AD-A256 405

WL-TR-92-7039

A Multigrid Approach to Embedded-Grid Solvers

Rudy A. Johnson

Wright Laboratory, Armament Directorate
Weapon Flight Mechanics Division
Aerodynamics Branch
Eglin AFB FL 32542-5000

DTIC
ELECTE
SEP 1 1 1992
C

AUGUST 1992

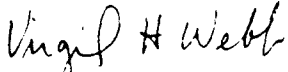INTERIM REPORT FOR PERIOD JANUARY 1991 - APRIL 1992

92-24976

92 9 1

WRIGHT LABORATORY, ARMAMENT DIRECTORATE
Air Force Material Command ∎ United States Air Force ∎ Eglin Air Force Base

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Virgil H Webb

VIRGIL H. WEBB, Col. USAF
Chief
Weapon Flight Mechanics Division

Even though this report may contain special release rights held by the controlling office, please do not request copies from the Wright Laboratory, Armament Directorate. If you qualify as a recipient, release approval will be obtained from the originating activity by DTIC. Address your request for additional copies to:

Defense Technical Information Center
Cameron Station
Alexandria VA 22304-6145

If your address has changed, if you wish to be removed from our mailing list, or if your organization no longer employs the addressee, please notify WL/MNAA , Eglin AFB FL 32542-5000, to help us maintain a current mailing list.

Do not return copies of this report unless contractual obligations or notice on a specific document requires that it be returned.

NOTE: This document was previously approved for public release (AFDTC/PA 92-148). It is releasable to the National Technical Information Service (NTIS), where it will be available to the general public, including foreign nationals.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE August 1992	3. REPORT TYPE AND DATES COVERED interim, Jan 91 to Apr 92

4. TITLE AND SUBTITLE A Multigrid Approach to Embedded-Grid Solvers	5. FUNDING NUMBERS PE 62602F PR 2567 TA 03 WU 29
6. AUTHOR(S) Rudy A. Johnson	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Wright Laboratory, Armament Directorate Weapon Flight Mechanics Division Aerodynamics Branch (WL/MNAA) Eglin AFB FL 32542-5000	8. PERFORMING ORGANIZATION REPORT NUMBER WL-TR-92-7039

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Same as above	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

This work published previously as a Master's Thesis at the University of Florida.

Not edited by TESCO, Inc.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE A

13. ABSTRACT (Maximum 200 words)

The explicit first order flux difference splitting (FDS) is used to solve the equations governing inviscid fluid flow on a single grid for the 5 degree ramp and the 5 degree ramp near a flat plate. Calculations were made for the Mach 2 case using local time stepping at a Courant number of 0.96.

The multigrid full approximation scheme (FAS) is applied to this non-linear problem to provide increased convergence rates and reduced central processing unit (CPU) time requirements.

Several systems of embedded grids are implemented to provide increased accuracy near shock waves. Embedded grids were aligned with the shocks to take advantage of the excellent shock capturing capability of the FDS scheme.

The nonaligned multigrid is introduced to provide increased convergence for the embedded grid systems by treating the individual grids as levels in the multigrid solution procedure. This technique is able to converge 69% faster than the embedded grid procedure for the 5 degree ramp, and 28% faster for a more complex reflected shock case.

14. SUBJECT TERMS Computational Fluid Dynamics (CFD), Multigrid, Chimera Embedded/Overset grids, Nonaligned Multigrid Euler Equations	15. NUMBER OF PAGES 86
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

PREFACE

This program was conducted in-house by the Computational Fluid Dynamics Section of Wright Laboratory, Armament Directorate, Eglin Air Force Base FL 32542-5000. Rudy Johnson managed this program for the Armament Directorate. This report covers the period from January 1991 to April 1992.

iii

PREFACE

This program was conducted in-house by the Computational Fluid Dynamics Section of Wright Laboratory, Armament Directorate, Eglin Air Force Base FL 32542-5000. Rudy Johnson managed this program for the Armament Directorate. This report covers the period from January 1991 to April 1992.

iii

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In recent years much of the work done in computational fluid dynamics (CFD) has been to produce high resolution flow solvers [25, 31, 32] and to develop domain decomposition techniques in order to accurately model the aerodynamics of complex geometries [4, 5, 11, 12, 13, 24]. Although these high resolution solvers can provide good answers they are still grid dependent and will usually require significant computer time to converge for all but the simplest problems.

Grid dependency can be taken care of by generating "good" grids, where "good" grid simply means any grid that causes the numerical solution to converge rapidly to the actual physical solution for a given geometry. Since most real geometries are not necessarily simple it is difficult, if not impossible, to generate a single "good" grid. Several techniques have been developed to allow the domain of interest to be approximated by a combination of smaller "good" grids as opposed to using a single complex grid. These domain decomposition techniques have been applied to the analytic solution of partial differential equations for many years.

One of the simpler forms of domain decomposition is referred to as composite blocked or multiblocked grid structures [4, 28, 29, 30]. In this method each block (separate grid) covers a given region of the domain and the interfaces between these blocked regions match physical grid points exactly, although the slopes of the grid lines through these boundaries may be discontinuous. In fact when a singularity is present on a block boundary multiple lines from one block may merge at the singularity and become a single grid line in the adjoining block.

Another widely used method is patched or zonal grids [18, 23, 24]. For zonal grids the boundaries between the zones must coincide; however, the grid lines i tersecting

these boundaries do not cross the boundary in a continuous manner. In fact the common boundary between zones may not contain the same number of points. One important restriction with this grid system is that a sufficient number of points must be on the zonal boundaries to maintain conservation principles across such boundaries. Blocked grids are a special case of zonal grids.

An even more general domain decomposition method is referred to as embedded, overset, or Chimera grids [3, 5, 13, 14, 22]. When using embedded grids no set boundary correspondence is required between grids. Grid boundaries that lie in the interior of the domain are set by interpolation. This method also requires sufficient grid points for intergrid communication to occur in a conservative manner. Grid points that are within the overlapped region and points that fall inside of solid surfaces are not used as a part of the solution. These points are referred to as blanked or I-blanked. Since these grids are generated independently they can be added, deleted, or moved in order to make configuration changes without having to regrid the entire domain. A great deal of work has gone into automating the search for the intergrid interpolation stencils and the grid movement procedure for grids in relative motion [13, 22]. Both zonal and blocked grids are special cases of embedded grids.

These domain decomposition techniques allow the solution of more complex geometries; however, the amount of computer time required usually increases with the number of artificial boundaries introduced into the domain. This has lead to the development of several techniques for improving convergence of the flow solvers.

One method for improving convergence of iterative schemes for solving systems of equations is multigrid [2, 9, 10, 15, 16, 21]. Regular multigrid simply uses several levels of increasingly coarser grids covering the same domain to solve a set of algebraic equations. The solution is generated on the finest level and corrections to the solution are calculated on the coarser levels. Multigrid techniques applied in CFD have shown significant improvements in convergence over single level methods [2, 3, 20, 27, 33].

Multigrid techniques have also been developed to provide increased resolution in portions of the domain. These local refinement techniques [9, 11, 15, 21] usually generate the local grid levels by quartering (in two dimensions) existing cells in the global grid and storing this refinement as a part of the next finer level. Boundary information on the local level is usually obtained by interpolation from a lower level. Accurate communication between these levels is critical [21]. These methods are typically automated such that additional levels are added during the solution process.

Multigrid has also been used to provide increased convergence on blocked and embedded grid systems. When this technique is applied to an embedded grid system, typically a complete multigrid cycle is made in each grid independent of the other grids [3] and the communications between grids is achieved on the finest grid level. Special treatment of blanked points is required on the the coarser levels. Numerical results show that when multigrid is applied to the blocked grid system it is better to make calculations and communicate between blocks on each level than to perform a full multigrid cycle in each block independently [33]. This would seem to indicate that if multigrid could be applied in a way that would allow the lower levels of the embedded grids to communicate that even better convergence rates would result. Reference [17] applies multigrid to an embedded grid system to solve an elliptic problem in this fashion.

The purpose of this study is to show the feasibility of using a nonaligned multigrid technique to both communicate between embedded grids and to simultaneously increase the convergence rate. Nonaligned multigrid [15] can be thought of as the global application of multigrid to a general embedded grid system. In this case the only blanked points will lie inside of solid surfaces and all communication between grids and grid levels will be done through the multigrid techniques. This means that overlapped regions will contain the solution on one grid and the other grids occupying the same region will be used to generate the corrections to that solution. The

nonaligned multigrid is basically a local refinement technique with the local levels generated independently of the global level. This method will be applied to the explicit flux difference splitting [26, 32] approximation of the two dimensional Euler equations for inviscid compressible flow.

In Chapter 2 the nondimensional Euler equations and the flux difference splitting (FDS) algorithm [26, 32] are presented in terms of a body conforming curvilinear coordinate system. These partial differential equations are stated in strong conservation law form and in quasi-linear form. The latter of these forms is necessary in order to obtain the eigenvalues and eigenvectors of the flux Jacobian matrices [4, 31] that are needed for the FDS algorithm. Stability requirements for the explicit FDS and initial conditions are stated. A phantom point formulation is also examined for the solid surface and supersonic inflow/outflow boundary conditions.

Multigrid methods are discussed in Chapter 3 and the formulation for the Full Approximation Scheme (FAS, sometimes referred to as the full approximation storage) [2, 9, 21] is presented. The use of bi-linear interpolation, applied in the computational domain, for the restriction and prolongation operations is examined. Grid scheduling and boundary conditions for different levels will also be discussed.

Chapter 4 presents a discussion of the embedded grid approach. This includes an examination of the stencil jumping, boundary interpolation, and blanking procedures.

