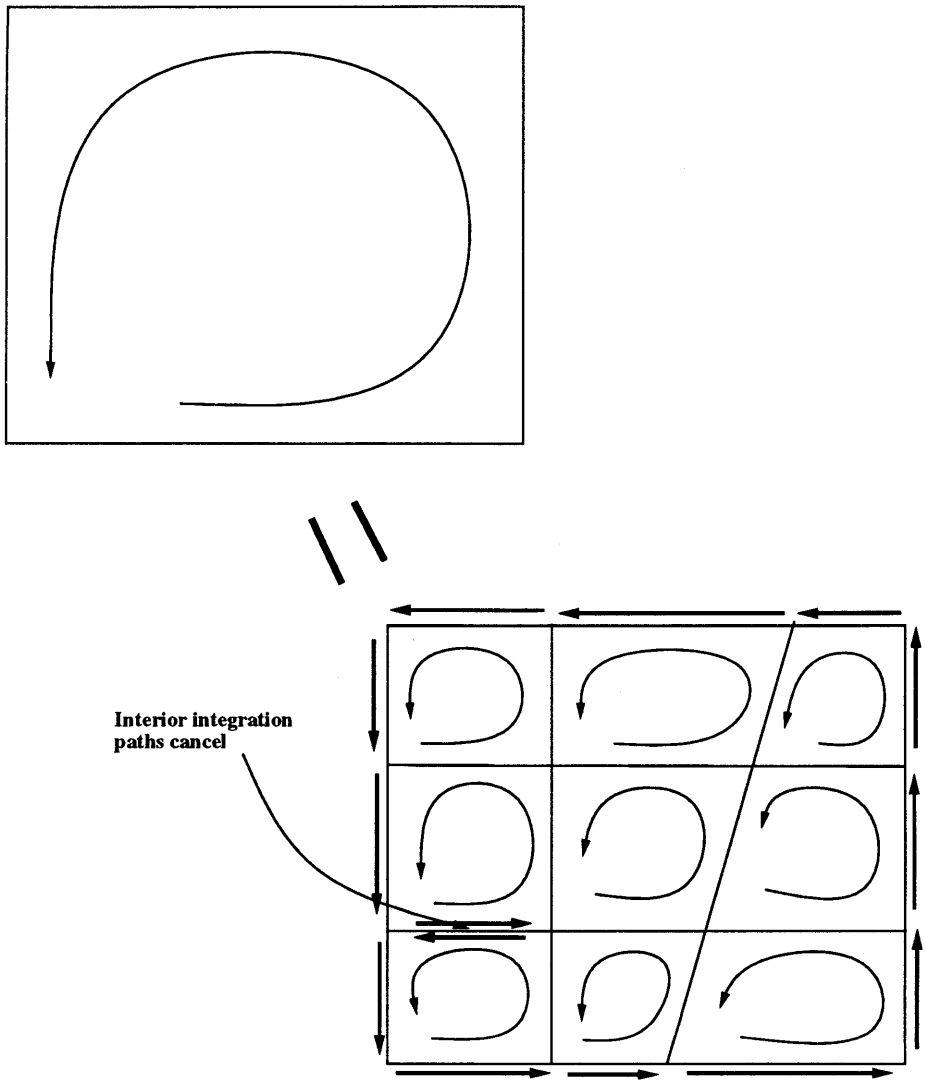
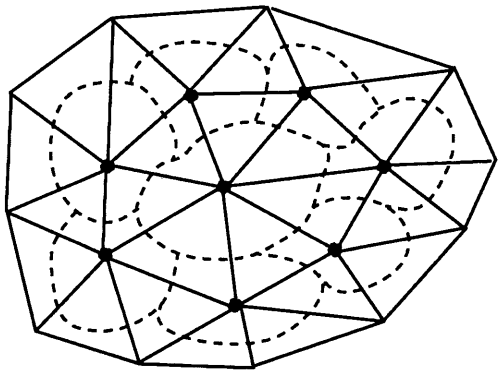


Figure 3.2: Integration over subvolumes

Generalized Control Volume



Restricted Generalized Control Volume

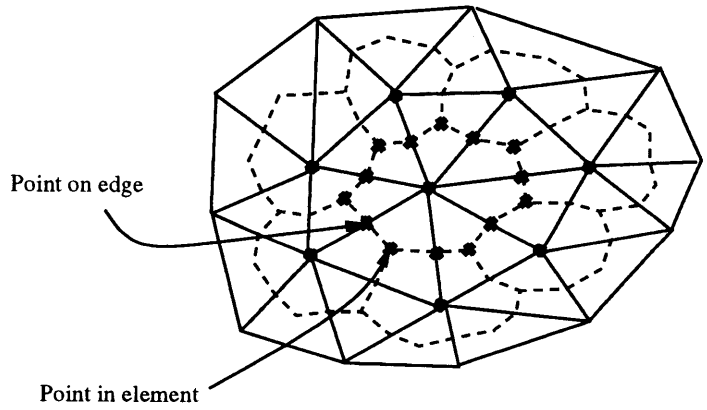


Figure 3.3: Generalized Control Volume

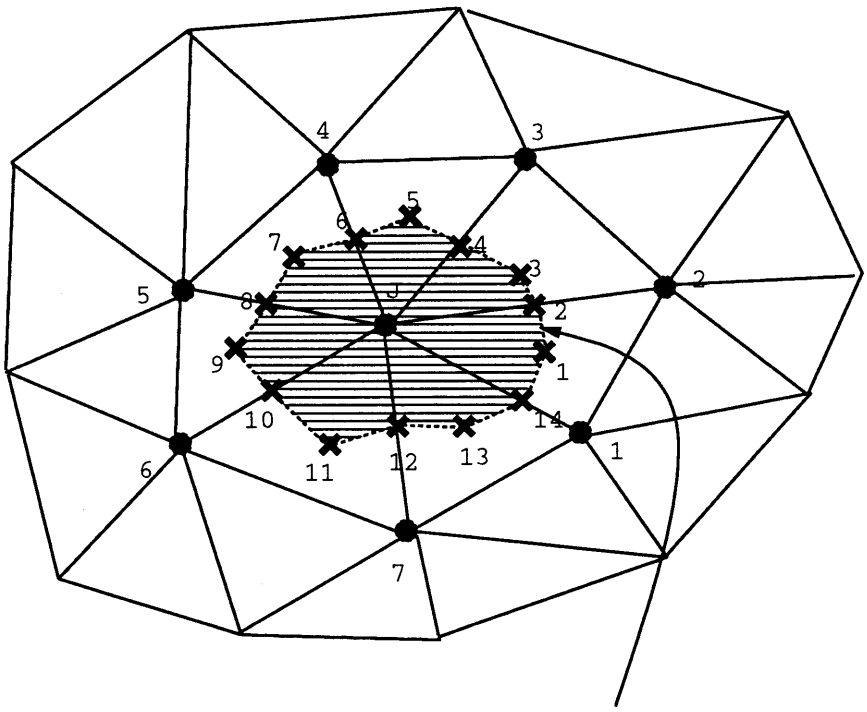


Figure 3.4: Numbered Control Volume

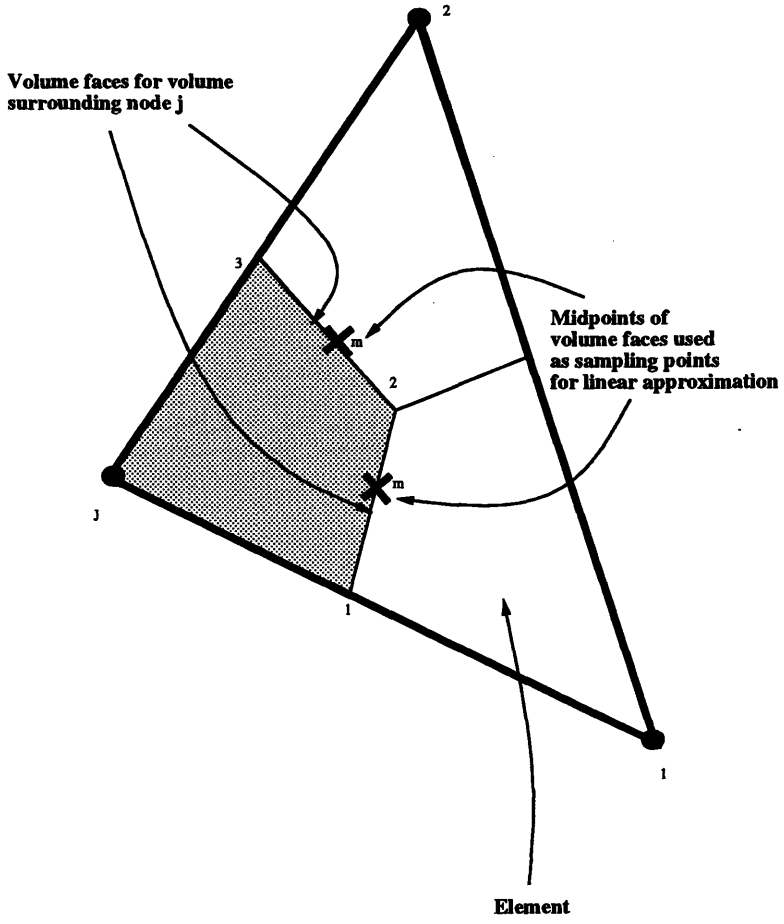


Figure 3.5: Element for Flux Evaluation Over Subvolume j

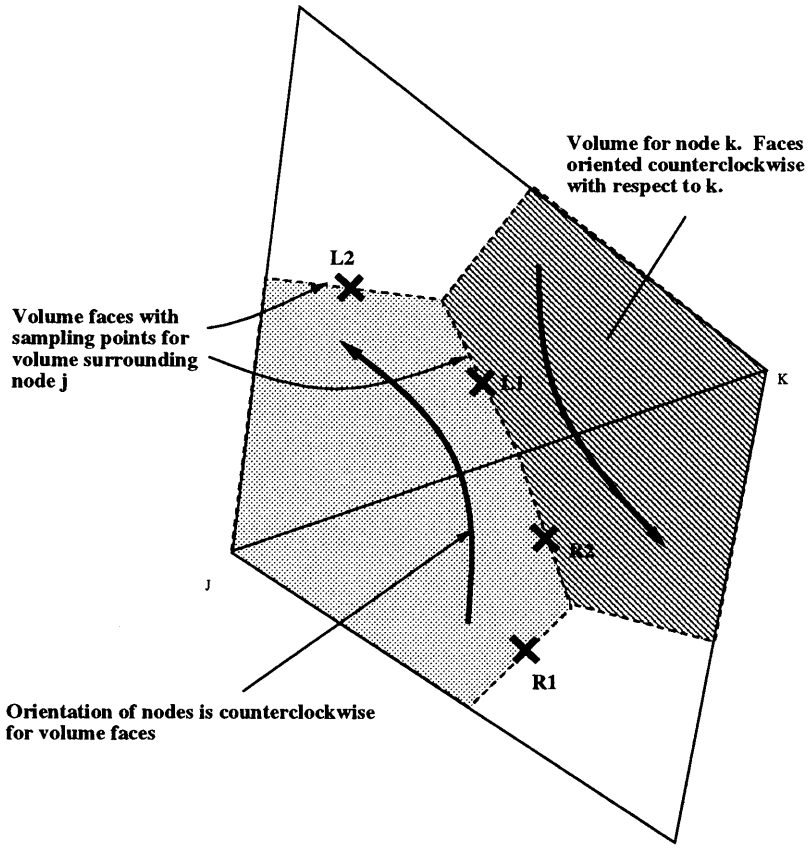


Figure 3.6: Edge for Flux Evaluation Over Subvolume j

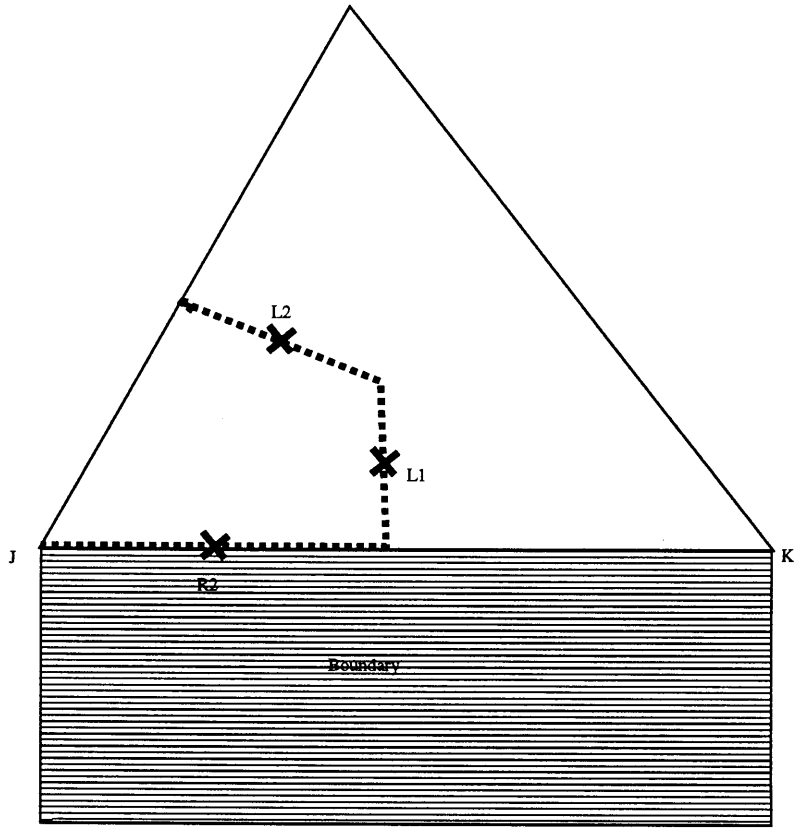


Figure 3.7: Edge for Flux Evaluation on Boundary j

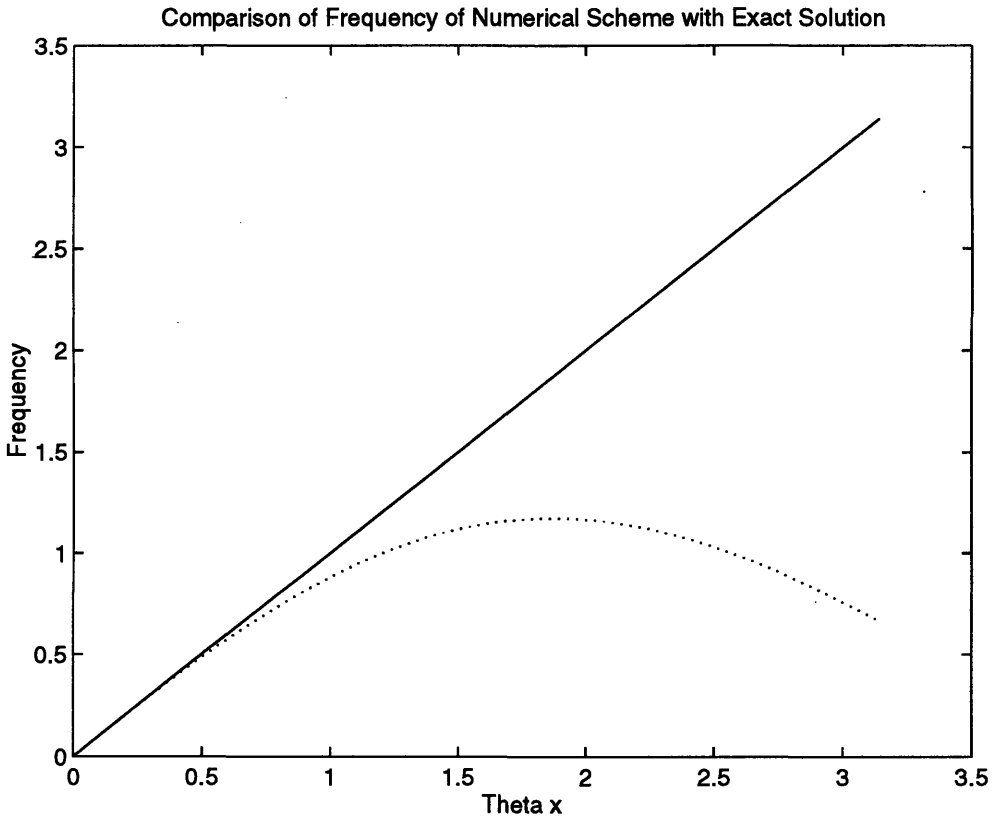


Figure 3.8: Phase Characteristics for Lumped Mass Matrix in x-direction

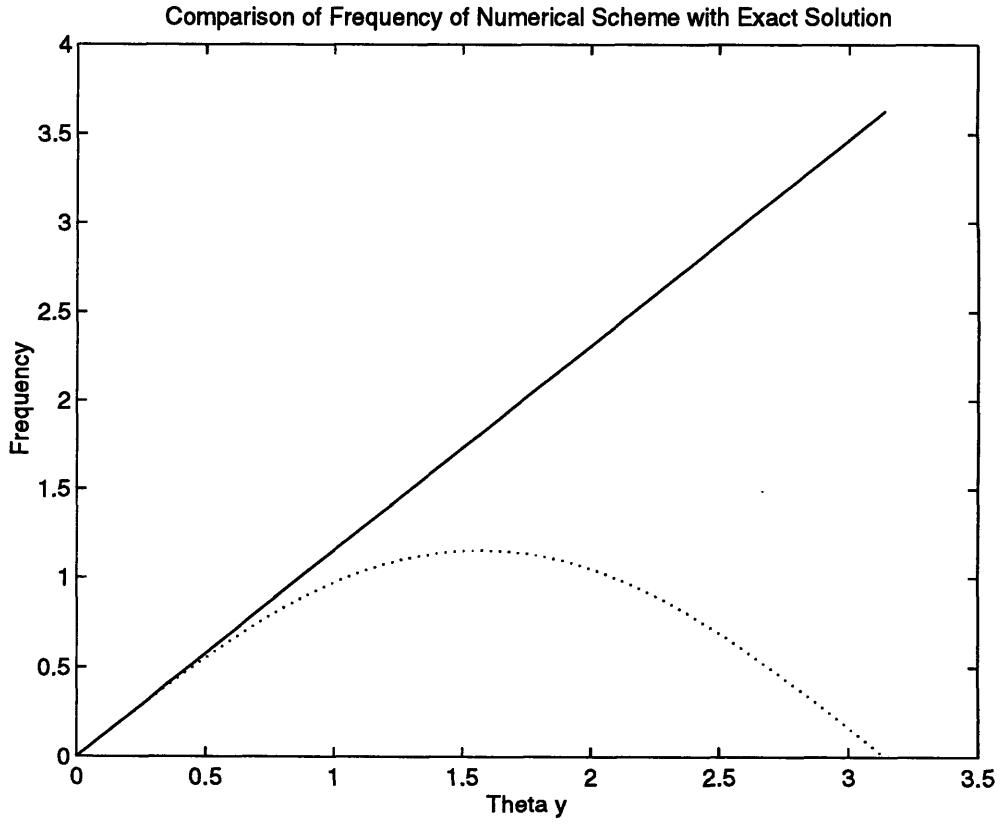


Figure 3.9: Phase Characteristics for Lumped Mass Matrix in y-direction

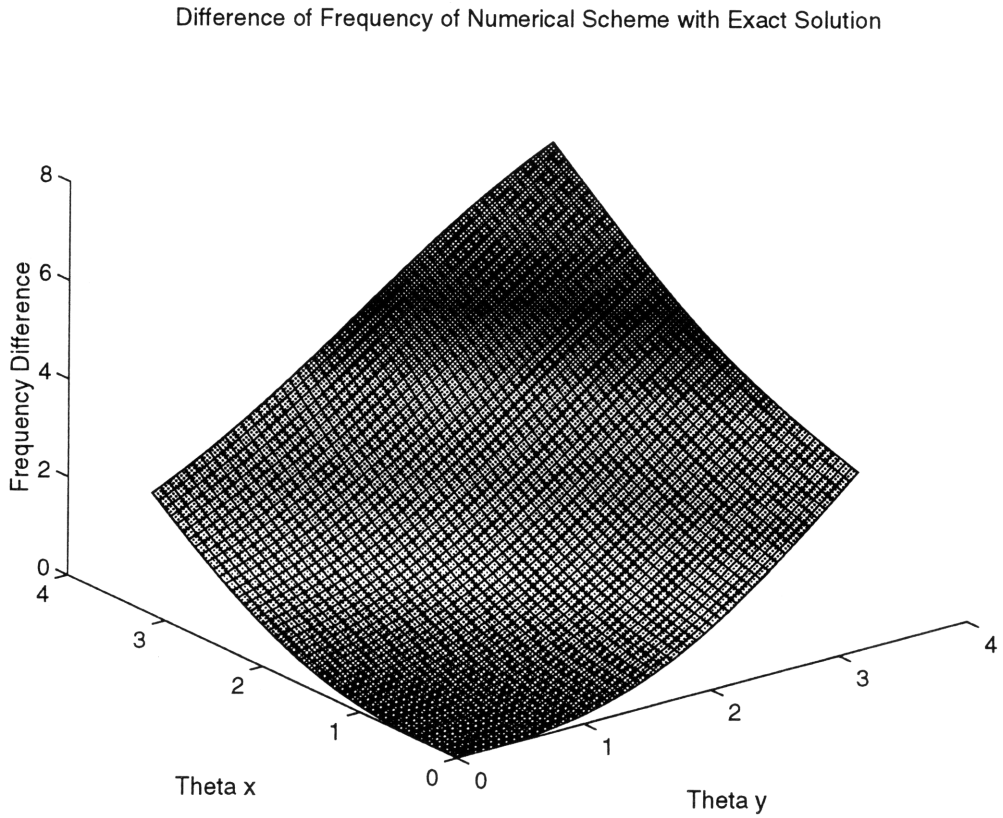


Figure 3.10: Surface Plot of Phase Characteristics for Lumped Mass Matrix

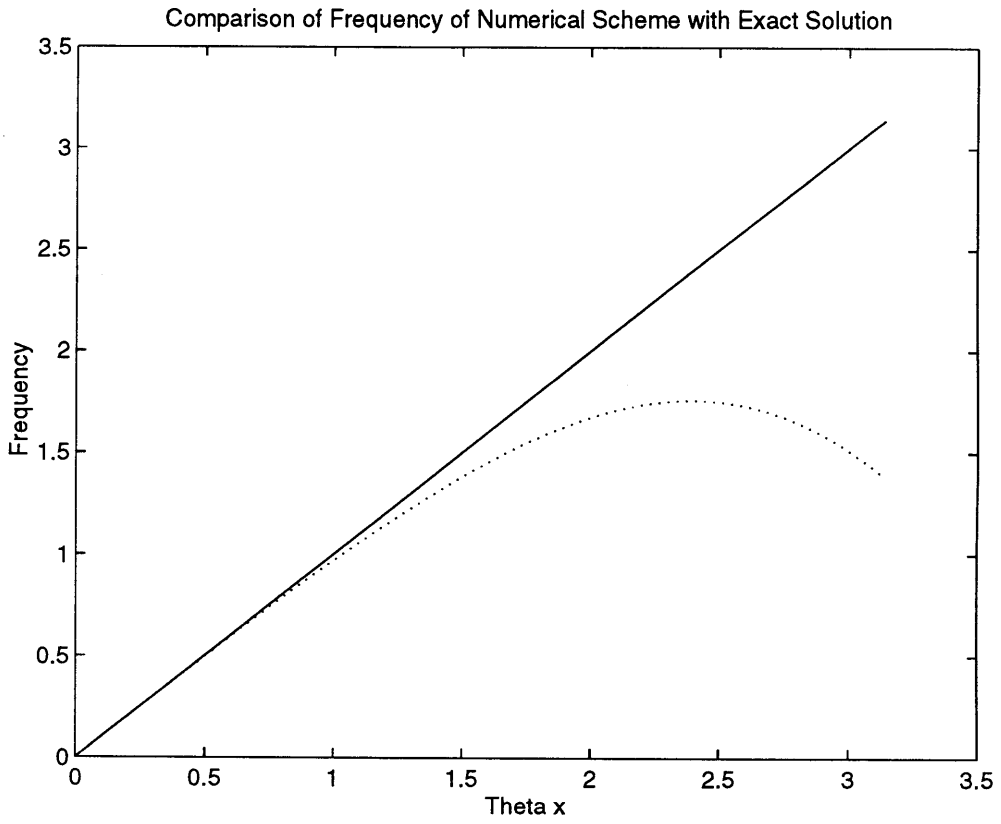


Figure 3.11: Phase Characteristics for Finite Volume Mass Matrix in x-direction

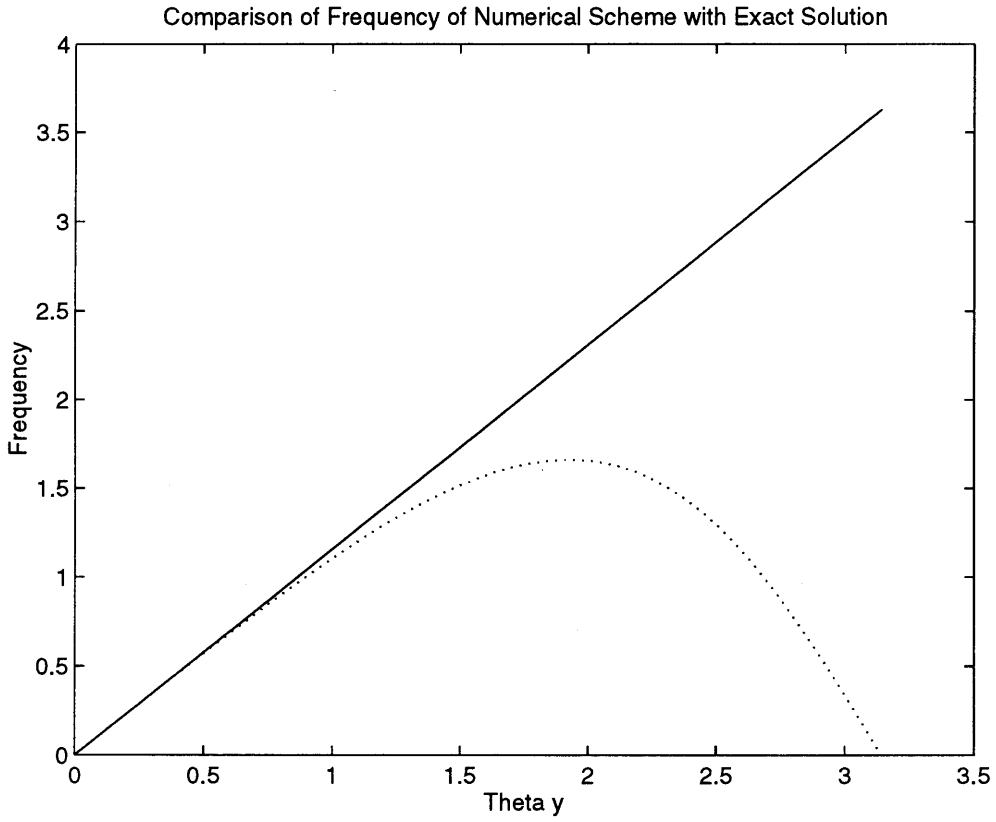


Figure 3.12: Phase Characteristics for Finite Volume Mass Matrix in y-direction

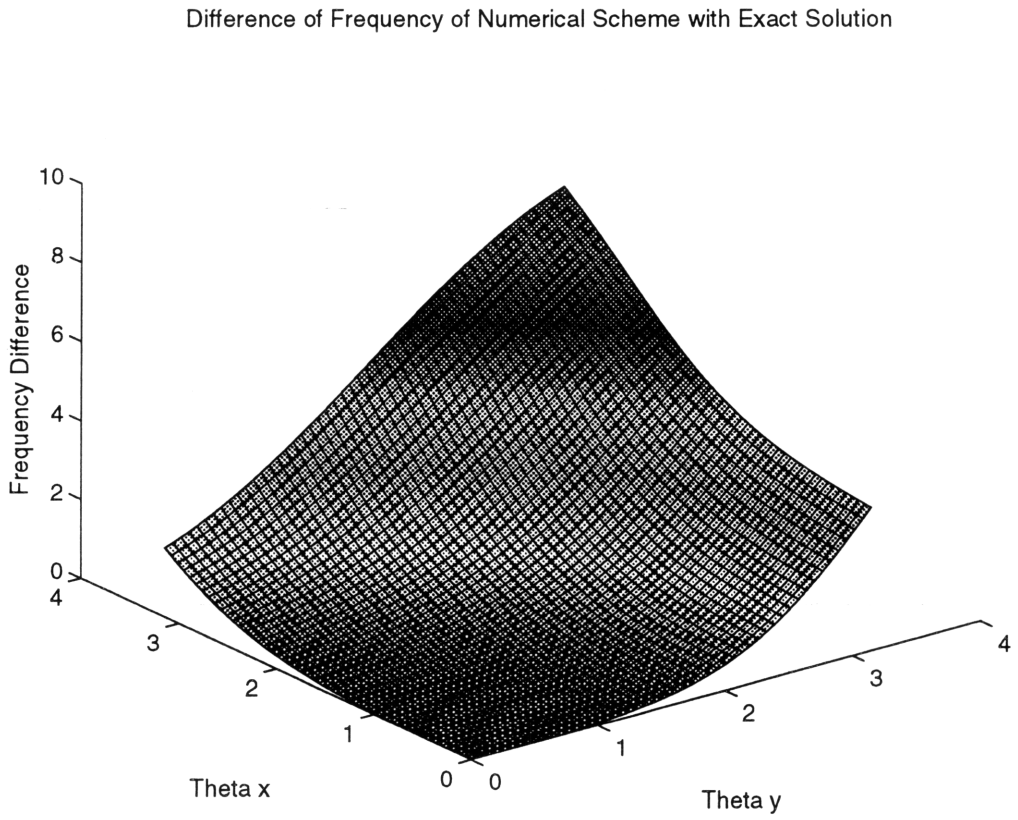


Figure 3.13: Surface Plot of Phase Characteristics for Finite Volume Mass Matrix

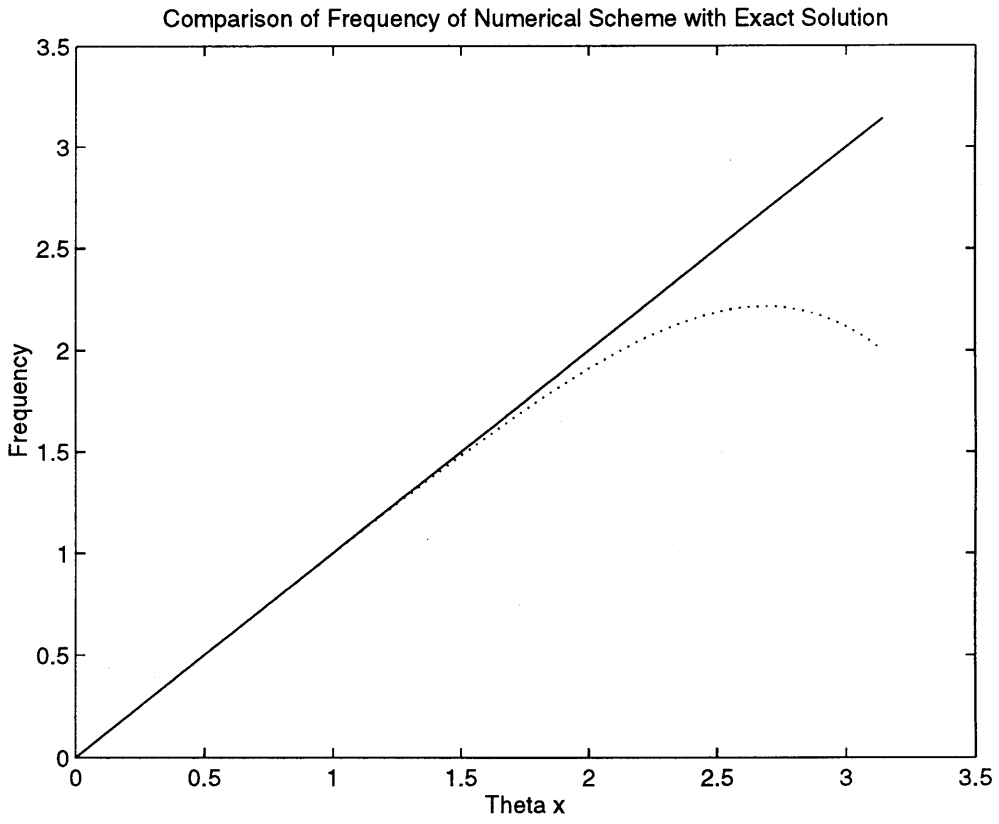


Figure 3.14: Phase Characteristics for Finite Element Mass Matrix in x-direction

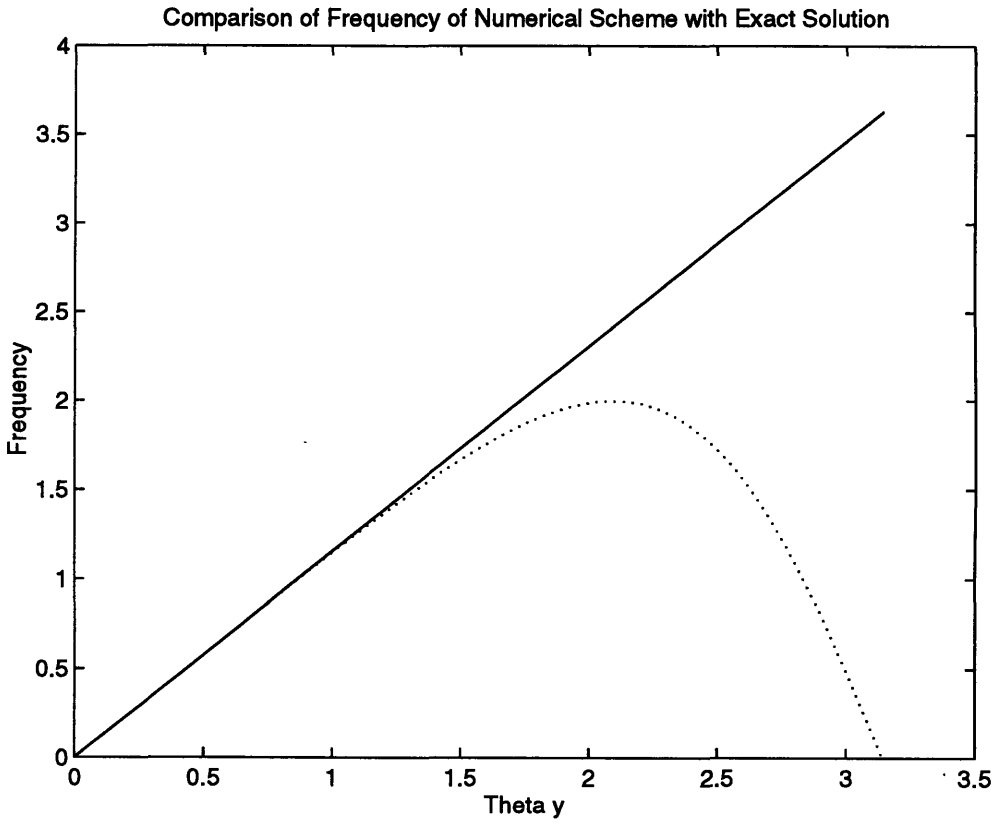


Figure 3.15: Phase Characteristics for Finite Element Mass Matrix in y-direction

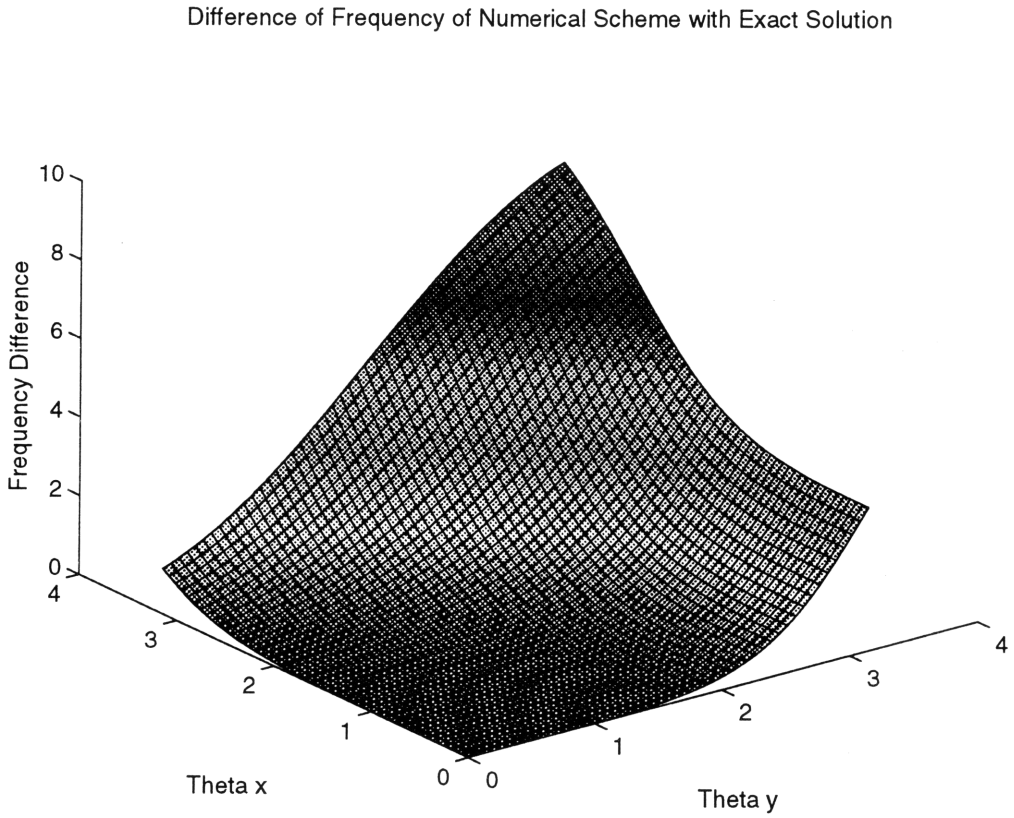


Figure 3.16: Surface Plot of Phase Characteristics for Finite Element Mass Matrix

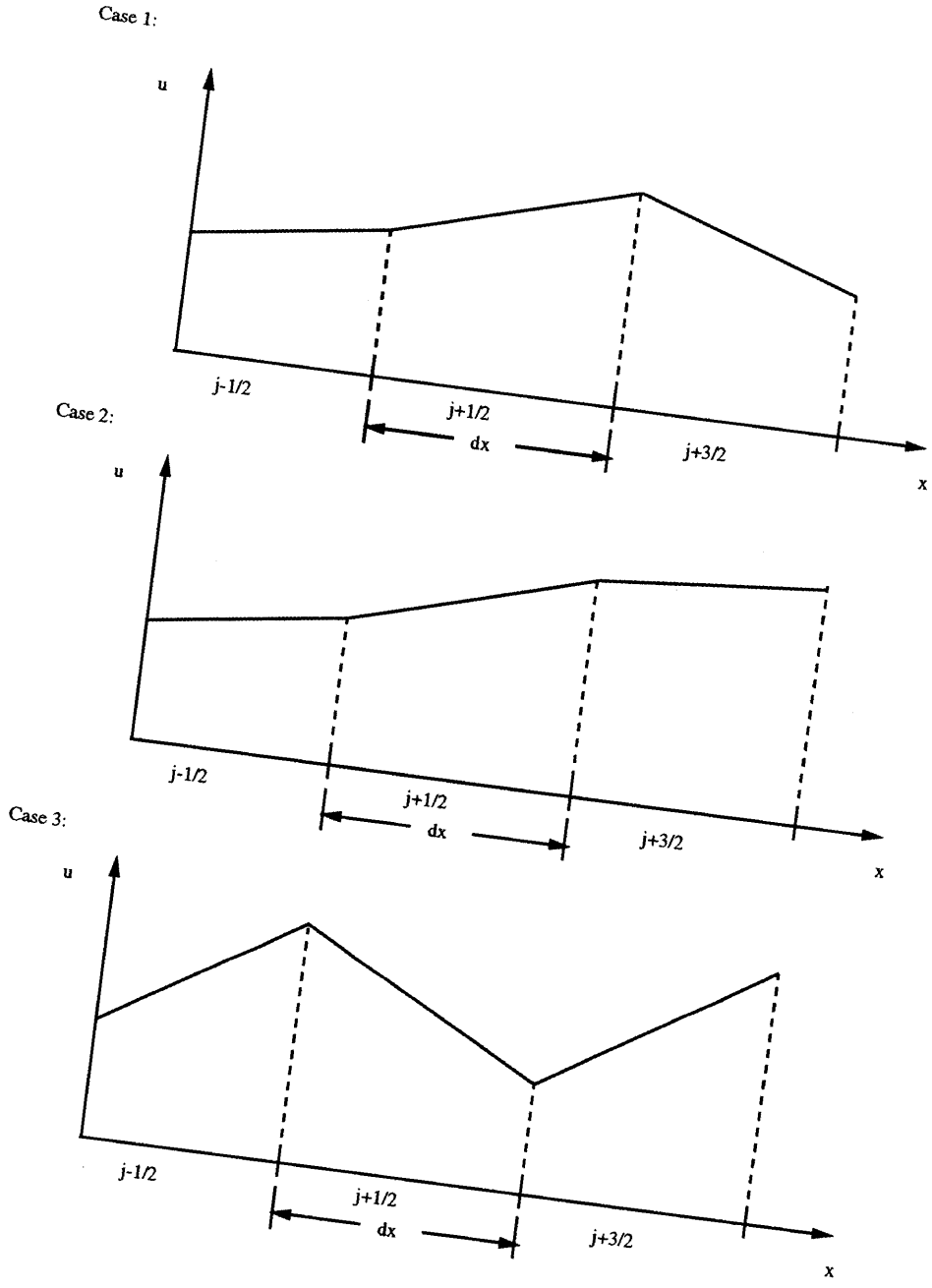


Figure 3.17: Consecutive Gradients

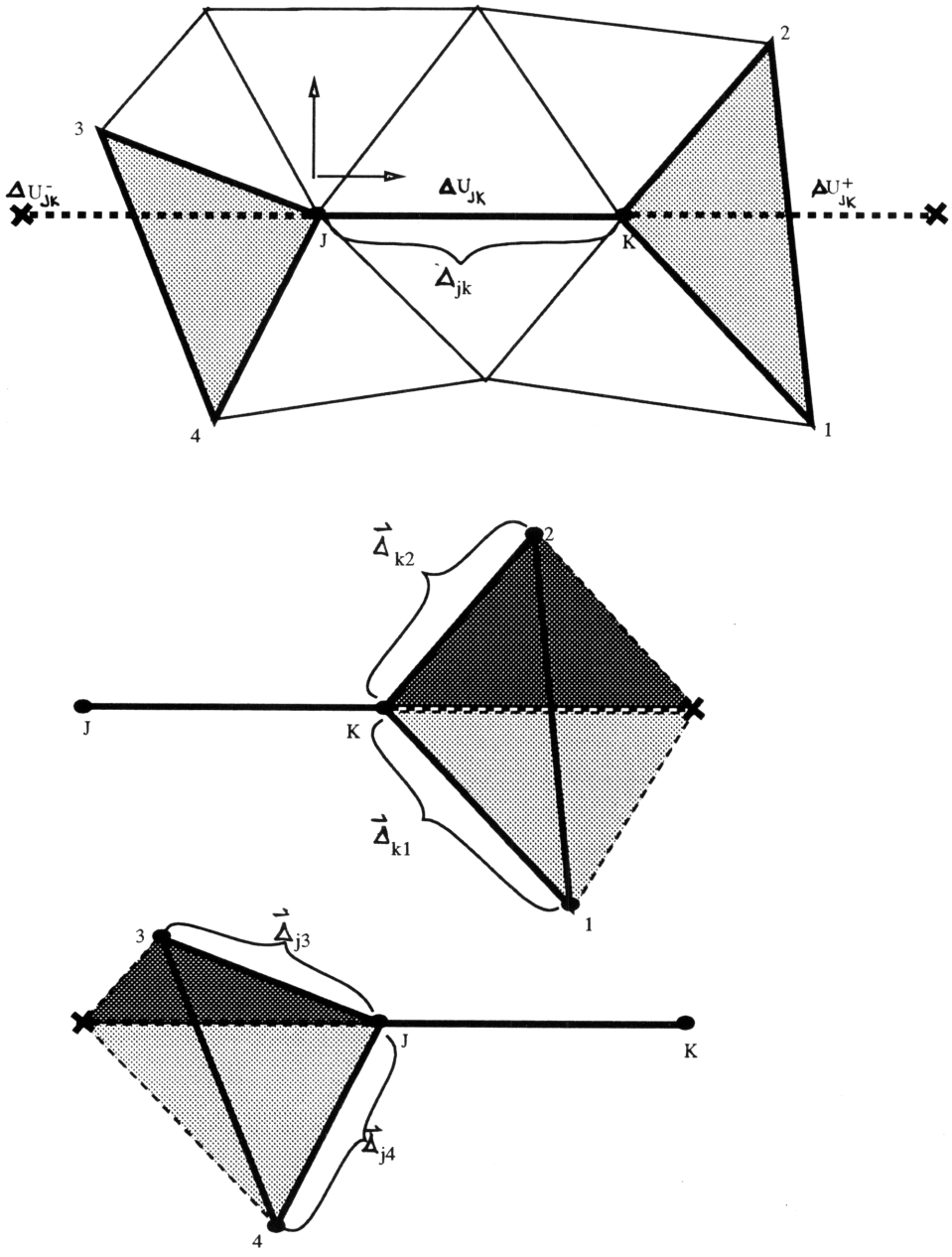


Figure 3.18: Gradient Reconstruction

Chapter 4
Code Validation

Two examples are used to validate the algorithm formulated in Chapter 3. Both test cases are transient applications where the exact solution can be determined. The first application is a shock tube and the second is a density perturbation in a uniform flow.