In Chapter 5 the nonaligned multigrid (NAM) approach is described using FAS for the nonlinear problem. The use of bi-linear interpolation, applied in the physical domain, for the restriction and prolongation operations is examined. Grid level ordering and boundary conditions will also be discussed for the nonaligned grids.

A discussion of the results for the $5°$ ramp and the $5°$ ramp near a flat plate is contained in Chapter 6. These include comparisons between single grid, multigrid, embedded grid, and nonaligned multigrid calculations. Comparisons will be made by looking at solutions, convergence histories, and CPU times for various methods.

Conclusions and observations about the numerical techniques used in this study are presented in Chapter 7.

CHAPTER 2
EULER EQUATIONS AND FLUX DIFFERENCE SPLITTING

This chapter states the strong conservation law form of the Euler equations in two dimensions in terms of a body conforming curvilinear coordinate system. The equation is converted to the quasi-linear form and the eigenvalues and eigenvectors are given for the flux Jacobian matrices. The first order FDS algorithm used to approximate the Euler equations and the resulting stability condition are presented. Time stepping, initial conditions, and boundary conditions used in this study are also discussed.

Euler Equations

The equations of motion in strong conservation law form for inviscid compressible flow are presented here for two dimensions. These equations are nondimensional and are expressed in terms of a general curvilinear (body fitted) coordinate system [4, 26, 31].

$$\frac{\partial Q}{\partial t} + \frac{\partial F_\xi}{\partial \xi} + \frac{\partial F_\eta}{\partial \eta} = 0 \tag{1}$$

Where the primary variables Q and the flux vectors F_k (k is ξ or η denoting the curvilinear component) are given by

$$Q = J \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} ; F_k = J \begin{bmatrix} \rho\theta_k \\ \rho u\theta_k + k_x p \\ \rho v\theta_k + k_y p \\ (e + p)\theta_k \end{bmatrix} . \tag{2}$$

In this notation ρ is the fluid density, u and v are the Cartesian components of velocity, p is the pressure, and e is the total energy per unit volume. Since there are

only four equations and five unknowns the perfect gas assumption is made in order to relate the energy and pressure.

$$e = \frac{p}{\gamma - 1} + \rho \frac{(u^2 + v^2)}{2} \tag{3}$$

The remainder of the terms in Eq. 2 are the metrics, k_x and k_y, of the transformation from Cartesian to curvilinear coordinates, the Jacobian of this transformation, J, and the contravariant velocity components θ_k.

$$\xi_x = J^{-1} y_\eta \tag{4}$$

$$\xi_y = -J^{-1} x_\eta \tag{5}$$

$$\eta_x = -J^{-1} y_\xi \tag{6}$$

$$\eta_y = J^{-1} x_\xi \tag{7}$$

$$J = x_\xi y_\eta - y_\xi x_\eta \tag{8}$$

$$\theta_k = u k_x + v k_y \tag{9}$$

The nondimensionalization [4, 26] of Eq. 1 is obtained from the following definitions. First, all dimensional quantities are indicated by a " ˆ " and a convenient length $\hat{l}$ is defined. The quantities subscripted by ∞ denote reference values in the undisturbed gas.

$$\begin{array}{cccc} x = \frac{\hat{x}}{\hat{l}}, & y = \frac{\hat{y}}{\hat{l}}, & t = \frac{\hat{t} \hat{c}_\infty}{\hat{l}}, & \rho = \frac{\hat{\rho}}{\hat{\rho}_\infty} \\ u = \frac{\hat{u}}{\hat{c}_\infty}, & v = \frac{\hat{v}}{\hat{c}_\infty}, & e = \frac{\hat{e}}{\hat{\rho}_\infty \hat{c}_\infty^2}, & p = \frac{\hat{p}}{\hat{\rho}_\infty \hat{c}_\infty^2} \end{array} \tag{10}$$

Where $\hat{c}_\infty = (\gamma \hat{p}_\infty / \hat{\rho}_\infty)^{1/2}$, is the speed of sound in the undisturbed gas. Applying this nondimensionalization to the Euler equations (Eq. 1) yields exactly the same equation in terms of nondimensional quantities.

The quasi-linear form of the Euler equations is obtained from Eq. 1 by defining the flux Jacobian matrices as $A_k = \partial F_k / \partial Q$ and applying the chain rule.

$$\frac{\partial Q}{\partial t} + A_\xi \frac{\partial Q}{\partial \xi} + A_\eta \frac{\partial Q}{\partial \eta} = 0 \tag{11}$$

When evaluated A_k becomes

$$A_k = \begin{bmatrix} 0 & k_x & k_y & 0 \\ k_x\phi - u\theta_k & \theta_k + k_x(2-\gamma)u & k_y u - k_x(\gamma-1)v & k_x(\gamma-1) \\ k_y\phi - v\theta_k & k_x v - k_y(\gamma-1)u & \theta_k + k_y(2-\gamma)v & k_y(\gamma-1) \\ (\phi+\beta)\theta_k & -k_x\beta - u(\gamma-1)\theta_k & -k_y\beta - v(\gamma-1)\theta_k & \gamma\theta_k \end{bmatrix}, \quad (12)$$

with $\beta = \phi - \gamma e/\rho$, and

$$\phi = \frac{(\gamma-1)}{2}(u^2 + v^2). \quad (13)$$

The eigenvalues and the left and right eigenvectors of this matrix are needed in order to implement the FDS algorithm presented later in this chapter. In the following equations λ_k are the eigenvalues, the columns of T_k are the right eigenvectors, and the rows of T_k^{-1} are the left eigenvectors [4, 31].

$$\lambda_k^1 = \theta_k \quad (14a)$$

$$\lambda_k^2 = \theta_k \quad (14b)$$

$$\lambda_k^3 = \theta_k + c(k_x^2 + k_y^2)^{1/2} \quad (14c)$$

$$\lambda_k^4 = \theta_k - c(k_x^2 + k_y^2)^{1/2} \quad (14d)$$

$$T_k = \begin{bmatrix} 1 & 0 & \frac{\rho}{\sqrt{2}c} & \frac{\rho}{\sqrt{2}c} \\ u & -\rho\tilde{k}_y & \frac{\rho}{\sqrt{2}}(\frac{u}{c} + \tilde{k}_x) & \frac{\rho}{\sqrt{2}}(\frac{u}{c} - \tilde{k}_x) \\ v & \rho\tilde{k}_x & \frac{\rho}{\sqrt{2}}(\frac{v}{c} + \tilde{k}_y) & \frac{\rho}{\sqrt{2}}(\frac{v}{c} - \tilde{k}_y) \\ \frac{\phi}{\gamma-1} & \rho(v\tilde{k}_x - u\tilde{k}_y) & \frac{\rho}{\sqrt{2}}(\tilde{\theta}_k + \frac{\phi+c^2}{(\gamma-1)c}) & \frac{\rho}{\sqrt{2}}(\frac{\phi+c^2}{(\gamma-1)c} - \tilde{\theta}_k) \end{bmatrix} \quad (15)$$

$$T_k^{-1} = \begin{bmatrix} \frac{c^2-\phi}{c^2} & \frac{u(\gamma-1)}{c^2} & \frac{v(\gamma-1)}{c^2} & -\frac{(\gamma-1)}{c^2} \\ \frac{u\tilde{k}_y - v\tilde{k}_x}{\rho} & -\frac{\tilde{k}_y}{\rho} & \frac{\tilde{k}_x}{\rho} & 0 \\ \Upsilon(\phi - \tilde{\theta}_k c) & \Upsilon(\tilde{k}_x c - (\gamma-1)u) & \Upsilon(\tilde{k}_y c - (\gamma-1)v) & \Upsilon(\gamma-1) \\ \Upsilon(\phi + \tilde{\theta}_k c) & -\Upsilon(\tilde{k}_x c + (\gamma-1)u) & -\Upsilon(\tilde{k}_y c + (\gamma-1)v) & \Upsilon(\gamma-1) \end{bmatrix} \quad (16)$$

Where "~" indicates division by $(k_x^2 + k_y^2)^{1/2}$ and

$$c = \sqrt{\gamma p/\rho} \quad (17)$$

$$\Upsilon = \frac{1}{\sqrt{2}\rho c}. \quad (18)$$

First Order Flux Difference Splitting

Now that the eigenvectors and eigenvalues of the flux Jacobian matrix have been defined the flux difference splitting can be used to approximate Eq. 1. The following equation is the explicit finite volume discretization [25, 26, 32] used for this study.

$$\frac{q_{i,j}^{n+1} - q_{i,j}^{n}}{\Delta t} + \frac{(\overline{f}_\xi)_{i+1/2,j}^{n} - (\overline{f}_\xi)_{i-1/2,j}^{n}}{\Delta \xi} + \frac{(\overline{f}_\eta)_{i,j+1/2}^{n} - (\overline{f}_\eta)_{i,j-1/2}^{n}}{\Delta \eta} = 0 \qquad (19)$$

Where $\overline{f}_k$ is the numerical flux in the k direction and $\Delta \xi = \Delta \eta = 1$ for the uniform curvilinear coordinate system. For a first order approximation the numerical flux is given by

$$(\overline{f}_\xi)_{i+1/2,j} = (f_\xi)_{i,j} + \sum_{m=1}^{4} (\alpha_\xi)_m \lambda_\xi^{-(m)} (T_\xi)_m \qquad (20a)$$

$$(\overline{f}_\eta)_{i,j+1/2} = (f_\eta)_{i,j} + \sum_{m=1}^{4} (\alpha_\eta)_m \lambda_\eta^{-(m)} (T_\eta)_m. \qquad (20b)$$

The flux vector F_k is evaluated at (i, j) to obtain f_k and $\lambda^{-(m)}$ is the mth non-positive eigenvalue given by

$$\lambda_k^{-(m)} = \frac{1}{2}(\lambda_k^{(m)} - |\lambda_k^{(m)}|). \qquad (21)$$

The subscript m on T_k indicates the corresponding right eigenvector contained in the mth column of the matrix T_k defined in Eq. 15. The α_k term in Eq. 20 is defined for the first order scheme as

$$(\alpha_\xi)_m = (T_\xi^{-1})_{m,l}(q_{i+1,j}^{n} - q_{i,j}^{n})_l, \qquad (22a)$$

$$(\alpha_\eta)_m = (T_\eta^{-1})_{m,l}(q_{i,j+1}^{n} - q_{i,j}^{n})_l. \qquad (22b)$$

In this case the subscript m indicates the left eigenvector corresponding to the eigenvalue (and right eigenvector) previously discussed. The left eigenvector is contained in the mth row of the T_k^{-1} matrix defined in Eq. 16. The subscript l indicates multiplication of the row vector of left eigenvectors by the column vector of differenced dependent variables.

These eigenvalues and eigenvectors given by Eqs. 14, 15, and 16 in the previous section are evaluated using the "Roe" averaged variables defined by

$$\rho = \sqrt{\rho_i \rho_{i+1}} \tag{23a}$$

$$u = \frac{\sqrt{\rho_i} u_i + \sqrt{\rho_{i+1}} u_{i+1}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{23b}$$

$$v = \frac{\sqrt{\rho_i} v_i + \sqrt{\rho_{i+1}} v_{i+1}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{23c}$$

$$c^2 = (\gamma - 1)[H - \frac{1}{2}(u^2 + v^2)] \tag{23d}$$

$$H = \frac{1}{\rho}(e + p) \tag{23e}$$

$$e = \frac{p}{\gamma - 1} + \rho \frac{(u^2 + v^2)}{2}. \tag{23f}$$

These equations are for the ξ coordinate direction. The terms are evaluated at the i index indicated and j. The equations for the η direction are obtained by substituting j for i.

The "Roe" averaging is selected to ensure uniform validity across discontinuities [25]. By using Roe's numerical fluxes shocks and contact discontinuities can be captured in as few as one to two grid cells [26, 32]. This method also allows the use of fewer grid points than the flux vector splitting schemes to capture viscous shear layers. The two (and three) dimensional applications of this method assumes that the characteristic waves propagate in a direction normal to the grid interfaces [32].

Choice of Time Step

Applying the von Neumann stability analysis [1] to Eq. 19 yields the following condition for stability (see the Appendix), analogous to the one dimensional Courant Friedrichs-Lewy (CFL) condition with the Courant number, (CFL), given by

$$CFL = \Delta t(|\theta_\xi| + c\sqrt{\xi_x^2 + \xi_y^2} + |\theta_\eta| + c\sqrt{\eta_x^2 + \eta_y^2}) \leq 1. \tag{24}$$

Two variations on this equation are implemented. The first is known as global time stepping and is necessary for time accurate calculations. In this approach the

time step, Δt, is chosen to satisfy the stability condition and all cells are advanced the same amount of time each iteration. In the local time stepping formulation the Courant number (labeled CFL) is selected and held constant. This method is only valid for steady state problems because each cell may be advanced by a different time step.

Initial and Boundary Conditions

The entire domain is initially assumed to be at free stream conditions. The Mach number is specified and the remainder of the free stream values are determined as follows for the nondimensionalized variables.

$$\rho_\infty = 1 \tag{25a}$$

$$(\rho u)_\infty = M_\infty \tag{25b}$$

$$(\rho v)_\infty = 0 \tag{25c}$$

$$e_\infty = \frac{1}{\gamma(\gamma - 1)} + \frac{1}{2}M_\infty^2 \tag{25d}$$

$$p_\infty = (\gamma - 1)(e_\infty - \frac{1}{2}M_\infty^2) \tag{25e}$$

The boundary conditions are applied using an explicit image point or phantom point formulation [31]. This formulation works by selecting a set of fictitious points just outside of the domain with each cell center on the boundary having a corresponding phantom point. Cell centers at the corners will have three phantom points, one for each coordinate direction and a corner point to fill out the boundary. This corner phantom point is not required or used by the FDS algorithm. Eqs. 26, 27, and 28 are used to set the values of these phantom points for the specified boundary conditions. In these equations the subscript a denotes the phantom point and subscript b the corresponding point in the domain.

Supersonic inflow

$$\rho_a = 1 \tag{26a}$$

$$(\rho u)_a = M_\infty \tag{26b}$$

$$(\rho v)_a = 0 \tag{26c}$$

$$e_a = \frac{1}{\gamma(\gamma-1)} + \frac{1}{2}M_\infty^2 \tag{26d}$$

$$p_a = (\gamma-1)(e_a - \frac{1}{2}M_\infty^2). \tag{26e}$$

Supersonic outflow

$$\rho_a = \rho_b \tag{27a}$$

$$(\rho u)_a = (\rho u)_b \tag{27b}$$

$$(\rho v)_a = (\rho v)_b \tag{27c}$$

$$e_a = e_b \tag{27d}$$

$$p_a = p_b. \tag{27e}$$

Solid surface (zero pressure gradient)

$$\rho_a = \rho_b \tag{28a}$$

$$(\rho u)_a = (\rho u)_b - 2(\rho\tilde{\theta}_k\tilde{k}_x)_b \tag{28b}$$

$$(\rho v)_a = (\rho v)_b - 2(\rho\tilde{\theta}_k\tilde{k}_y)_b \tag{28c}$$

$$e_a = e_b \tag{28d}$$

$$p_a = p_b. \tag{28e}$$

CHAPTER 3
MULTIGRID APPROACH

The idea behind multigrid is to accelerate the convergence of a relaxation scheme by adding corrections to the fine grid solution based on solutions generated on coarser grids [9, 10, 21]. In order to describe why multigrid works two important pieces of information need to be recognized. First, most relaxation schemes eliminate the high frequency error in as little as 2 or 3 iterations, while the same scheme can take thousands of iterations and may never satisfactorily eliminate the lower frequency errors. Secondly the low frequency errors on a fine grid appear as high frequency errors on coarser grids. It would then appear that nearly all of the error could be treated as high frequency error when the correct sequencing of grids is used. The following is a description of the multigrid method used in this work.

Full Approximation Scheme (FAS)

The full approximation scheme [2, 8, 9, 16] is used a great deal in computational fluid dynamics (CFD) due to its ability to handle nonlinear problems such as the Euler or Navier-Stokes equations. FAS restricts both the residual and dependent variables to the coarser grids. This differs from the multigrid designed for linear systems of equations which only restricts the residual to the coarser grid.

Letting a symbol superscripted by h denote that the symbol belongs to the fine grid level then the grid sequence is denoted by superscripts $(h, 2h, 4h, \ldots, nh)$. Where $2h$ indicates a grid over the same domain as h with half the number of grid points in each direction and recursively for each subsequent grid. In order to apply FAS, Eq. 19 is written in the following form

$$L^h(Q^h) = 0 \tag{29}$$

where Q^h is the exact steady state solution to the finite difference equation and L^h is the finite difference operator on the fine grid, defined for the explicit flux difference splitting algorithm as

$$L^h(q^h) = (\overline{f}_\xi)^n_{i+1/2,j} - (\overline{f}_\xi)^n_{i-1/2,j} + (\overline{f}_\eta)^n_{i,j+1/2} - (\overline{f}_\eta)^n_{i,j-1/2}. \qquad (30)$$

Applying an iterative scheme

$$L^h(q^h) = R^h \qquad (31)$$

where R^h is the residual obtained by subtracting Eq. 31 from Eq. 29.

$$L^h(Q^h) - L^h(q^h) = -R^h \qquad (32)$$

This equation is then approximated on the next coarser grid as

$$L^{2h}(Q^{2h}) = II^{2h}_h(-R^h) + L^{2h}(I^{2h}_h q^h) \qquad (33)$$

with the restriction operators II^{2h}_h and I^{2h}_h to be discussed in the next section. Eq. 31 is then used to substitute for R^h.

$$L^{2h}(Q^{2h}) = L^{2h}(I^{2h}_h q^h) - II^{2h}_h(L^h q^h) \qquad (34)$$

The right hand side of this equation is now the defect correction (or the relative truncation error) between the solution on finest (level h) and the coarse (level $2h$) grids. This defect correction acts as a forcing function on the coarser levels in order to maintain the accuracy of the finest level. Note that if the $L^h(q^h) = 0$, then $q^h = Q^h$ and the exact solution for Q^{2h} in Eq. 34 is $I^{2h}_h q^h = I^{2h}_h Q^h$. Designating the defect correction as τ^{2h} the previous equation becomes

$$L^{2h}(Q^{2h}) = \tau^{2h}. \qquad (35)$$

Equations for any number of levels can be generated by this same procedure. For example, treating Eq. 35 with the same procedure used on Eq. 29 the equation on the $4h$ level becomes

$$L^{4h}(Q^{4h}) = \tau^{4h}. \qquad (36)$$

The defect correction on this level is actually a measure of the relative truncation error between level h and $4h$.

$$\tau^{4h} = L^{4h}(I_{2h}^{4h}q^{2h}) - II_{2h}^{4h}(L^{2h}(q^{2h})) - II_{2h}^{4h}\tau^{2h} \qquad (37)$$

Observing Eqs. 29, 35, and 36 it can be seen that the same system of equations are solved on each level ($\tau^h = 0$) making FAS an easy technique to program (see references [2, 8] for more details). These equations also indicate that the residual (obtained from the iterative procedure) minus the defect correction is driven to zero on the coarse grids in the FAS procedure. The only real drawback to this procedure is the need to store the solution at all levels. Fortunately each grid level requires only a quarter (in two dimensions) of the previous level's storage requirement.