4.1 Shock Tube

The shock tube application is a good application for testing the scheme's ability to capture shocks and transient phenomena. The exact solution to the problem is outlined in Appendix D and is illustrated in figure D.1.

The initial conditions for the shock tube problem used for testing are given in table 4.1. The exact solution is shown in figures 4.2, 4.3, 4.4 and 4.5.

Numerical simulations are performed for various meshes and combinations of algorithm parameters which are summarized in table 4.2. The boundary conditions used for the numerical simulation are illustrated in figure 4.1 and discussed in Appendix C. The meshes are uniform meshes comprised of quadrilaterals cells split into two triangular elements. Periodic boundary conditions are used on the upper and lower surfaces of the shock tube to help enforce the one-dimensionality of the problem and also verify their implementation in the numerical scheme. The Euler equations are non-dimensionalized as discussed in Appendix E using the density and speed of sound of the driven gas as the reference density and reference speed. The length of the shock tube is chosen as the reference length.

4.1.1 Case 1

This simulation uses the lumped mass matrix and soft limiting. The results are presented in figures 4.6,4.7, 4.8 and 4.9. The numerical simulation resolves the shock and expansion regions adequately. The contact discontinuity between regions two and three is not resolved very well and is diffused more than the shock. A possible explanation for this effect is that the contact discontinuity is not a nonlinearly driven flow feature. The numerical dissipation does not recognize the difference between this feature and the shock. Therefore, it applies a similar amount of dissipation to both flow features, resulting in a loss in resolution of the contact discontinuity.

4.1.2 Case 2

This simulation is identical to Case 1 except that it uses a smaller coefficient, Vc_4, for the higher order dissipation. The results are shown in figures 4.10,4.11, 4.12 and 4.13. A closer inspection of the results between Case 1 and Case 2 (see figures 4.15, 4.14 and 4.16) shows that Case 2 diffuses the flow more than Case 1. The gradient at the shock is lessened as is apparent in figure 4.16. This is expected using the smaller coefficient in Case 2.

4.1.3 Case 3

This simulation uses the finite element mass matrix with the soft limiter and higher order dissipation coefficient the same as for Case 1. The results are shown in figures 4.17, 4.18, 4.19 and 4.20. Comparing these results with the lumped mass matrix results shows a slight difference in the expansion region of the flow. The lumped solution exhibits a lag in the density plot, shown in figure 4.6. The finite element mass matrix solution does not exhibit this behavior, but appears dissipative near the tail of the expansion wave which could be a result of the limiting procedure.

4.1.4 Case 4

This simulation uses the finite element mass matrix with a smaller higher order dissipation coefficient. The results are shown in figures 4.21, 4.22, 4.23 and 4.24. The solution looks very similar to the Case 3 solution except in the expansion region where it appears slightly more diffused.

4.1.5 Case 5

This simulation uses the finite volume mass matrix. The results are shown in figures 4.25, 4.26,4.27 and 4.28. The same coefficients for the dissipation and the soft limiter are used as for Case 1 and Case 3. Comparing the results with the finite element and lumped mass matrices shows a small variation between the three. The finite volume mass matrix solution appears to lie between the lumped and finite element mass matrix solutions in the expansion region. This can be expected from the Fourier analysis found in Appendix B and the discussion in Chapter 3. Overall, the results agree quite well with the exact solution.

4.1.6 Case 6

This simulation uses a mesh with twice as many nodes as the previous six cases. The results are shown in figures 4.29, 4.30 and 4.31. The figures show increased resolution of the contact discontinuity and the shock wave as expected. These results are in excellent agreement with the exact solution.

4.2 Density Perturbation

This application is chosen to test the scheme's ability to convect small perturbations. It consists of a uniform flowfield with a density perturbation in the center of the domain. The density perturbation is convected uniformly downstream at the freestream velocity. The exact solution for this application is determined using the linearized Euler equations and is found in [21] and in Appendix D. Figure 4.33 shows a schematic of the simulation.

Several different simulations are made for both a Gaussian profile and a square profile perturbation. These are summarized in table 4.3. Cases 1, 2 and 3 focus on the differences between the mass matrices. Cases 4, 5 and 6 focus on the soft limiting aspects of the flux limiter. Cases 7, 8 and 9 focus on the use of different higher order flux coefficients, Vc_4. All cases use a freestream Mach number of .5. The boundary conditions are the locally one dimensional Riemann Invariant treatment found in Appendix C. The Euler equations are non-dimensionalized using the freestream density, freestream speed of sound and the length of the flow domain.

4.2.1 Case 1

This simulation uses the finite element mass matrix with no limiting. This means that only the higher order diffusive flux is used so the perturbation is not significantly diffused as it is convected. The result is illustrated in figure 4.34, showing the perturbation magnitude versus the streamwise location.

The ability of the finite element mass matrix scheme to adequately convect the density perturbation is clearly evident. The numerical solution agrees very nicely with the exact solution with little attenuation.

4.2.2 Case 2

This simulation is the same as Case 1, but uses the finite volume mass matrix. Figure 4.35 shows the result of the numerical solution. The numerical solution exhibits some lag in

the convection of the pertubation which is not evident in the finite element mass matrix solution. This agrees with the Fourier analysis presented in Appendix B and Chapter 3 which show the finite volume mass matrix scheme deviates from the correct wave speed for phase angles greater than one. The finite element mass matrix is able to replicate the correct wave speed of phase angles up to 1.5. This simulation re-inforces the Fourier analysis.

4.2.3 Case 3

This simulation is the same as Cases 1 and 2, but uses the lumped mass matrix. Figure 4.36 shows the numerical solution. The solution exhibits a large deviation in convecting the perturbation and shows oscillations on the upstream side. This oscillation could be a result of the scheme's inability to properly propagate the higher frequency modes that comprise the perturbation. Since these waves propagate at a slower speed than the low frequency modes as shown in the Fourier analysis in Appendix B and Chapter 3, they tend to separate from the other modes. Therefore, the oscillation can be a result of this effect.

4.2.4 Case 4

This simulation tests the effect of the soft limiting procedure. It uses the finite element mass matrix with the soft limiting perturbation coefficient, $\Psi = 1.e - 6$. The result is shown in figure 4.37. The scheme convects the perturbation and agrees well with the exact solution. The limiter tends to smooth the high gradient regions without too large an attenuation in the flow structure.

4.2.5 Case 5

This simulation increases the soft limiting perturbation coefficient to $\Psi = 1.e^{-2}$, one order of magnitude larger than the density perturbation. The result is shown in figure 4.38 and shows oscillations at the large gradient areas of the flow structure. Both oscillations appear on the upwind side (recall the flow direction is from left to right). The soft limiter allows these perturbations in the solution as compared with Case 4 which does not exhibit this behavior. Therefore, this indicates that the soft limiter is effective in allowing small perturbations. The oscillations in the figure are very small, of the order $1.e{-}4$. They should not be as evident for a larger density perturbation.

4.2.6 Case 6

This simulation is the same as Case 5, but with a larger density perturbation. The result is shown in figure 4.39 and shows no apparent oscillation in the large gradient regions. Very small deviations appear at the same locations as the oscillations in Case 5. It is an acceptable compromise in allowing the limiting procedure to have small perturbations which can be acoustic phenomena. Overall, the scheme performs adequately in capturing the flow solution.

4.2.7 Case 7

This simulation uses the standard limiting procedure, finite element mass matrix, no soft limiting (i.e. $\Psi = 0$) and $Vc_4 = 0$. The result is shown in figure 4.40 and shows a large attenuation of the density perturbation. This is due to the limiting procedure smoothing the peak of the perturbation as it is convected downstream. This indicates a soft limiting procedure is needed to allow the perturbation to pass unattenuated.

4.2.8 Case 8

This simulation uses the soft limiter and a higher order dissipation coefficient, $Vc_4 = .5$. The result is given in figure 4.41 and shows very little attenuation of the disturbance as compared to Case 7. The soft limiting is capable of allowing the disturbance to convect without the smoothing effect of the normal limiting procedure.

4.2.9 Case 9

This simulation is the same as Case 8, but with the higher order dissipation coefficient, Vc_4, set to .99. The result is given in figure 4.42 and shows a smaller attenuation of the disturbance. This behavior is expected with the limiter applying less dissipation than in Case 8.

4.3 Conclusions

This chapter tests the numerical scheme for the solution of the Euler equations developed in the previous section. The shock tube simulation illustrates the shock capturing capability of the soft limiting procedure for all three schemes: lumped, finite volume and finite element mass matrix. Although the numerical dissipation is based on the lumped mass matrix form, the finite element and finite volume mass matrices exhibit no oscillations in

the vicinity of the shock. Small variations between the three schemes are noticed in the expansion region of the flow. Also, refining the mesh increases the resolution of the shock and interface as expected.

The convected density perturbation reinforces the conclusions drawn from the Fourier analysis of Chapter 3 and Appendix B. This simulation shows the finite element mass matrix's improved ability over the finite volume and lumped mass matrices to accurately propagate a disturbance. Finally, the soft limiting procedure shows a large improvement over the standard limiting procedure in reducing the attenuation of a disturbance as it is convected.

Table 4.1: Initial Shock Tube Conditions

p_1	1 atm
ρ_1	1.225 $\frac{kg}{sec}$
$\frac{p_4}{p_1}$	10
$\frac{\rho_4}{\rho_1}$	8
γ	1.4
shock tube length	3 m
shock tube height	1 m
diaphragm location	1.5 m
elapsed time	2.0862 msec

Table 4.2: Inputs for the Numerical Simulation of the Shock Tube

Case No.	Matrix	Ψ	Vc_4	Mesh
1	Lumped	.01	.9	100x33
2	Lumped	.01	.45	100x33
3	Finite Element	.01	.9	100x33
4	Finite Element	.01	.45	100x33
5	Finite Volume	.01	.9	100x33
6	Lumped	.01	.9	200x67

Table 4.3: Inputs for the Numerical Simulation of the Density Perturbation

Case No.	Matrix	Ψ	Vc_4	Mesh	Profile	$\Delta\rho$
1	Finite Element	No limiting	.999	50x50	Gaussian	.001
2	Finite Volume	No limiting	.999	50x50	Gaussian	.001
3	Lumped	No limiting	.999	50x50	Gaussian	.001
4	Finite Element	.01	.45	50x50	Square	.001
5	Finite Element	.01	.9	50x50	Square	.001
6	Finite Element	.01	.9	50x50	Square	.1
7	Finite Element	0.	0.	50x50	Gaussian	.001
8	Finite Element	No limiting	.999	50x50	Gaussian	.001
9	Finite Element	No limiting	.999	50x50	Gaussian	.001

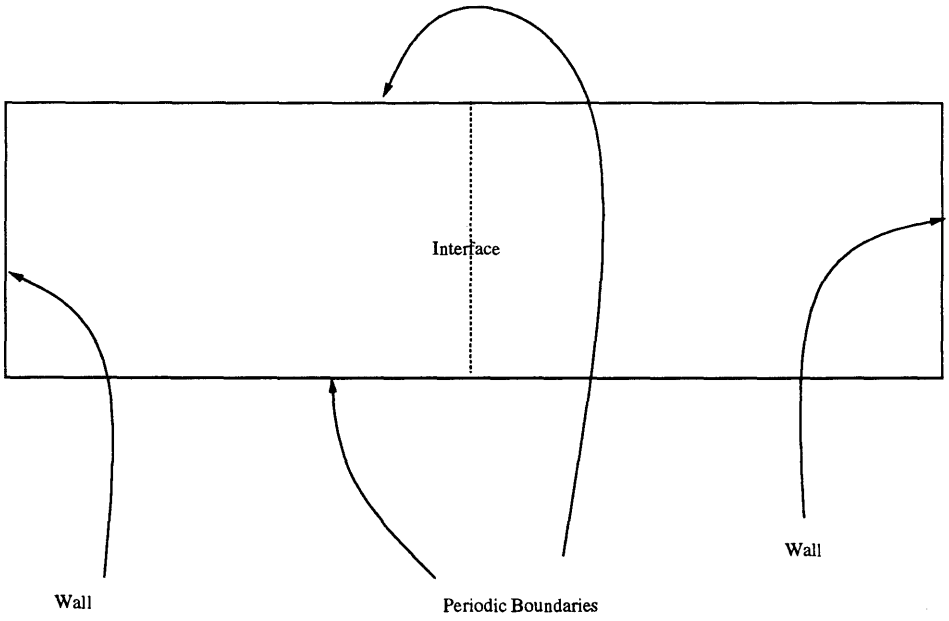


Figure 4.1: Shock Tube with Boundary Conditions

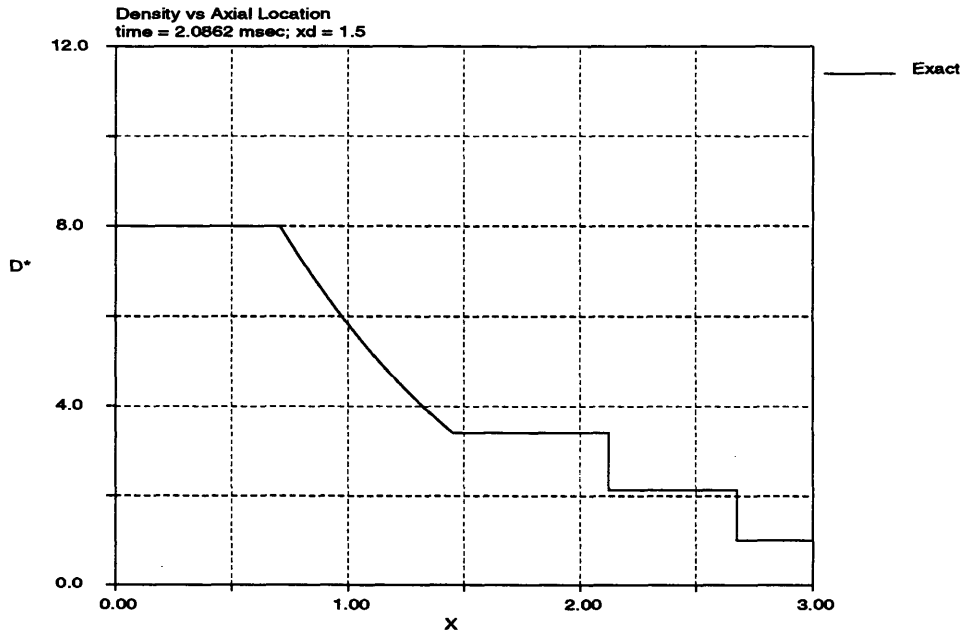


Figure 4.2: Density Distribution in Shock Tube

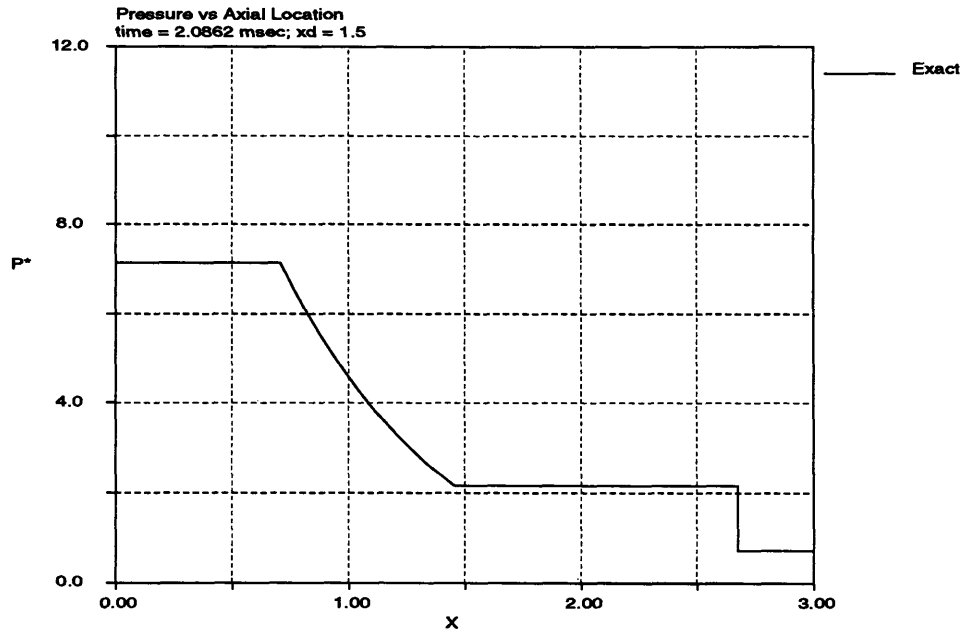


Figure 4.3: Pressure Distribution in Shock Tube

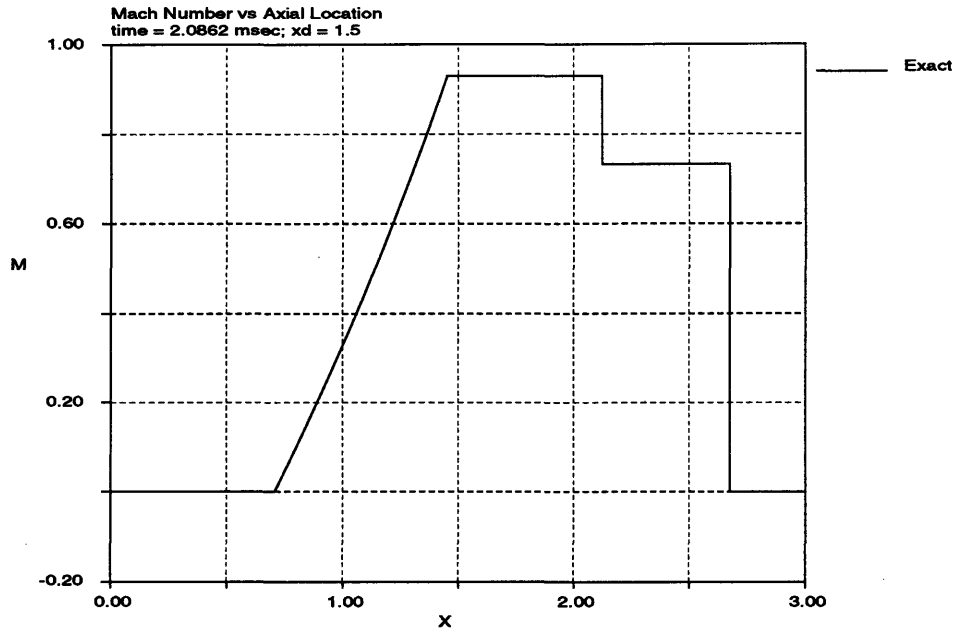


Figure 4.4: Mach Number Distribution in Shock Tube

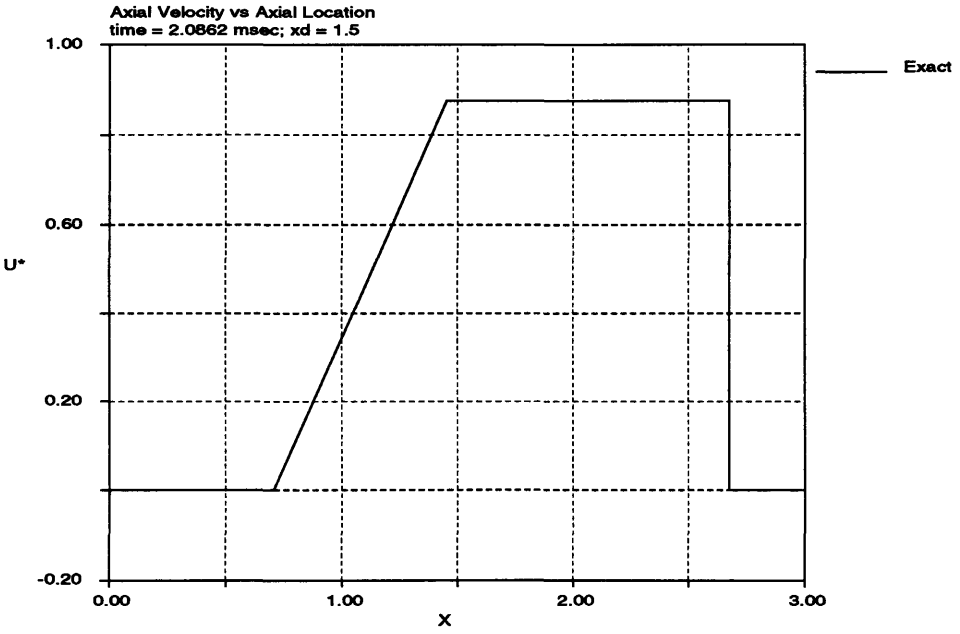


Figure 4.5: Axial Velocity Distribution in Shock Tube

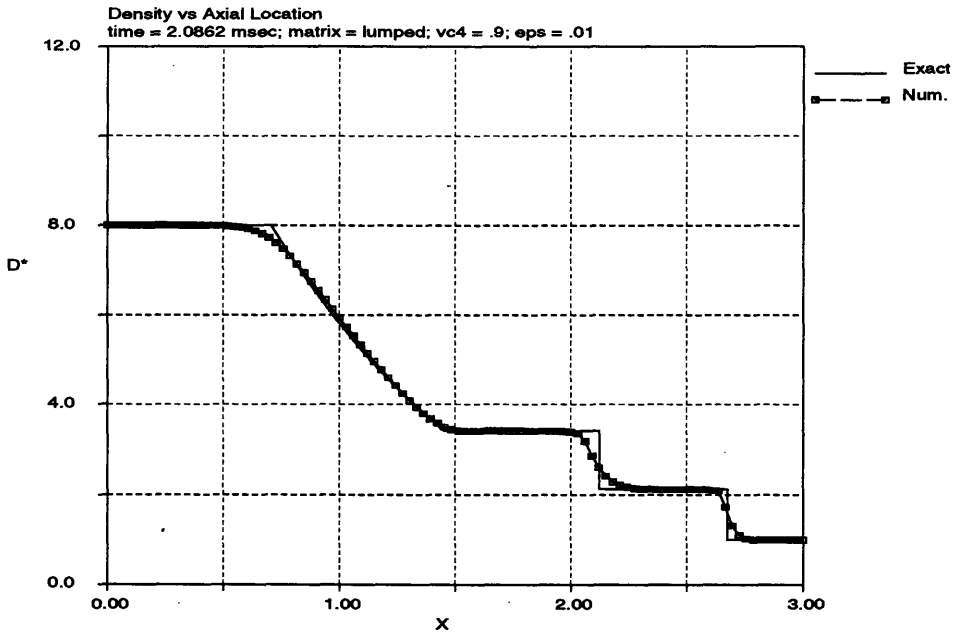


Figure 4.6: Density Distribution in Shock Tube for Case 1

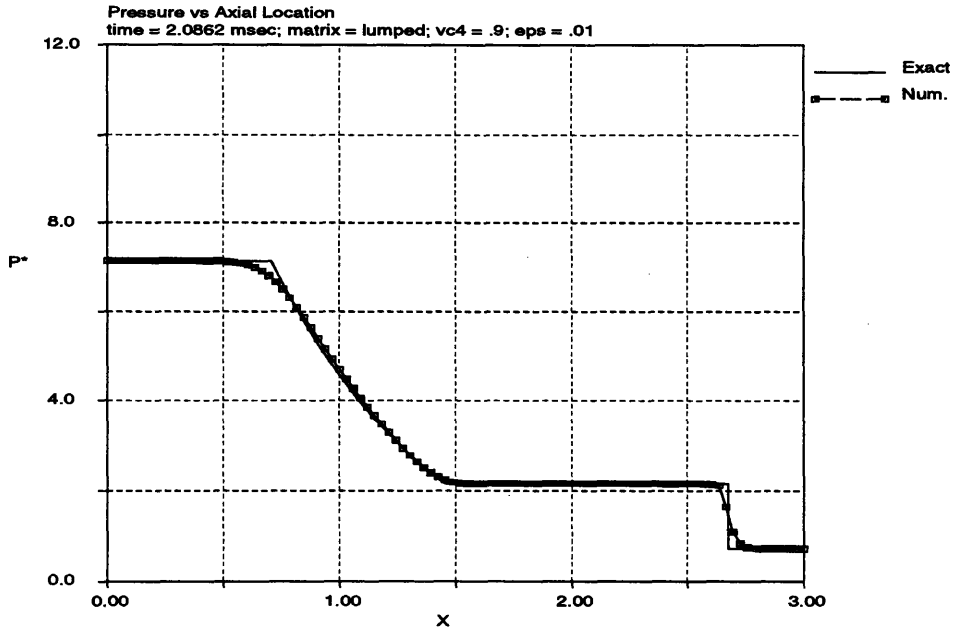


Figure 4.7: Pressure Distribution in Shock Tube for Case 1

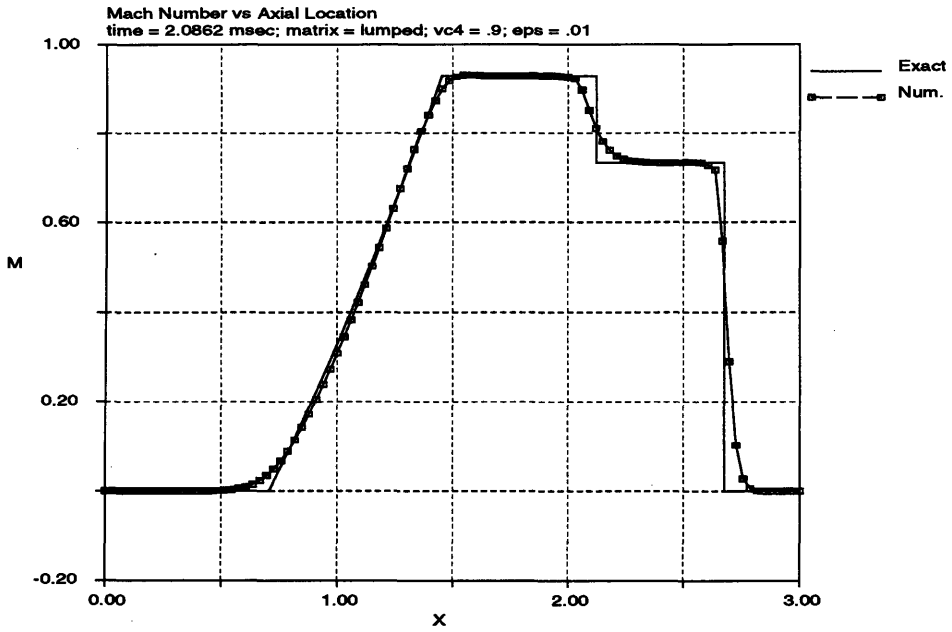


Figure 4.8: Mach Number Distribution in Shock Tube for Case 1

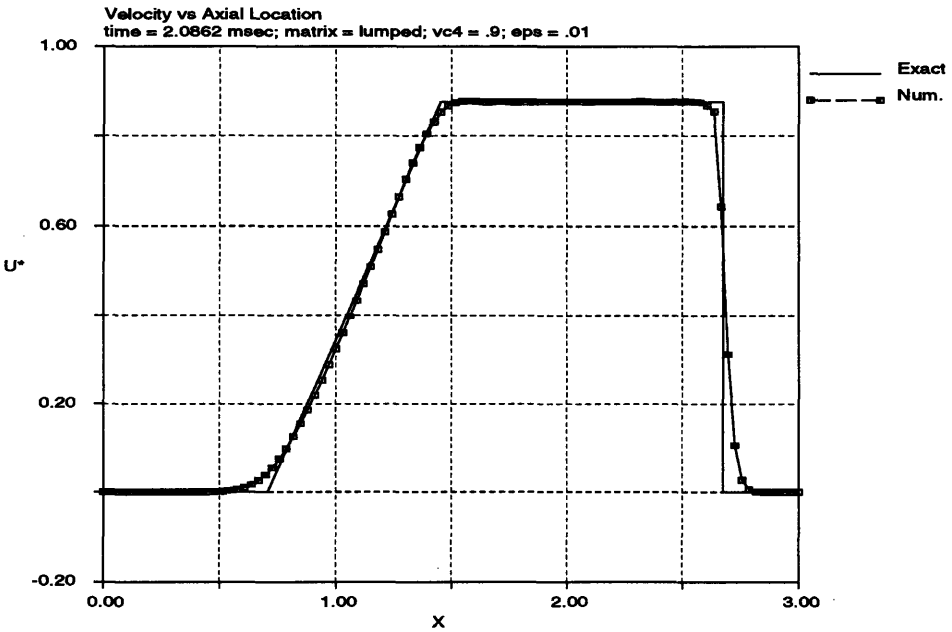


Figure 4.9: Axial Velocity Distribution in Shock Tube for Case 1

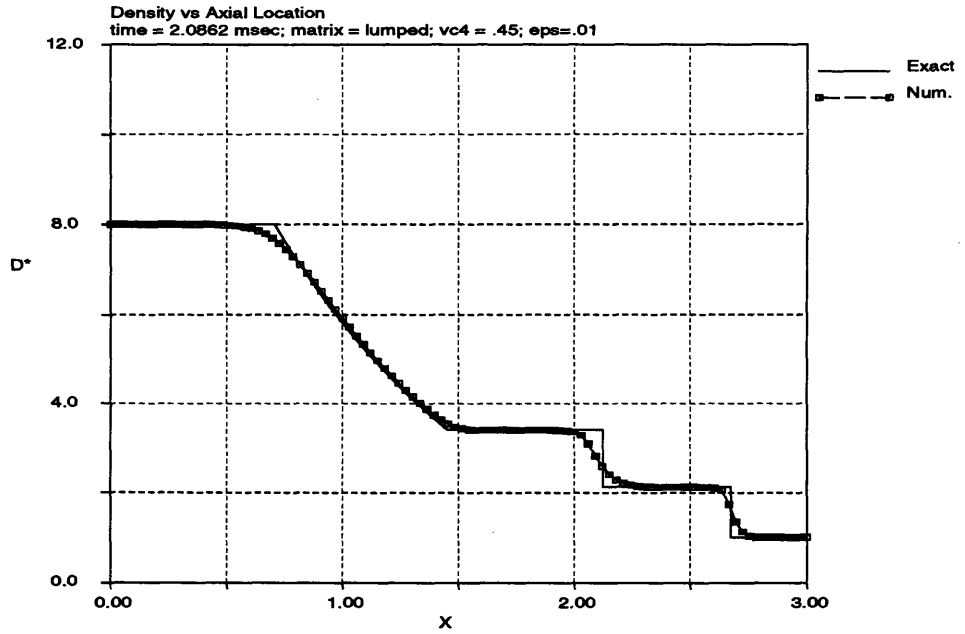


Figure 4.10: Density Distribution in Shock Tube for Case 2

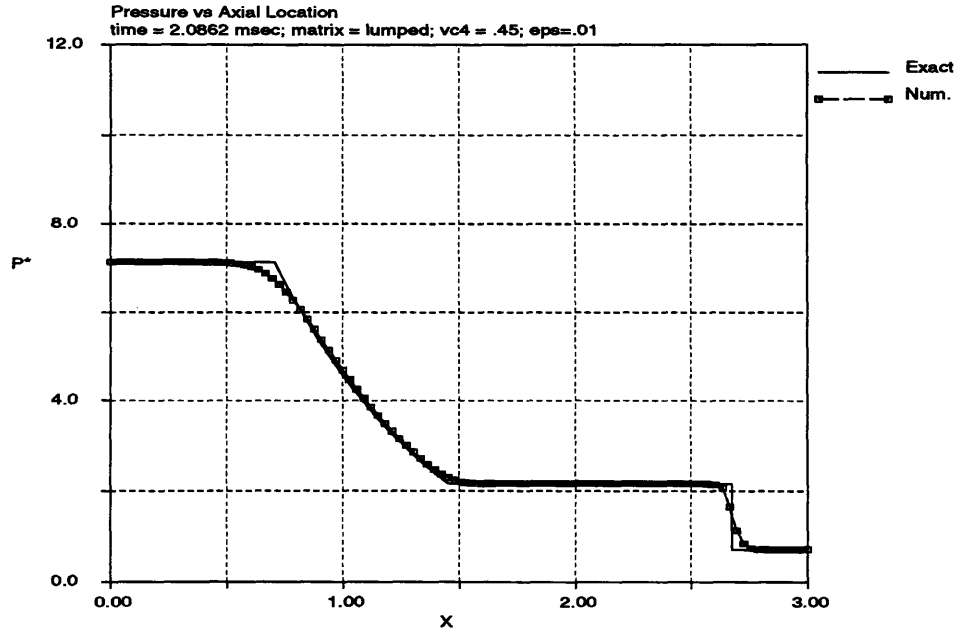


Figure 4.11: Pressure Distribution in Shock Tube for Case 2

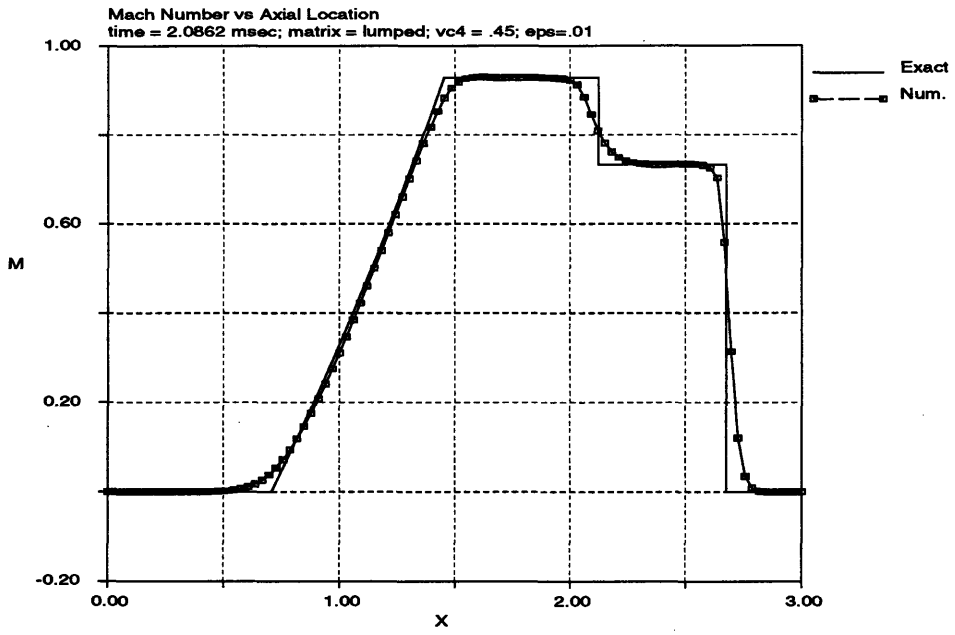


Figure 4.12: Mach Number Distribution in Shock Tube for Case 2

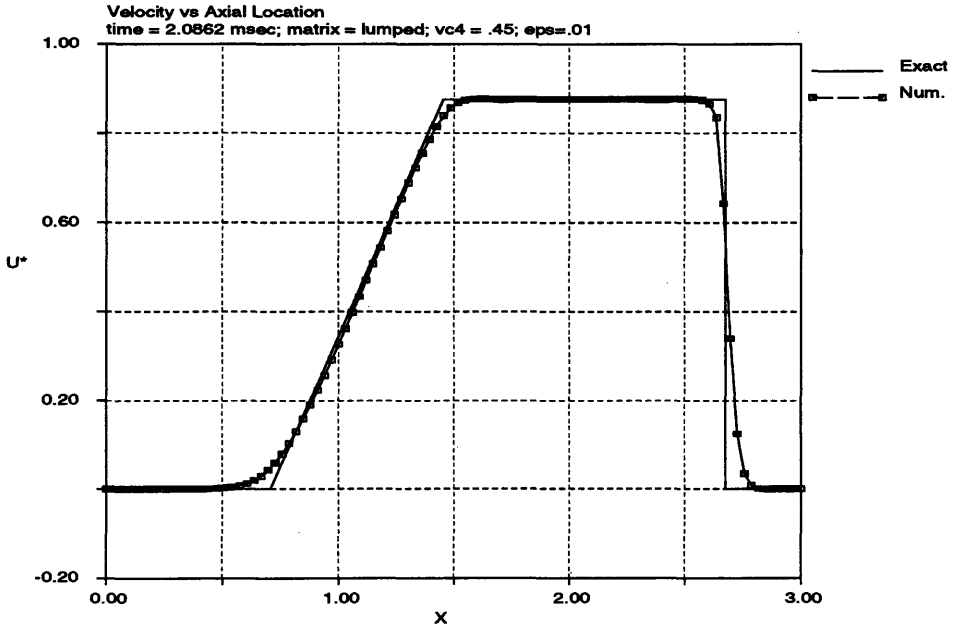


Figure 4.13: Axial Velocity Distribution in Shock Tube for Case 2

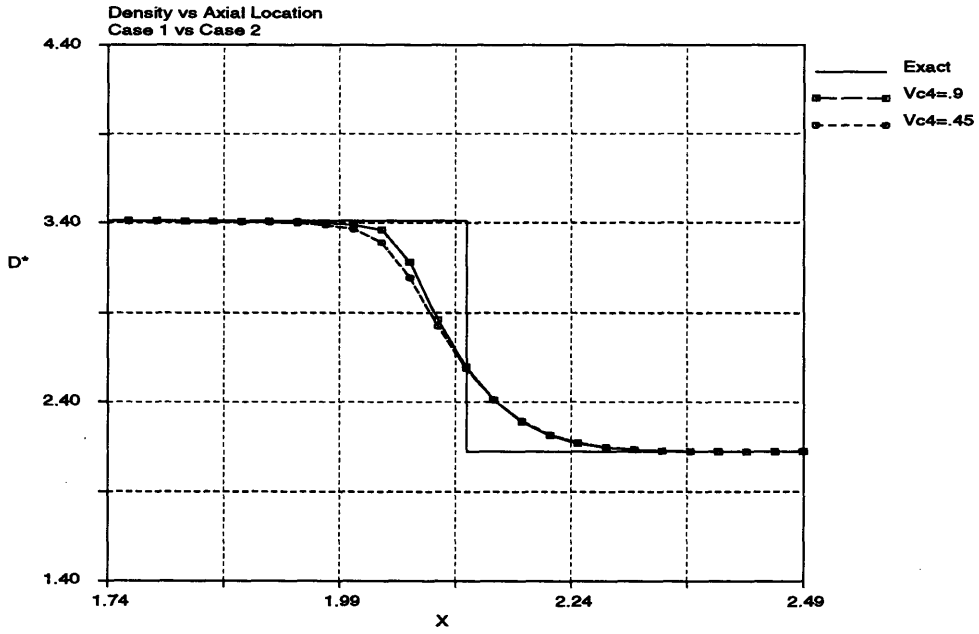


Figure 4.14: Density Distribution Comparison of Case 1 vs Case 2 in the Interface Region