The remainder of the FAS procedure is to calculate the coarse grid correction, V, and pass it to the next finer level. The following equations are used to calculate the correction on the $4h$ level and to prolongate (interpolate) it to the $2h$ level.

$$V^{4h} = Q^{4h} - I_{2h}^{4h}q^{2h} \qquad (38)$$

$$q^{2h} = q^{2h} + I_{4h}^{2h}V^{4h} \qquad (39)$$

Restriction and Prolongation Operators

These operators simply define the interpolation needed to move quantities between grids. The type of interpolation to be used is fairly arbitrary and there is no requirement that the restriction and prolongation operators be of the same type or order. It is actually better to chose a prolongation operator that will not pass high frequency errors due to the increased iteration cost on the finer grid levels [9].

The restriction operator for a point quantity given by,

$$(I_h^{2h}q^h)_{i,j} = \frac{1}{4}(q_{2i-1,2j-1}^h + q_{2i,2j-1}^h + q_{2i-1,2j}^h + q_{2i,2j}^h), \qquad (40)$$

is bi-linear interpolation applied in the computational domain. Since the computational domain is uniform all of the coefficients are equal and constant. Defining the

restriction operator in this way will allow conservative transfer of mass, momentum, and energy only when the grid is uniform. The subscripts (i, j) denote a general point in the coarser grid.

The restriction operator for the residual,

$$(II_h^{2h}R^h)_{i,j} = (R_{2i-1,2j-1}^h + R_{2i,2j-1}^h + R_{2i-1,2j}^h + R_{2i,2j}^h), \qquad (41)$$

simply sums the residuals on the fine grid that lie inside of the coarse grid cell. This formulation gives an exact conservative transfer of residuals [2] (reference [2] also uses a conservative restriction operator for the conserved variables).

Bi-linear interpolation applied in the computational domain is also used for the prolongation operator. In this case point quantities are transferred back to the finer grid levels using

$$(I_{2h}^h V^{2h})_{2i-1,2j-1} = \frac{1}{16}(9V_{i,j}^{2h} + 3(V_{i-1,j}^{2h} + V_{i,j-1}^{2h}) + V_{i-1,j-1}^{2h}) \qquad (42a)$$

$$(I_{2h}^h V^{2h})_{2i,2j-1} = \frac{1}{16}(9V_{i,j}^{2h} + 3(V_{i+1,j}^{2h} + V_{i,j-1}^{2h}) + V_{i-1,j-1}^{2h}) \qquad (42b)$$

$$(I_{2h}^h V^{2h})_{2i-1,2j} = \frac{1}{16}(9V_{i,j}^{2h} + 3(V_{i-1,j}^{2h} + V_{i,j+1}^{2h}) + V_{i-1,j-1}^{2h}) \qquad (42c)$$

$$(I_{2h}^h V^{2h})_{2i,2j} = \frac{1}{16}(9V_{i,j}^{2h} + 3(V_{i+1,j}^{2h} + V_{i,j+1}^{2h}) + V_{i+1,j-1}^{2h}). \qquad (42d)$$

Boundary Conditions

The boundary conditions applied to the fine grid are given in Chapter 2, with the exception of the corner phantom points which are set by taking the average of the two nearest phantom points. These points are not used in the flux difference splitting but they do come into play during the transfer of information between grid levels.

The fine grid boundary conditions cannot be applied directly to the coarser grid levels without causing large truncation errors [16]. This can be corrected by a boundary condition defect correction generated in a manner similar to the defect correction

for the interior region. The problems with this are that the boundary conditions must have a flux formulation, the calculation of the correction itself can be very complex, and a correction of this type can make the program problem dependent [16]. An alternative to this is to pass the boundary conditions from the fine grid then perform relaxation on the boundaries of the coarser levels [9]. Another option is to update the boundary conditions only on the fine grid and freeze them on the coarser levels [16]. For finite volume schemes using a phantom point boundary condition formulation simply applying the fine grid conditions to the coarse grid is acceptable when only the change on the coarse grid is transferred to the fine grid [16]. This last method of applying boundary conditions was selected in order to avoid the need for special interpolation (and extrapolation) that would be necessary to update the phantom points.
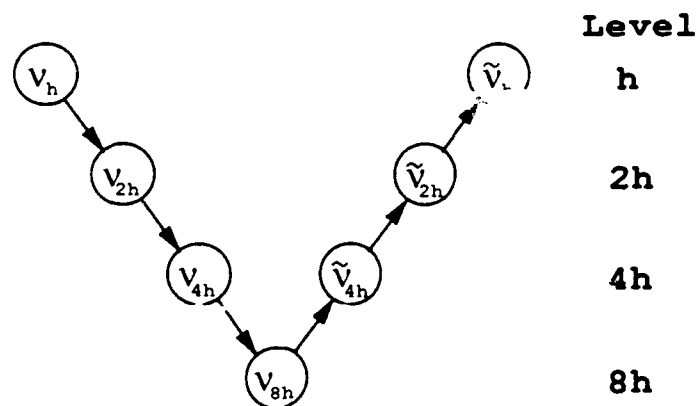
<u>V-Cycle Procedure</u>

Figure 1: FAS V-Cycle Grid Schedule

Several methods of scheduling levels can be used for multigrid [2, 10]. The V-Cycle procedure is used in this study. This procedure is recursive and is easily applied to any number of levels. The actual grid scheduling for a four level scheme is shown in Figure 1 and the corresponding solution procedure follows.

1. Iterate ν_h times on $\Delta q^h = -\Delta t(L^h(q^h))$

   for (n = 2h; n $\leq$ 8h; n = 2*n) {

2. Calculate the residual $R^{n/2} = L^{n/2}(q^{n/2})$

3. Restrict solution to coarser level $q^n = I_{n/2}^n q^{n/2}$

4. Restrict residual minus defect correction to coarser level $R^n = II_{n/2}^n(R^{n/2} - \tau^{n/2})$

   (recall $\tau^h = 0$)

5. Calculate the defect correction $\tau^n = L^n(I_{n/2}^n q^{n/2}) - R^n$

6. Iterate ν_n times on $\Delta q^n = -\Delta t(L^n(q^n) - \tau^n)$

   }

   for (n = 8h; n $\geq$ 2h; n = n/2) {

7. Calculate the coarse grid correction(CGC) $V^n = Q^n - I_{n/2}^n q^{n/2}$

8. Prolongate CGC and add to solution on finer level $q^{n/2} = q^{n/2} + \hat{I}_n^{n/2} V^n$

9. Iterate $\bar{\nu}_n$ times on $\Delta q^{n/2} = -\Delta t(L^{n/2}(q^{n/2}) - \tau^{n/2})$

   }

10. If solution is not converged go to 1

CHAPTER 4
EMBEDDED GRID APPROACH

Embedded grids are also referred to as Chimera or overset grids. This domain decomposition method uses independently generated grids to accurately model different regions of the domain. Since these grids are generated independently, configuration modifications will not require regridding the entire problem. This also makes it easy to add components to the configuration, to move two bodies relative to one another, and to add higher resolution grids in areas of interest.

Blanking or I-blanking as it is sometimes called is used to avoid updating the solution inside of holes. These holes come about in several ways. The most obvious way is when part of an embedded grid lies outside of the domain of interest (as occurs when a cell is inside of a solid boundary). This case is not encountered in this study but is mentioned for completeness. Holes are also caused when two or more grids cover the same region in the domain. This is done to prevent the large number of interpolations that would be necessary if all of the points in the overlapping region of one of the grids were to be set by the other overlapping grid. Not only could a large number of interpolations be expensive, they could also degrade the global accuracy when cells of the overlapping grids are different sizes [5]. In this case the grid that provides the best physical solution should be used for the solution procedure in the overlap region. This can be difficult to determine in most cases and is usually up to the engineer to decide which grid will yield the best solution.

Once a point is determined to be in a hole or on a hole boundary it becomes blanked. The residuals calculated for these blanked points are set to zero so that no correction is made to the solution. Since no corrections are made to the solution for blanked points boundary conditions need to be applied on the hole boundaries.

19

This is accomplished by interpolating the dependent variables from the grid that is causing the blanked points. Grid boundaries that lie within another grid must also obtain boundary information in this way. This requires interpolation stencils to be found in the grid causing the hole for these boundary points. It can be difficult to find valid interpolation stencils. Valid stencils do not use points that are themselves set by interpolation. In the case where no valid stencil can be found the point in question is referred to as an "orphan point" [12]. This requirement usually leads to an overlap of two to three cells depending on the interpolation stencil. Best results tend to be obtained when grid cells are approximately the same size in overlapping grids [5, 12].

In order to carry out the intergrid communication a method of finding the interpolation stencils must be employed. The "stencil jumping" or "stencil walking" [5, 12] technique is a very efficient method of finding the nearest neighboring point. Since the FDS algorithm is a finite volume method (ie. the dependent variables are stored at the cell centers) the cell centered meshes must be constructed in order to search for interpolation stencils.

This stencil jumping technique is actually an inverse transfinite interpolation problem. By assuming a linear variation in $\hat{\xi}$ and $\hat{\eta}$ between 0 and 1 in each interpolation stencil with cell centers as corners Eq. 43 can be obtained.

$$\begin{bmatrix} \Delta X^c \\ \Delta Y^c \end{bmatrix} = \begin{bmatrix} \frac{\partial X^c}{\partial \hat{\xi}} & \frac{\partial X^c}{\partial \hat{\eta}} \\ \frac{\partial Y^c}{\partial \hat{\xi}} & \frac{\partial Y^c}{\partial \hat{\eta}} \end{bmatrix} \begin{bmatrix} \Delta \hat{\xi} \\ \Delta \hat{\eta} \end{bmatrix} \tag{43}$$

Where

$$\Delta X^c = X^c_{i,j} - X^c_p \tag{44a}$$

$$\Delta Y^c = Y^c_{i,j} - Y^c_p \tag{44b}$$

$$\Delta \hat{\xi} = \hat{\xi}_{i,j} - 1/2 \tag{44c}$$

$$\Delta \hat{\eta} = \hat{\eta}_{i,j} - 1/2 \tag{44d}$$

and $X_{i,j}^c, Y_{i,j}^c$ is the lower left corner point of the cell being searched. The point X_p^c, Y_p^c is the cell center from another grid that requires an interpolation stencil. The derivatives in the matrix in Eq. 43 are given by

$$\frac{\partial X^c}{\partial \hat{\xi}} = \frac{1}{2}\left(X_{i,j}^{rc} - X_{i+1,j}^{rc} + X_{i+1,j+1}^{rc} - X_{i,j+1}^{rc}\right) + X_{i+1,j}^{rc} - X_{i,j}^{rc} \tag{45a}$$

$$\frac{\partial X^c}{\partial \hat{\eta}} = \frac{1}{2}\left(X_{i,j}^c - X_{i+1,j}^c + X_{i+1,j+1}^c - X_{i,j+1}^c\right) + X_{i,j+1}^c - X_{i,j}^c \tag{45b}$$

$$\frac{\partial Y^c}{\partial \hat{\xi}} = \frac{1}{2}\left(Y_{i,j}^c - Y_{i+1,j}^c + Y_{i+1,j+1}^c - Y_{i,j+1}^c\right) + Y_{i+1,j}^c - Y_{i,j}^c \tag{45c}$$

$$\frac{\partial Y^c}{\partial \hat{\eta}} = \frac{1}{2}\left(Y_{i,j}^c - Y_{i+1,j}^c + Y_{i+1,j+1}^c - Y_{i,j+1}^c\right) + Y_{i,j+1}^c - Y_{i,j}^c. \tag{45d}$$

Eq. 43 is in the form of a Newton's method and will converge at nearly a quadratic rate as ΔX^c and ΔY^c go to zero. This method works well on fairly uniform grids without singularities.

Once the nearest neighbor is found interpolation can be applied to obtain values at the boundary points. This study uses bi-linear interpolation given by

$$q_p = (1 - \hat{\eta})\left[(1 - \hat{\xi})q_{i,j} + \hat{\xi}q_{i+1,j}\right] + \hat{\eta}\left[(1 - \hat{\xi})q_{i,j+1} + \hat{\xi}q_{i+1,j+1}\right]. \tag{46}$$

This interpolation procedure in general does not maintain conservation principles but it is commonly used in this fashion with satisfactory results [5]. The use of conservative techniques (such as those in references [6, 20]) would require a substantial programming effort in order to determine areas of intersection between overlapping cells. This would also complicate interpolation stenciles and can cause stability problems.

Note that in the equations in this section (i, j) denotes a general point on the cell centered grid.

CHAPTER 5
NONALIGNED MULTIGRID (NAM)

The idea here is to treat the overlapping grids of the embedded approach as independent levels of a multigrid FAS procedure. As with the FAS scheme, defect corrections and dependent variables are restricted to the lower levels and corrections are prolongated to the upper levels. Unlike the standard FAS where one level directly communicates with only the next coarser level, NAM levels may directly restrict or prolongate information to any or all levels depending on the particular system of grids. NAM is actually a generalized local refinement method [9, 11, 21] in the sense that the various levels are in general not related and may not have any information restricted to portions of their domain. The lack of relationship between levels covering the same domain is similar to the multigrid methods applied to unstructured grids [20]. Unlike embedded grids where blanked hole points are eliminated from the solution, NAM uses the regions of overlapping grids to generate lower level corrections producing the faster convergence of a multilevel scheme. This is possible because the defect correction stops grid cells of different sizes from degrading the global solution. Blanking is still appropriate in NAM for cases where the cell centers of one grid level are located inside of a solid surface on another level or when a portion of one of the levels is outside of the domain of interest. Problems with blanked points in the lower levels of a multigrid procedure are discussed in references [3] and [17].

The communications between levels can be illustrated by applying the NAM procedure to the system of embedded grids depicted in Figure 2. Treating this system with the NAM procedure the levels are as indicated, assuming that level 1 provides the best resolution and that level 2 provides better resolution than level 3. In this case Level 1 restricts information to level 2 in region 1 and the remainder of level

1 restricts information to level 3. Level 2 restricts information to level 3. The restricted information in region 1 contains information from level 1 in the level 2 defect correction. On the way back up level 3 prolongates the lower level correction to all of level 2 (including region 1) and to level 1 except in region 1. Level 2 prolongates a correction to level 1 in region 1. A grid schedule analogous to the FAS V-Cycle diagram is shown in Figure 4 for the three level treatment of grids just described.

Another interesting aspect of local refinements in general is having multiple grids on the same level as indicated in Figure 3 where level 1 contains two grids as indicated. This grid is treated as part of level 1 because it receives no information from a higher level. When this occurs the numerical algorithm is applied to both grids independently and information is restricted to levels 2 and 3 as required. In general, multiple grids can occur on every level and when coding the NAM procedure no designation needs to be made between levels and grids. This means that grid 2 could be treated as just another level without any prolongation or restriction with non-overlapping levels.

For faster convergence of this problem any of the grids may be coarsened and added as additional levels. Care should be used when coarsening levels other than the lowest level. For example, coarsening level 1 and inserting it as level 2 may result
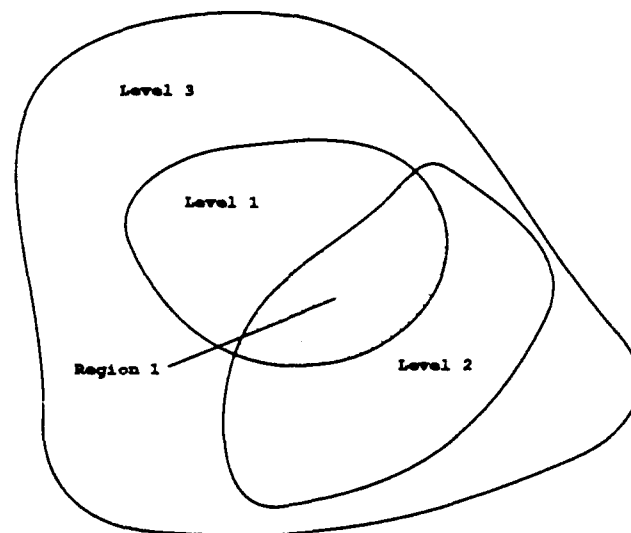

Figure 2: NAM Three Level System

in a grid with worse resolution than the level receiving the restricted information. This would result in forcing lower resolution onto the lower levels. The lowest level (typically the global grid) is usually the safest to coarsen. Coarsening the global grid will in general provide faster convergence by allowing larger time steps for a more rapid development of the flow structure.
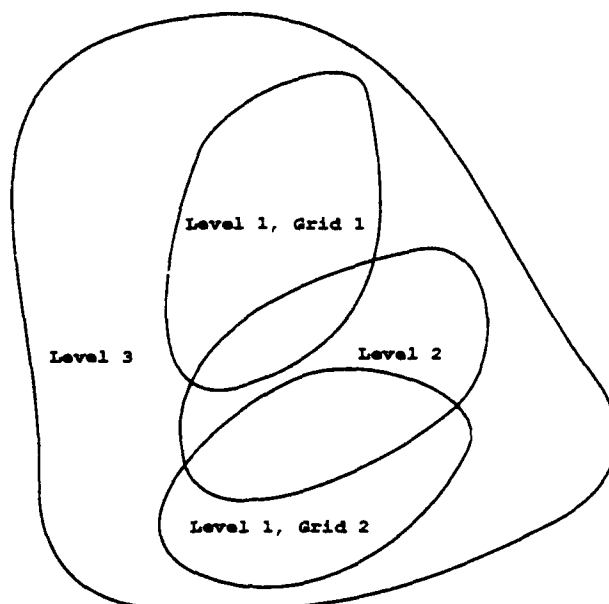

Figure 3: NAM Three Level System with Multiple Grids per Level

Restriction and Prolongation Operators

As in Chapter 3 these operators define the interpolation needed to move quantities between grids. The arbitrary orientations permissible with NAM require a more general set of operators. The interpolation stencils are found using the "stencil jumping" technique discussed in Chapter 4. Bi-linear interpolation is used for both restriction and prolongation operators as indicated in Eqs. 47, 48, and 49. The subscripts (i, j) denote a general point in the higher level and (k, l) denotes a general point in the lower level.

$$(I_n^{n+1} q^n)_{k,l} = (1 - \hat{\eta}) \left[(1 - \hat{\xi}) q_{i,j}^n + \hat{\xi} q_{i+1,j}^n \right] + \hat{\eta} \left[(1 - \hat{\xi}) q_{i,j+1}^n + \hat{\xi} q_{i+1,j+1}^n \right] \qquad (47)$$

$$(II_n^{n+1}R^n)_{k,l} = \vartheta_{k,l}^{n+1}(1-\hat{\eta})\left[(1-\hat{\xi})(R/\vartheta)_{i,j}^n + \hat{\xi}(R/\vartheta)_{i+1,j}^n\right] +$$

$$\vartheta_{k,l}^{n+1}\hat{\eta}\left[(1-\hat{\xi})(R/\vartheta)_{i,j+1}^n + \hat{\xi}(R/\vartheta)_{i+1,j+1}^n\right] \qquad (48)$$