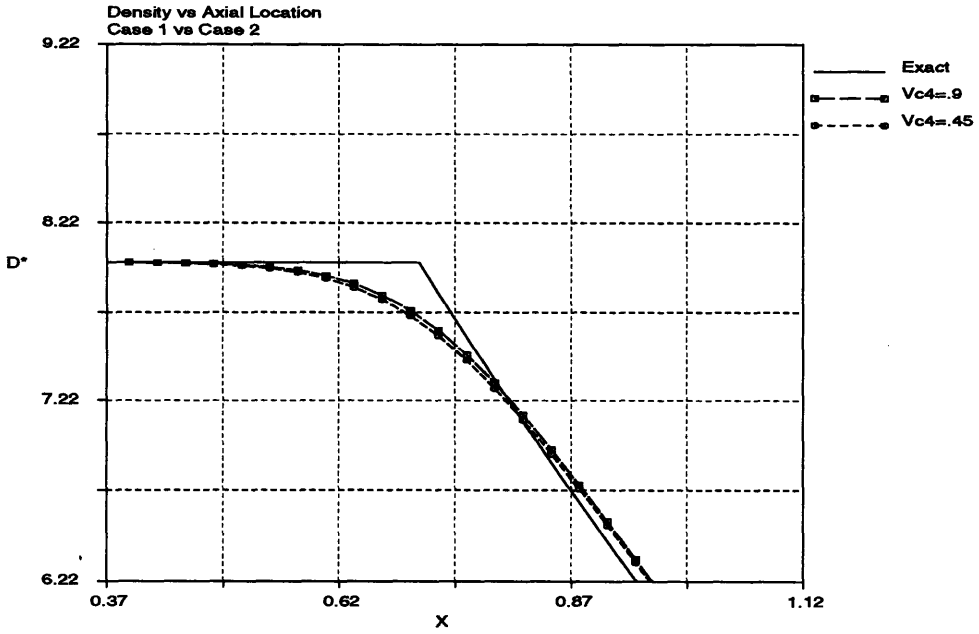


Figure 4.15: Density Distribution Comparison of Case 1 vs Case 2 in the Expansion Region

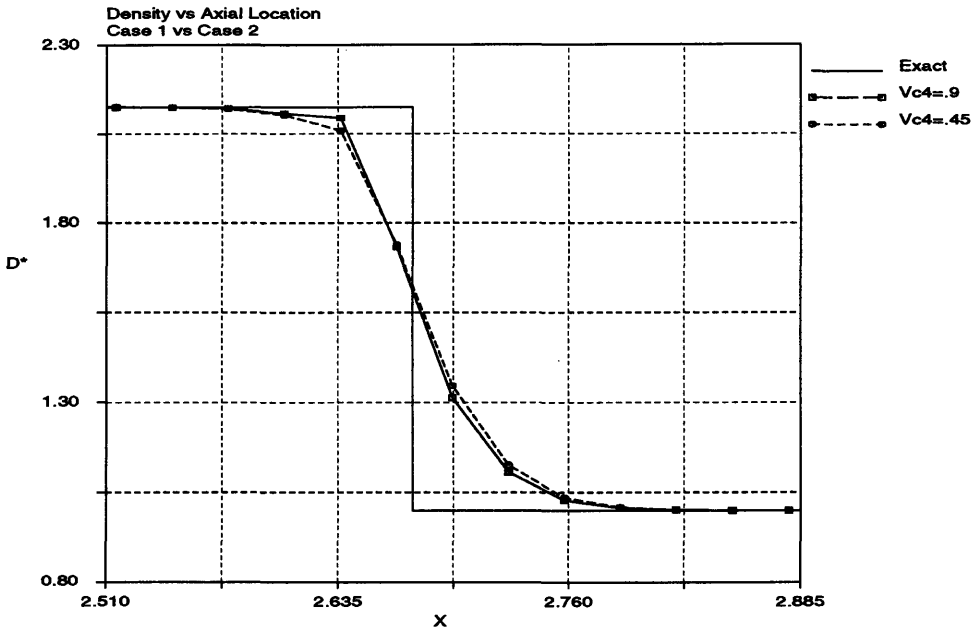


Figure 4.16: Density Distribution Comparison of Case 1 vs Case 2 in the Shock Region