In the previous equation ϑ is the volume. The appearance of the volume is due to treating the residuals as point quantities during the interpolation process.

$$(\hat{I}_{n+1}^n q^n)_{i,j} = (1-\hat{\eta})\left[(1-\hat{\xi})q_{k,l}^{n+1} + \hat{\xi}q_{k+1,l}^{n+1}\right] + \hat{\eta}\left[(1-\hat{\xi})q_{k,l+1}^{n+1} + \hat{\xi}q_{k+1,l+1}^{n+1}\right] \qquad (49)$$

The bi-linear interpolation is not conservative, but is selected here based on its ability to handle communications with embedded grids [5]. In Chapter 6 FAS produces the same results as the single grid calculation using nonconservative prolongation and restriction operators. These results combined with successful applications on embedded grids would tend to indicate that bi-linear interpolation is worth trying. Alternatives include the conservative techniques used in unstructured multigrid [20] and the techniques discussed in references [6, 7].
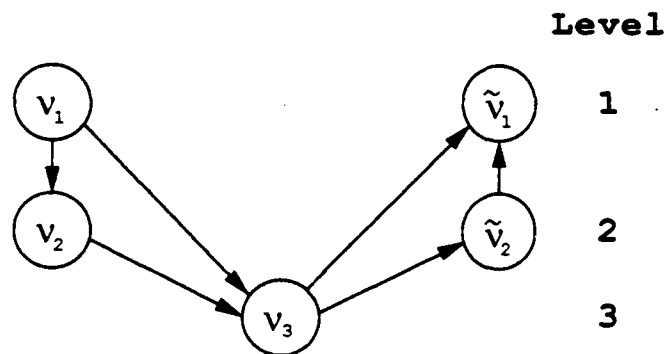
V-Cycle Procedure

Figure 4: NAM Three Level Grid Schedule

The cycling procedure and grid scheduling is similar to that of the FAS method and FAS is actually a special case of NAM. The NAM grid schedule shown in Figure 4 is for a three level configuration as in Figure 2. Several methods of scheduling are possible depending on the system of grids to be used. The solution procedure follows for the grid schedule in Figure 4.

1. Iterate ν_1 times on $\Delta q^1 = -\Delta t(L^1(q^1))$

   for (n = 2; n $\leq$ 3; n = n+1) {

2. Calculate the residual $R^{n-1} = L^{n-1}(q^{n-1})$

   for (m = n+1; m $\leq$ 3; m = m+1) {

3. Restrict solution to lower level $q^m = I^m_{n-1}q^{n-1}$

4. Restrict residual minus defect correction to lower level $R^m = II^m_{n-1}(R^{n-1} - \tau^{n-1})$

   (recall $\tau^1 = 0$)

   }

5. Calculate the defect correction $\tau^n = L^n(I^n_{n-1}q^{n-1}) - R^n$

6. Iterate ν_n times on $\Delta q^n = -\Delta t(L^n(q^n) - \tau^n)$

   }

   for (n = 3; n $\geq$ 2; n = n-1) {

7. Calculate the lower level correction(LLC) $V^n = Q^n - I^n_{n-1}q^{n-1}$

   for (m = n-1; m $\geq$ 1; m = m-1) {

8. Prolongate LLC and add to solution on finer level $q^m = q^m + \hat{I}^m_n V^n$

   }

9. Iterate $\tilde{\nu}_n$ times on $\Delta q^{n-1} = -\Delta t(L^{n-1}(q^{n-1}) - \tau^{n-1})$

   }

10. If solution is not converged go to 1

CHAPTER 6
RESULTS

The two dimensional 5° ramp near a flat plate shown in Figure 5 provides the test case for comparing the techniques discussed in Chapters 2 through 5. By accurately predicting the flow field for this test configuration the feasibility of the nonaligned multigrid technique is shown. Solutions generated by the rest of the methods discussed are to show the relative improvement of the nonaligned multigrid and to provide an understanding of how the various components of the nonaligned multigrid work. The algorithms are coded in "C" and the numerical solutions are generated using double precision numbers on an Iris 4D/320VGX workstation. No attempt was made to parallelize on this two processor machine.

Theoretical Solution

The theoretical solution to this test case is obtained from the oblique shock charts and the normal shock tables for isentropic flow [19]. The weak shock solution is given in Table 1 and the shock angles are depicted in Figure 5.

Table 1: Theoretical Solution

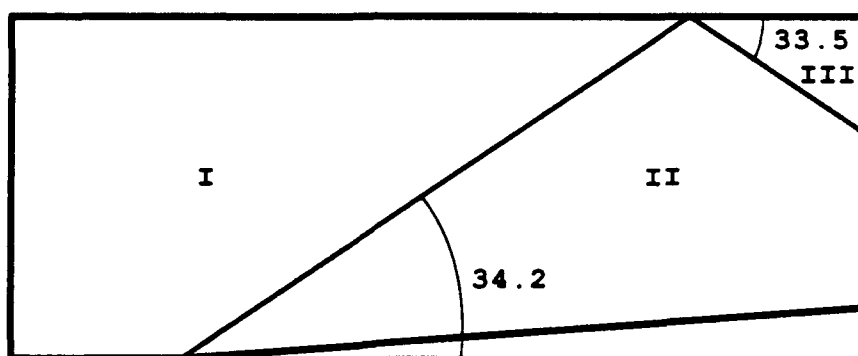|	Region I	Region II	Region III
M	2.00	1.83	1.60
ρ	1.00	1.20	1.49
ρu	2.00	2.28	2.58
ρv	0.00	0.20	0.00
e	3.79	4.49	5.37
p	0.71	0.93	1.25

Figure 5: Problem Geometry

Single Grid Calculations

A single grid solution is obtained by applying the flux difference splitting (FDS) algorithm to the 40 by 32 grid in Figure 6. Density contours from the solution are plotted in Figure 7 after converging to machine zero. This solution does show the general features of the shock structure; however, the first order FDS and the fairly coarse grid combine to smear the shock waves and the zero pressure gradient boundary conditions cause the shocks to curve near the solid surfaces. In order to gain better accuracy grids with dimensions 80 by 64 and 160 by 128 are also used to generate solutions. The 80 by 64 grid plotted in Figure 8 is obtained by quartering the grid cells in the 40 by 32 grid. The 160 by 128 grid is too dense to be plotted clearly and is obtained by quartering the 80 by 64 grid cells. Density contours of the resulting solutions are shown in Figures 9 and 10. The contour plots show that as the number of grid points is increased the dissipation becomes less and the curving near the solid boundaries decreases causing the solution to come closer to the theoretical values in Table 1.

The next area of interest is to determine the efficiency of the solution procedure and the overall cost. In order to do this the following measure of the change in the solution is generated for each iteration.

$$R = \log_{10}\left[\sqrt{\sum_{j=1}^{jmax}\sum_{i=1}^{imax}\left[\sum_{l=1}^{4}\left((\overline{f}_\xi)_{i+1/2,j}^n - (\overline{f}_\xi)_{i-1/2,j}^n + (\overline{f}_\eta)_{i,j+1/2}^n - (\overline{f}_\eta)_{i,j-1/2}^n\right)_l^2\right]}\right] \quad (50)$$

By looking at the history of this residual, plotted in Figure 11, and the CPU times in Table 2 the cost of using an explicit scheme becomes apparent. The explicit FDS scheme converges in a fairly slow manner due to the stability condition indicated in Eq. 24. Since this is a steady state problem local time stepping is applied with a Courant number of 0.96 in order to provide the largest time step in each grid cell. With the addition of grid points the already slow convergence rate becomes even slower due to the stability condition requiring smaller time steps as the grid spacing decreases. This addition of grid points not only decreases the convergence rates, but it also increases the CPU time as indicated in Table 2. Although these differences in CPU time and convergence rates may not seem that important for a problem of this size, these factors become very limiting for more complicated three dimensional problems.
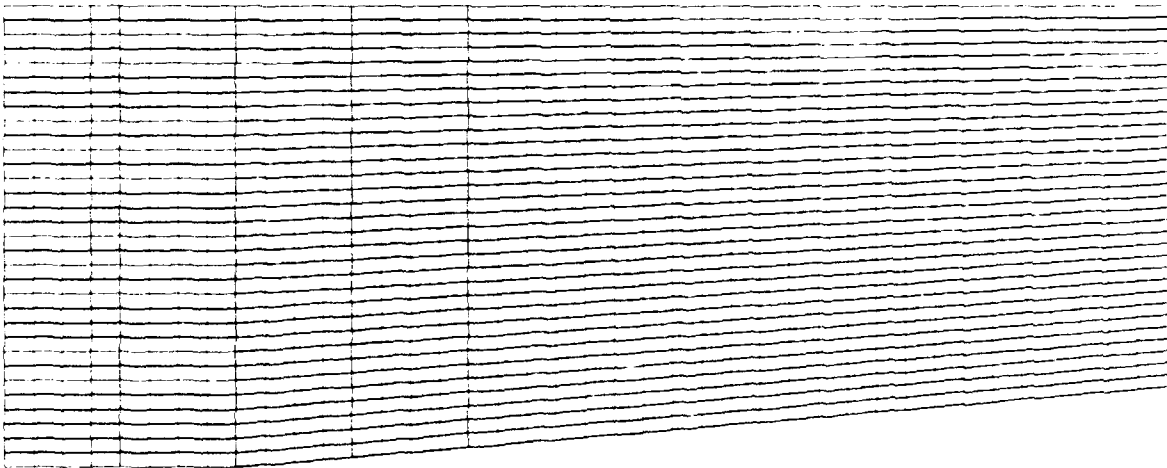

Figure 6: 40 X 32 Single Grid

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
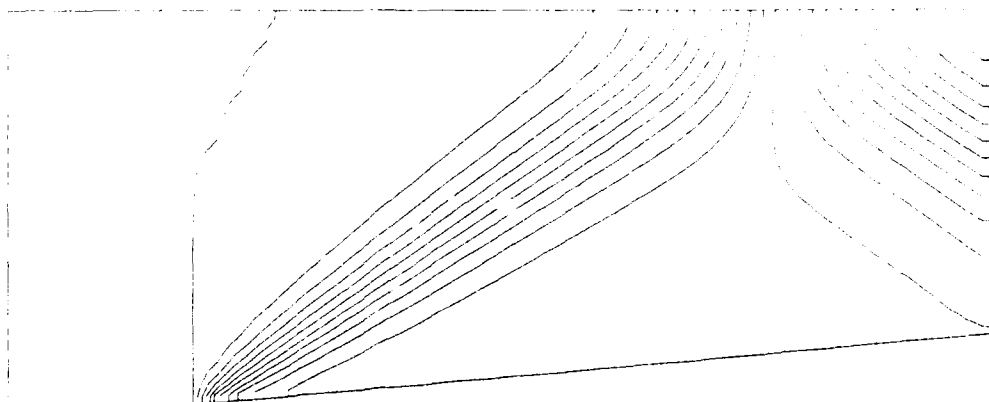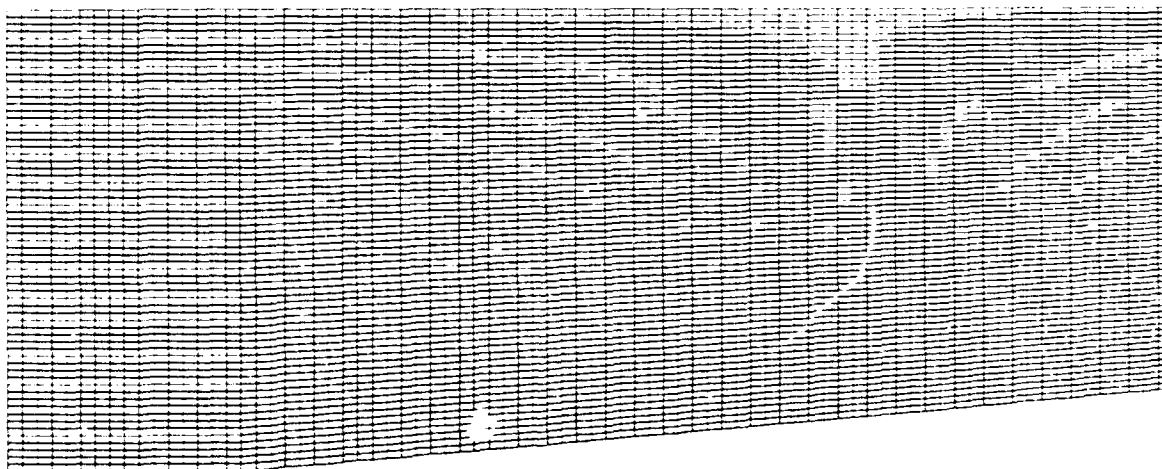1.34000
1.36000
1.38000
1.40000
1.42000
1.44000

Figure 7: 40 X 32 Single Grid Density Contours

Figure 8: 80 X 64 Single Grid

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
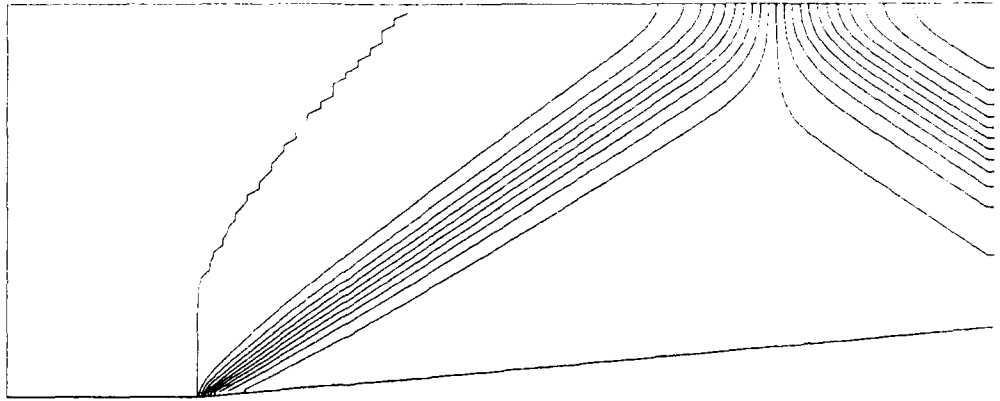1.36000
1.38000
1.40000
1.42000
1.44000
1.46000

Figure 9: 80 X 64 Single Grid Density Contours

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
1.40000
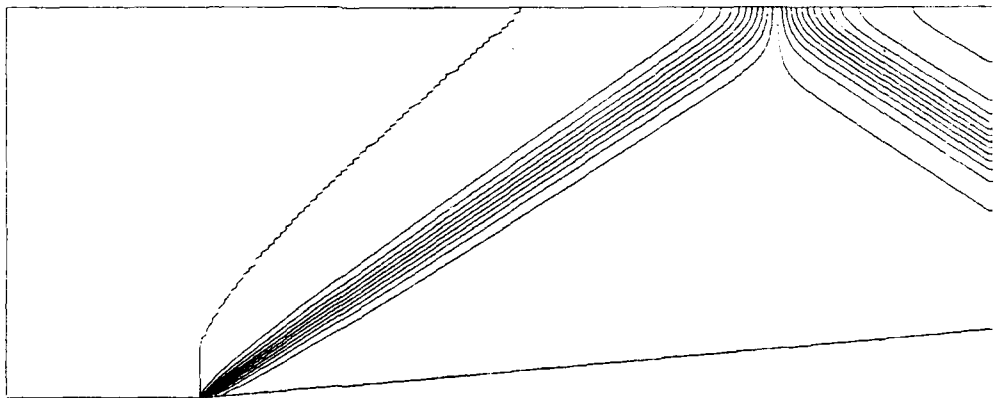1.42000
1.44000
1.46000
1.48000

Figure 10: 160 X 128 Single Grid Density Contours

Table 2: Single Grid Calculation Statistics

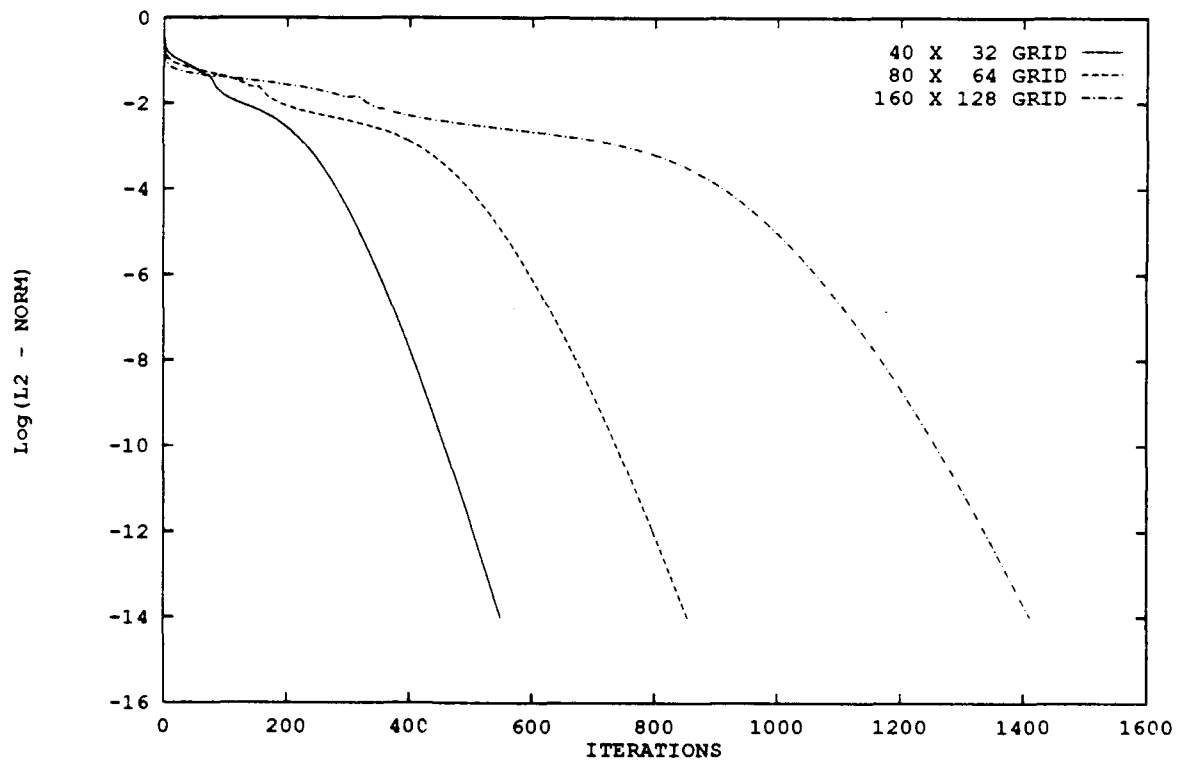| Grid	Iteration	CPU Time
40 X 32	550	131.6 sec
80 X 64	855	900.2 sec
160 X 128	1412	5863.0 sec

Figure 11: Single Grid Residual Histories

Multigrid Calculations

The full approximation scheme (FAS) discussed in Chapter 3 is applied to the grids of the previous section in order to obtain faster convergence.