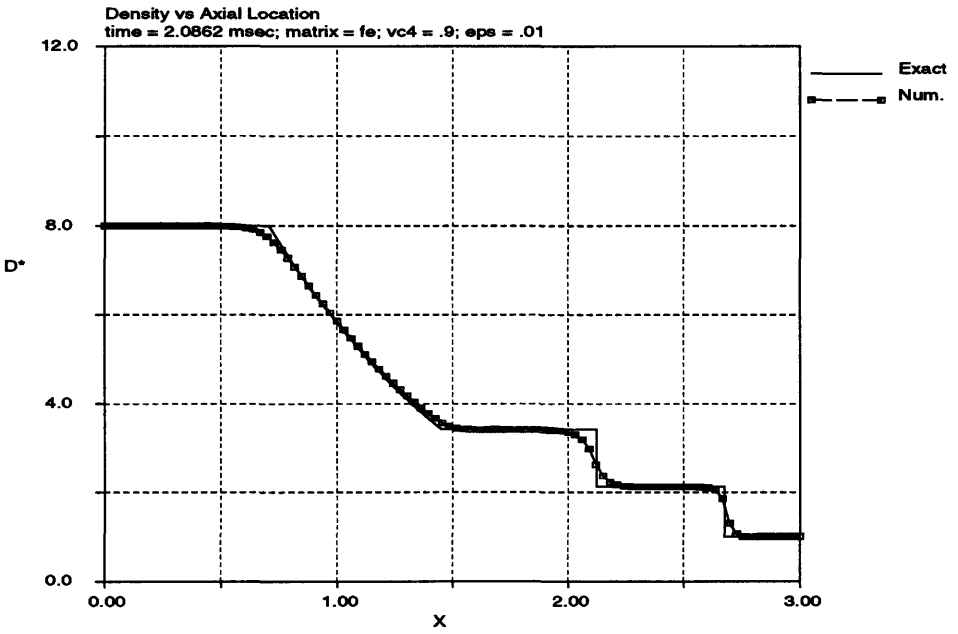


Figure 4.17: Density Distribution in Shock Tube for Case 3

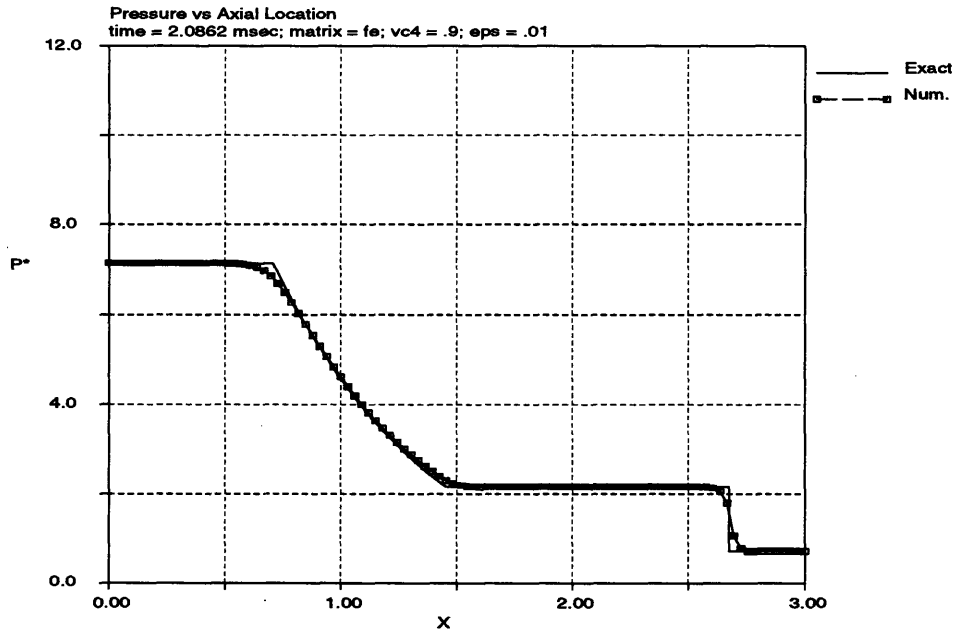


Figure 4.18: Pressure Distribution in Shock Tube for Case 3

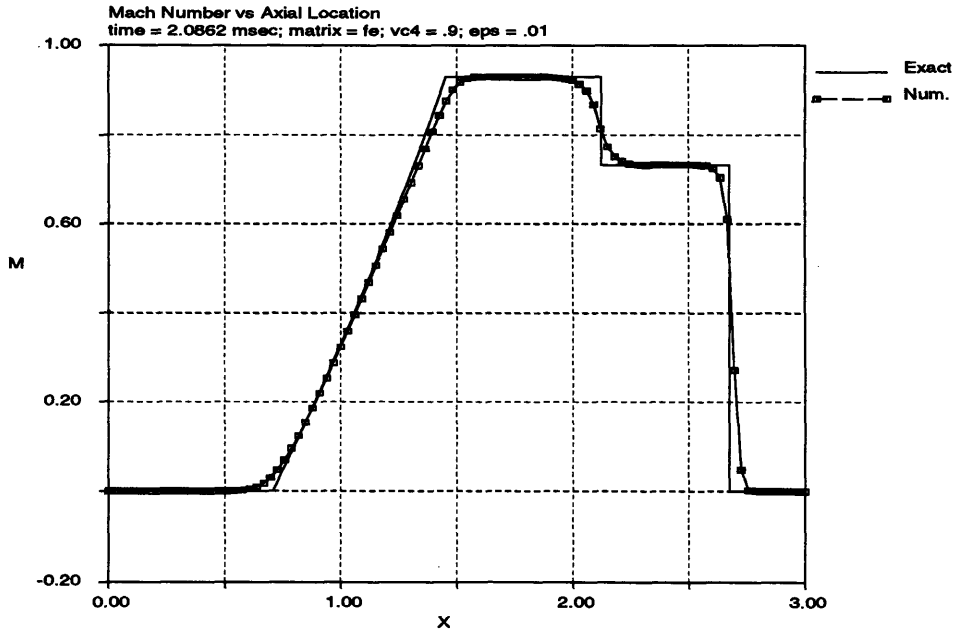


Figure 4.19: Mach Number Distribution in Shock Tube for Case 3

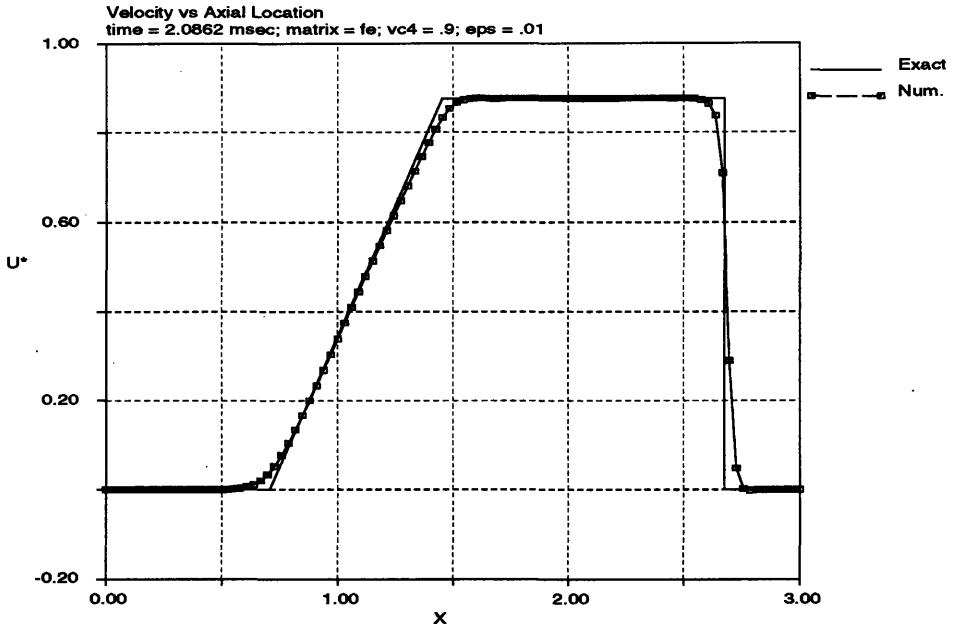


Figure 4.20: Axial Velocity Distribution in Shock Tube for Case 3

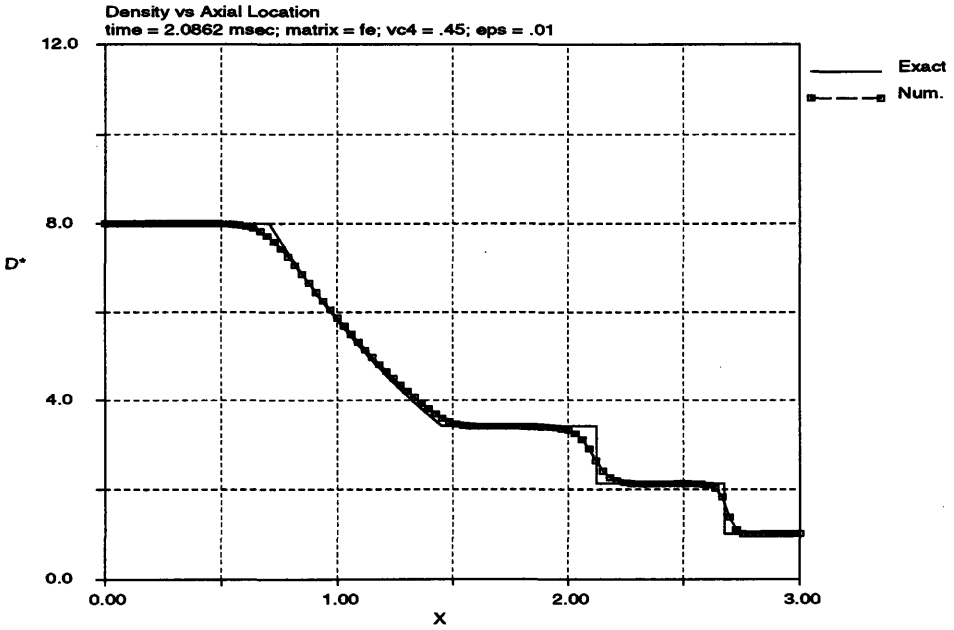


Figure 4.21: Density Distribution in Shock Tube for Case 4

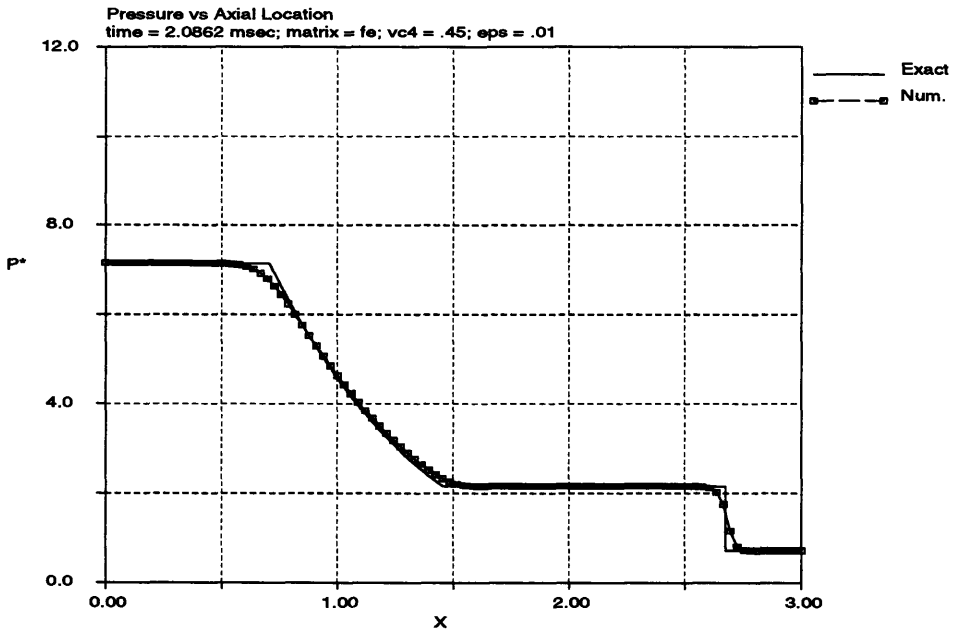


Figure 4.22: Pressure Distribution in Shock Tube for Case 4

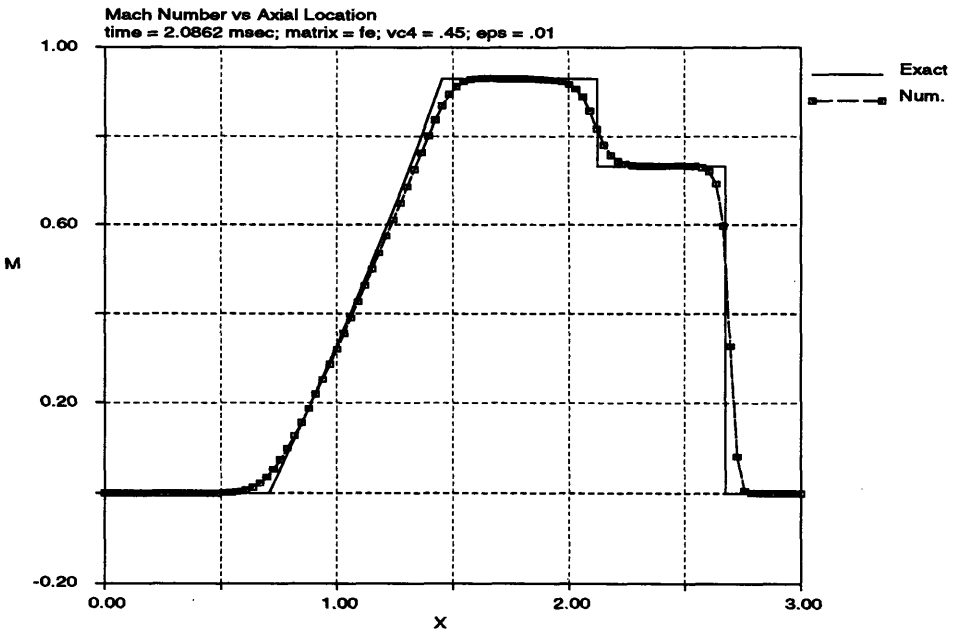


Figure 4.23: Mach Number Distribution in Shock Tube for Case 4

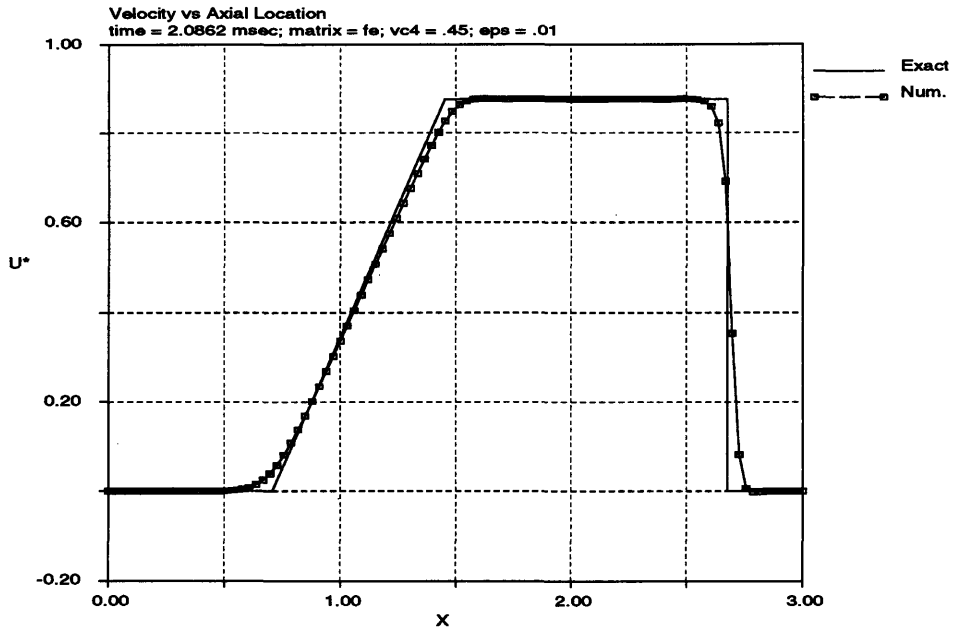


Figure 4.24: Axial Velocity Distribution in Shock Tube for Case 4

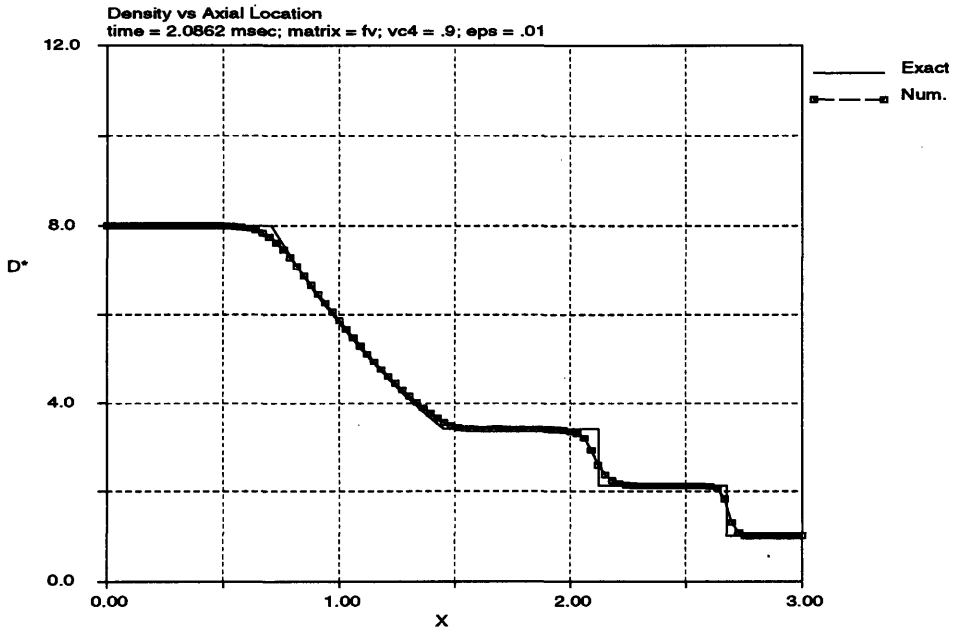


Figure 4.25: Density Distribution in Shock Tube for Case 5

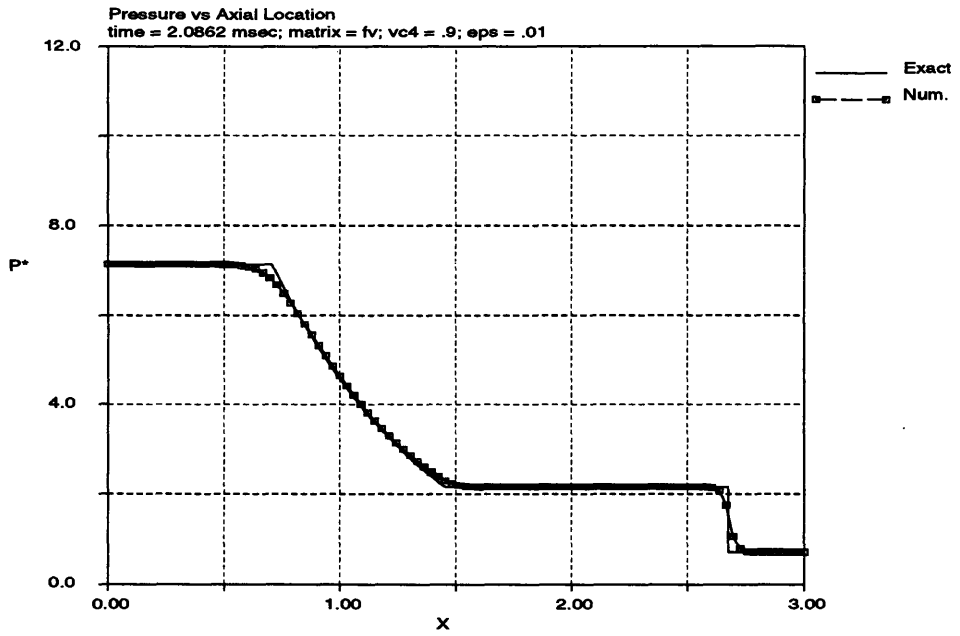


Figure 4.26: Pressure Distribution in Shock Tube for Case 5

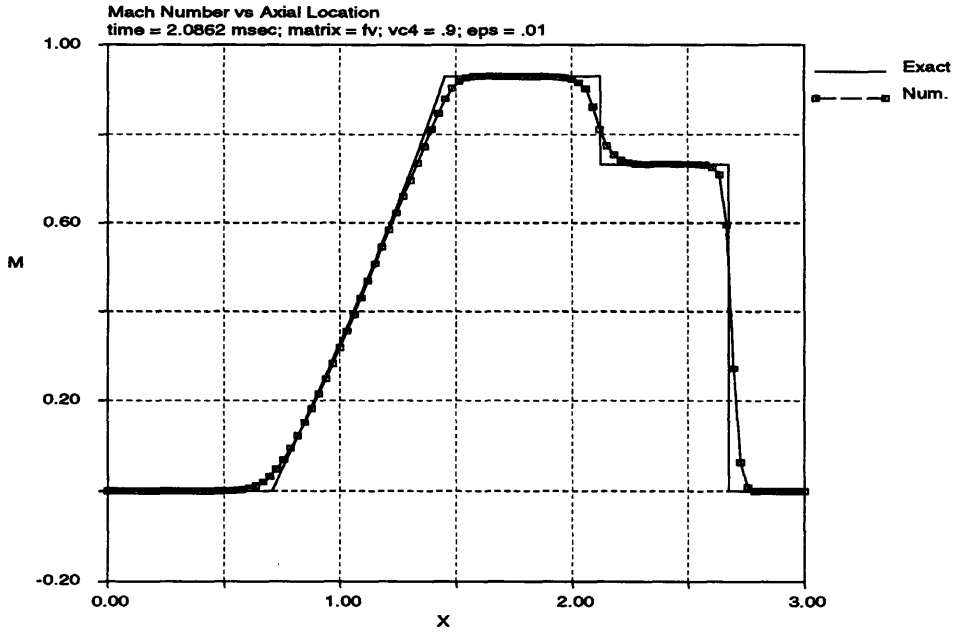


Figure 4.27: Mach Number Distribution in Shock Tube for Case 5

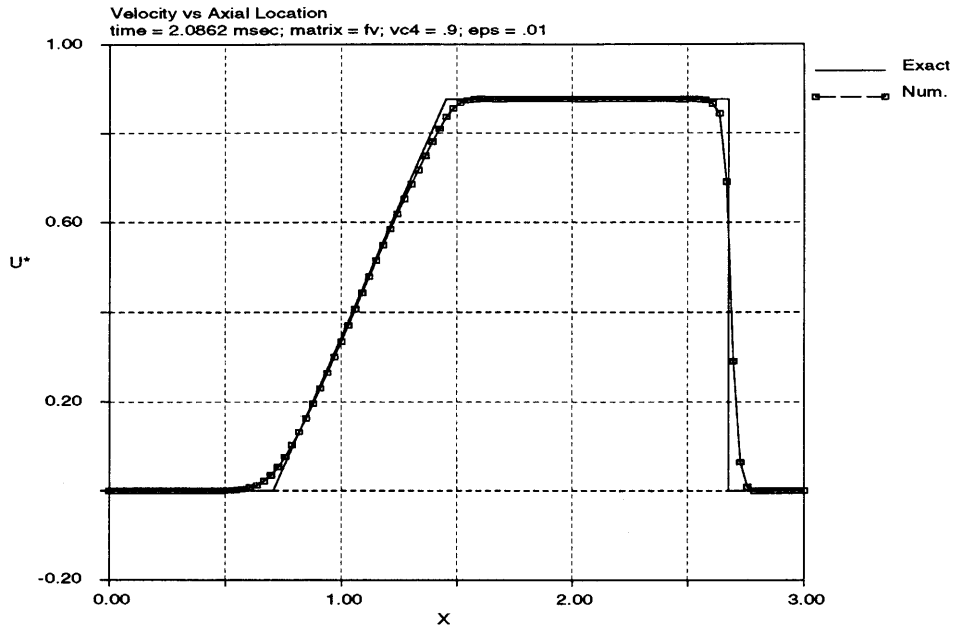


Figure 4.28: Axial Velocity Distribution in Shock Tube for Case 5

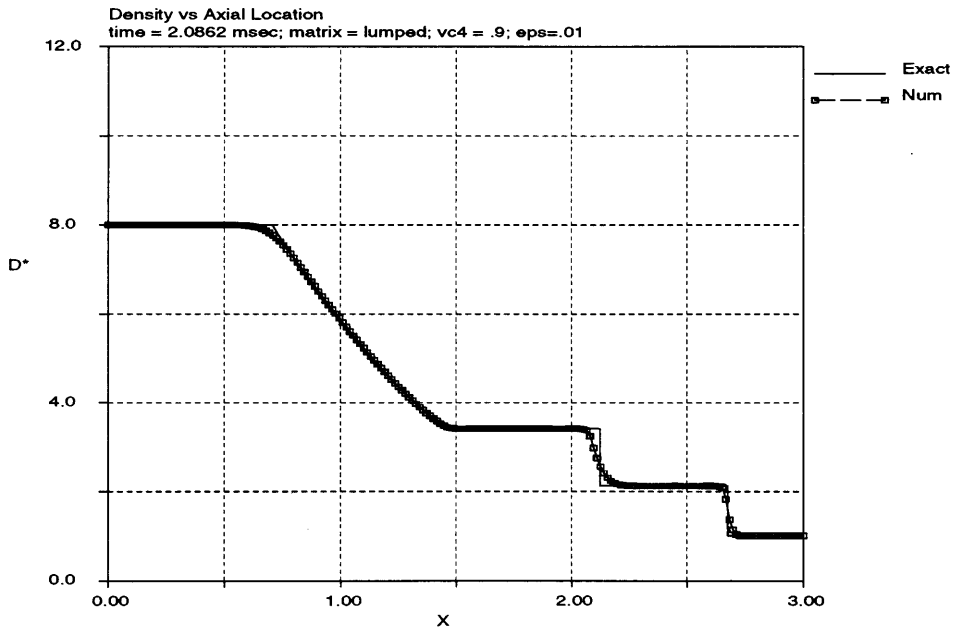


Figure 4.29: Density Distribution in Shock Tube for Case 6

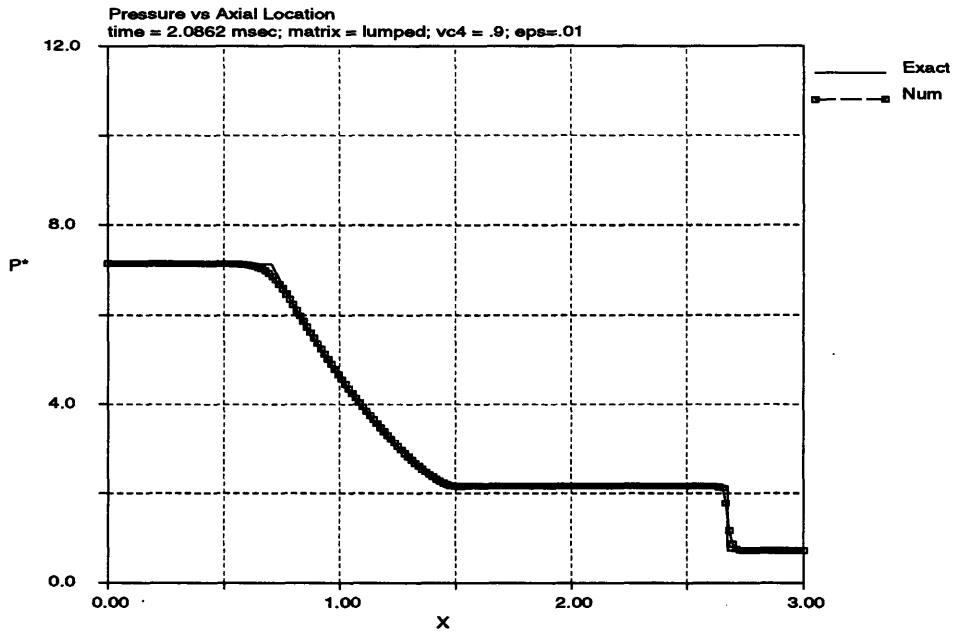


Figure 4.30: Pressure Distribution in Shock Tube for Case 6

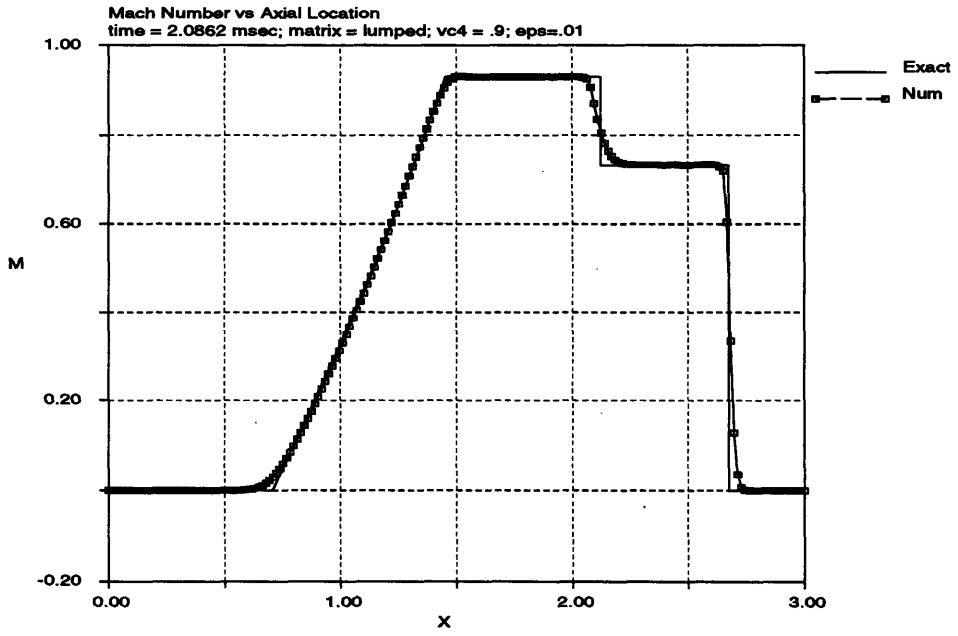


Figure 4.31: Mach Number Distribution in Shock Tube for Case 6

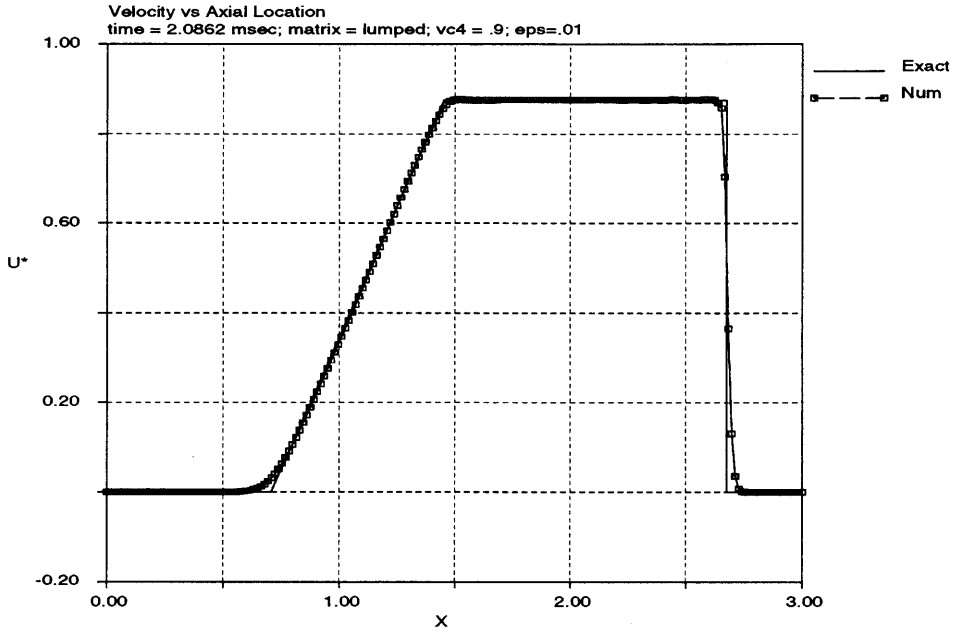


Figure 4.32: Axial Velocity Distribution in Shock Tube for Case 6

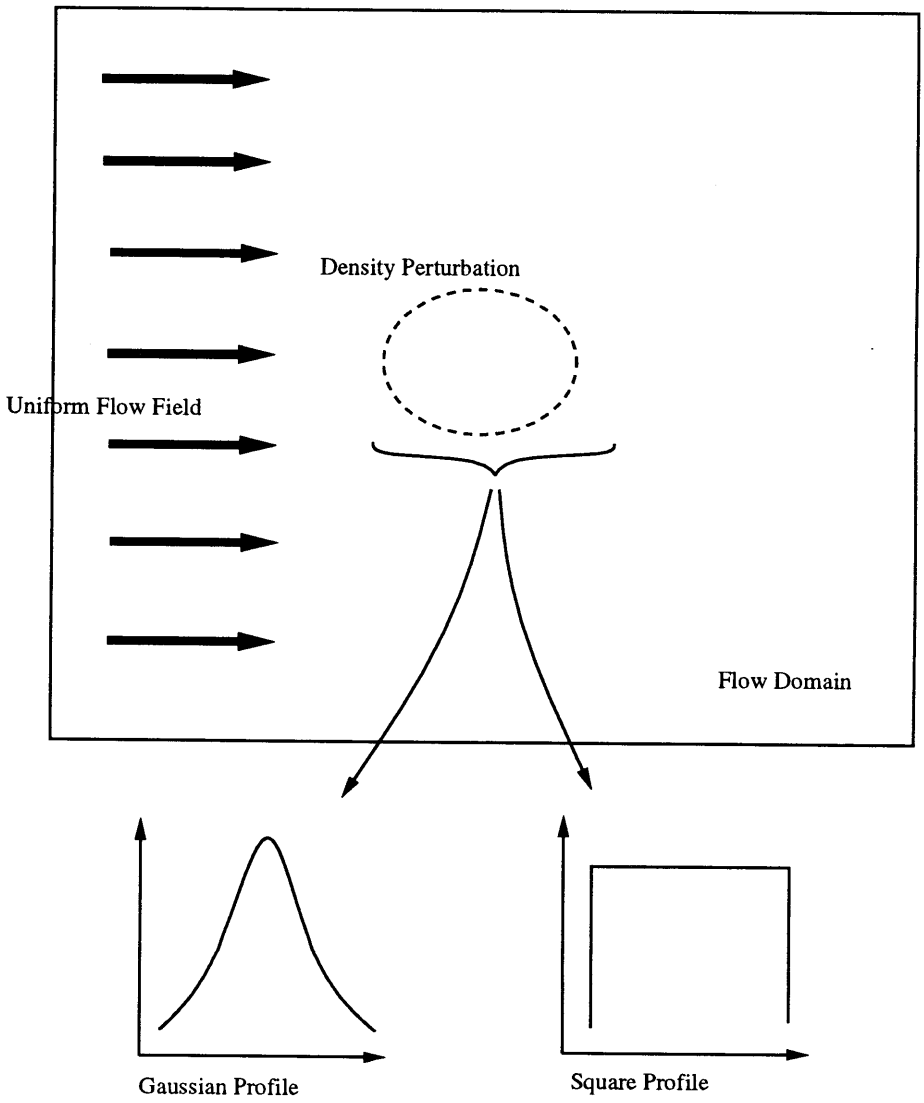


Figure 4.33: Schematic of Density Perturbation Test Case

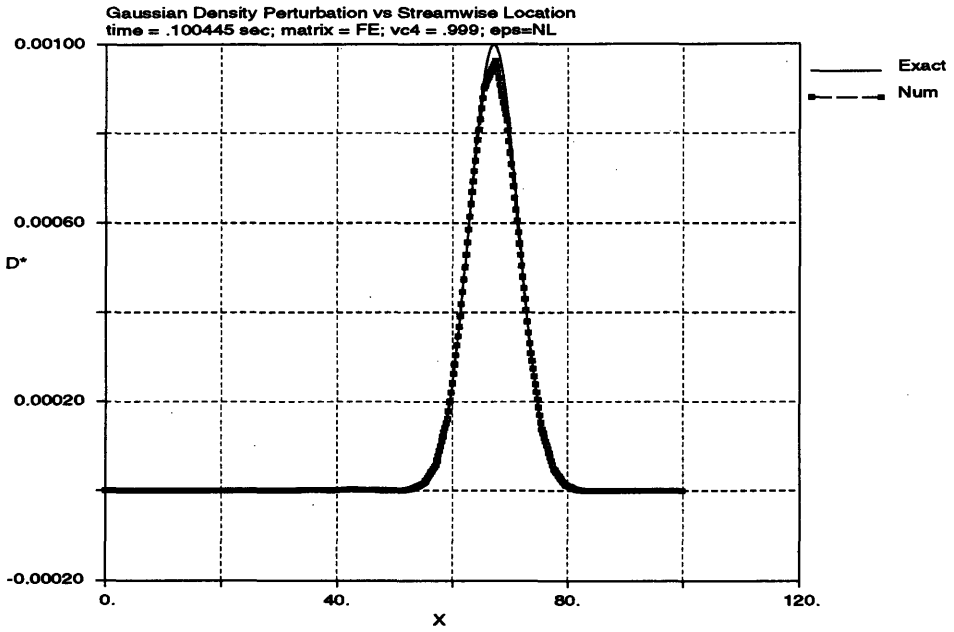


Figure 4.34: Streamwise Density Perturbation for Case 1

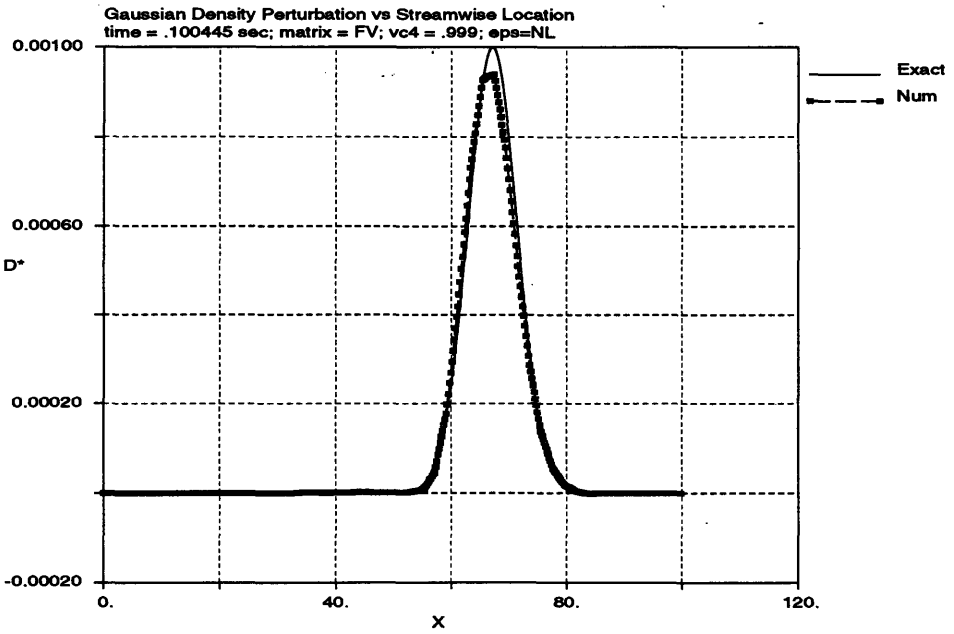


Figure 4.35: Streamwise Density Perturbation for Case 2

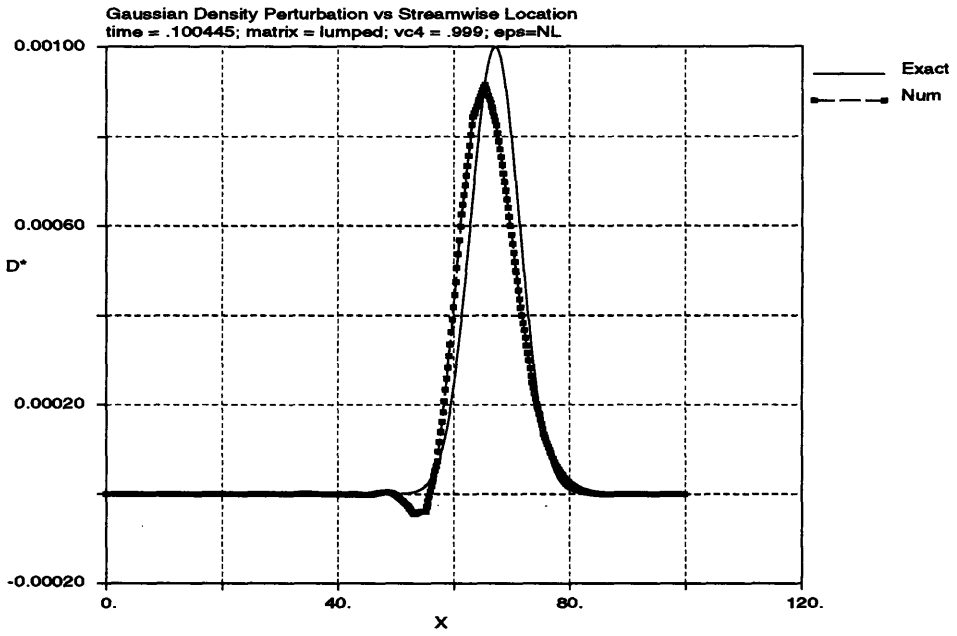


Figure 4.36: Streamwise Density Perturbation for Case 3

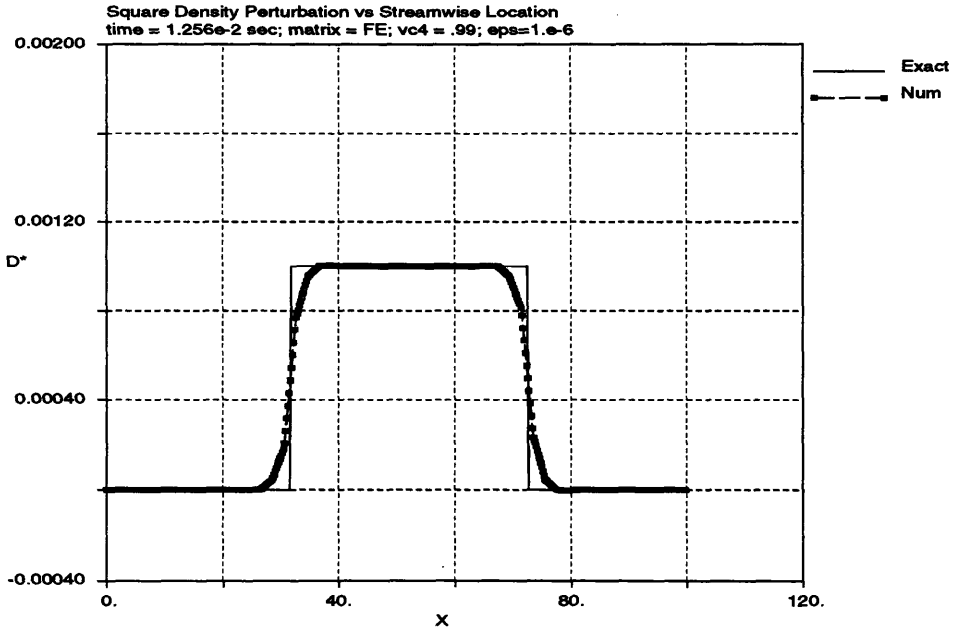


Figure 4.37: Streamwise Density Perturbation for Case 4

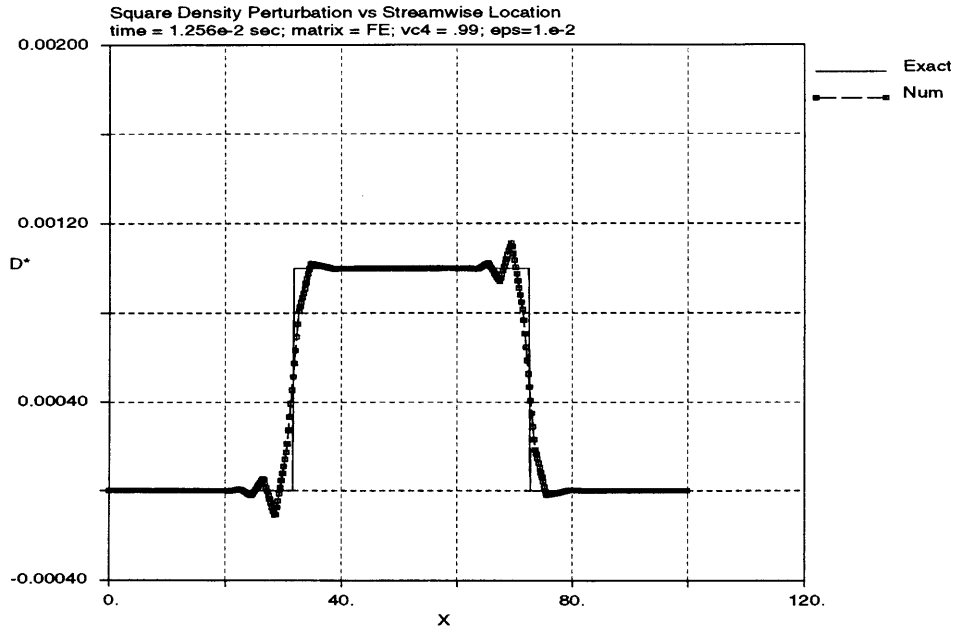


Figure 4.38: Streamwise Density Perturbation for Case 5

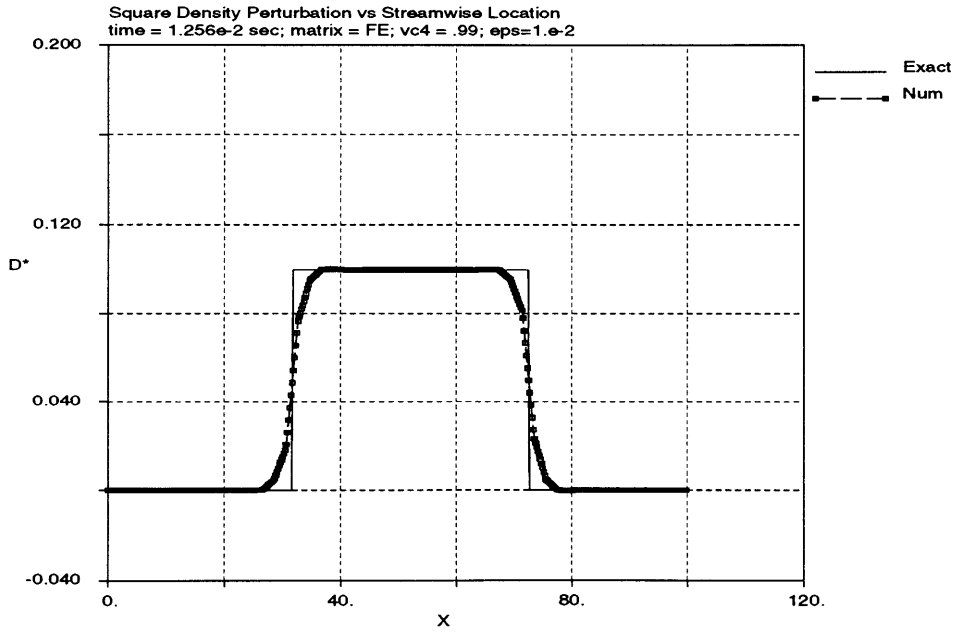


Figure 4.39: Streamwise Density Perturbation for Case 6

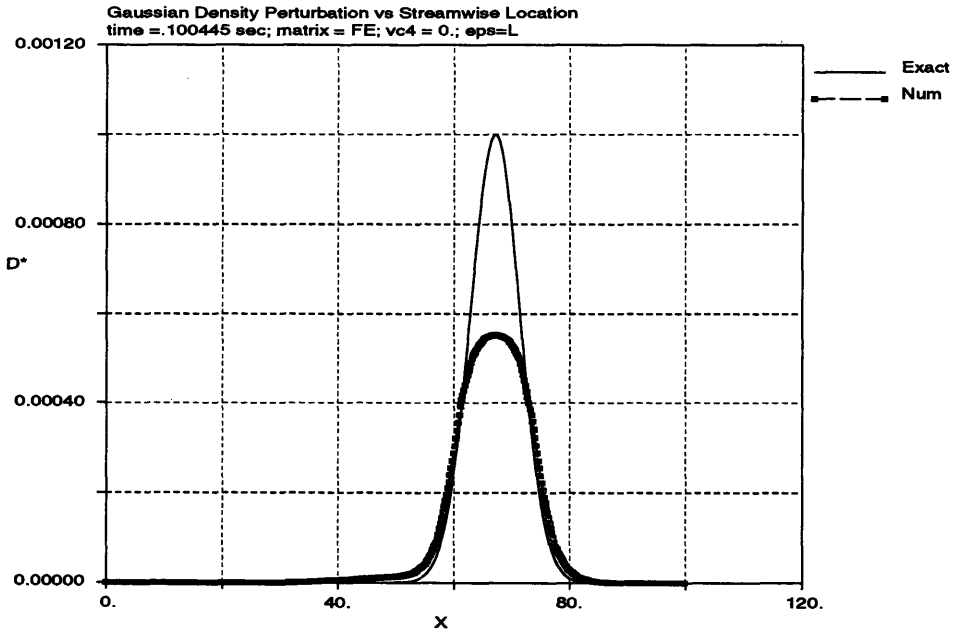


Figure 4.40: Streamwise Density Perturbation for Case 7

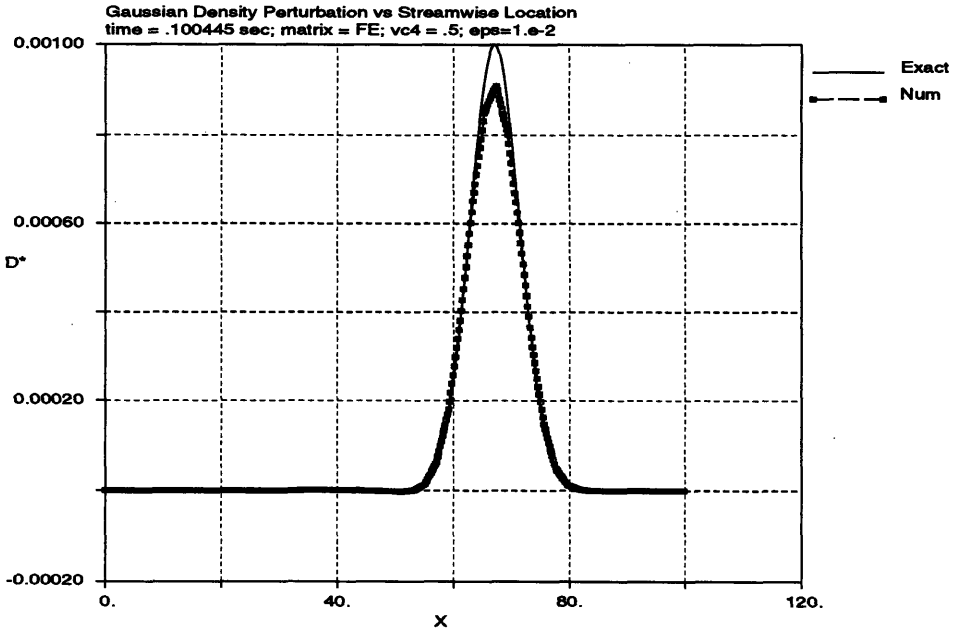


Figure 4.41: Streamwise Density Perturbation for Case 8

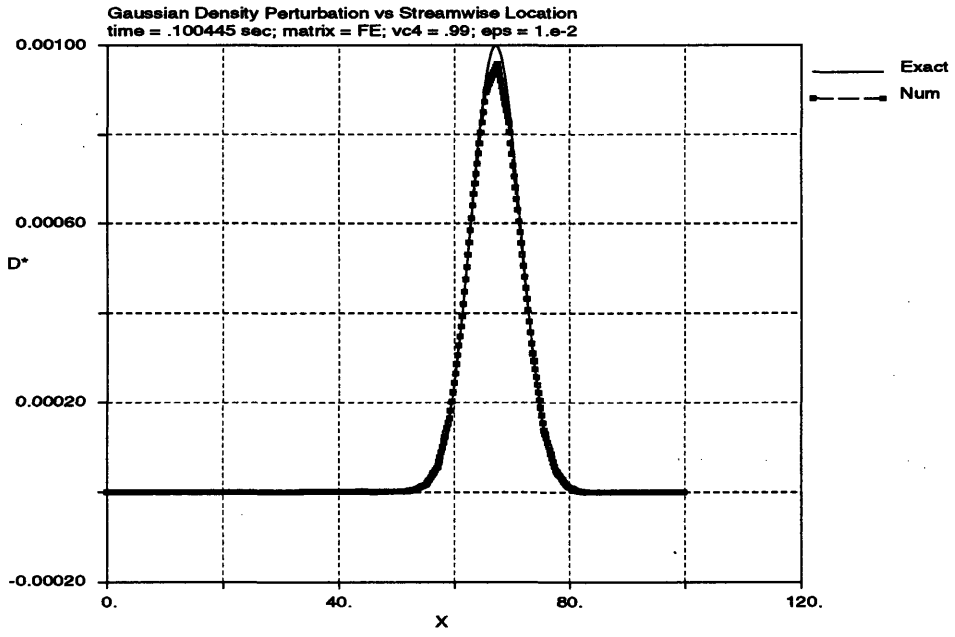


Figure 4.42: Streamwise Density Perturbation for Case 9

Chapter 5
Compressor Simulation

5.1 Background

5.1.1 Compressor

The compressor modeled in the numerical simulation is a two-stage axial fan designed by General Electric Aircraft Engines. This is a low aspect ratio fan used primarily for research such as flowfield prediction. The data for the first stage blades, provided by the Compressor Research Facility (CRF) located at Wright Labs, is actually the location of sensors at various cross sections of the blade and is used to reconstruct the blade shape. There are 16 blades in the first stage rotor. The design speed of the compressor is 13,288 rpm.

5.1.2 Experimental Setup

Figure 5.1 gives a diagram of the experimental setup for the first stage rotor of the compressor rig. A distortion duct is placed at 0.3834 m ahead of the leading edge at 85 percent span. The duct provides an angularly periodic total pressure pattern and is used for a forced response compressor test.

5.2 Numerical Simulation

The numerical simulation is two dimensional. The blade profile at 85 percent span of the blade is used for the simulation. The span is the length of the blade from the root or hub to the blade tip as illustrated in figure 5.1. This location is chosen since the streamline deviation from the leading to the trailing edge of the blade is less than one percent as measured in the experiment. Therefore, the flow in this region can be modeled as two dimensional.

The compressible Euler equations are used to model the flow with the computations performed in the rotating frame of reference, i.e. moving with the compressor. The two dimensional Euler equations are still valid in the rotating frame of reference.

The Euler equations are non-dimensionalized using inlet stagnation quantities in the stationary frame of reference. These quantities are the stagnation speed of sound, a_t, the stagnation density, ρ_t and a reference length of unity. The stagnation density is determined from the specified inlet axial stagnation pressure and the stagnation speed of sound. Each value is non-dimensionalized as discussed in Appendix E. Therefore, the non-dimensional, axial inlet stagnation quantities are:

$$P_t^* = \frac{1}{\gamma} \, , \; \rho_t^* = 1$$

The coefficient of pressure, C_p, is based on the axial stagnation pressure, stagnation density and stagnation speed of sound as follows:

$$C_p = \frac{P - P_t}{\frac{1}{2}\rho_t a_t^2}$$

Therefore, using the non-dimensional parameters this becomes:

$$C_p = 2\,(P^* - P_t^*)$$

5.2.1 Blade and Flow Domain Representation

The flow domain is represented with an unstructured mesh created by the mesh generation process outlined in Chapter 2. Elements are clustered around the leading and trailing edges of the blade to resolve the blade profile in these areas. A periodic boundary is constructed in the center of the flow passage between successive blades, with the blade lying in the center of the mesh. The flow passage between the blades is refined to help resolve flow features within the passage. The inlet boundary is placed at 0.5 m ahead of the leading edge of the blade. Likewise, the exit boundary is placed at 0.5 m behind the trailing edge of the blade. The mesh is shown in figure 2.14. Multiple blade passages are created by patching replicas of the mesh together (see figure 2.16).

The profile of the blade is extracted from the three dimensional blade description using linear interpolation between the data points provided. Unfortunately, the data does not contain points on the leading and trailing edges of the blade. Therefore, these curves of the blade are created using the splining technique outlined in Chapter 2 to create a curve segment for these regions. This results in slightly extending the profile as illustrated in figure 5.2.

5.2.2 Boundary Conditions

Three types of boundary conditions are used in the numerical simulation: prescribed back pressure, prescribed axial total pressure and periodicity. Figure 5.4 shows the boundaries of the flow domain associated with these conditions and Appendix C describes the application of each of these boundary conditions in the numerical algorithm.

At the inlet, the axial total pressure is prescribed. This is chosen to numerically represent the conditions of the experiment. In the experiment, the compressor draws mass flow from a reservoir through an inlet. This reservoir is typically the atmosphere and the flow ingested into the compressor is usually an axially aligned flow. Therefore, the axial total pressure measured by a pitot probe is the atmospheric pressure. By specifying the total axial flow conditions at the inlet boundary, the compressor in the numerical simulation is allowed to draw in mass flow. Since the flow calculations are performed in the rotating frame of reference, the compressor speed is also specified at the inlet boundary.

The static pressure is specified at the exit boundary of the flow domain. This boundary condition acts as a throttle to control the amount of mass flow the compressor draws into the domain.

The periodic boundary conditions are specified at the upper and lower boundaries of the domain. These conditions are used to enforce periodicity in the flow solution. Therefore, only one blade passage is needed for the numerical simulations that use a uniform axial total pressure at the inlet.

5.2.3 Startup

The compressor is started from rest and its rotational speed slowly increased until the final operating speed is reached. This approach is taken because starting the compressor impulsively at the operating speed results in leading edge stall. Therefore, slowly increasing the speed of the compressor gives the flow time to react to the conditions and allows the compressor time to draw in more mass flow from the inlet boundary. Also, a starting axial flow velocity is specified to give an initial mass flow, helping to reduce the tendency of the compressor to stall.

The back pressure is slowly increased as the compressor speed is increased. This gives the flow time to establish itself within the blade passage. Otherwise, by instantaneously setting the back pressure at the operating value, a large pressure wave propagates upstream. If the flow is not established, this pressure wave decreases the axial momentum to a degree that the compressor blade stalls.

5.2.4 Clean Flow Simulations

The clean flow simulations have a constant axial total pressure at the inlet. These simulations are carried out using two compressor speeds: 67 percent and 85 percent corrected design speed. Corrected speed, used to account for changes in atmospheric conditions, is defined as:

$$N_c = \frac{N}{\sqrt{\theta}}$$

where θ is the square root of the ratio of the total inlet temperature and a reference temperature (usually the standard atmospheric temperature) as follows:

$$N_c = \sqrt{\frac{T}{T_{ref}}}$$

The reference conditions for the compressor simulations are found in table 5.1. One blade passage was simulated numerically using periodic boundary conditions with a series of increasing back pressures.

The flow solver uses local time stepping to accelerate convergence of the solution to steady state. Also, the lumped mass matrix version of the solver is used since time accuracy is not an issue in this simulation.

67 Percent Corrected Design Speed

The inlet conditions for these simulations are given in table 5.2.

Results

Back Pressure = 1.185 atm This simulation results in a choked flow within the blade passage. The Mach number distribution along the blade is shown in figure 5.5. A shock appears near the trailing edge of the blade on the suction surface and crosses the passage, striking the pressure surface of the blade above it approximately midway along the blade. The shock strength on the pressure surface of the blade is weaker than on the suction surface as seen from the lower Mach number ratio. The Mach contours are shown in figure 5.17. This is a passage shock and not a bow shock that has been ingested into the passage. Since the relative Mach number (i.e. the Mach number in the rotating frame of reference) is 1.045, this creates a bow shock ahead of the upper blade that strikes the suction surface at approximately one third the chord from the leading edge. The Mach number profile shows this weaker shock causes the flow to become subsonic. This confirms that the shock at the trailing edge is definitely a passage shock caused by the flow becoming critical within the passage for the specified back pressure. The leading and trailing edges of the blades exhibit sharp gradients in the Mach number distribution. Note that the Mach number near the leading edge does not reach zero at the stagnation point, but does attain a very small value.

The C_p distribution along the blade is shown in figure 5.11. The shock is clearly shown on the suction surface near the trailing edge of the blade and on the pressure surface about midway along the chord. The bow shock is also clearly visible in the figure. The large gradients shown at the leading and trailing edges are due to the inviscid approximation of

the fluid mechanics. The numerical algorithm appears to adequately represent the shock structures with no oscillations evident in the vicinity of the shocks.

The convergence history of the numerical simulation is shown in figure 5.19. This plot shows the natural log of the maximum density residual (normalized by the reference density) over the entire mesh versus the iteration. The numerical scheme exhibits good convergence characteristics for this simulation. The mass flow at the inlet boundary is shown in figure 5.20. The sharp drop in mass flow at iteration 1250 is caused by the pressure wave from the back pressure boundary condition reaching the inlet boundary.

Back Pressure = 1.25 atm The back pressure is increased to 1.25 atm in an attempt to remove the passage shock. The simulation is started from the previously converged solution for 1.185 atm. The Mach number distribution on the blade is shown in figure 5.6. The figure shows the passage shock moving upstream within the passage and is weaker than the 1.185 atm simulation. The bow shock ahead of the blade does not appear to move or change in strength.

The C_p distribution is shown in figure 5.12 and correlates with the Mach number distribution, showing the passage shock moving forward and the bow shock remaining the same. The mass flow does not change from the 1.185 atm back pressure specification, signifying that the compressor is choked.

Back Pressure = 1.30, 1.301, 1.31 atm The back pressure is increased to 1.30, 1.301 and 1.31 atm. The Mach number distributions, shown in figures 5.7, 5.8 and 5.9, illustrate the passage shock continually moving forward and weakening within the passage. Again, the bow shock remains in the same position with the same strength.

The C_p distributions are shown in figures 5.13, 5.14 and 5.15. These figures reflect the same behavior as the Mach number figures. The mass flow remains the same for these simulations (a choked condition), but the total pressure ratio increases as expected.

Back Pressure = 1.33 atm The flow is unchoked at this back pressure, as seen from the Mach number distribution in figure 5.10. The bow shock increases in strength and moves slightly forward within the passage. This is also reflected in the C_p distribution of figure 5.16.

The mass flow decreases at this back pressure signaling the compressor is at an unchoked condition. Figure 5.18 shows a black and white density contour of the compressor with the high density areas appearing dark and the low density areas lighter. The expansion around the leading edge of the blade is apparent with the density increasing along the blade passage.

Comparison to Euler Turbine Equation The Euler turbine equation for a two dimensional analysis states [13]:

$$\Delta H_t = \omega r \Delta v$$

where ΔH_t is the change in stagnation enthalpy; ωr is the speed of the compressor; and Δv is the change in tangential velocity. A comparison of the flow solution to the Euler turbine equation is used to validate the solver. The non-dimensional compressor speed is $\omega r = .835701$. Table 5.3 shows the average measured values of Δv and ΔH_{t_n} from the inlet boundary to the exit obtained from the numerical simulation. The value of the numerical stagnation enthalpy change is compared with the Euler turbine equation for the calculated Δv. The numerical simulation agrees with the Euler turbine equation for the change in stagnation enthalpy. This gives added confidence to the numerical simulations results.

85 Percent Corrected Design Speed

The inlet conditions for these simulations are given in table 5.4.

Results The results are very similar to the 67 percent corrected design speed simulations, therefore not much detail will be spent on describing the similar flow behavior. The back pressures used are: 1.205, 1.25, 1.275, 1.30, 1.35 and 1.40 atm. The Mach number distributions on the blade are shown in figures 5.21, 5.22, 5.23, 5.24, 5.25 and 5.26. The C_p distributions are shown in figures 5.27, 5.28, 5.29, 5.30, 5.31 and 5.32.

The compressor is choked for all the back pressures with the appearance of a strong passage shock. The difference between this simulation and the 67 percent simulation is the formation of a stronger bow shock ahead of the blade because of the higher relative Mach number of 1.456. This shock is ingested into the blade passage and reflects off the suction surface ahead of the passage shock. This reflected shock interacts with the passage shock as seen in figure 5.33 showing the Mach contours. Note the reflected shock does not pass through the passage shock, but forms a λ type structure with the passage shock.

The convergence history for the 1.205 atm back pressure simulation shows poor convergence as illustrated in figure 5.35. The density residual decreases only two orders of magnitude from the initial residual. A possible reason for this behavior is the location of the passage shock at the trailing edge of the blade as seen from the Mach number and C_p distributions in figures 5.21 and 5.27 respectively. This region of the blade is ill-defined for an inviscid flow calculation since it is blunt. The presence of the shock can cause instabilities which do not allow the numerical scheme to converge satisfactorily.

When the back pressure is increased to 1.40 atm, the shock moves into the passage. The convergence history of this simulation is likewise shown in figure 5.35 and shows that the

density residual converges four orders of magnitude from the initial residual. This result supports the explanation given for the poor convergence.

The inlet mass flow history is shown in figure 5.36. A sharp decrease in mass flow is again evident as the pressure wave from the exit boundary condition reaches the inlet.

Comparison to Euler Turbine Equation The non-dimensional compressor speed is $\omega r = 1.08256$. Table 5.5 shows the average measured values of Δv and ΔH_{t_n} from the inlet to the exit boundary obtained from the numerical simulation. Again, the numerical simulation agrees with the Euler turbine equation.

5.2.5 Distortion Simulation at 67 Percent Corrected Design Speed

This simulation is performed to replicate a forced response test in the experiment. The inlet boundary is farther away from the leading edge than the distortion duct in the experiment. This is to ensure that the bow shock from the compressor does not interfere with the inlet boundary conditions.

The distortion pattern is an axial total pressure variation that is periodic over two blade passages. Because this pattern is fixed in the stationary frame of reference, the blades pass through the distortion pattern in the relative frame. The pattern is shown in figure 5.37 as a function of the angle, θ. This function is:

$$p_t = p_{t_{max}} \left(1 - \frac{1}{2}\Delta p_t(1 - cos(N\theta(t)))\right)$$

where $p_{t_{max}}$ is the maximum axial total pressure; Δp_t is the magnitude of the axial total pressure deficit as a percentage of the maximum axial total pressure; N is the number of distortion patterns per revolution; θ is the angular location of the boundary point as a function of time; and t is the elapsed time. (Since the calculations are performed in the relative frame it is consistent to think of the inlet boundary as moving at the compressor speed. Therefore, the position of the distortion at the inlet boundary is a function of time, i.e. $\theta(t)$.)

Two blade passages are needed for the simulation. The converged, unchoked, steady state solution for the 67 percent corrected design speed solution at a back pressure of 1.33 atm is used as the start of the simulation. The flow solver uses the finite element mass matrix because of its proven unsteady capabilities in the previous chapters along with the minimum allowable time step on the mesh.

Inlet Conditions

The inlet conditions are the same as for the 67 percent clean flow simulations with $p_{t_{max}}$ replacing p_t in table 5.2. The magnitude of the distortion used is 10 percent (i.e. $\Delta p_t = .1$).

Results

Transient pressure distributions for the lower and upper blade (i.e. the lower blade is the blade that appears in the lower half of the domain) are shown in figures 5.38 and 5.39. The total force on the blades is calculated by linearly integrating the pressure distribution over the blade. This result is shown in figure 5.40 for both blades. The figure shows that the total force on the upper and lower blades are out of phase. This is expected from the inlet distortion since one blade experiences the total pressure deficit while the other experiences no deficit. Therefore, while the loading on one blade decreases the other blade experiences an increased loading.

The frequency of the distortion in the relative frame is the wheel speed divided by the length of the inlet since the distortion pattern is periodic over this length. This yields:

$$f = \frac{1.99}{.835699} = 3.357$$

This is non-dimensionalized by the stagnation speed of sound at the inlet which is 346.68 m/s. The dimensional frequency is then 1.1644 kHz. Measuring the frequency of the total force variation on the blades from the figures gives a frequency of approximately 3.333 or 1.1556 kHz. This is in good agreement and the deviation is most likely caused by the measurement from the plot. Stationary oscillations were not achieved due to the lack of time and computer resources during this simulation.

Total pressure contours in the relative frame at 1.152 msec into the simulation are shown in figure 5.41. This corresponds to the non-dimensional time of 64.545 in figure 5.40. At this time the lower blade is experiencing a decreased loading while the upper blade an increase. As shown in the contour plot, the total pressure contours on the suction surface of the upper blade cover a larger area behind the bow shock than the contours on the suction surface of the lower blade. These contours appear misleading in that the upper blade experiences a large total pressure loss. Measuring the values in those areas shows that the loss is greater on the lower blade than the upper blade.

The total pressure contours at the earlier time of .7358 msec into the simulation are shown in figure 5.42. This corresponds to the non-dimensional time of 64.440 in figure 5.40 which shows the lower blade's loading decreases and the upper blade increases. The total pressure contours show the opposite behavior discussed previously. The Mach contours are shown in figure 5.43.

5.3 Conclusions

The numerical algorithm developed in the previous chapters applies well to the compressor simulations performed in this chapter. The steady state simulations show excellent shock capturing capability with no oscillations apparent in the vicinity of the shocks. The convergence to steady state for the 67 percent corrected design speed simulations is very good, but not for some of the 85 percent corrected design speed simulations. One possible explanation for this behavior is the appearance of a passage shock at the trailing edge of the blade. The work performed by the compressor is measured from the numerical simulation and agrees with the Euler turbine equation. The numerical algorithm is able to effectively simulate the progress from choked to unchoked conditions in the 67 percent corrected design speed simulations.

A forced response test is simulated using a periodically varying axial total pressure at 67 percent corrected design speed. The numerical algorithm performs adequately in simulating these conditions. The frequency of the measured total force variation on the blades agrees well with the frequency of the distortion pattern. This simulation needs a longer run time to achieve a stationary oscillatory total force variation on the blades. This was not available at the time this simulation was performed. Also, experimental data was not available for comparison. Overall, the unstructured flow solver developed in the previous chapters can be used in the simulation of two dimensional compressor cascade flows for a variety of applications.

Table 5.1: Compressor Reference Conditions

Reference Pressure, p_{ref}	1 atm
Reference Temperature, T_{ref}	288.149 K

Table 5.2: Inlet Conditions for 67% Corrected Design Speed Simulation

Axial Total Pressure at Inlet, p_t	.968702 atm
Total Temperature at Inlet, T_t	299.45 K
Compressor Speed, ω	8733.33 rpm

Table 5.3: Comparison of the 67% Corrected Design Speed Simulation with the Euler Turbine Equation

Back Pressure	Δv	ΔH_{t_n}	$\Delta H_{t_{Euler}}$
1.185 atm	-0.4014	0.335	0.335
1.25 atm	-0.4238	0.354	0.354
1.30 atm	-0.4384	0.366	0.366
1.301 atm	-0.4388	0.367	0.367
1.31 atm	-0.4413	0.369	0.369
1.33 atm	-0.4596	0.384	0.384

Table 5.4: Inlet Conditions for 85% Corrected Design Speed Simulation

Axial Total Pressure at Inlet, p_t	.8151986 atm
Total Temperature at Inlet, T_t	293.4 K
Compressor Speed, ω	11193.29 rpm

Table 5.5: Comparison of 85% Corrected Design Speed Simulation with Euler Turbine Equation

Back Pressure	Δv	ΔH_{t_n}	$\Delta H_{t_{Euler}}$
1.205 atm	-0.5860	0.634	0.634
1.25 atm	-0.6230	0.674	0.674
1.275 atm	-0.6363	0.688	0.688
1.3 atm	-0.6481	0.702	0.702
1.35 atm	-0.6650	0.719	0.719
1.40 atm	-0.6816	0.738	0.738

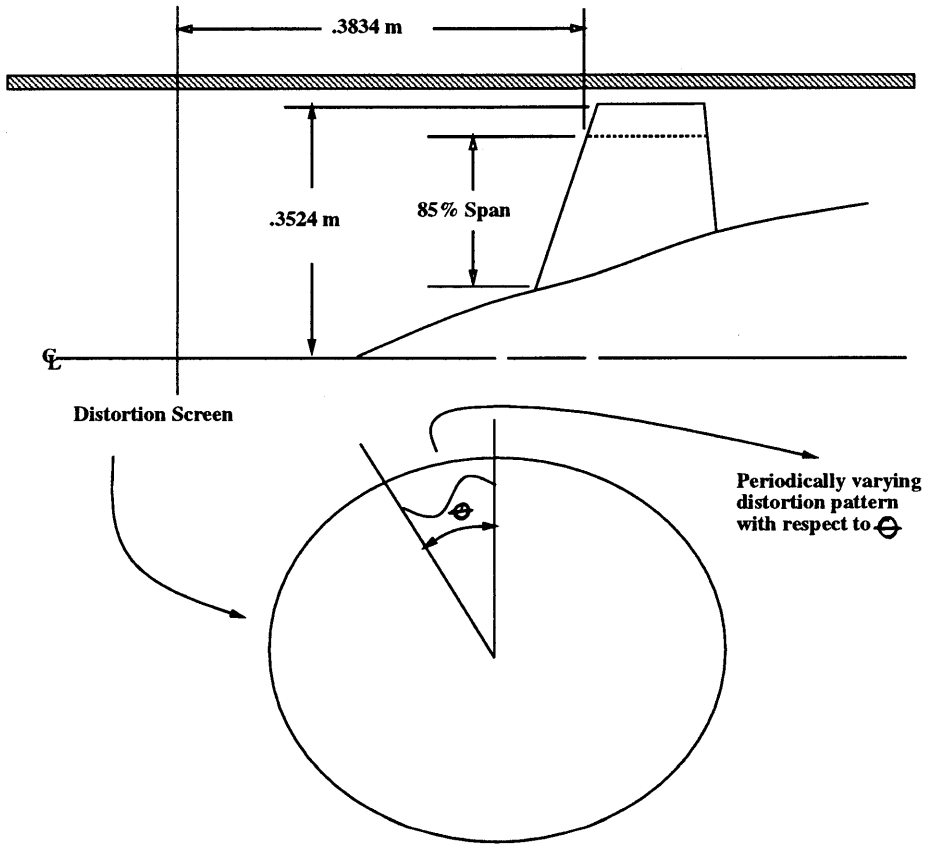


Figure 5.1: Experimental Setup

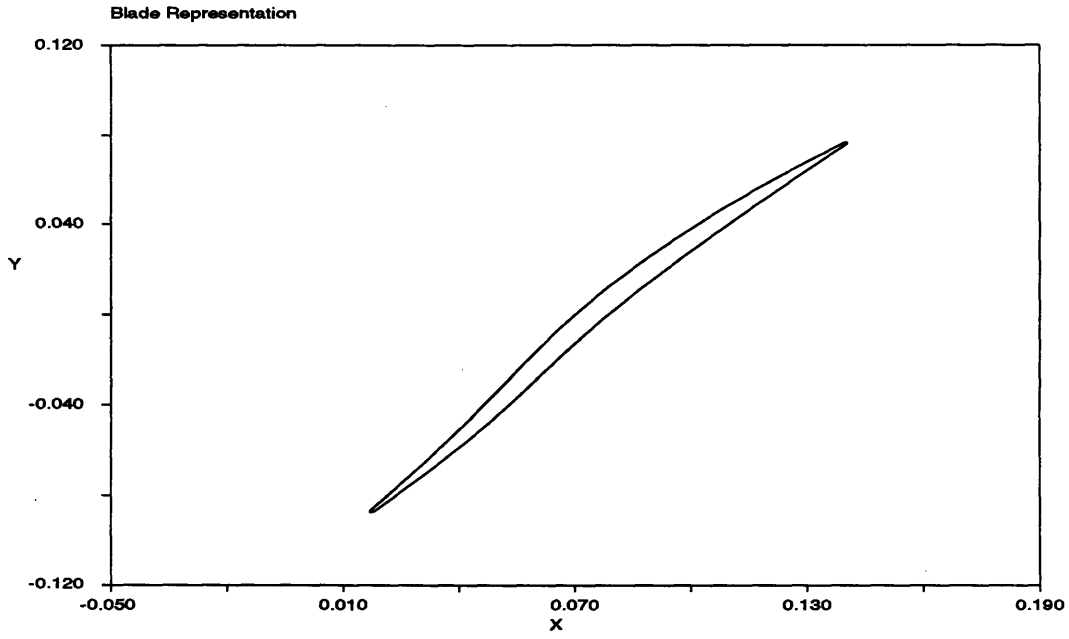


Figure 5.2: Blade Representation

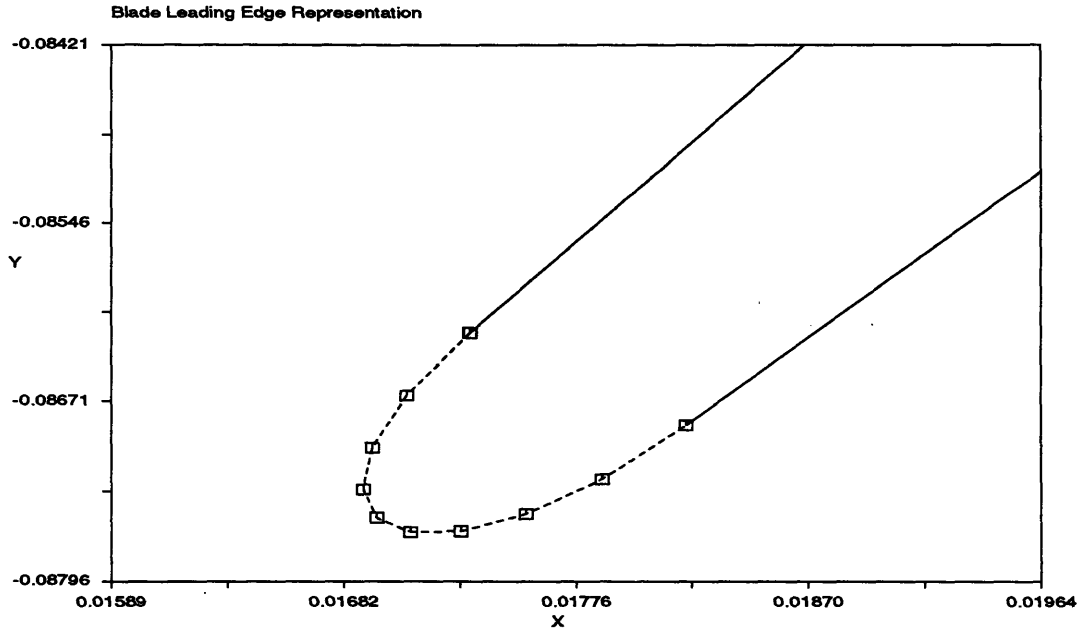


Figure 5.3: Blade Construction at Leading Edge

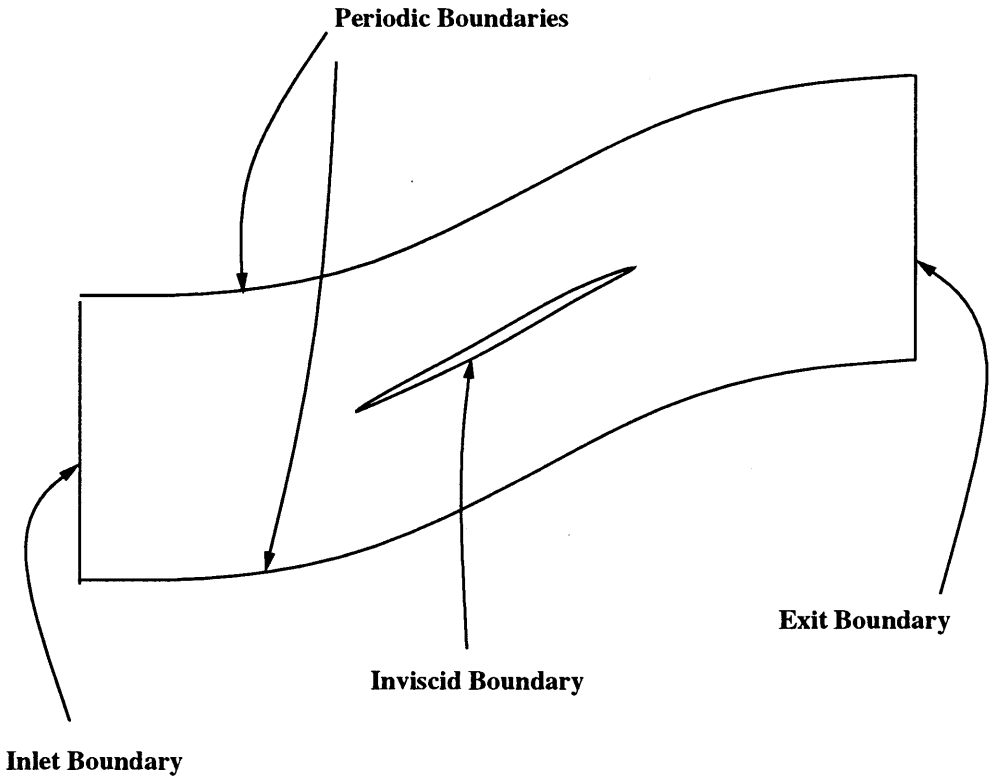


Figure 5.4: Boundary Conditions for Numerical Simulation

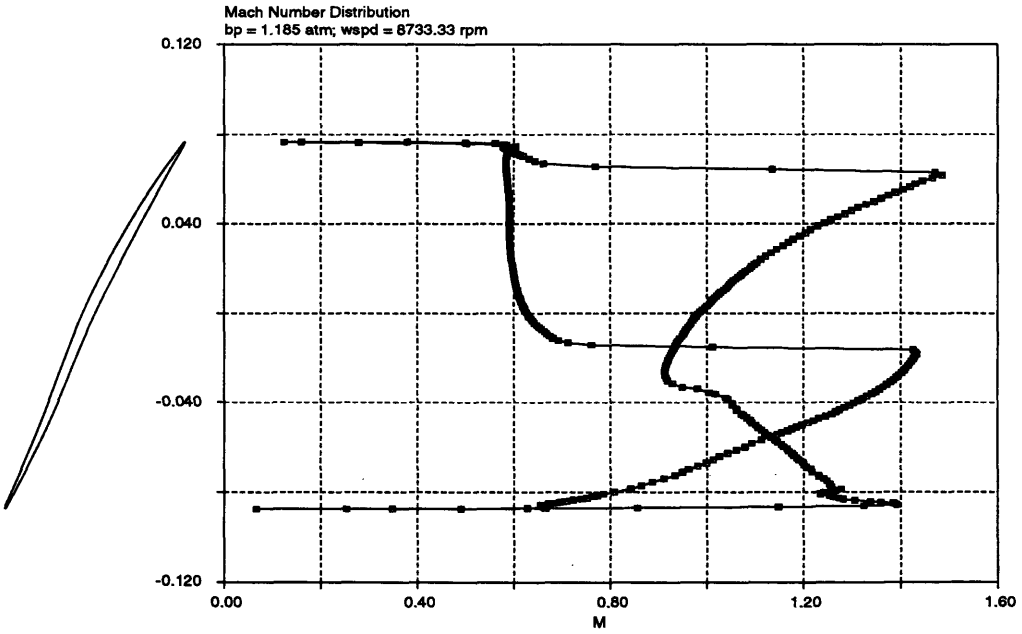


Figure 5.5: Mach Number Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.185 atm

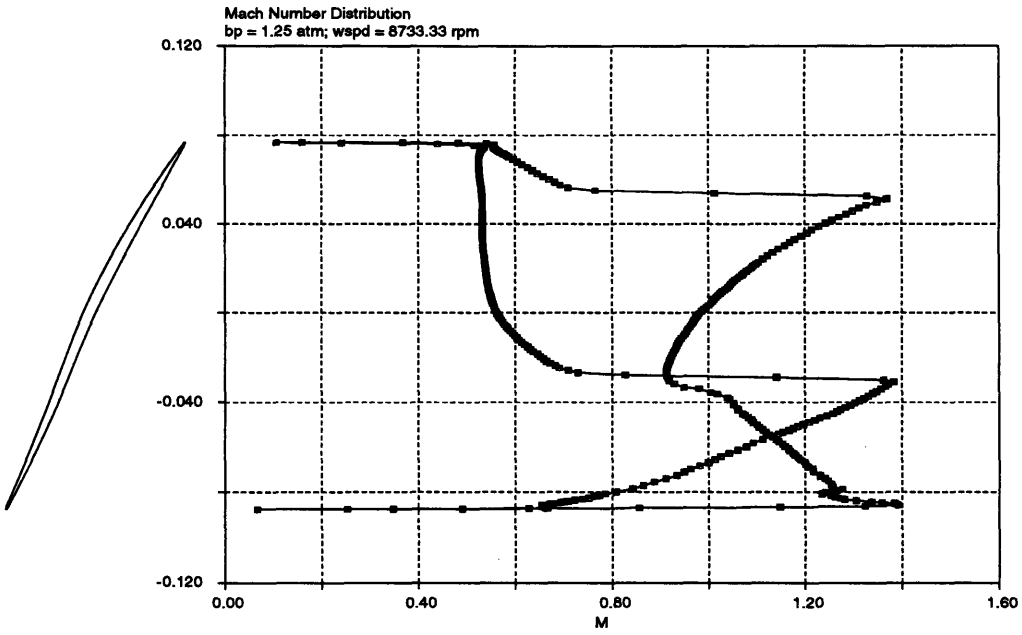


Figure 5.6: Mach Number Distribution on Blade for 67% N_c with Clean Inlet flow and Back Pressure of 1.25 atm

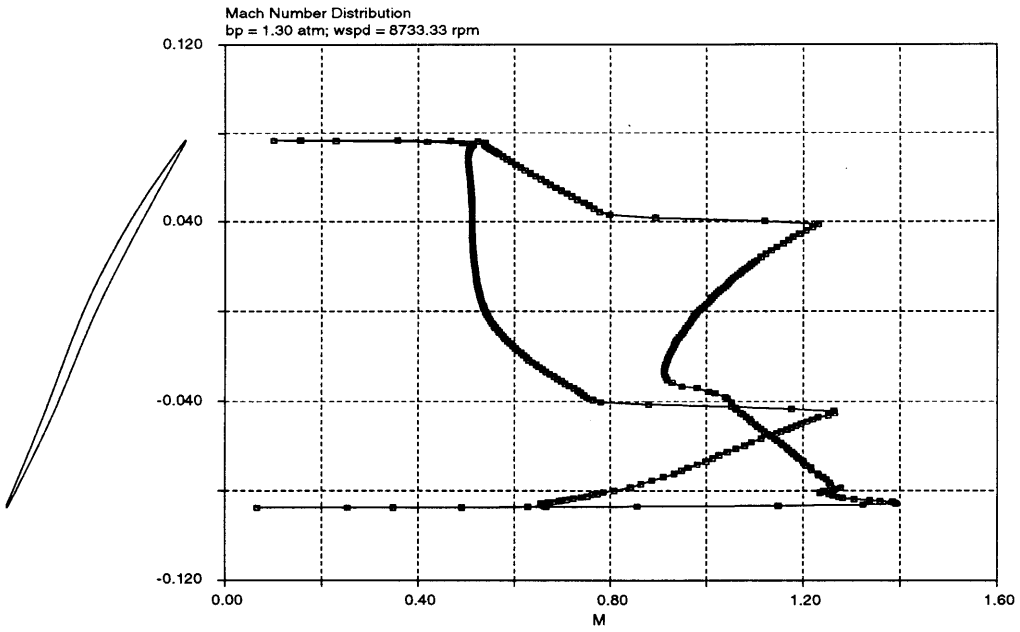


Figure 5.7: Mach Number Distribution on Blade for 67% N_c with Clean Inlet flow and Back Pressure of 1.30 atm

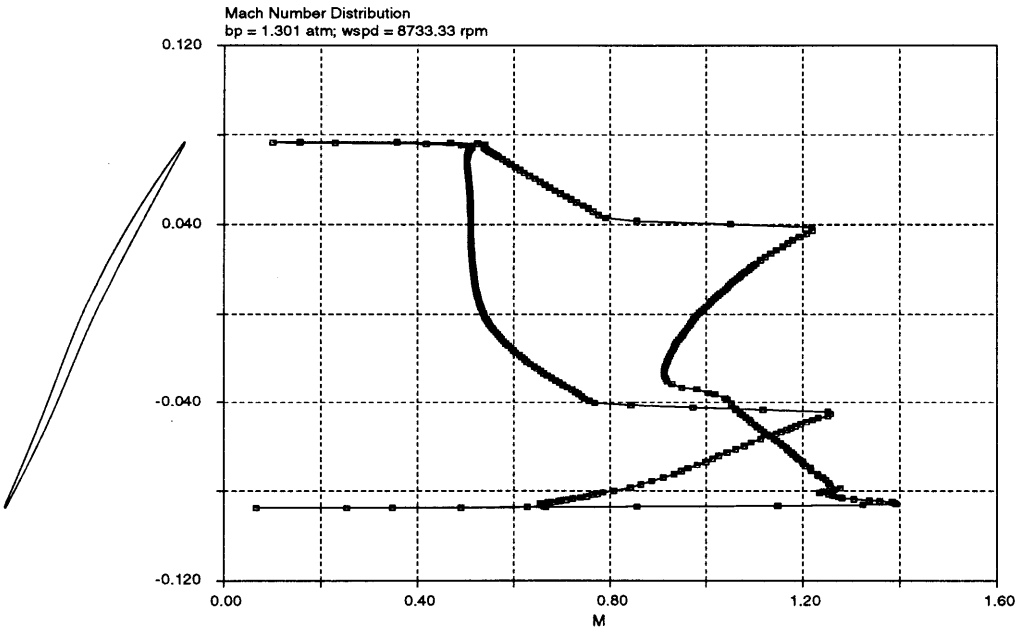


Figure 5.8: Mach Number Distribution on Blade for 67% N_c with Clean Inlet flow and Back Pressure of 1.301 atm

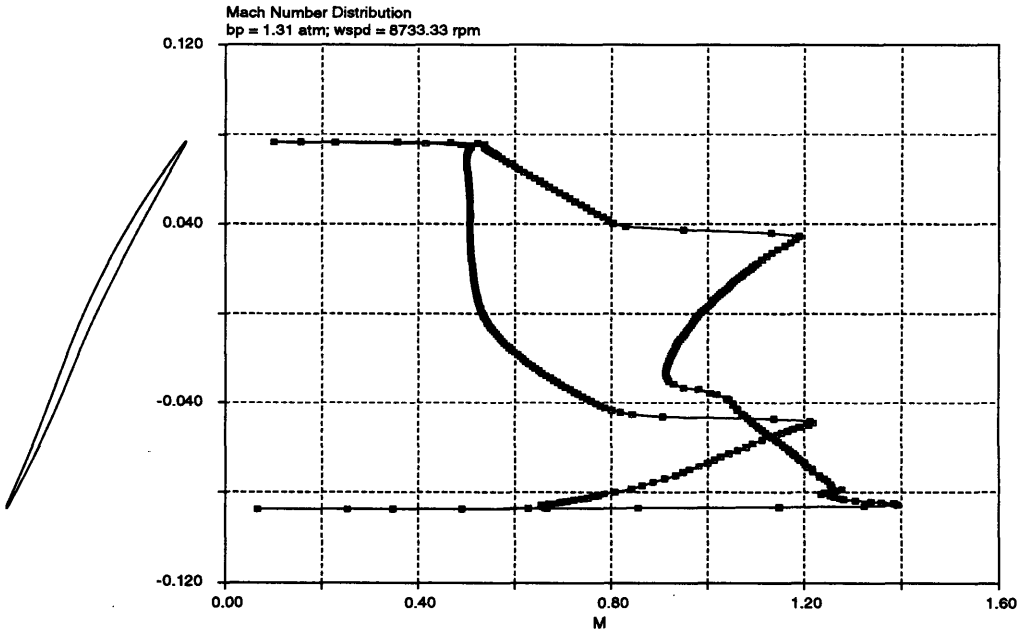


Figure 5.9: Mach Number Distribution on Blade for 67% N_c with Clean Inlet flow and Back Pressure of 1.31 atm

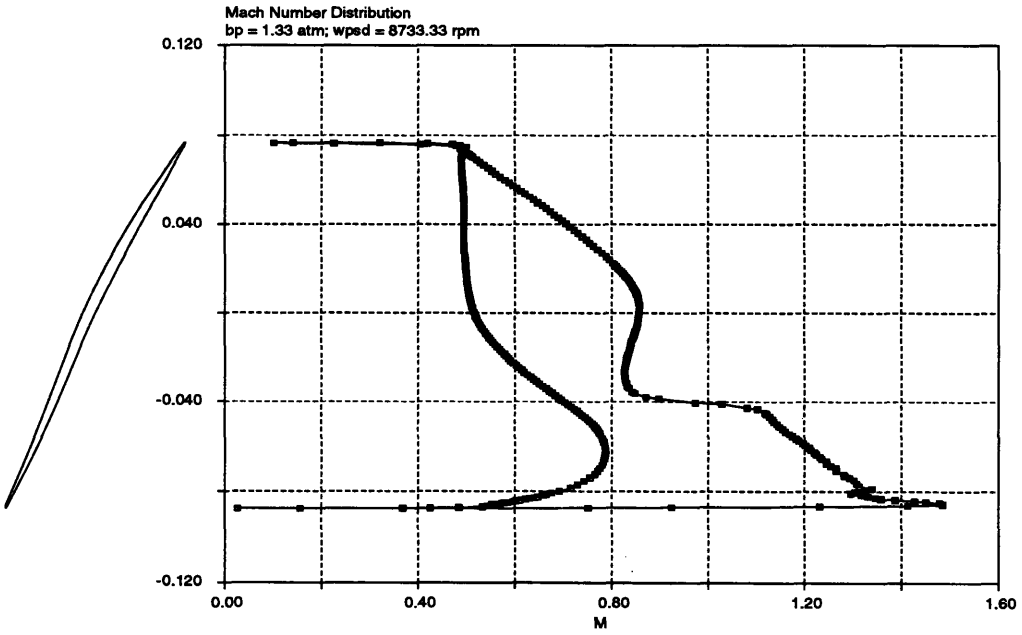


Figure 5.10: Mach Number Distribution on Blade for 67% N_c with Clean Inlet flow and Back Pressure of 1.33 atm

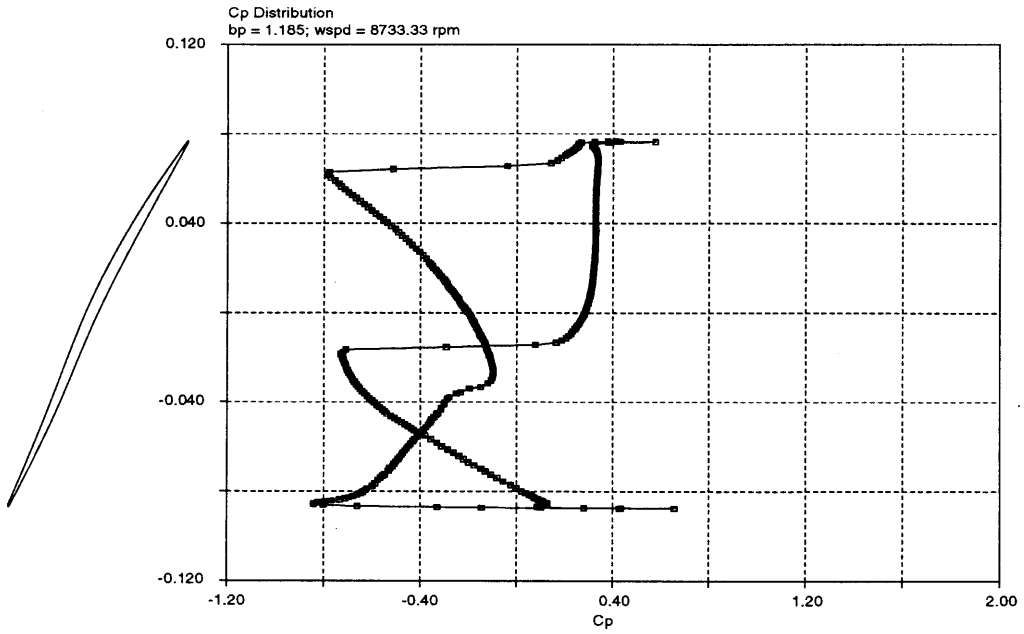


Figure 5.11: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.185 atm

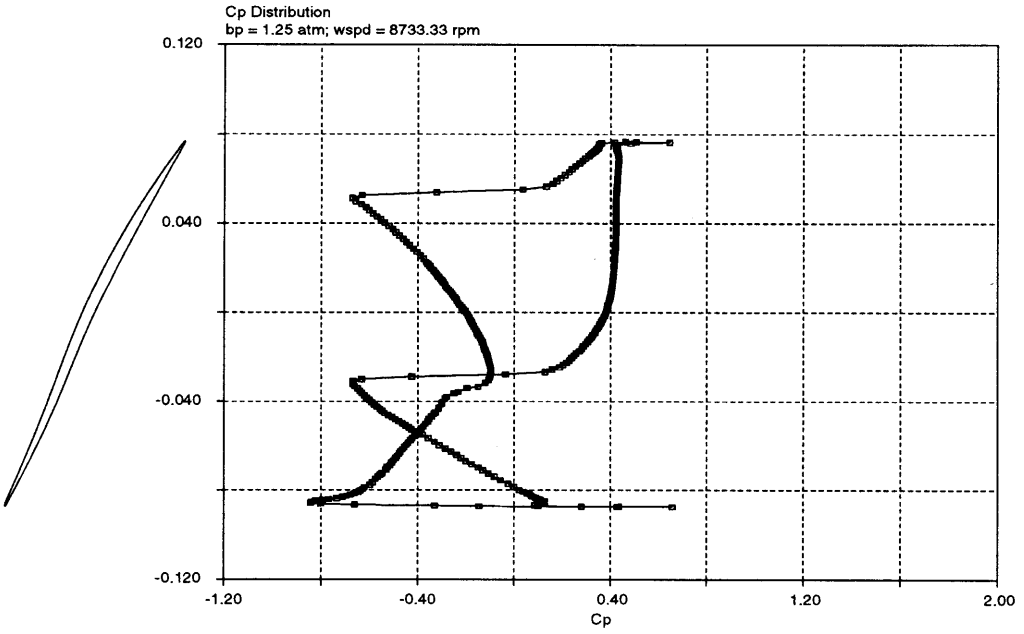


Figure 5.12: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.25 atm

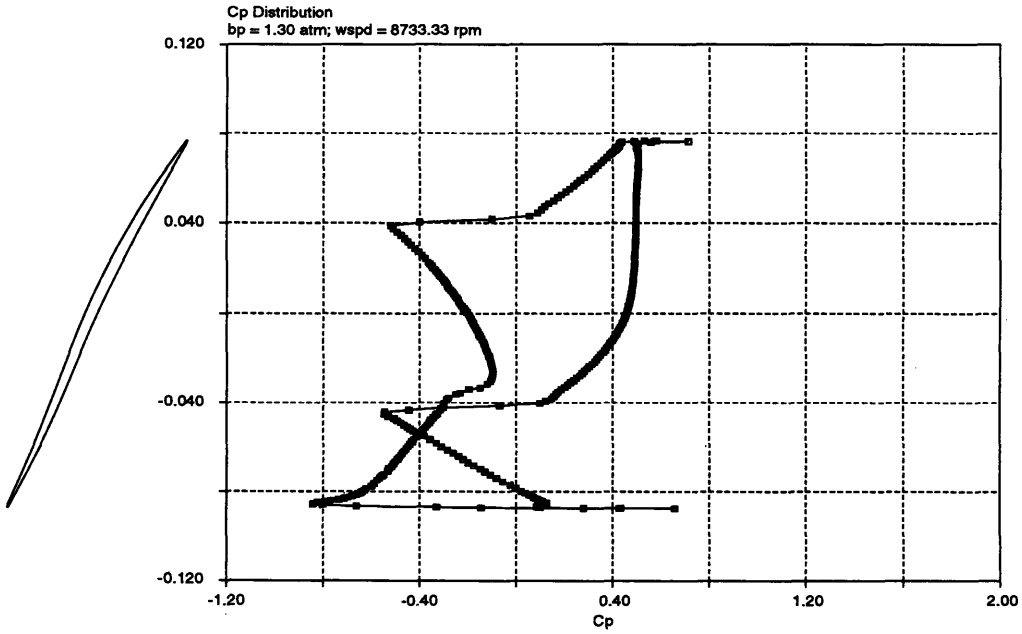


Figure 5.13: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.30 atm

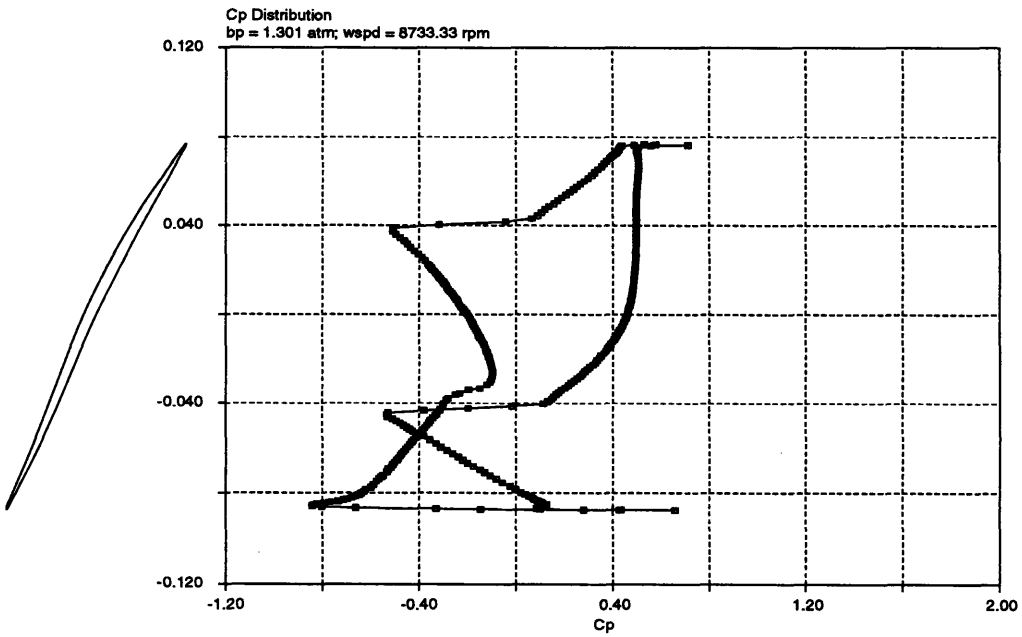


Figure 5.14: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.301 atm

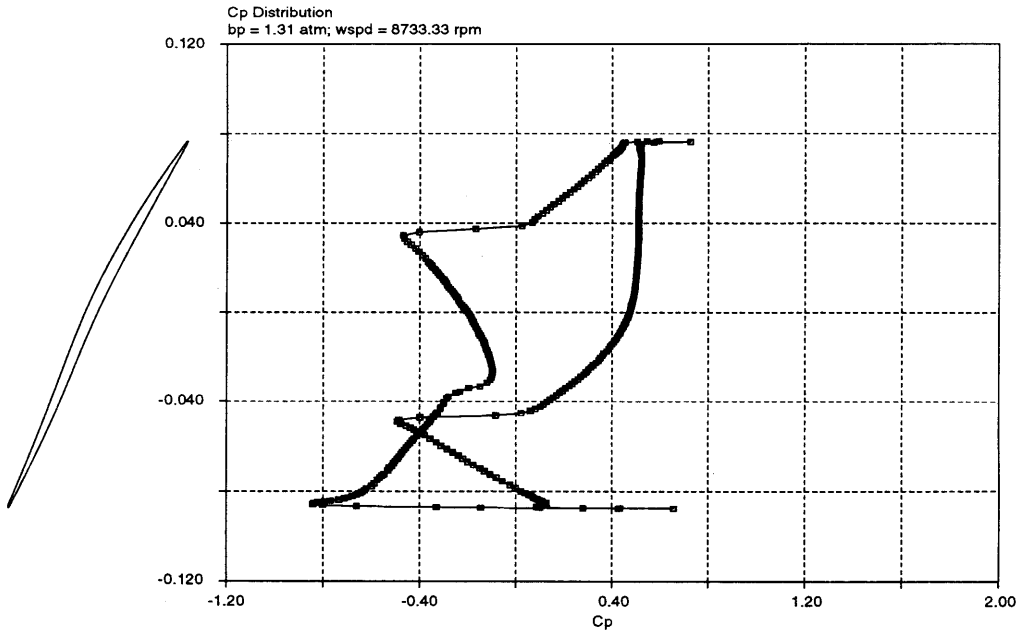


Figure 5.15: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.31 atm

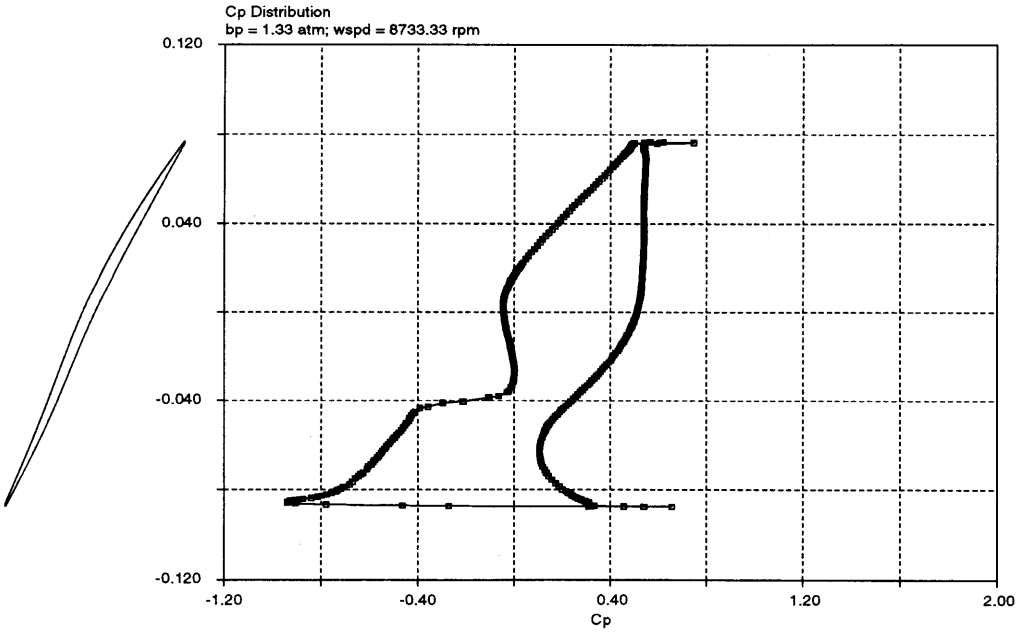


Figure 5.16: Cp Distribution on Blade for 67% N_c with Clean Inlet Flow and Back Pressure of 1.33 atm

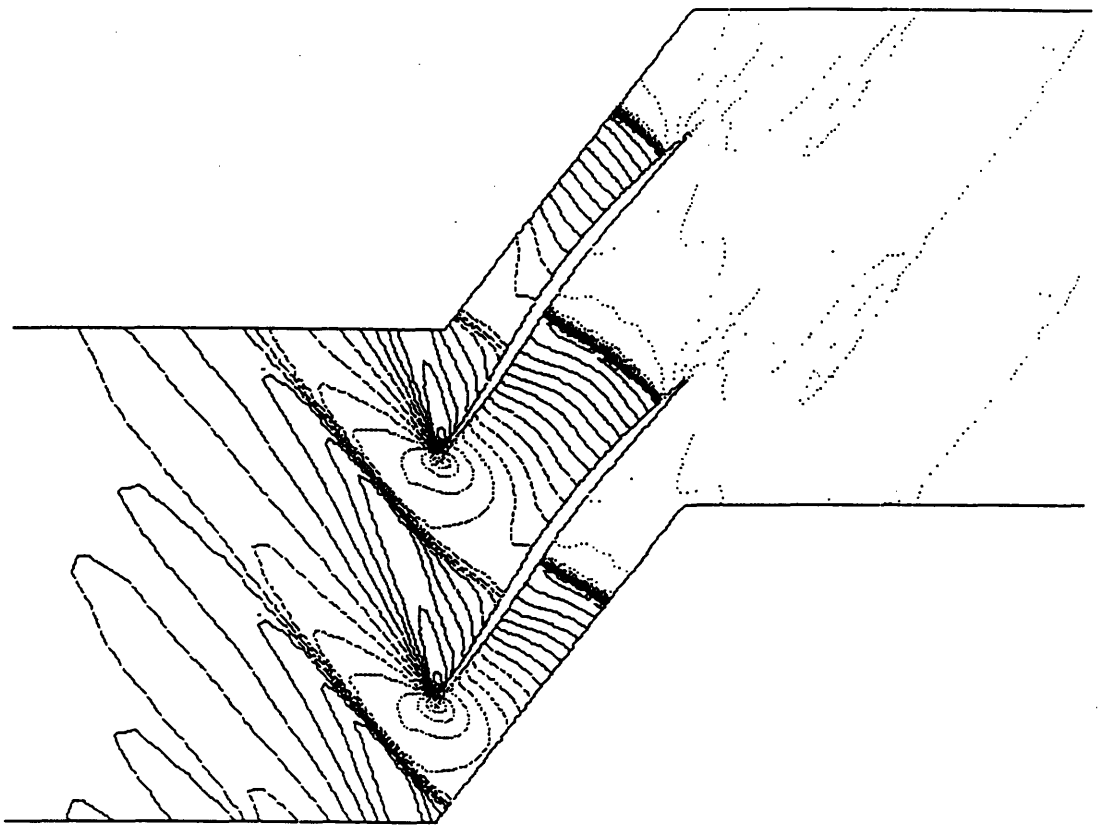


Figure 5.17: Mach Contours for 67% Corrected Design Speed Choked Condition with Back Pressure at 1.185 atm

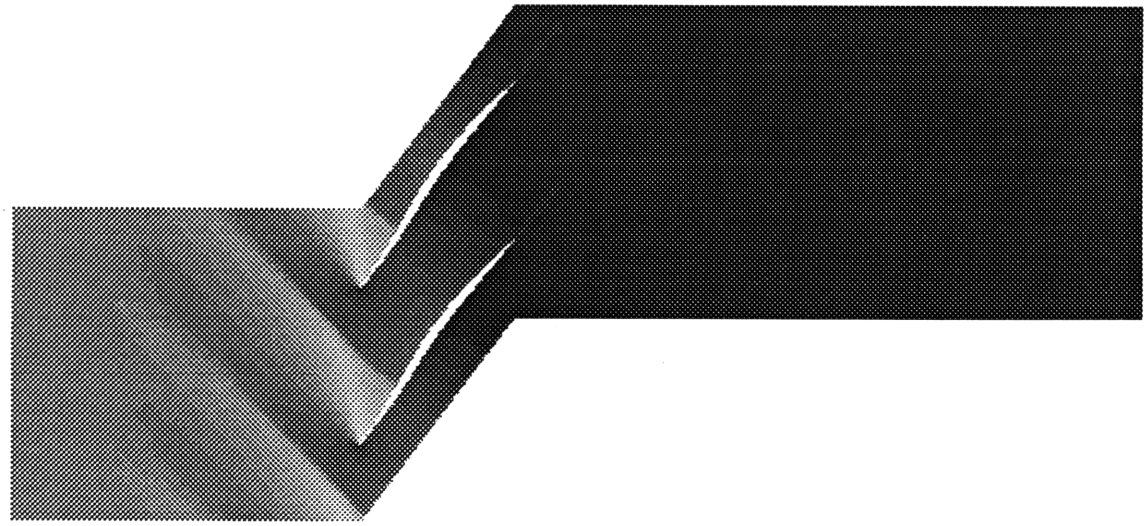


Figure 5.18: Density Contours for 67% Corrected Design Speed with Back Pressure at 1.33 atm

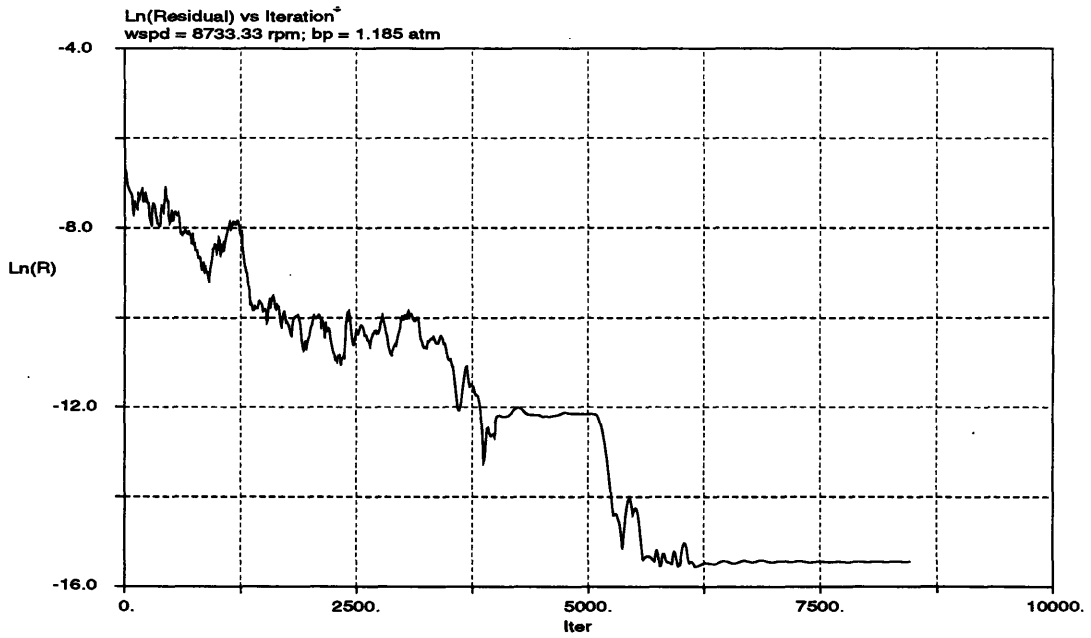


Figure 5.19: Convergence History for 67% Corrected Design Speed with Backp Pressure at 1.185 atm

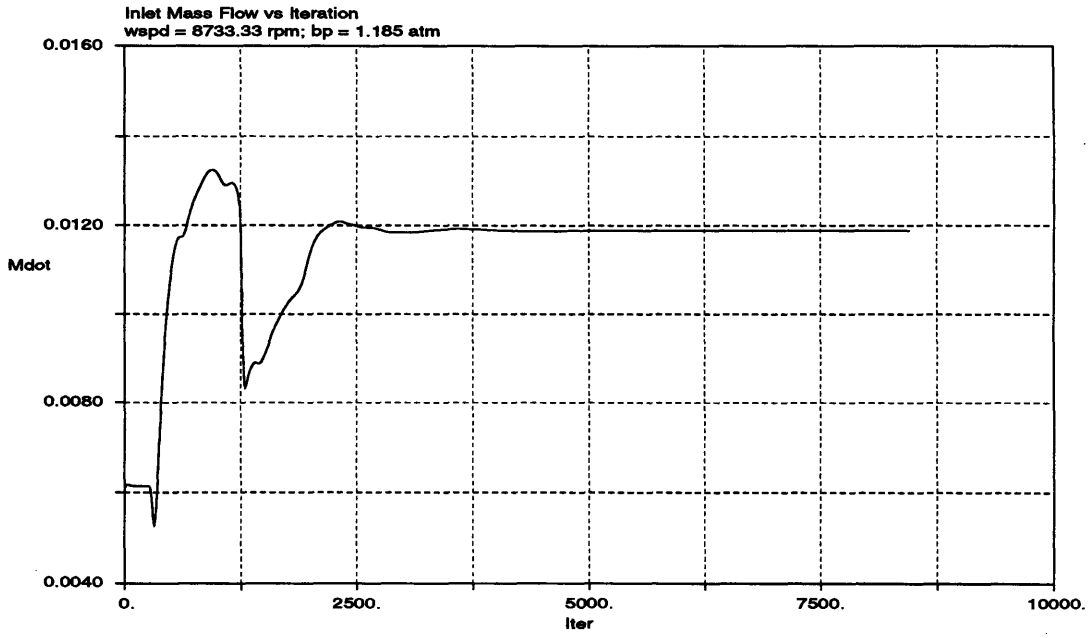


Figure 5.20: Inlet Mass Flow History for 67% Corrected Design Speed with Back Pressure at 1.185 atm

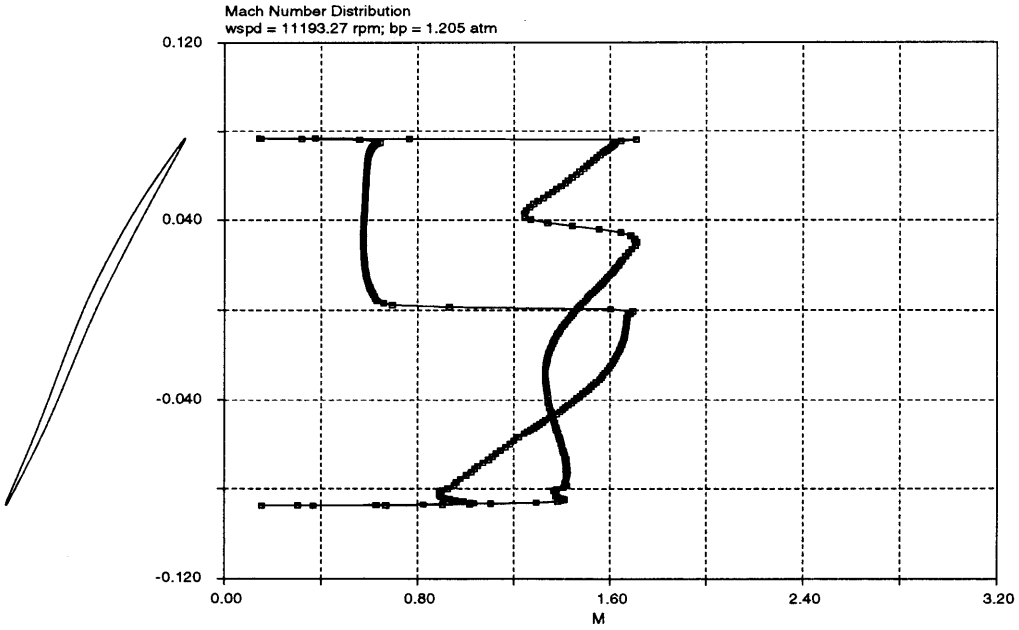


Figure 5.21: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.205 atm

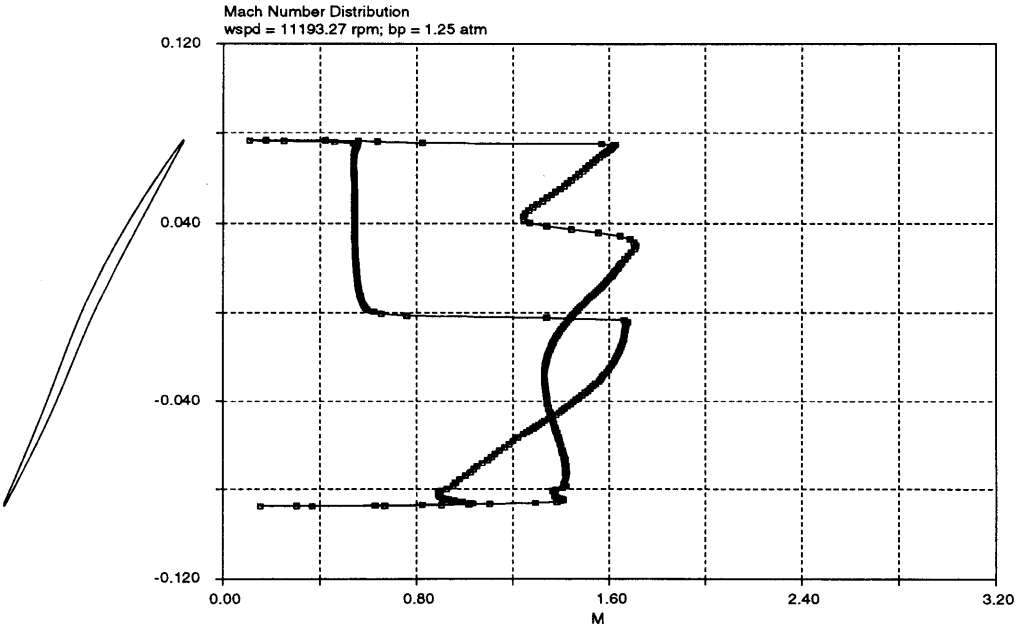


Figure 5.22: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.25 atm

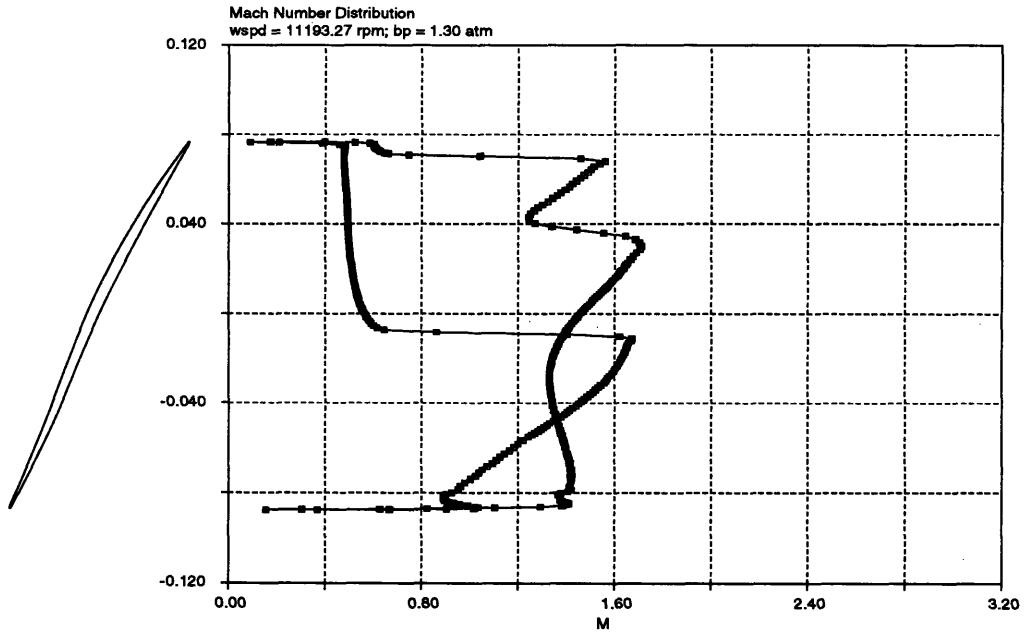


Figure 5.23: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.30 atm

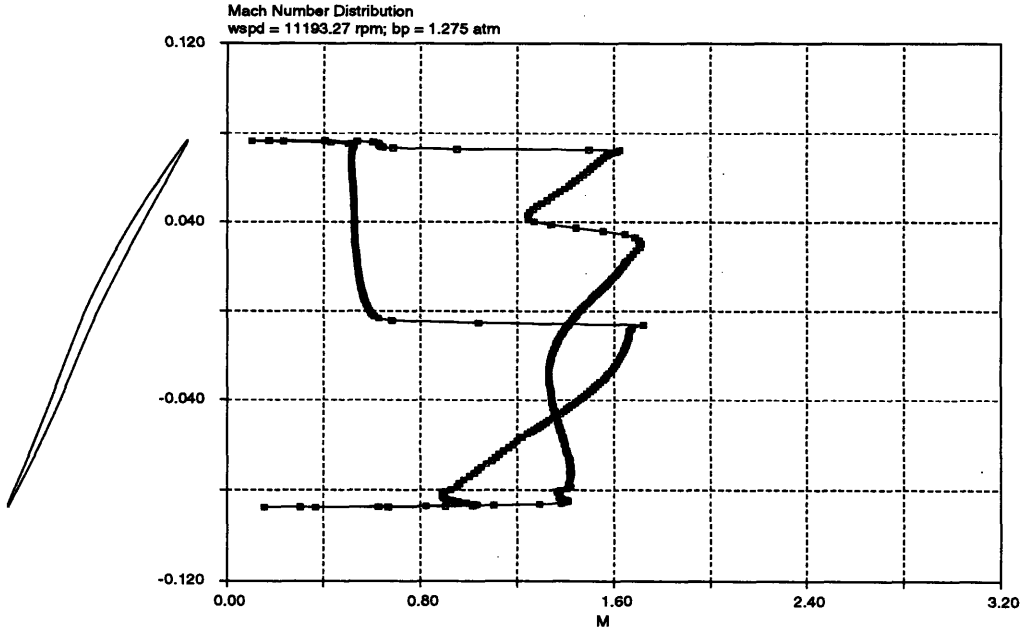


Figure 5.24: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.275 atm

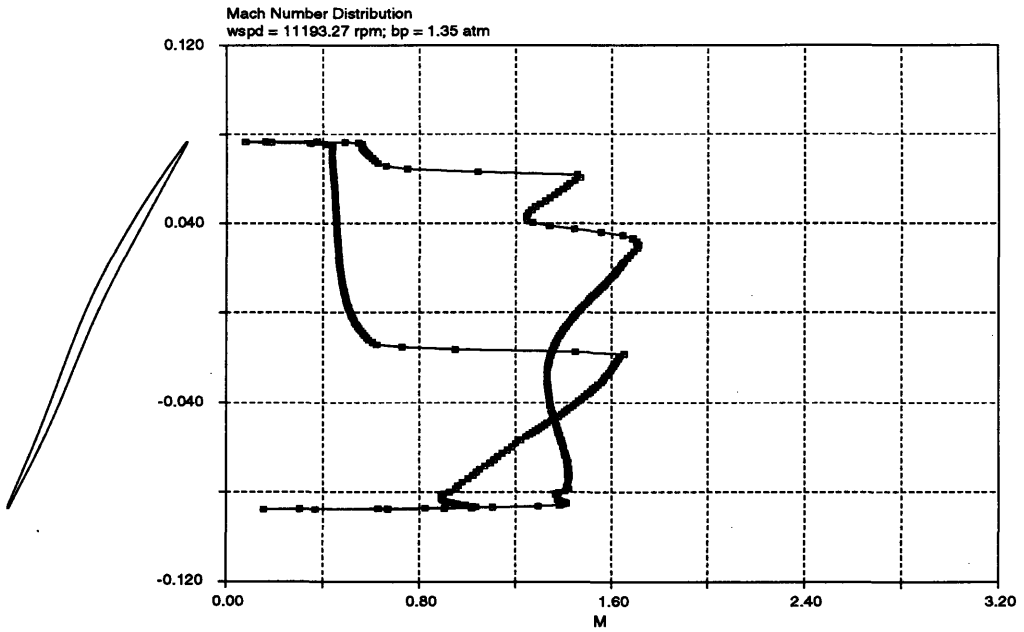


Figure 5.25: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.35 atm

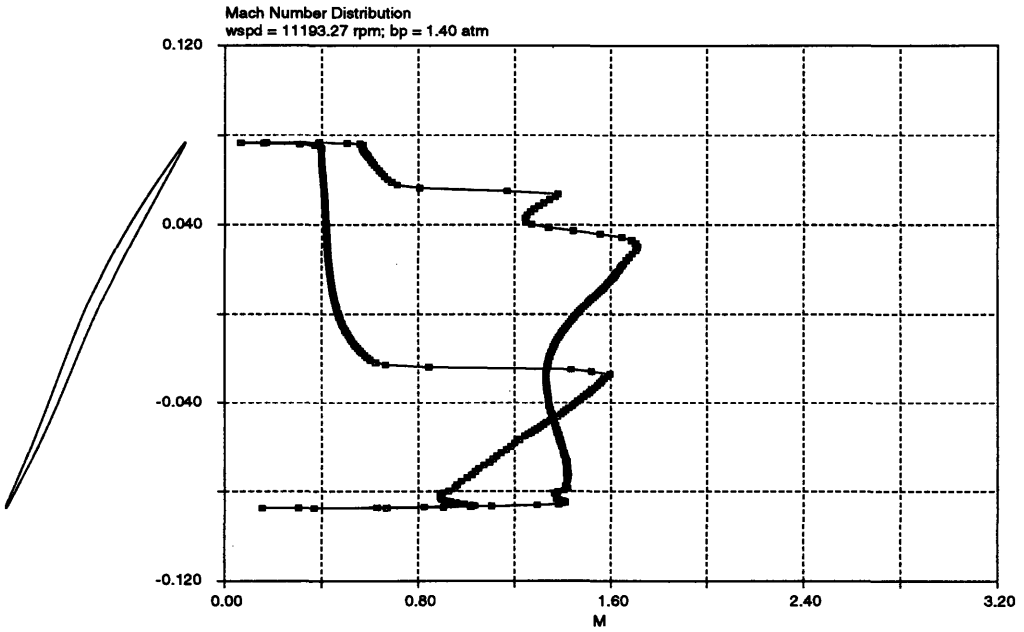


Figure 5.26: Mach Number Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.40 atm

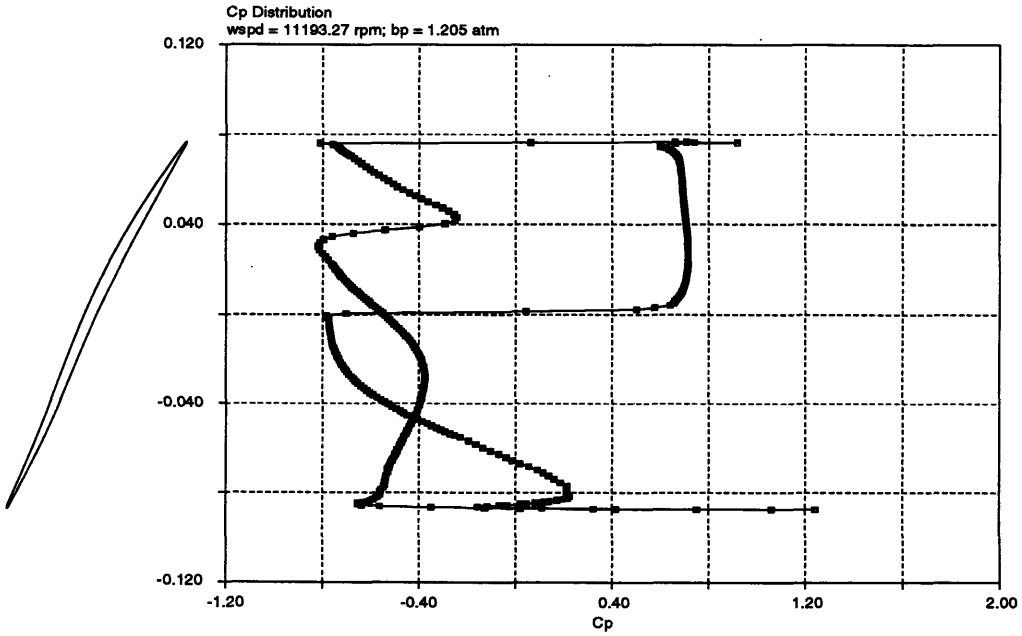


Figure 5.27: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.205 atm

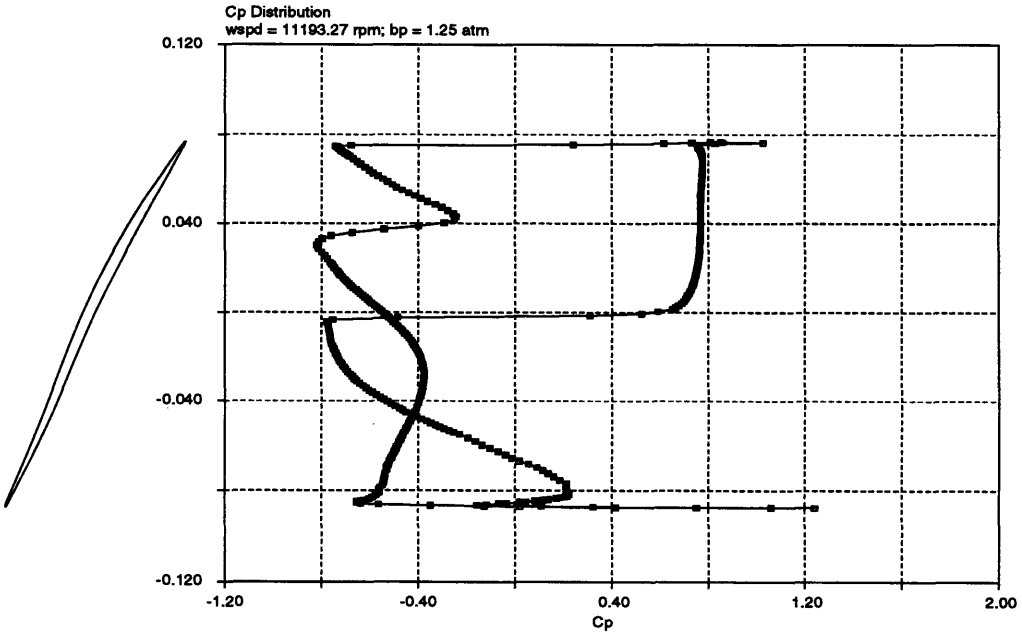


Figure 5.28: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.25 atm

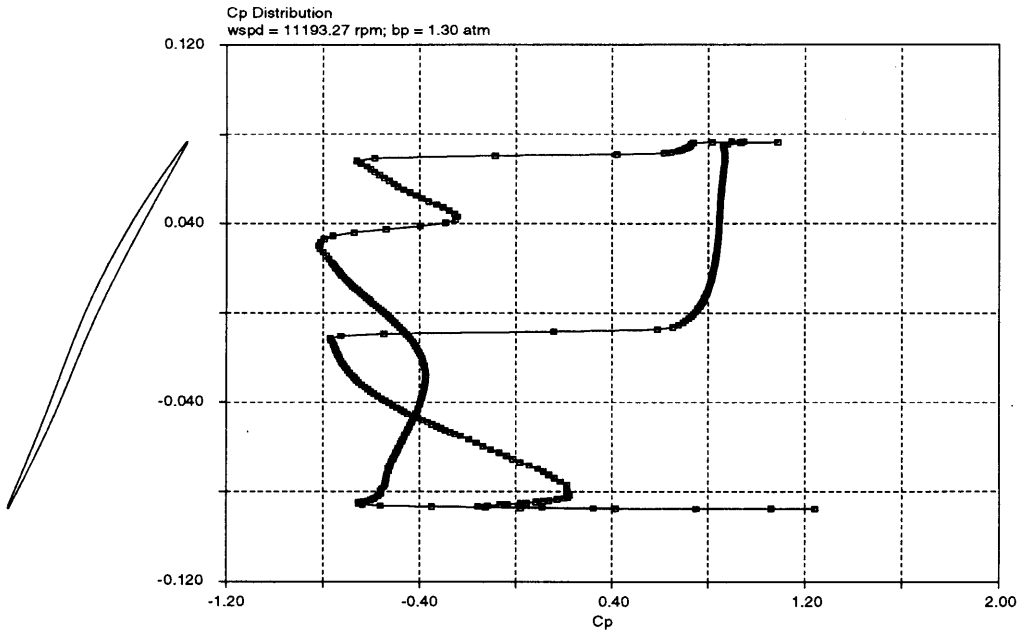


Figure 5.29: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.30 atm

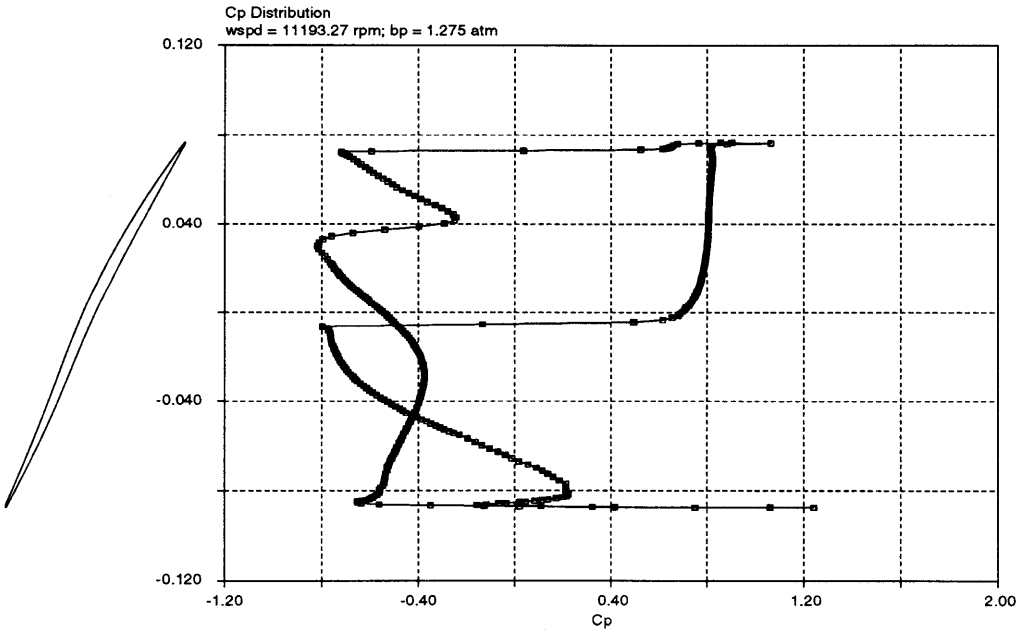


Figure 5.30: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.275 atm

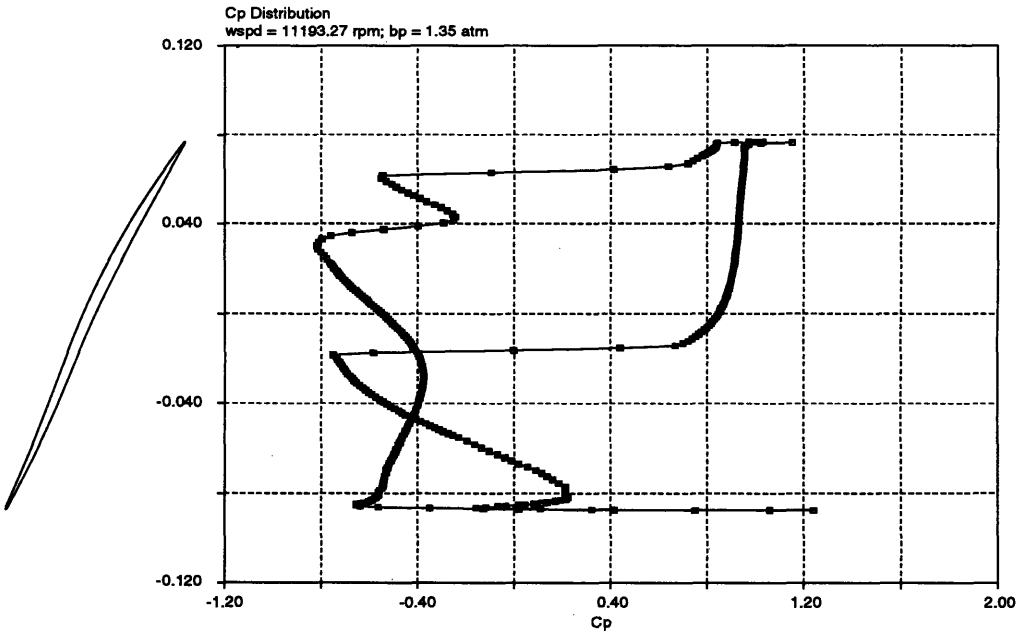


Figure 5.31: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.35 atm

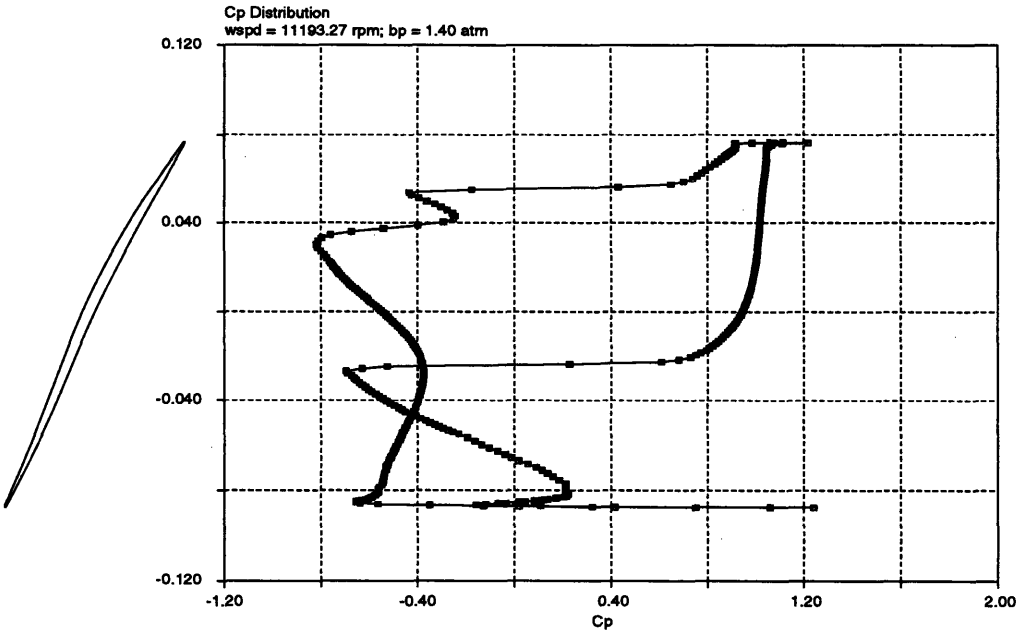


Figure 5.32: Cp Distribution on Blade for 85% N_c with Clean Inlet Flow and Back Pressure of 1.40 atm

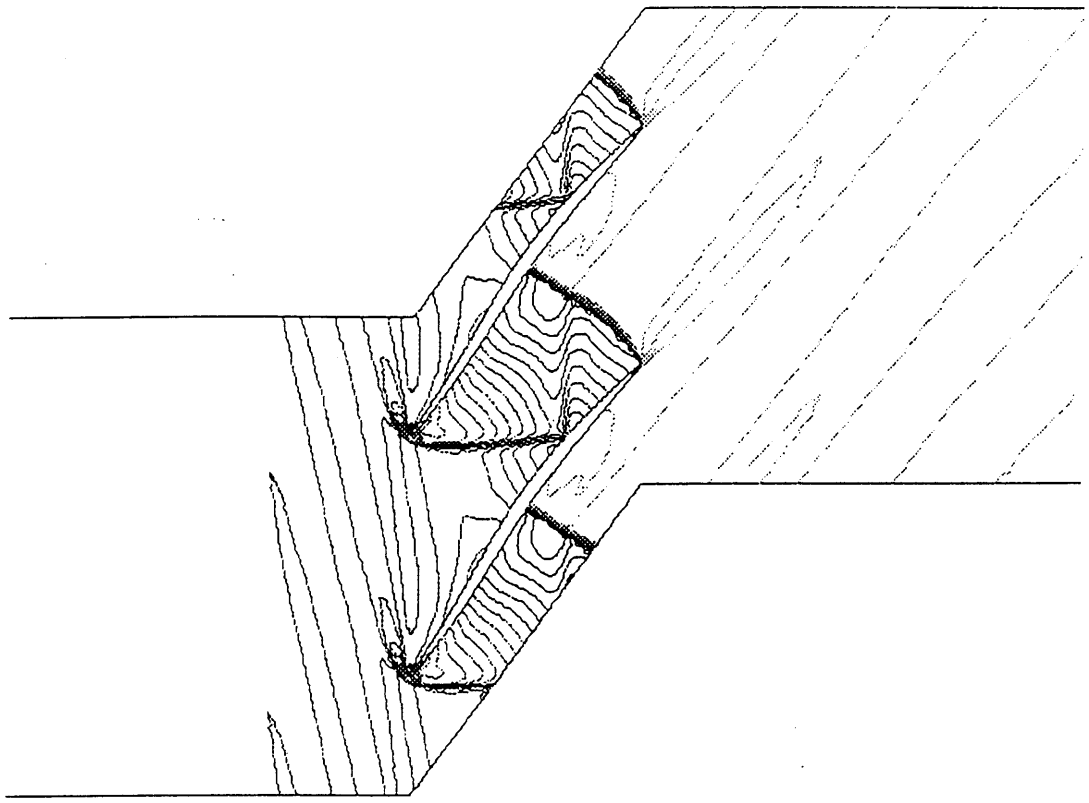


Figure 5.33: Mach Contours for 85% Corrected Design Speed Choked Condition with Back Pressure at 1.205 atm

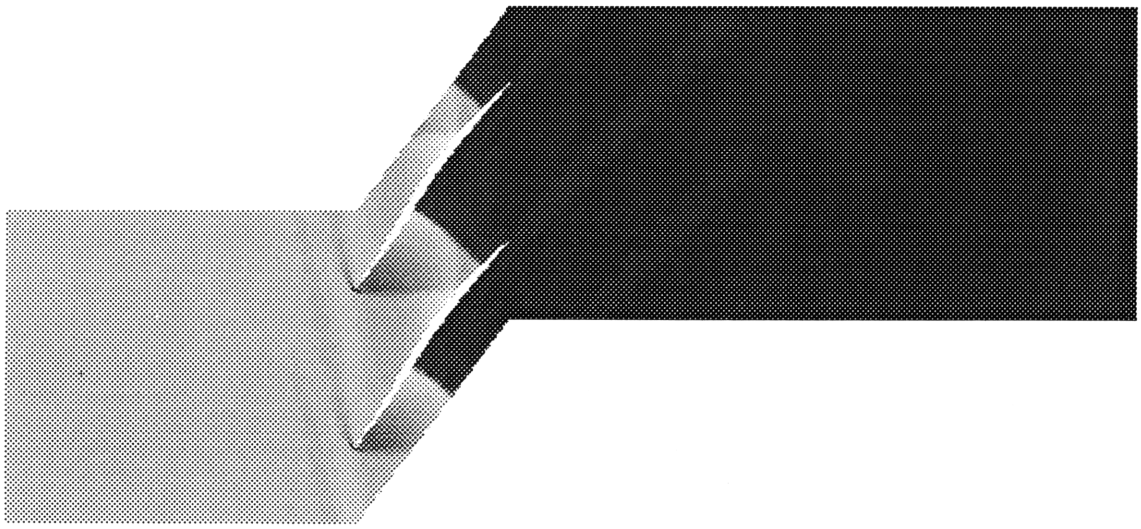


Figure 5.34: Density Contours for 85% Corrected Design Speed with Back Pressure at 1.45 atm

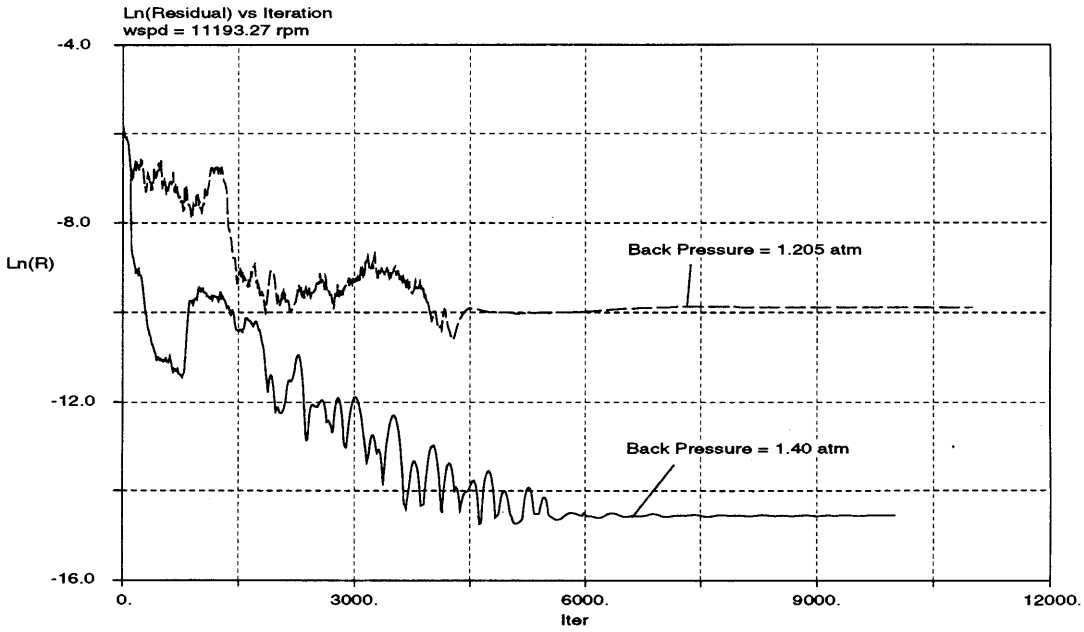


Figure 5.35: Residual History for 85% Corrected Design Speed with Back Pressures at 1.205 atm and 1.40 atm

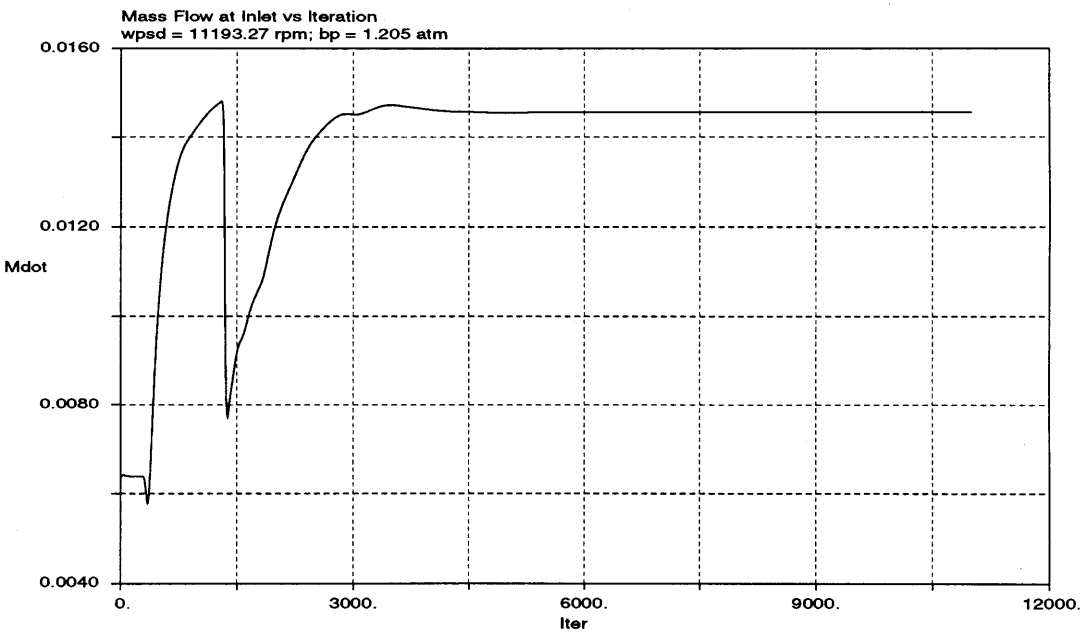


Figure 5.36: Inlet Mass Flow History for 85% Corrected Design Speed with Backp Pressure at 1.1205 atm

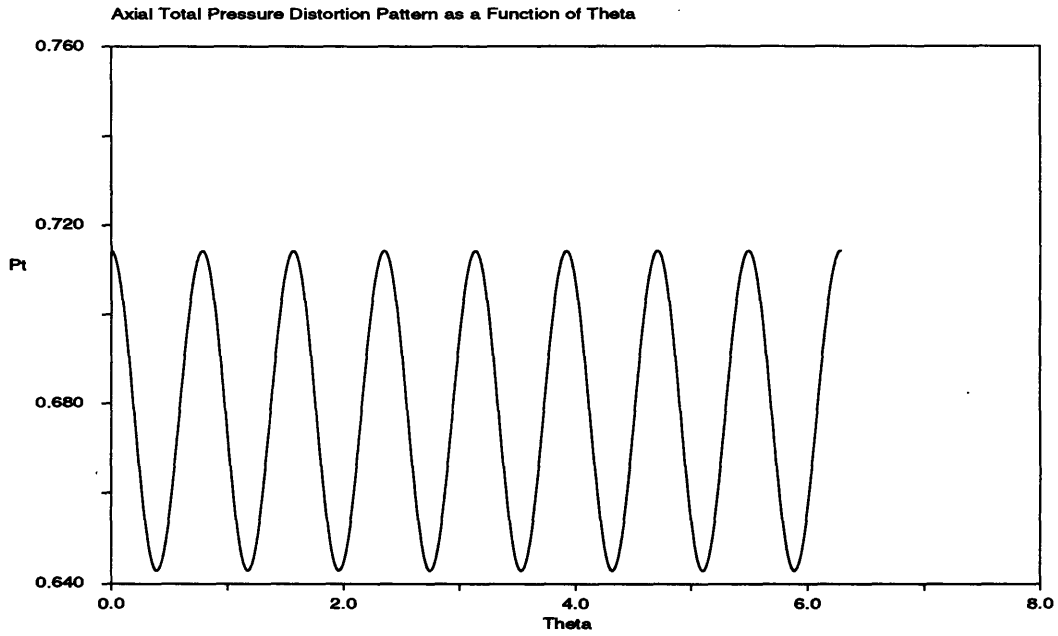


Figure 5.37: Distortion Pattern at Inlet

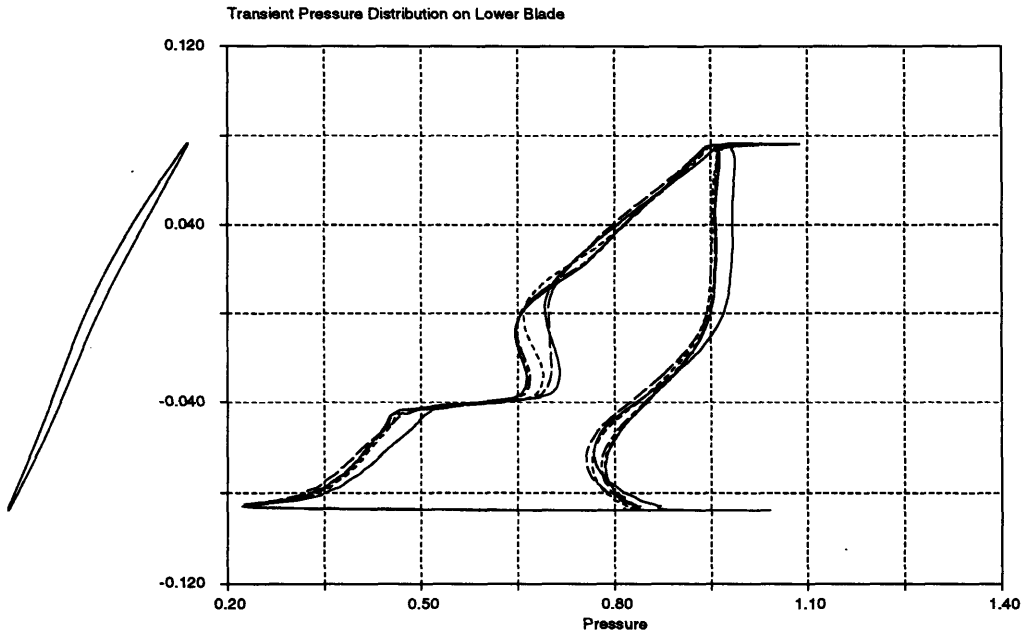


Figure 5.38: Transient Pressure Distribution on Lower Blade

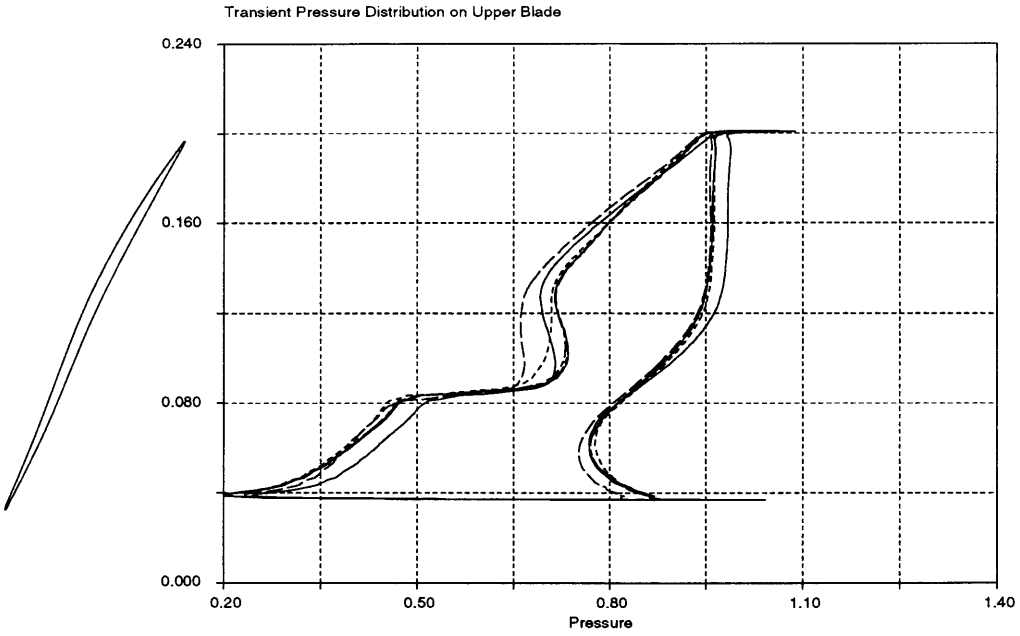


Figure 5.39: Transient Pressure Distribution on Upper Blade

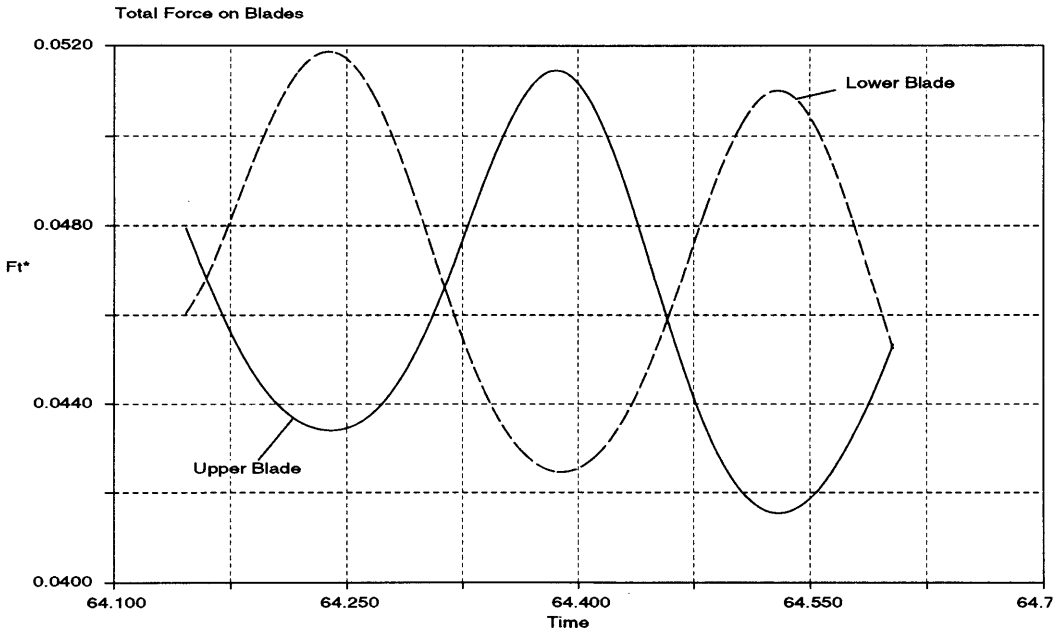


Figure 5.40: Total Force Variation on Blades

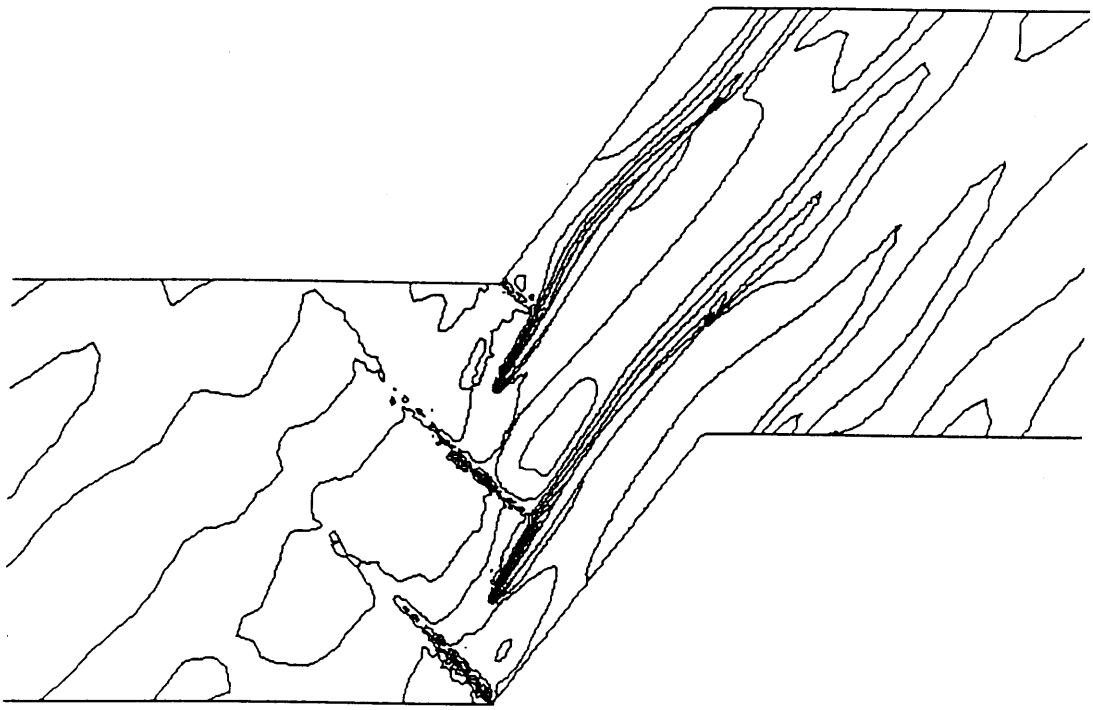


Figure 5.41: Total Pressure Contours for 67% Corrected Design Speed with Inlet Distortion at 1.152 msec

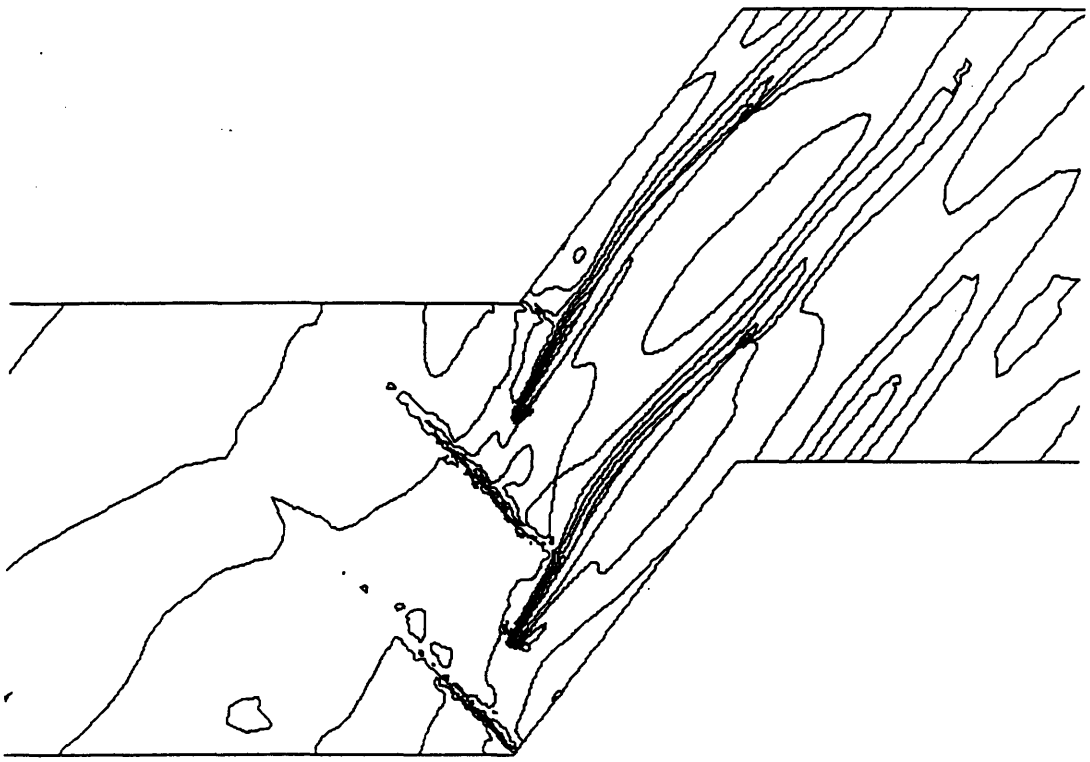


Figure 5.42: Total Pressure Contours for 67% Corrected Design Speed with Inlet Distortion at 0.7358 msec

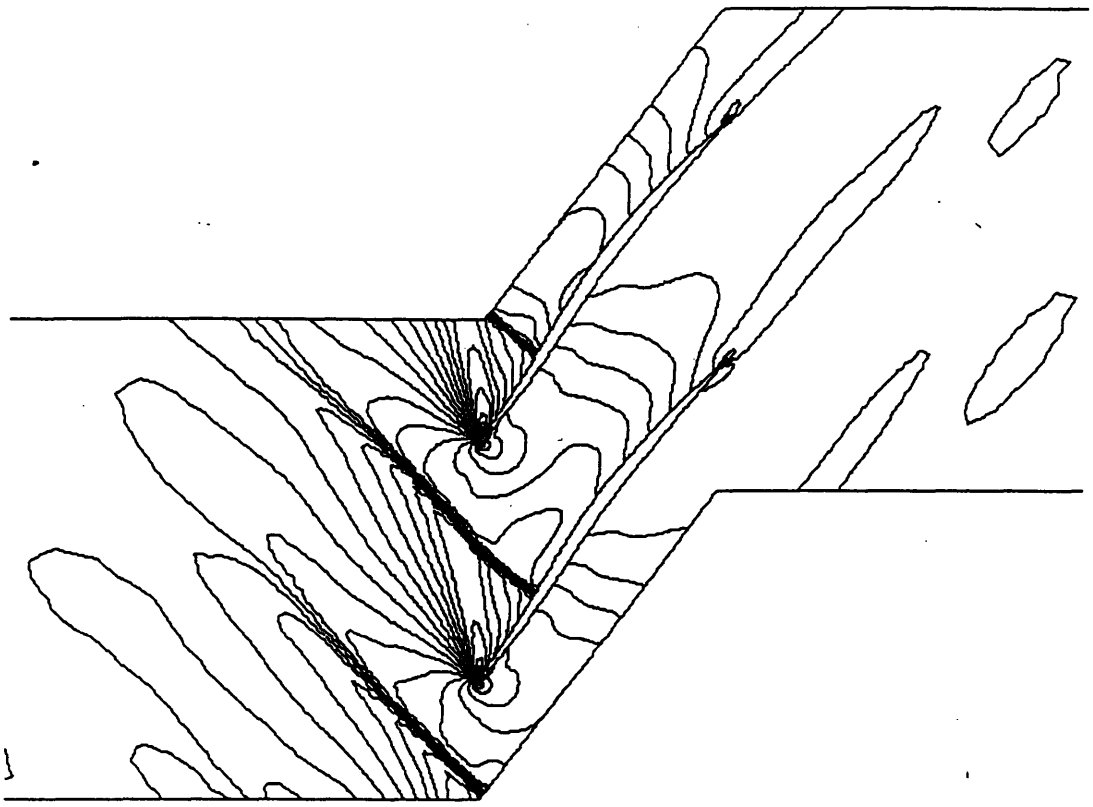


Figure 5.43: Mach Contours for 67% Corrected Design Speed with Inlet Distortion at 0.7358 msec

Chapter 6
Conclusions and Recommendations

6.1 Conclusions

A numerical algorithm for the solution of the two dimensional, compressible Euler equations for turbomachinery applications was successfully developed. This algorithm used a finite volume approach on unstructured meshes and a linear approximation to spatially approximate the unknowns. The integration was performed on generalized control volumes that were arbitrarily defined by a point lying within each element and a point on each edge of the mesh.

A Fourier analysis was used to investigate the dispersion characteristics of the numerical algorithm. It was shown that these characteristics strongly depend on the spatial integration of the time derivative term. The classical piecewise constant form of integration, found in many finite volume approximations, showed inadequate behavior for reproducing unsteady phenomena. A piecewise linear integration yielded a matrix that coupled the nodes of the mesh and showed an improvement over the piecewise constant approximation. It was shown that by restricting the generalized control volumes to be defined by the centroids of the elements and the midpoints of the edges, a combination of a piecewise constant and piecewise linear approximation could be used to reproduce the finite element mass matrix used in the finite element method. This method yielded far better dispersion behavior than the piecewise constant and piecewise linear approximations. It was concluded that this form of integration should be used when modeling unsteady flows.

The addition of numerical dissipation to stabilize the scheme, capture shocks without oscillations and accurately model the solution in smooth regions of the flow was addressed. This dissipation was based on the Symmetric Limited Positive (SLIP) scheme developed by Jameson [11, 12] which uses a flux limiting procedure. This procedure was modified so that the limiter was not active at smooth extrema and was less dissipative in smooth regions of the flow. This was verified by simulating a shock tube and convected density perturbation which showed that the procedure captured shocks without oscillations and allowed a small perturbation to convect without serious attenuation. Also, it was shown that the choice of control volumes used in the formulation could add a diffusion-like term into the numerical algorithm. This term was incorporated into the numerical dissipation.

Finally, a transonic axial compressor simulation was performed. Steady state solutions were obtained for two operating speeds with varying back pressures. The solutions were compared with the Euler turbine equation and showed excellent agreement. Also, an

unsteady forced response test was simulated using a periodic axial total pressure variation. The frequency of the total force on the compressor blades agreed well with the frequency of the distortion.

6.2 Recommendations

There are a number of issues concerning the numerical algorithm which can be explored. One issue is the effect of using different types of control volumes on the numerical solution. Although the formulation for the scheme presented used generalized control volumes, the volumes used throughout this work were defined by the centroids of the elements and midpoints of the edges. Another issue is formulating a time accurate, implicit version of the algorithm. This would have been very useful for the transonic, axial compressor simulation since the time step restriction on the nodes at the leading and trailing edges of the blade governed the simulation. The data obtained for the forced response test took approximately five days to compute on a SGI Indigo workstation. Finally, the extension of the algorithm for the solution of three dimensional, inviscid flows should be straightforward.

The inclusion of non-reflecting boundary conditions in the simulation of axial compressor flows would also be a great benefit. It was noticed that the boundary conditions applied in the compressor simulation created reflections of outgoing waves. These reflections affected the convergence of the steady state solutions and would be an impedance for unsteady simulations since reflected waves would corrupt the solution.

Bibliography

[1] J. D. Anderson. *Modern Compressible Flow.* McGraw-Hill, 1982.

[2] T. J. Barth. Aspects of unstructured grids and finite volume solvers for the euler and navier-stokes equations. AGARD Report 787.

[3] J. Bonet and J. Peraire. An alternating digital tree (adt) algorithm for 3d geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering*, 31:1–17, 1991.

[4] A. Bowyer. Computing dirichlet tessellations. *The Computer Journal*, 24(2), 1981.

[5] J. Ferguson. Multivariable curve interpolation. *J. ACM*, 11(2):221–228, 1964.

[6] P. L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering*, pages 269–288, 1991.

[7] M. B. Giles. Energy stability analysis of multi-step methods on unstructured meshes. Technical Report CFDL-TR-87-1, MIT, 1987.

[8] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.

[9] C. Hirsch. *Numerical Computations of Internal and External Flows*, volume 1. John Wiley and Sons, 1988.

[10] C. Hirsch. *Numerical Computations of Internal and External Flows*, volume 2. John Wiley and Sons, 1988.

[11] A. Jameson. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flows. In *11th AIAA Computational Fluid Dynamics Conference*, Orlando, FL, July 1993.

[12] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1 artificial diffusion, upwind biasing, limiters and their effect on accuracy of multigrid convergence. *Submitted for publication in International Journal of Computational Fluid Dynamics*, August 1994.

[13] J. L. Kerrebrock. *Aircraft Engines and Gas Turbines.* The MIT Press, 2 edition, 1992.

[14] P. D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* SIAM Publications, 1973.

[15] H. W. Liepmann and A. Roshko. *Elements of Gasdynamic.* John Wiley and Sons, Inc., 1957.

[16] J. Peraire. Class notes from 16.160, Advanced Computational Fluid Dynamics, Massachusetts Institute of Technology, 1994.

[17] J. Peraire, J. Peiro, and K. Morgan. *Felisa: A Reference Manual*, August 1993.

[18] W.H. Press, S. A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in FORTRAN: The Art of Scientific Computing.* Press Syndicate of the University of Cambridge, 2 edition, 1992.

[19] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[20] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[21] C. K. W. Tam, J. C. Webb, and Z. Dong. A study of the short wave components in computational acoustics. *Journal of Computational Acoustics*, 1(1):1–30, 1993.

[22] E. Turkel. Improving the accuracy of central difference schemes. Technical report, NASA CR-181712, 1988.

[23] D.F. Watson. Computing the n-dimensional delaunay tessellation with applicaton to voronoi polytopes. *The Computer Journal*, 24(2), 1981.

[24] N.P. Weatherhill. A method for generating irregular computational grids in multiply connected planar domains. *International Journal for Numerical Methods in Fluids*, 8:181–197, 1988.

[25] O. C. Zienkeiwicz and K. Morgan. *Finite Elements and Approximation.* John Wiley and Sons, Inc., 1983.

Appendix A
Mass Matrices

When evaluating the integral

$$\int\int_{\Omega} \frac{\partial u}{\partial t} d\Omega \tag{A.1}$$

two approximations are used: a piecewise linear and a combination of piecewise linear/piecewise constant.

A.1 Finite Volume Mass Matrix

The integration is performed over all the generalized control volumes in the mesh. This is accomplished by looping over all the elements of the mesh and integrating over the portion of each volume contained in the element. Figure A.1 refers to the notation used. The integrals for each node of the element shown in the figure are:

$$Node\ 1: \quad \int\int_{\Omega_1} \frac{\partial u}{\partial t} d\Omega_1 + \int\int_{\Omega_2} \frac{\partial u}{\partial t} d\Omega_2 \tag{A.2}$$

$$Node\ 2: \quad \int\int_{\Omega_3} \frac{\partial u}{\partial t} d\Omega_3 + \int\int_{\Omega_4} \frac{\partial u}{\partial t} d\Omega_4 \tag{A.3}$$

$$Node\ 3: \quad \int\int_{\Omega_5} \frac{\partial u}{\partial t} d\Omega_5 + \int\int_{\Omega_6} \frac{\partial u}{\partial t} d\Omega_6 \tag{A.4}$$

Evaluation of these integrals is performed by mapping each subelement, Ω_1, Ω_2, Ω_3, Ω_4, Ω_5 and Ω_6, into a parametric space [25]. The coordinates in real space as a function of the parametric coordinates are:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} x_2 & x_3 \\ y_2 & y_3 \end{bmatrix} \begin{bmatrix} \xi_2 \\ \xi_3 \end{bmatrix}$$

where the ξ_1, ξ_2 axes are shown in the figure. This gives:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_2 & x_3 \\ y_2 & y_3 \end{bmatrix} \begin{bmatrix} d\xi_2 \\ d\xi_3 \end{bmatrix}$$

The area integral over subelement Ω_1 becomes:

$$\int\int_{\Omega_1} \frac{\partial u}{\partial t} d\Omega_1 = \int_0^1 \int_0^{1-\xi_2} \frac{\partial u}{\partial t} \underbrace{2A_{\Omega_1} d\xi_2 d\xi_3}_{dxdy}$$

where A_{Ω_1} is the area of the subelement. A linear approximation to $\frac{\partial u}{\partial t}$ found in Chapter 3 is substituted into the above equation. The approximation is:

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{n} \frac{du_i}{dt} N(x,y)_i$$

where $N(x,y)_i$ are the hat functions described in Chapter 3. The only nodes influencing the integral are those defining the element. Therefore, the integral for subelement Ω_1 becomes:

$$\int_0^1 \int_0^{1-\xi_2} \frac{\partial u}{\partial t} 2A_{\Omega_1} d\xi_2 d\xi_3 = \; 2A_{\Omega_1} \int_0^1 \int_0^{1-\xi_2} \left[\frac{du_1}{dt} N_1 + \frac{du_2}{dt} N_2 + \frac{du_3}{dt} N_3\right] d\xi_2 d\xi_3 \quad \text{(A.5)}$$

Each term in the sum can be evaluated separately. The first term in the sum, N_1, is written as a function of the coordinates in the mapped space, (ξ_1, ξ_2, ξ_3). This gives:

$$N(\xi_2, \xi_3)_1 = \xi_1 + N_{p1}^{(1)} \xi_2 + N_{p2}^{(1)} \xi_3$$

but $\xi_1 = 1 - \xi_2 - \xi_3$, therefore:

$$N(\xi_2, \xi_3)_1 = 1 + (N_{p1}^{(1)} - 1)\xi_2 + (N_{p2}^{(1)} - 1)\xi_3$$

where the subscripts and superscripts on the weights N are:

$$N_{p1}^{(1)} \;=\; \textit{the weight of node } 1 \textit{ evaluated at point } p1$$

The first term in equation A.5 is evaluated by substituting the above expression for N_1. This yields:

$$2A_{\Omega_1} \int_0^1 \int_0^{1-\xi_2} \frac{du_1}{dt} N_1 d\xi_2 d\xi_3 \;=\; \frac{A_{\Omega_1}}{3} \left[1 + N_{p1}^{(1)} + N_{p2}^{(1)}\right] \frac{du_1}{dt}$$

The other terms in equation A.5 are determined in the same manner by noting that for subelement Ω_1:

$$\begin{aligned} N(\xi_2, \xi_3)_2 &= N_{p1}^{(2)} \xi_2 + N_{p2}^{(2)} \xi_3 \\ N(\xi_2, \xi_3)_3 &= N_{p1}^{(3)} \xi_2 + N_{p2}^{(3)} \xi_3 \end{aligned}$$

Substituting these in the integral in equation A.5 yields:

$$\begin{aligned} 2A_{\Omega_1} \int_0^1 \int_0^{1-\xi_2} \frac{du_2}{dt} N_2 d\xi_2 d\xi_3 &= \frac{A_{\Omega_1}}{3} \left[N_{p1}^{(2)} + N_{p2}^{(2)}\right] \frac{du_2}{dt} \\ 2A_{\Omega_1} \int_0^1 \int_0^{1-\xi_2} \frac{du_3}{dt} N_3 d\xi_2 d\xi_3 &= \frac{A_{\Omega_1}}{3} \left[N_{p1}^{(3)} + N_{p2}^{(3)}\right] \frac{du_3}{dt} \end{aligned}$$

The second integral in equation A.2 is evaluated in the same manner by noting that for subelement 2:

$$\begin{aligned} N(\xi_2, \xi_3)_1 &= 1 + (N_{p2}^{(1)} - 1)\xi_2 + (N_{p3}^{(1)} - 1)\xi_3 \\ N(\xi_2, \xi_3)_2 &= N_{p2}^{(2)} \xi_2 + N_{p3}^{(2)} \xi_3 \\ N(\xi_2, \xi_3)_3 &= N_{p2}^{(3)} \xi_2 + N_{p3}^{(3)} \xi_3 \end{aligned}$$

Substituting these expressions into the integral over subelement 2 gives the following:

$$
\begin{aligned}
2A_{\Omega_2}\int_0^1\int_0^{1-\xi_2}\frac{du_1}{dt}N_1 d\xi_2 d\xi_3 &= \frac{A_{\Omega_2}}{3}\left[1+N_{p2}^{(1)}+N_{p3}^{(1)}\right]\frac{du_1}{dt}\\
2A_{\Omega_2}\int_0^1\int_0^{1-\xi_2}\frac{du_2}{dt}N_2 d\xi_2 d\xi_3 &= \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(2)}+N_{p3}^{(2)}\right]\frac{du_2}{dt}\\
2A_{\Omega_2}\int_0^1\int_0^{1-\xi_2}\frac{du_3}{dt}N_3 d\xi_2 d\xi_3 &= \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(3)}+N_{p3}^{(3)}\right]\frac{du_3}{dt}
\end{aligned}
$$

Combining the two integrals for node one gives the total contribution from the element to the volume surrounding node one:

$$
\left[\begin{array}{c|c|c|}
\frac{A_{\Omega_1}}{3}\left[1+N_{p1}^{(1)}+N_{p2}^{(1)}\right]+ & \frac{A_{\Omega_1}}{3}\left[N_{p1}^{(2)}+N_{p2}^{(2)}\right]+ & \frac{A_{\Omega_1}}{3}\left[N_{p1}^{(3)}+N_{p2}^{(3)}\right]+ \\
\frac{A_{\Omega_2}}{3}\left[1+N_{p2}^{(1)}+N_{p3}^{(1)}\right] & \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(2)}+N_{p3}^{(2)}\right] & \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(3)}+N_{p3}^{(3)}\right]
\end{array}\right]
\left[\begin{array}{c}\frac{du_1}{dt}\\ \frac{du_2}{dt}\\ \frac{du_3}{dt}\end{array}\right] \tag{A.6}
$$

The other integrals for nodes two and three in equations A.3 and A.4 are performed in the same manner. The functions $N(x,y)$ for the integrals are:

$$
\begin{aligned}
&Node\ 2: \quad Subelement\ 3\\
&\qquad N(\xi_2,\xi_3)_2 = 1+(N_{p1}^{(2)}-1)\xi_2+(N_{p2}^{(2)}-1)\xi_3\\
&\qquad N(\xi_2,\xi_3)_1 = N_{p1}^{(1)}\xi_2+N_{p2}^{(1)}\xi_3\\
&\qquad N(\xi_2,\xi_3)_3 = N_{p1}^{(3)}\xi_2+N_{p2}^{(3)}\xi_3\\
&Node\ 2: \quad Subelement\ 4\\
&\qquad N(\xi_2,\xi_3)_2 = 1+(N_{p2}^{(2)}-1)\xi_2+(N_{p4}^{(2)}-1)\xi_3\\
&\qquad N(\xi_2,\xi_3)_1 = N_{p2}^{(1)}\xi_2+N_{p4}^{(1)}\xi_3\\
&\qquad N(\xi_2,\xi_3)_3 = N_{p2}^{(3)}\xi_2+N_{p4}^{(3)}\xi_3\\
&Node\ 3: \quad Subelement\ 5\\
&\qquad N(\xi_2,\xi_3)_3 = 1+(N_{p4}^{(3)}-1)\xi_2+(N_{p2}^{(3)}-1)\xi_3\\
&\qquad N(\xi_2,\xi_3)_1 = N_{p4}^{(1)}\xi_2+N_{p2}^{(1)}\xi_3\\
&\qquad N(\xi_2,\xi_3)_2 = N_{p4}^{(2)}\xi_2+N_{p2}^{(2)}\xi_3\\
&Node\ 3: \quad Subelement\ 6\\
&\qquad N(\xi_2,\xi_3)_3 = 1+(N_{p2}^{(3)}-1)\xi_2+(N_{p3}^{(3)}-1)\xi_3\\
&\qquad N(\xi_2,\xi_3)_1 = N_{p2}^{(1)}\xi_2+N_{p3}^{(1)}\xi_3\\
&\qquad N(\xi_2,\xi_3)_2 = N_{p2}^{(2)}\xi_2+N_{p3}^{(2)}\xi_3
\end{aligned} \tag{A.7}
$$

Evaluating the integrals gives the following matrix for the element corresponding to the volumes for nodes one, two and three:

$$\left[\begin{array}{c|c|c} \frac{A_{\Omega_1}}{3}\left[1+N_{p1}^{(1)}+N_{p2}^{(1)}\right]+ & \frac{A_{\Omega_1}}{3}\left[N_{p1}^{(2)}+N_{p2}^{(2)}\right]+ & \frac{A_{\Omega_1}}{3}\left[N_{p1}^{(3)}+N_{p2}^{(3)}\right]+ \\ \frac{A_{\Omega_2}}{3}\left[1+N_{p2}^{(1)}+N_{p3}^{(1)}\right] & \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(2)}+N_{p3}^{(2)}\right] & \frac{A_{\Omega_2}}{3}\left[N_{p2}^{(3)}+N_{p3}^{(3)}\right] \\ \hline \frac{A_{\Omega_3}}{3}\left[N_{p1}^{(1)}+N_{p2}^{(2)}\right]+ & \frac{A_{\Omega_3}}{3}\left[1+N_{p1}^{(2)}+N_{p2}^{(2)}\right]+ & \frac{A_{\Omega_3}}{3}\left[N_{p1}^{(3)}+N_{p2}^{(3)}\right]+ \\ \frac{A_{\Omega_4}}{3}\left[N_{p2}^{(1)}+N_{p4}^{(1)}\right] & \frac{A_{\Omega_4}}{3}\left[1+N_{p2}^{(2)}+N_{p4}^{(2)}\right] & \frac{A_{\Omega_4}}{3}\left[N_{p2}^{(3)}+N_{p4}^{(3)}\right] \\ \hline \frac{A_{\Omega_5}}{3}\left[N_{p2}^{(1)}+N_{p4}^{(1)}\right]+ & \frac{A_{\Omega_5}}{3}\left[N_{p2}^{(2)}+N_{p4}^{(2)}\right]+ & \frac{A_{\Omega_5}}{3}\left[1+N_{p2}^{(3)}+N_{p4}^{(3)}\right]+ \\ \frac{A_{\Omega_6}}{3}\left[N_{p2}^{(1)}+N_{p3}^{(1)}\right] & \frac{A_{\Omega_6}}{3}\left[N_{p2}^{(2)}+N_{p3}^{(2)}\right] & \frac{A_{\Omega_6}}{3}\left[1+N_{p2}^{(3)}+N_{p3}^{(3)}\right] \end{array}\right] \left[\begin{array}{c} \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \end{array}\right] \tag{A.8}$$

This matrix is simplified further by noting:

$$\begin{aligned} N_{p1}^{(1)}+N_{p2}^{(1)} &= 2N_{m1}^{(1)} \\ N_{p1}^{(2)}+N_{p2}^{(2)} &= 2N_{m1}^{(2)} \\ N_{p1}^{(3)}+N_{p2}^{(3)} &= 2N_{m1}^{(3)} \\ N_{p2}^{(1)}+N_{p3}^{(1)} &= 2N_{m2}^{(1)} \\ N_{p2}^{(2)}+N_{p3}^{(2)} &= 2N_{m2}^{(2)} \\ N_{p2}^{(3)}+N_{p3}^{(3)} &= 2N_{m2}^{(3)} \\ N_{p3}^{(1)}+N_{p4}^{(1)} &= 2N_{m3}^{(1)} \\ N_{p3}^{(2)}+N_{p4}^{(2)} &= 2N_{m3}^{(2)} \\ N_{p3}^{(3)}+N_{p4}^{(3)} &= 2N_{m3}^{(3)} \end{aligned}$$

where the subscript m corresponds to the midpoints of the flux faces in figure A.1. This yields:

$$\left[\begin{array}{c|c|c} \frac{A_{\Omega_1}}{3}\left[1+2N_{m1}^{(1)}\right]+ & \frac{A_{\Omega_1}}{3}\left[2N_{m1}^{(2)}\right]+ & \frac{A_{\Omega_1}}{3}\left[2N_{m1}^{(3)}\right]+ \\ \frac{A_{\Omega_2}}{3}\left[1+2N_{m2}^{(1)}\right] & \frac{A_{\Omega_2}}{3}\left[2N_{m2}^{(2)}\right] & \frac{A_{\Omega_2}}{3}\left[2N_{m2}^{(3)}\right] \\ \hline \frac{A_{\Omega_3}}{3}\left[2N_{m1}^{(1)}\right]+ & \frac{A_{\Omega_3}}{3}\left[1+2N_{m1}^{(2)}\right]+ & \frac{A_{\Omega_3}}{3}\left[2N_{m1}^{(3)}\right]+ \\ \frac{A_{\Omega_4}}{3}\left[2N_{m3}^{(1)}\right] & \frac{A_{\Omega_4}}{3}\left[1+2N_{m3}^{(2)}\right] & \frac{A_{\Omega_4}}{3}\left[2N_{m3}^{(3)}\right] \\ \hline \frac{A_{\Omega_5}}{3}\left[2N_{m3}^{(1)}\right]+ & \frac{A_{\Omega_5}}{3}\left[2N_{m3}^{(2)}\right]+ & \frac{A_{\Omega_5}}{3}\left[1+2N_{m3}^{(3)}\right]+ \\ \frac{A_{\Omega_6}}{3}\left[2N_{m2}^{(1)}\right] & \frac{A_{\Omega_6}}{3}\left[2N_{m2}^{(2)}\right] & \frac{A_{\Omega_6}}{3}\left[1+2N_{m2}^{(3)}\right] \end{array}\right] \left[\begin{array}{c} \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \end{array}\right] \tag{A.9}$$

This gives the elemental mass matrix using a linear approximation. The entire system matrix, $\bar{M}$, may be determined by looping over all the elements and sending their contributions to the correct place in the matrix. For the solver developed in Chapter 3, the elemental matrix was broken into an edge-based form since the scheme uses edges.

A.2 Finite Element Mass Matrix

The finite element mass matrix used in the finite element method has the form [25]:

$$\frac{A_e}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}\begin{bmatrix} \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \end{bmatrix}$$

where A_e is the area of the element. An approximation of $\frac{du}{dt}$ using a combination of piecewise constant and piecewise linear functions is:

$$\frac{du}{dt} = \sum_{k=1}^{n} \frac{du_k}{dt}\left[\beta N(x,y)_k + (1-\beta)\, P(x,y)_k\right] \tag{A.10}$$

where $N(x,y)$ are the linear hat functions defined previously; $P(x,y)$ are piecewise constant hat functions which have the value of unity over the whole volume and zero everywhere else; and β is a constant.

A particular form of the finite volume elemental mass matrix is found by restricting the points p1, p2, p3 and p4 in figure A.1 to lie at the centroid of the element and midpoints of the edges. The values of the weights, N, are:

$$\begin{aligned} N^1_{m1} = N^2_{m1} = N^1_{m2} = N^3_{m2} = N^2_{m2} = N^3_{m2} &= \frac{5}{12} \\ N^3_{m1} = N^2_{m2} = N^1_{m2} &= \frac{1}{6} \\ A_{\Omega_1} = A_{\Omega_2} = A_{\Omega_3} = A_{\Omega_4} = A_{\Omega_5} = A_{\Omega_6} &= \frac{A_e}{6} \end{aligned}$$

The finite volume elemental mass matrix becomes:

$$\frac{A_e}{18}\begin{bmatrix} \frac{11}{3} & \frac{7}{6} & \frac{7}{6} \\ \frac{7}{6} & \frac{11}{3} & \frac{7}{6} \\ \frac{7}{6} & \frac{7}{6} & \frac{11}{3} \end{bmatrix}$$

If the approximation in equation A.10 is used to evaluate the integral in equation A.1, the result is a combination of the finite volume matrix and its lumped form as follows:

$$\beta\frac{A_e}{18}\begin{bmatrix} \frac{11}{3} & \frac{7}{6} & \frac{7}{6} \\ \frac{7}{6} & \frac{11}{3} & \frac{7}{6} \\ \frac{7}{6} & \frac{7}{6} & \frac{11}{3} \end{bmatrix} + (1-\beta)\frac{A_e}{18}\begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

(The lumped form is the sum of the row of the matrix placed on the diagonal and zeroes placed on the off-diagonals.) It is left to determine β so that the resulting matrix is the finite element mass matrix. Since β does not affect the off-diagonal terms in the lumped matrix because they are zero, it is sufficient to look at one of the off-diagonal terms in

the first matrix and equate it to the corresponding off-diagonal term in the finite element mass matrix. This leads to:

$$\beta \frac{A_e}{18} \frac{7}{6} = \frac{A_e}{12}$$

Solving for β yields $\beta = \frac{9}{7}$. Substituting this back into the previous matrix equation recovers the finite element mass matrix. A plot of the function in equation A.10 for a one dimensional case with $\beta = \frac{9}{7}$ is shown in figure A.2. The effect of this approximation is to place more weight on the off-diagonal terms in the element mass matrix and less on the diagonal terms.

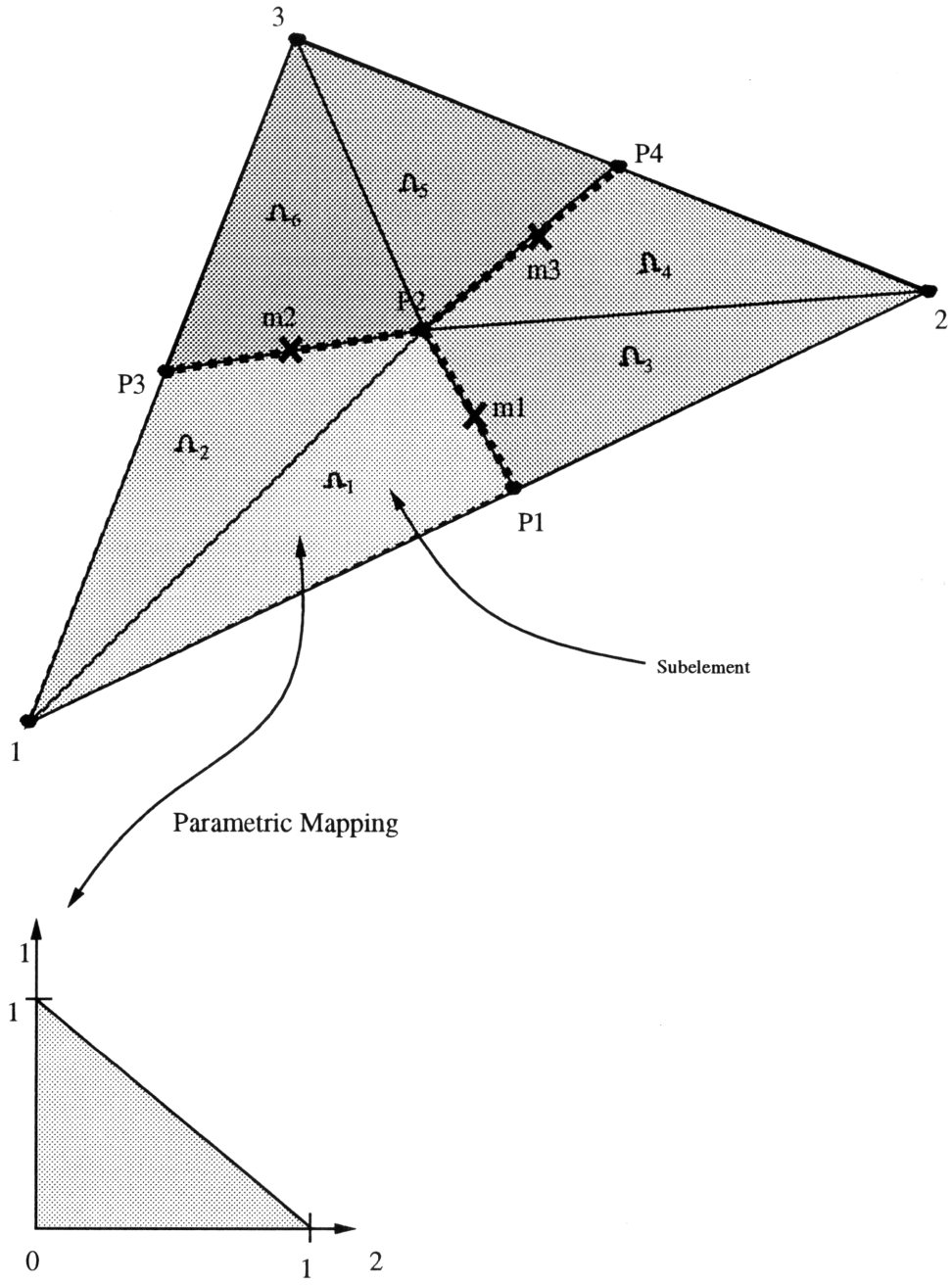


Figure A.1: Element Matrix Stencil

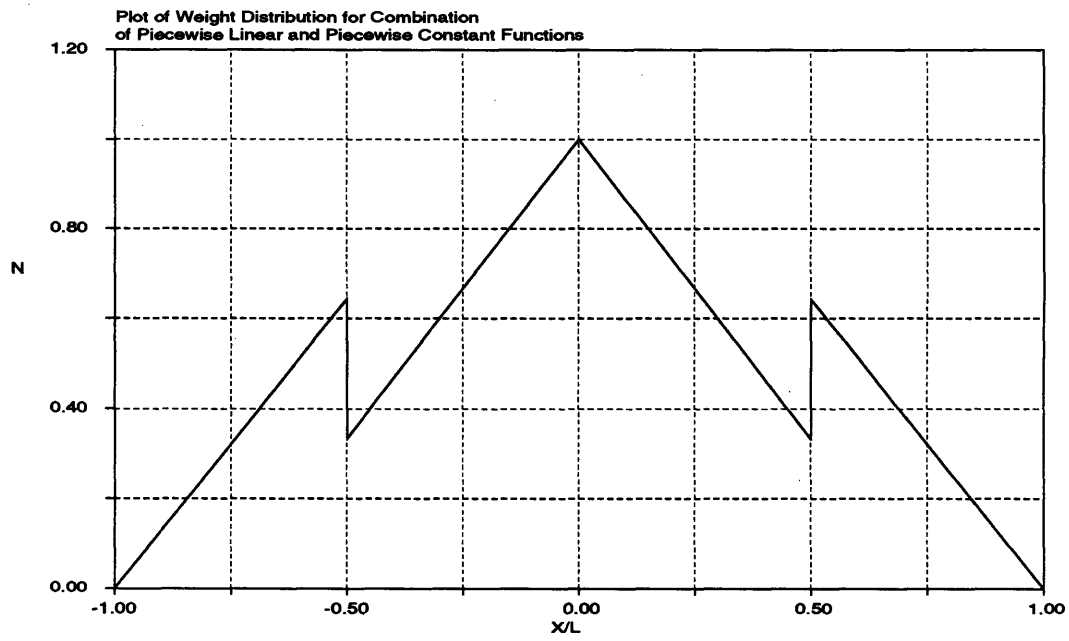


Figure A.2: Combination of Piecewise Linear and Piecewise Constant Functions

Appendix B
Fourier Analysis

B.1 Lumped Mass Matrix

The model convection equation for a two dimensional flow is given by:

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} + b\,\frac{\partial u}{\partial y} = 0 \tag{B.1}$$

where a and b are constant. Applying the finite volume algorithm developed in Chapter 3 to this equation yields:

$$A_j \frac{du_j}{dt} = -\sum_{k=1}^{n} (a\,c_{jk} - b\,l_{jk})\,\Delta u_{jk} \tag{B.2}$$

where n is the number of edges connected to node j. A semi-discrete Fourier analysis on a discretized set of equations assumes periodic boundary conditions and a uniform mesh [9]. Structured meshes provide this uniformity, but unstructured meshes have a similar analog. If a mesh of equilateral triangles is constructed as shown in figure B.1 then a similar analysis can be performed. A Fourier mode of the form:

$$u_j(t) = \hat{u}(l, m, t)\exp(i(lx + my)) \tag{B.3}$$

where:

$$\begin{aligned} l &= \textit{wave number in the } x - \textit{direction} \\ k &= \textit{wave number in the } y - \textit{direction} \\ \hat{u} &= \textit{amplitude of the } l, k \textit{ harmonic} \end{aligned}$$

may be substituted into equation B.2. This gives:

$$\begin{aligned} A_j \frac{d\hat{u}_j}{dt}\exp(i(lx + my)) &= \sum_{k=1}^{n} (a\,c_{jk} - b\,l_{jk})\,\hat{u}(l, m, t) \\ &\quad (\exp(i(l(x + \Delta x) + m(y + \Delta y))) - \exp(i(lx + my))) \\ &= \hat{u}(l, m, t)\exp(i(lx + my)) \sum_{k=1}^{n} (a\,c_{jk} - b\,l_{jk})\exp(i(\theta_x + \theta_y) - 1) \end{aligned}$$

where $\theta_x = l\Delta x$ and $\theta_y = m\Delta y$ are phase angles. This yields:

$$A_j \frac{du_j}{dt} = \hat{u}(l, m, t) \sum_{k=1}^{n} (a\,c_{jk} - b\,l_{jk})\exp(i(\theta x + \theta y) - 1) \tag{B.4}$$

The edge coefficients, c_{jk} and l_{jk}, can be explicitly determined for the uniform mesh of figure B.1. Figure B.2 shows all the elements which contain the volume surrounding node j. The points defining the control volume j are marked with triangular symbols in the figure. These points are restricted to the centroids of the elements and the midpoints of all the edges. In this figure the following geometric property is used:

$$\Delta s = \frac{\sqrt{3}}{6}\Delta$$

The definition of c_{jk} and l_{jk}, found in Chapter 3 and repeated here for convenience, is:

$$\begin{aligned} c_{k,j} &= N_j^{R1}\Delta y^{R1} + N_j^{R2}\Delta y^{R2} + N_j^{L1}\Delta y^{L1} + N_j^{L2}\Delta y^{L2} \\ l_{k,j} &= N_j^{R1}\Delta x^{R1} + N_j^{R2}\Delta x^{R2} + N_j^{L1}\Delta x^{L1} + N_j^{L2}\Delta x^{L2} \end{aligned}$$

corresponding to figure 3.6 for notation. Figure B.2 shows the sampling points marked with X's. The weights, N_k, which express the influence of node k on the flux for a given face of the volume can be determined. Since the elements are equilateral triangles, these weights are:

$$\begin{aligned} N_k^{R,1} &= \frac{1}{6} \\ N_k^{R,2} &= \frac{5}{12} \\ N_k^{L,1} &= \frac{5}{12} \\ N_k^{L,2} &= \frac{1}{6} \end{aligned}$$

The increments Δx and Δy of the flux faces are:

$$\begin{aligned} \Delta x^{L,1} &= -\Delta s \sin(\alpha_k) \\ \Delta y^{L,1} &= \Delta s \cos(\alpha_k) \\ \Delta x^{L,2} &= -\Delta s \sin(\alpha_k + 60°) \\ \Delta y^{L,2} &= \Delta s \cos(\alpha_k + 60°) \\ \Delta x^{R,1} &= -\Delta s \sin(\alpha_k - 60°) \\ \Delta y^{R,1} &= \Delta s \cos(\alpha_k - 60°) \\ \Delta x^{R,2} &= -\Delta s \sin(\alpha_k) \\ \Delta y^{R,2} &= \Delta s \cos(\alpha_k) \end{aligned}$$

where α_k is the angle of the edge with respect to the x-axis. Substituting the above expressions into the definitions of c_{jk} gives:

$$\begin{aligned} c_{jk} &= \Delta s \left[\frac{5}{12}\cos(\alpha_k) + \frac{1}{6}\cos(\alpha_k + 60°) + \frac{1}{6}\cos(\alpha_k - 60°) + \frac{5}{12}\cos(\alpha_k)\right] \\ &= \Delta s \cos(\alpha_k) \\ &= \frac{\sqrt{3}}{6}\Delta \cos(\alpha_k) \end{aligned}$$

Likewise:

$$\begin{aligned} l_{jk} &= \Delta s\left[\frac{5}{12}\sin(\alpha_k)+\frac{1}{6}\sin(\alpha_k+60°)+\frac{1}{6}\sin(\alpha_k-60°)+\frac{5}{12}\sin(\alpha_k)\right] \\ &= -\Delta\frac{\sqrt{3}}{6}\sin(\alpha_k) \end{aligned}$$

Therefore, the edge coefficients are functions of the edge length and of the angle of the edge with the x-axis. The equilateral mesh contains six edges connected to each node at the angles, $\alpha_k = 0°, 60°, 120°, 180°, 240°, 300°$. Using the following relation:

$$\sum_{k=1}^{6} c_{jk} = 0 \ , \ \sum_{k}^{6} l_{jk} = 0$$

equation B.2 can be written as:

$$A_j\frac{du_j}{dt} = \frac{\sqrt{3}}{6}\Delta\hat{u}(l,m,t)\sum_{k=1}^{6}(a\cos(\alpha_k)+b\sin(\alpha_k))\exp(i(\theta_x+\theta_y)) \qquad \text{(B.5)}$$

The phase angles θ_x and θ_y are also evaluated based on the angle α_k. Without the loss of generality, the node j in figure B.2 can be assumed to lie at the cartesian coordinate (0,0). Therefore, the edge increments Δx and Δy defining the phase angles are:

$$\begin{aligned} \Delta x &= \Delta\cos(\alpha_k) \\ \Delta y &= \Delta\sin(\alpha_k) \end{aligned}$$

The phase angles become:

$$\begin{aligned} \theta_x &= l\Delta\cos(\alpha_k) \\ \theta_y &= m\Delta\sin(\alpha_k) \end{aligned}$$

Substituting this in equation B.5 yields:

$$A_j\frac{du_j}{dt} = \frac{\sqrt{3}}{6}\Delta\hat{u}(l,m,t)\sum_{k=1}^{6}(a\ \cos(\alpha_k)+b\ \sin(\alpha_k))\exp\left(i\Delta(l\cos(\alpha_k)+m\sin(\alpha_k))\right) \qquad \text{(B.6)}$$

This summation is now evaluated by summing over all the edges attached to node j. Equation B.6 is simplified to:

$$A_j\frac{d\hat{u}_j}{dt} = -i\frac{\sqrt{3}}{6}\Delta\ \hat{u}(l,m,t)\left(a\ (\sin(\theta_x)+\sin(\frac{1}{2}\theta_x)\cos(\theta_y))+\sqrt{3}b\ \sin(\theta_y)\cos(\frac{1}{2}\theta_x)\right)$$

The area, A_j, is determined by the edge length of the elements, Δ, and is:

$$\begin{aligned} A_j &= \frac{1}{3}\sum_{k=1}^{6}\frac{\sqrt{3}}{4}\Delta^2 \\ &= \frac{\sqrt{3}}{2}\Delta^2 \end{aligned}$$

Substituting the expression for A_j into the previous equation and integrating gives:

$$\hat{u}(l,m,t) = \hat{u}(l,m,0)\exp\left(-i\omega(\theta_x,\theta_y)_n\, t\right) \tag{B.7}$$

where the function $\omega(\theta_x,\theta_y)_n$ is the numerical frequency:

$$\omega(\theta_x,\theta_y)_n = \frac{2}{3\Delta}\left(a\,\left(\sin(\theta_x)+\sin(\frac{1}{2}\theta_x)\cos(\theta_y)\right)+\sqrt{3}b\,\sin(\theta_y)\cos(\frac{1}{2}\theta_x)\right)$$

Equation B.7 represents the solution of the unstructured, lumped, finite volume algorithm (developed in Chapter 3) for a Fourier mode substituted into the model convection equation B.1. The exact solution is found by substituting the same Fourier mode into the model convection equation. This gives:

$$\frac{d\hat{u}}{dt} + i\hat{u}(l,m,t)\,(a\,l + b\,m) = 0$$

Integration of the above equation results in:

$$\hat{u}(l,m,t) = \hat{u}(l,m,0)\exp\left(-i(\frac{a}{\Delta x}\theta_x + \frac{b}{\Delta y}\theta_y)\right)$$

This is simplified further by noting that $\Delta x = \Delta$ and $\Delta y = \frac{\sqrt{3}}{2}\Delta$. Substituting this into the above expression gives the final result:

$$\hat{u_e}(l,m,t) = \hat{u_e}(l,m,0)\exp\left(-i\frac{1}{\Delta}(a\,\theta_x + \frac{2}{3}\sqrt{3}\theta_y)t\right) \tag{B.8}$$

The exact frequency is therefore:

$$\omega_e(\theta_x,\theta_y) = \frac{1}{\Delta}\left(a\,\theta_x + \frac{2}{3}\sqrt{3}b\,\theta_y\right)$$

A comparison is made between the frequency, ω_n, of the numerical solution for a given mode with the exact frequency, ω_e, in Chapter 3.

B.2 Finite Volume Mass Matrix

The finite volume mass matrix version of the scheme couples node j to the surrounding nodes. Equation B.2 takes the form:

$$\beta_j\frac{d\hat{u}_j}{dt} + \sum_{k=1}^{6}\beta_k\frac{d\hat{u}_k}{dt} \;=\; -\sum_{k=1}^{n}(a\,c_{jk} - b\,l_{jk})\,\Delta u_{jk} \tag{B.9}$$

where the coefficients, β, are the entries on the row of the mass matrix corresponding to node j. Referring to Appendix A and figure A.1 for the form of the finite volume mass

matrix, the entries are determined from the geometric relations found in the previous section. The entries are:

$$\begin{aligned} A_{\Omega_1} = A_{\Omega_2} = A_{\Omega_3} = A_{\Omega_4} = A_{\Omega_5} = A_{\Omega_6} &= \frac{\sqrt{3}}{24}\Delta^2 \\ N_{p1}^{(1)} = N_{p1}^{(2)} = N_{p3}^{(2)} = N_{p3}^{(3)} = N_{p2}^{(1)} = N_{p2}^{(3)} &= \frac{5}{12} \\ N_{p3}^{(1)} = N_{p2}^{(2)} = N_{p1}^{(3)} &= \frac{1}{6} \end{aligned}$$

which yields the following elemental matrix:

$$M_e = \frac{\sqrt{3}}{24}\Delta^2 \begin{bmatrix} \frac{22}{6} & \frac{7}{6} & \frac{7}{6} \\ \frac{7}{6} & \frac{22}{6} & \frac{7}{6} \\ \frac{7}{6} & \frac{7}{6} & \frac{22}{6} \end{bmatrix}$$

Summing over all the elements attached to node j gives the complete row of the system mass matrix for node j. Peforming this summation gives the following row multiplied by the column of nodes attached to and including node j:

$$\frac{\sqrt{3}}{24}\Delta^2 \begin{bmatrix} 22 & \frac{7}{3} & \frac{7}{3} & \frac{7}{3} & \frac{7}{3} & \frac{7}{3} & \frac{7}{3} \end{bmatrix} \begin{bmatrix} \frac{du_j}{dt} \\ \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \\ \frac{du_4}{dt} \\ \frac{du_5}{dt} \\ \frac{du_6}{dt} \end{bmatrix}$$

The β coefficients in equation B.9 are:

$$\begin{aligned} \beta_j &= 11\frac{\sqrt{3}}{12}\Delta^2 \\ \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 &= \frac{7}{3}\frac{\sqrt{3}}{24}\Delta^2 \end{aligned}$$

The evaluation of the extra terms on the left side of equation B.9 requires the same Fourier analysis as in the previous section for the lumped mass matrix. The analysis of the previous section was completely general for any node in the mesh. Therefore, the evaluation of these extra terms is the same as for node j, with the coordinates shifted relative to j. For instance, if the same analysis is performed for node one of the mesh in figure B.2 the following results:

$$A_1\frac{du_1}{dt} = \frac{\sqrt{3}}{6}\Delta\hat{u}(l,m,t)\exp(i\ l\ \Delta)\sum_{k=1}^{6}(a\ \cos(\alpha_k) + b\ \sin(\alpha_k))\exp\left(i\Delta(l\cos(\alpha_k) + m\sin(\alpha_k))\right)$$

This is the Fourier analysis about node j as shown in equation B.6 shifted by $\exp(i\ l\ \Delta)$. Therefore:

$$\frac{d\hat{u}_1}{dt} = \underbrace{\exp(i\ l\ \Delta)}_{shift\ relative\ to\ j}\frac{d\hat{u}_j}{dt}$$

Likewise, the other terms in equation B.9 are:

$$\begin{aligned}
\frac{d\hat{u}_2}{dt} &= \exp(i(\frac{1}{2}\theta_x + \frac{\sqrt{3}}{2}\theta_y))\frac{d\hat{u}_j}{dt} \\
\frac{d\hat{u}_3}{dt} &= \exp(i(-\frac{1}{2}\theta_x + \frac{\sqrt{3}}{2}\theta_y))\frac{d\hat{u}_j}{dt} \\
\frac{d\hat{u}_4}{dt} &= \exp(-i\theta_x)\frac{d\hat{u}_j}{dt} \\
\frac{d\hat{u}_5}{dt} &= \exp(-i(\frac{1}{2}\theta_x + \frac{\sqrt{3}}{2}\theta_y))\frac{d\hat{u}_j}{dt} \\
\frac{d\hat{u}_6}{dt} &= \exp(i(\frac{1}{2}\theta_x - \frac{\sqrt{3}}{2}\theta_y))\frac{d\hat{u}_j}{dt}
\end{aligned}$$

recalling the definitions of θ_x and θ_y from the previous section. Substituting these expressions into the left side of equation B.9 gives:

$$\frac{d\hat{u}_j}{dt} = -i\,\hat{u}(l,m,t)\underbrace{\frac{24}{\Delta}\frac{\left[a\,(\sin(\theta_x)+\sin(\frac{1}{2}\theta_x)\cos(\theta_y))+\sqrt{3}b\,\sin(\theta_y)\cos(\frac{1}{2}\theta_x)\right]}{\left[22+\frac{7}{3}(2\cos(\theta_x)+4\cos(\frac{1}{2}\theta_x)\cos(\theta_y))\right]}}_{numerical\ frequency,\ \omega_{fv}}$$

Integrating gives:

$$\hat{u}(l,m,t) = \hat{u}(l,m,0)\exp(-i\,\omega_{fv}\,t)$$

A comparison is made with the exact frequency in Chapter 3.

B.3 Finite Element Mass Matrix

The analysis of the scheme's finite element mass matrix form proceeds in the exact manner as for the finite volume mass matrix in the previous section. The finite element mass matrix for the equilateral mesh is:

$$\frac{\sqrt{3}}{24}\Delta^2\begin{bmatrix} 3 & \frac{3}{2} & \frac{3}{2} \\ \frac{3}{2} & 3 & \frac{3}{2} \\ \frac{3}{2} & \frac{3}{2} & 3 \end{bmatrix}$$

Summing for all the elements attached to node j leads to:

$$\frac{\sqrt{3}}{24}\Delta^2\begin{bmatrix} 18 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix}\begin{bmatrix} \frac{du_j}{dt} \\ \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \\ \frac{du_4}{dt} \\ \frac{du_5}{dt} \\ \frac{du_6}{dt} \end{bmatrix}$$

The β coefficients in equation B.9 are now known. Proceeding in the same manner as for the finite volume mass matrix gives the final result:

$$\frac{d\hat{u}_j}{dt} = -i\,\hat{u}(l,m,t)\;\underbrace{\frac{24}{\Delta}\frac{\left[a\,(\sin(\theta_x)+\sin(\tfrac{1}{2}\theta_x)\cos(\theta_y))+\sqrt{3}b\,\sin(\theta_y)\cos(\tfrac{1}{2}\theta_x)\right]}{\left[18+3\,(2\cos(\theta_x)+4\cos(\tfrac{1}{2}\theta_x)\cos(\theta_y))\right]}}_{numerical\ frequency,\ \omega_{fe}}$$

Integrating gives:

$$\hat{u}(l,m,t) = \hat{u}(l,m,0)\exp(-i\,\omega_{fe}\,t)$$

A comparison is made with the exact frequency in Chapter 3.

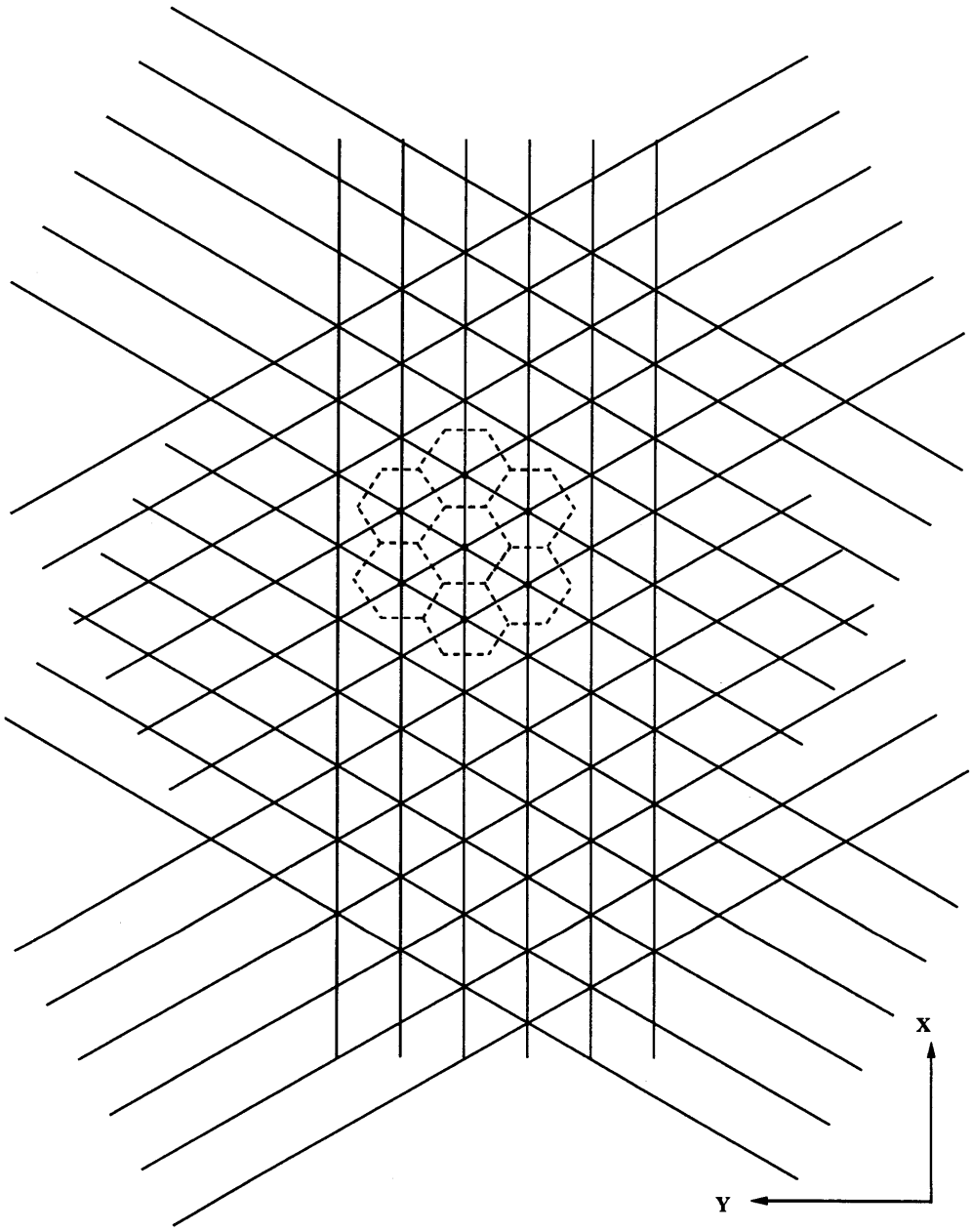


Figure B.1: Equilateral Mesh

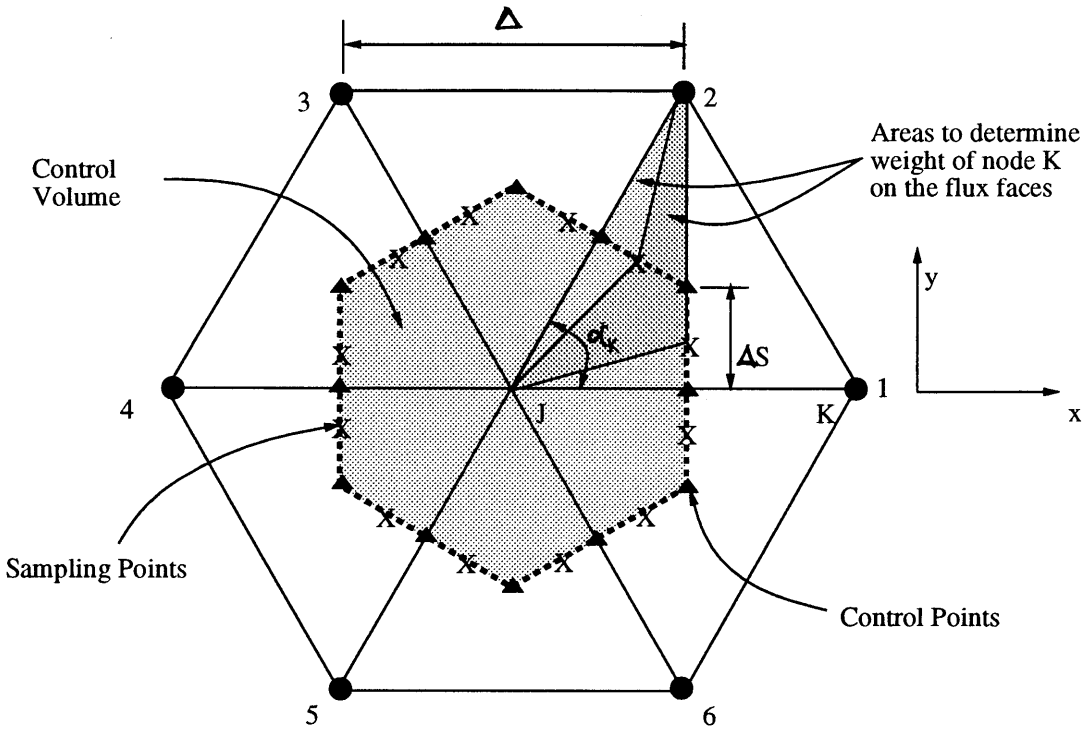


Figure B.2: Enlarged Mesh Patch with Subvolume

Appendix C
Boundary Conditions

Three types of boundaries are used in the numerical simulations: inviscid wall boundaries, inflow/outflow boundaries and periodic boundaries. The boundary conditions at the inflow/outflow boundaries are based on a one dimensional characteristic treatment of the flow at the boundary. Three types of inflow/outflow boundary conditions are used: specified pressure, specified total pressure and complete one dimensional characteristic treatment of flow. The periodic boundaries are used in the compressor simulation to enforce the periodicity of the solution between the blade passages.

C.1 Inviscid Boundary Conditions

The normal component of the velocity at a wall must be removed. This is accomplished using the following equation:

$$\vec{V}^* = \vec{V} - \left(\vec{V} \cdot \hat{n}\right) \hat{n}$$

where $\hat{n}$ is the unit normal vector of the boundary and $\vec{V}^*$ is the new velocity. This ensures the velocity is tangent to the boundary and that there is no flux across a solid boundary.

C.2 Inflow/Outflow Boundary Conditions

The inflow/outflow boundary conditions are applied using a one dimensional characteristic treatment of the flow at the boundaries. Assuming no variation in the y-direction and no shocks, the two dimensional, compressible Euler equations [16] become:

$$\frac{\partial}{\partial t}\begin{bmatrix} R^+ \\ R^- \\ v \\ S \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} u+a & 0 & 0 & \frac{-a^2}{\gamma(\gamma-1)} \\ 0 & u-a & 0 & \frac{-a^2}{\gamma(\gamma-1)} \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{bmatrix}\begin{bmatrix} R^+ \\ R^- \\ v \\ S \end{bmatrix} = 0$$

where S is the entropy, [u,v] are the cartesian velocity components and a is the speed of sound. R^+ and R^- are the characteristic quantities known as the Riemann invariants

defined as:

$$R^+ = u + \frac{2a}{\gamma - 1} \tag{C.1}$$

$$R^- = u - \frac{2a}{\gamma - 1} \tag{C.2}$$

If $\frac{\partial S}{\partial x}$ is considered small then a locally isentropic flow assumption can be made and the above matrix reduces to a diagonal matrix. The system of equations reduces to an uncoupled set of four equations, each one representing a nonlinear wave equation. R^+ and R^- are convected at the speed $u + a$ and $u - a$ respectively. R^+ and R^- are constant along the J^+ and J^- characteristic lines respectively in the x-t plane [9, 1] defined as:

$$J^+ : \ \frac{dx}{dt} = u + a \tag{C.3}$$

$$J^- : \ \frac{dx}{dt} = u - a \tag{C.4}$$

The other two quantities, the entropy and tangential velocity, are convected with speed u.

Four conditions may exist: subsonic inflow, subsonic outflow, supersonic inflow and supersonic outflow. Figure C.1 illustrates the four conditions and the slope of the characteristic lines in the x-t plane.

Once the flow is known at the boundary, it is possible to determine whether the boundary is an inflow or an outflow and in which direction the characteristic information is traveling. To use the locally one dimensional characteristic treatment, the normal component of the velocity, u, and the normal Mach number, M, are used at the boundary. The direction of the normal to the boundary is taken to be pointing out of the flow domain. With the normals defined in this way, the specification of the characteristic quantities for the four conditions is shown in table C.1. The flow variables at the boundary are updated from

Table C.1: Flux Boundary Conditions

Condition	Interior Specifies	Farfield Specifies
$u > 0,\ M < 1$	R^+, S, v	R^-
$u < 0,\ M < 1$	R^-	R^+, S, v
$u > 0,\ M > 1$	R^+, R^-, S, v	$None$
$u < 0,\ M > 1$	$None$	R^+, R^-, S, v

R^+, R^-, v and S in the following manner:

$$u^* = \frac{1}{2}(R^+ + R^-)$$

$$a^* = \frac{\gamma - 1}{4}(R^+ - R^-)$$

where $*$ represents the updated value. The other necessary variables are determined from the entropy and tangential velocity in the following manner:

$$S = \frac{p}{\rho^\gamma}$$

$$\begin{aligned} \rho^* &= \left[\frac{a^2}{\gamma S}\right]^{\frac{1}{\gamma-1}} \\ p^* &= \frac{\rho^* a^2}{\gamma} \\ v^* &= v \end{aligned}$$

This procedure is used to specify conditions such as back pressure and stagnation pressure, and is outlined in the next two sections.

C.2.1 Back Pressure

The compressor application requires the specification of a given back pressure at the outflow exit boundary. Since this boundary is a subsonic boundary, R^+, v and S are determined from the interior calculated solution. R^- is chosen so the pressure at the boundary is the specified pressure. The equations used are:

$$\begin{aligned} p &= \textit{specified quantity} \\ \rho &= \left[\frac{p}{S}\right]^{\frac{1}{\gamma}} \\ a &= \sqrt{\frac{\gamma p}{\rho}} \\ u &= R^+ - \frac{2a}{\gamma - 1} \end{aligned}$$

C.2.2 Specified Total Pressure

The inlet boundary condition for the compressor is a resevoir-type boundary condition. Stagnation quantities are known at the inlet. These quantities are the axial total pressure, axial stagnation temperature and flow angle. The one dimensional treatment at the inlet boundary is a good approximation since the flow is axial and one dimensional in the stationary frame of reference. The static thermodynamic quantities (i.e. p, ρ, T) are the same in either a rotating or stationary frame. Therefore, at the inlet, R^+ is:

$$R^+ = -\frac{2a_0}{\gamma - 1}$$

where a_0 is the stagnation speed of sound based on the total temperature and $u = 0$. This yields:

$$\begin{aligned} u &= \frac{1}{2}(R^+ + R^-) \\ a &= \frac{\gamma - 1}{4}(R^+ - R^-) \end{aligned}$$

The entropy condition is replaced by a stagnation pressure condition. The Mach number is defined from the updated speed of sound, a, and updated velocity, u, as:

$$M = \frac{u}{a}$$

The pressure is determined from the isentropic relation so that the axial total pressure, p_t, is the specified value. This becomes:

$$p = \frac{p_t}{\left[1 + \frac{\gamma-1}{2}M^2\right]^{\frac{\gamma}{\gamma-1}}}$$

The density is determined using the updated speed of sound and pressure:

$$\rho = \frac{\gamma p}{a^2}$$

This preserves the specified axial stagnation pressure at the boundary.

C.3 Periodic Boundary Conditions

Periodic boundary conditions are easily created with the algorithm developed in Chapter 3. The edge coefficients can be used to create periodic boundaries which are transparent to the flow solver. These coefficients define the integral about a part of the control volume surrounding the node of a given edge. It is possible to complete the control volume for a periodic node by adding the corresponding portion of the control volume from the other periodic node. Since the algorithm is developed for an edge-based data structure, it becomes easier to work with periodic edges rather than periodic nodes.

Figure C.2 shows two periodic nodes and their attached elements. Since the edge coefficients, (c_{jk}, l_{jk}, c_{kj} and l_{kj}), contain the integration paths, finding the periodic edges and adding their corresponding coefficients completes the integration. The integration paths along the boundaries cancel since they are in opposing directions. After all the periodic edges (edges E1 and E2) are determined and their corresponding edge coefficients added, the edges that define one of the periodic boundaries are removed from the mesh (edge E1). The nodes defining the removed set of periodic edges are also removed (nodes N1 and N2). The edges attached to these nodes (edges E3, E4, E5, E6 and E7) are modified to correspond to the other periodic node. This causes the algorithm to send the flux contribution from the modified edges to the periodic node not removed from the database. This removes some degrees of freedom from the system.

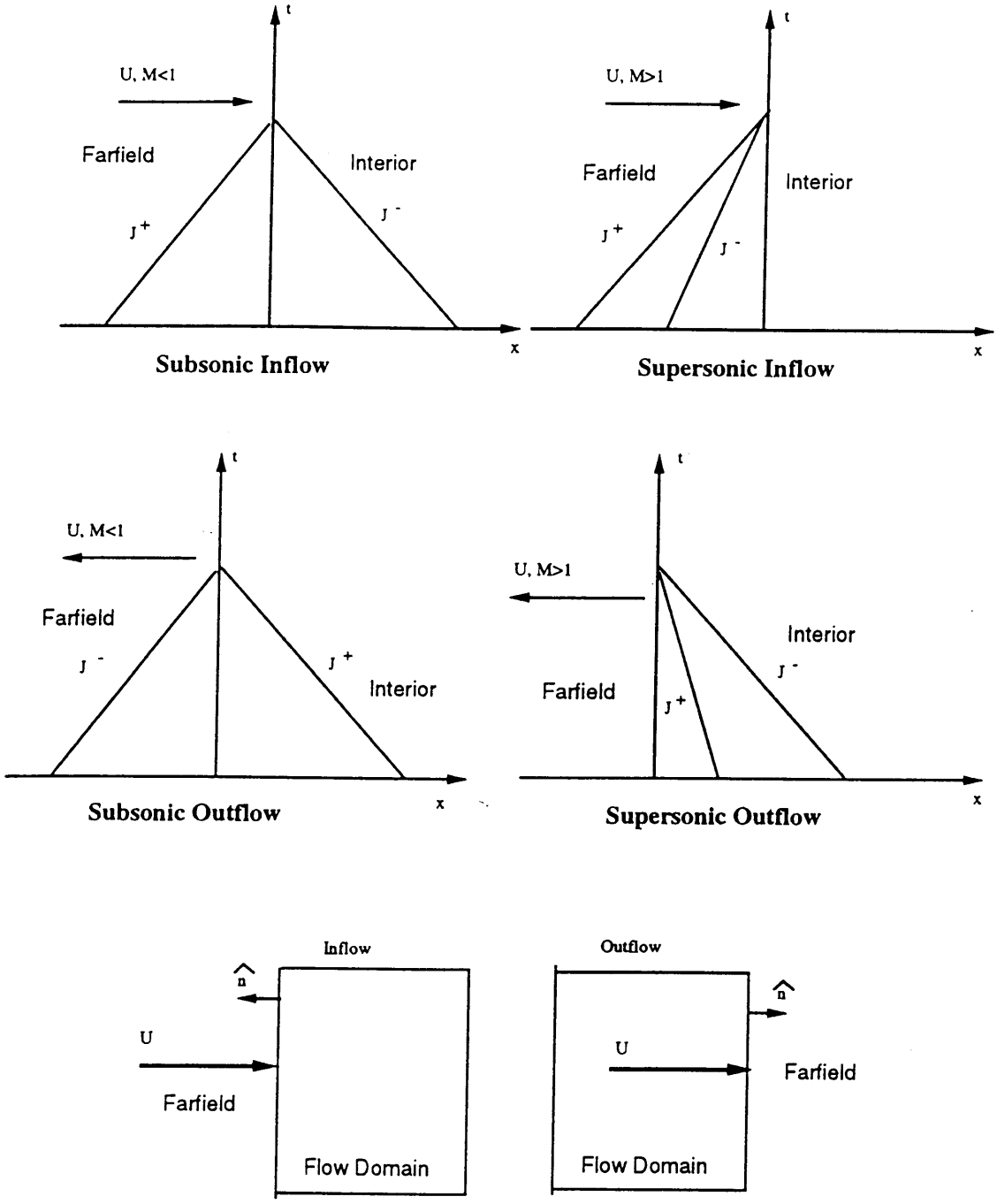


Figure C.1: Characteristics

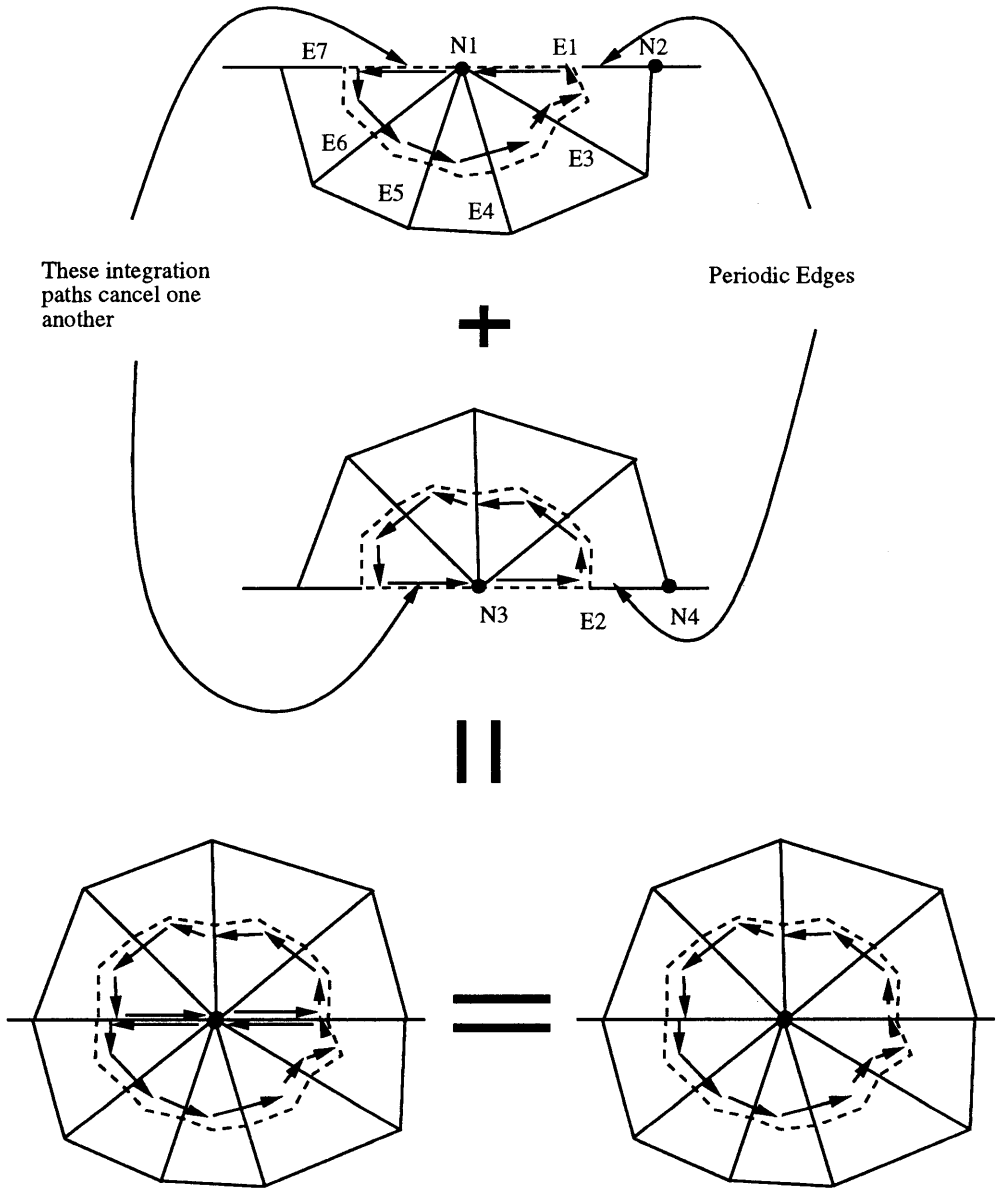


Figure C.2: Periodic Boundary Conditions

Appendix D
Test Case Solutions

D.1 Exact Solution to Shock Tube Problem

The exact solution of the shock tube problem is determined from a one dimensional analysis. This is detailed in [1, pp176–205]. Figure D.1 gives a schematic of the problem with reference to the following equations needed to solve the problem. The one dimensional, normal shock relations for an inviscid flow in the frame of reference of the shock are [1]:

$$\begin{aligned} \textit{Mass}: \quad & \rho_1 U_s &&= \rho_2 \underbrace{(U_s - U_i)}_{\textit{velocity relative to shock}} \\ \textit{Momentum}: \quad & p_1 + \rho_1 U_s^2 &&= p_2 + \rho_2 (U_s - U_i)^2 \\ \textit{Energy}: \quad & h_1 + \frac{U_s^2}{2} &&= h_2 + \frac{(U_s - U_i)^2}{2} \end{aligned}$$

where ρ = *density*, p = *pressure*, U_s = *shock speed*, U_i = *contact discontinuity speed* and h = *enthalpy*.

The shock speed for a calorically perfect gas (i.e. $e = c_v T$ and $p = \rho RT$) is determined from the normal shock equations [1, pp177].

$$U_s = a_1 \sqrt{\frac{\gamma + 1}{2\gamma} \left(\frac{p_2}{p_1} - 1 \right) + 1} \tag{D.1}$$

where $a_1 = \sqrt{\gamma RT}$ is the speed of sound of the stagnant gas of region 1.

The speed of the contact discontinuity between regions one and two is determined from the conservation of mass.

$$U_i = \frac{a_1}{\gamma} \left(\frac{p_2}{p_1} - 1 \right) \sqrt{\frac{\frac{2\gamma}{\gamma+1}}{\frac{p_2}{p_1} + \frac{\gamma-1}{\gamma+1}}} \tag{D.2}$$

The local flow speed in the expansion region of the shock tube is determined from nonlinear wave theory and the method of characterstics [1, pp235].

$$u = \frac{2}{\gamma + 1} \left(a_4 + \frac{x}{t} \right) \tag{D.3}$$

where a_4 is the speed of sound in region four, x is the position and t is the elapsed time after the diaphragm is broken.

The temperature in the expansion region as a function of the local flow speed is determined from the adiabatic relation:

$$\frac{T}{T_4} = \left[1 - \frac{\gamma - 1}{2}\left(\frac{u}{a_4}\right)\right] \tag{D.4}$$

The pressure and density are determined from the isentropic relation:

$$\frac{p_4}{p} = \left(\frac{T_4}{T}\right)^{\frac{\gamma}{\gamma-1}} = \left(\frac{\rho_4}{\rho}\right)^{\gamma}.$$

The matching conditions between the regions of the shock tube are:

$$p_3 = p_2$$
$$u_3 = u_2 = U_i$$

A relationship between the shock strength and the initial static pressure ratio $\frac{p_4}{p_1}$ is determined from the normal shock relations and the matching conditions [1, pp237]:

$$\frac{p_4}{p_1} = \frac{p_2}{p_1}\left[1 - \frac{(\gamma-1)(\frac{a_1}{a_4})(\frac{p_2}{p_1} - 1)}{\sqrt{2\gamma\left[2\gamma + (\gamma+1)(\frac{p_2}{p_1} - 1)\right]}}\right] \tag{D.5}$$

assuming that $\gamma_1 = \gamma_4 = \gamma$.

The complete solution to the shock tube problem using the above equations is summarized in the following steps:

1. Specify p_4, p_1, ρ_4, ρ_1. This determines a_4 and a_1.
2. Determine $\frac{p_2}{p_1}$ from equation D.5 using an iterative solver (e.g. a Newton-Raphson solver). This determines the shock speed from equation D.1.
3. Given a time, t, the position of the shock is determined from the shock speed and its initial position at the diaphragm location. The position of the head of the expansion wave is also known because its speed is $-a_4$.
4. The speed of the contact discontinuity between regions two and three is determined from equation D.2. Therefore, the position of the tail of the expansion is determined since its speed is $U_i - a_3$ and a_3 is:

$$a_3 = \sqrt{\gamma \frac{\frac{p_3}{p_1} p_1}{\frac{\rho_3}{\rho_1}\rho_1}}$$

The matching conditions give $\frac{p_3}{p_1} = 1$ and $\frac{\rho_3}{\rho_1}$ is determined from the isentropic expansion relation in equation D.4. (Note $\frac{\rho_3}{\rho_1} = \frac{\rho_3}{\rho_2}\frac{\rho_2}{\rho_1}$ with $\frac{\rho_2}{\rho_1}$ determined from the normal shock relations.)

5. The positions of regions one, two, three and four are now known. The conditions in regions one and four are the initial conditions. Region two is determined from normal shock relations. The expansion in region three is determined from equation D.3 for the velocity and equation D.4 for the remaining thermodynamic quantities. Finally, the remainder of region three is known from the previous steps.

D.2 Exact Solution to Density Perturbation

The exact solution to the density perturbation problem is determined from the linearized Euler equations [21]. For the Gaussian perturbation and a freestream velocity in the x-direction, the exact solution is:

$$\begin{aligned} \rho(x,y,t) &= \rho_0 + \Delta\rho \exp(-\alpha((x - Mt) + y)) \\ p(x,y,t) &= \textit{constant} \\ u(x,y,t) &= \textit{constant} \\ v(x,y,t) &= \textit{constant} \end{aligned}$$

where (x_0, y_0) is the initial location of the perturbation; ρ_0 is the initial density in the unperturbed flow; $\Delta\rho$ is the perturbation magnitude; M is the freestream Mach number in the x-direction; p is the pressure; $[u, v]$ are the cartesian velocity components; and t is the elapsed time. The term α is defined as:

$$\alpha = \frac{\ln(2)}{b^2}$$

where b is the Gaussian half width of the perturbation. The square perturbation solution is convected with the freestream flow velocity in the same manner.

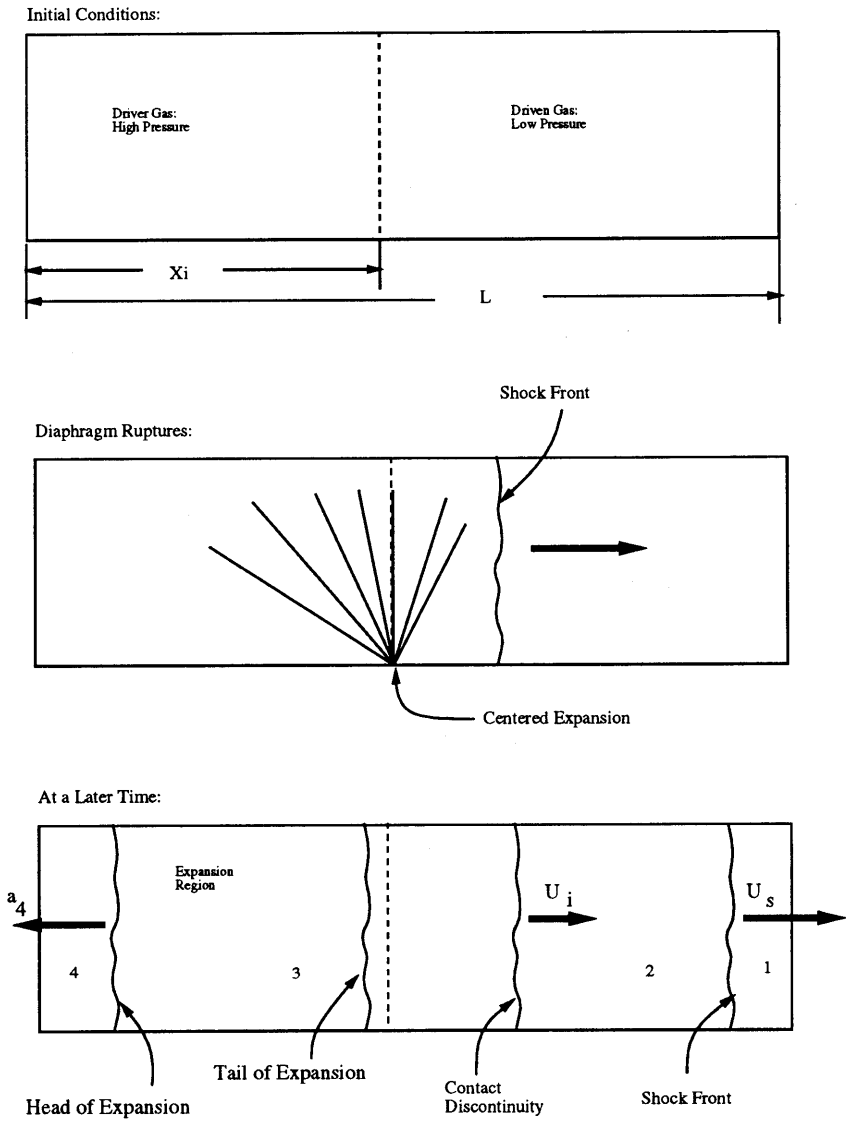


Figure D.1: Schematic of Shock Tube

Appendix E
Non-dimensionalization of the Euler Equations

The variables in the Euler equations are non-dimensionalized using the parameters outlined in table E.1. The paramters ρ_∞, a_∞ and L_{ref} are a reference density, speed and length respectively. These are usually chosen to be the freestream values for external flows. Rewriting the Euler equations in these non-dimensional variables gives the exact same set of equations in non-dimensional form. The Euler equations become:

$$\int\int_\Omega \frac{\partial \vec{U}}{\partial t} \partial\Omega + \oint_S (\vec{F} \cdot n_x + \vec{G} \cdot n_y) dS = 0$$

where $\vec{U}, \vec{F}$ and $\vec{G}$ are now:

$$\vec{U} = \begin{bmatrix} \rho^* \\ \rho^* u^* \\ \rho^* v^* \\ \rho^* E^* \end{bmatrix} \quad \vec{F} = \begin{bmatrix} \rho^* u^* \\ \rho^* u^{*2} + p^* \\ \rho^* u^* v^* \\ \rho^* u^* H^* \end{bmatrix} \quad \vec{G} = \begin{bmatrix} \rho^* v^* \\ \rho^* u^* v^* \\ \rho^* v^{*2} + p^* \\ \rho^* v^* H^* \end{bmatrix}$$

Table E.1: Non-dimensional Variables for the Euler Equations

Variable	Non-dimensional Parameter	Non-dimensional Variable
ρ	ρ_∞	ρ^*
p	$\rho_\infty\ a_\infty^2$	p^*
u,v	a_∞	$u^*,\ v^*$
E,H	a_∞^2	$E^*,\ H^*$
t	$\frac{L_{ref}}{a_\infty}$	t^*