Grid Coarsening

The first step is to coarsen the grids as indicated in Figure 12 for the 40 by 32 grid. The coarsening used here simply removes every other grid point on the current level in each curvilinear coordinate direction to produce the next coarser grid which will contain 1/4 the number of points of the current level [9, 10]. When doing this it is important to preserve the geometry on the lower levels. This is accomplished here by placing a sufficient number of grid points in front of the ramp such that the leading edge point will not be eliminated by the coarsening procedure for the number of levels that are used in these calculations.

Work Units

Before making any calculations the multigrid work unit [2, 9, 10] needs to be defined. The work unit is the ratio of the number of grid points on the current level to the number of grid points on the fine level. This means the work units are 1 on level h, 1/4 on level 2h, and 1/16 on level 4h for the grid coarsening discussed above. With this definition the work required for the restriction and prolongation operations is neglected. This is the standard multigrid work unit definition which seems reasonable since the cost of interpolation between levels is insignificant compared to the cost of the FDS solution procedure. With this definition the work unit provides a means of comparison between the single grid and multigrid convergence histories.

FAS Results

Applying the FAS procedure to the grids (40 by 32, 80 by 64, and 160 by 128) discussed in the previous section yields the same solution (to double precision accuracy) as those in Figures 7, 9, and 10, respectively. This result would seem to indicate that the nonconservative restriction and prolongation operators defined in Chapter 2 are accurate enough to keep from introducing significant error into the solution.

The residual histories for the two and three level calculations on the 40 by 32 grid are plotted in Figure 13 along with the single grid residual. Several calculations were made varying the iteration count on each level in order to find the maximum gain in convergence. It was fairly easy to find a good two level calculation, but when an additional level is added the possible number of variations becomes large. The iteration counts used for Figure 13 are given in Table 3. Comparing the two and three level residuals indicates that the three level calculation could not achieve an increase in performance over the two level calculation. Also note that the best convergence on the three level calculation was achieved by running one iteration per multigrid cycle on the coarsest level. The combination of these two would seem to indicate that the 4h level is too coarse to handle the shock wave. The wider cells on the 4h level, depicted as the bottom grid in Figure 12, are more likely to cause the restriction and prolongation operators to straddle the shock wave resulting in faulty communication between levels [9]. A conservative set of operators should help to overcome this problem.

Similar results are obtained by looking at the convergence histories for the 80 by 64 grid in Figure 8. Figure 14 plots the residual histories for the single grid calculation and for the two and three level FAS calculations. In this case the convergence is slightly better for the three level calculation. The residual history indicates that the third level helps to accelerate convergence early in the calculations, but the gain is lost as the solution becomes more accurate. No four level calculations are attempted

for this case since the three level results on the 40 by 32 grid did not provide any improvement. The iteration counts used in these calculations are given in Table 3.

Figure 15 shows the residual histories for the single grid and the two, three, and four level calculations on the 160 by 128 grid. In this case the four level V-Cycle provides the best results as expected.

Several observations can be made from the convergence histories. First, the best results occur when smoothing iterations are made ($\bar{\nu} \neq 0$) when prolongating the coarse grid correction. Secondly, the 40 by 32 grid results show that there is a practical limit as to how course the grids can be and still help to reduce the number of work units for the hyperbolic problem when nonconservative interpolation operators are used. Additionally, a method of automatically determining the iteration count on the various levels [9, 17] needs to be implemented based on the number of runs made to find the iteration counts for the calculations presented here. .

A direct consequence of the faster convergence rate provided by multigrid is the reduced CPU time. As noted previously the restriction and prolongation operations were neglected in the work unit calculations but are included in the CPU time. This explains why the savings in CPU time indicated in Tables 3, 4, and 5 may not be as large as the savings indicated by the work units.

Table 3: FAS Calculation Statistics on the 40 X 32 Grid

Levels	ν_h	ν_{2h}	ν_{4h}	$\bar{\nu}_{2h}$	$\bar{\nu}_h$	Work Units	Multigrid Cycles	CPU Time
1	1	—	—	—	—	550.000	—	131.6 sec
2	2	6	—	—	2	396.000	61	94.8 sec
3	3	5	1	4	2	411.000	48	96.3 sec

Table 4: FAS Calculation Statistics on the 80 X 64 Grid

Levels	ν_h	ν_{2h}	ν_{4h}	$\tilde{\nu}_{2h}$	$\tilde{\nu}_h$	Work Units	Multigrid Cycles	CPU Time
1	1	–	–	–	–	855.000	–	900.2 sec
2	2	4	–	–	2	552.000	92	547.9 sec
3	4	4	14	4	4	545.625	45	547.0 sec

Table 5: FAS Calculation Statistics on the 160 X 128 Grid

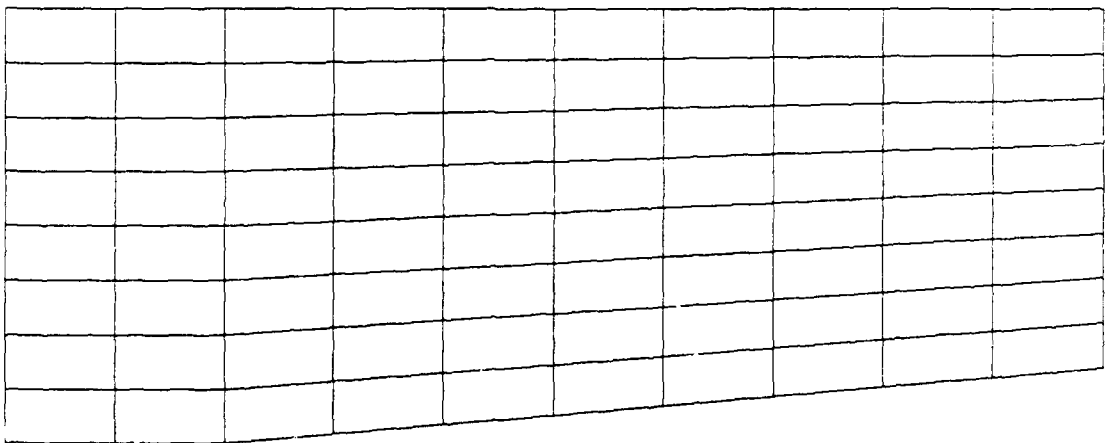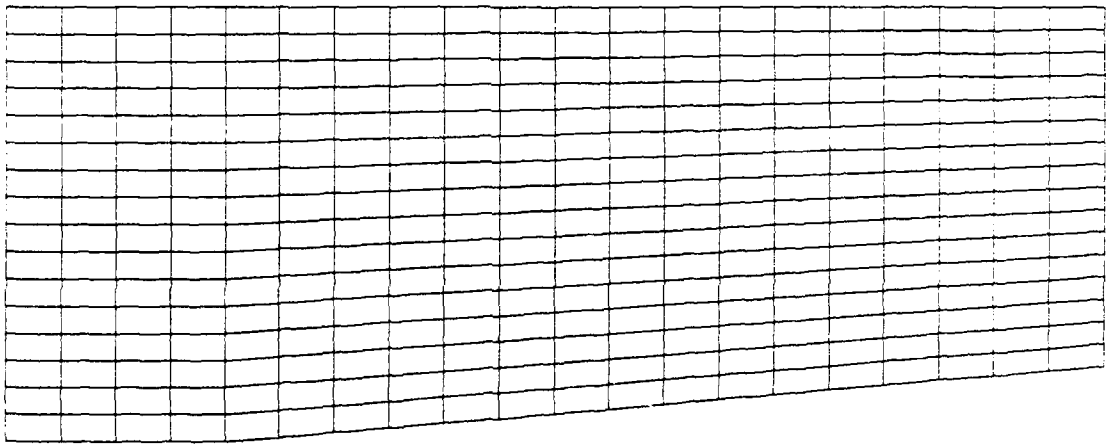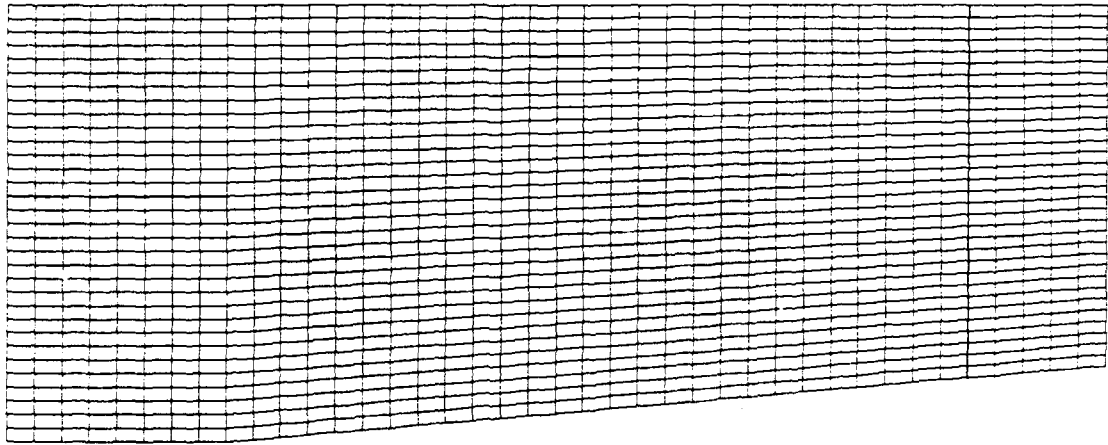| Levels	ν_h	ν_{2h}	ν_{4h}	ν_{8h}	$\tilde{\nu}_{4h}$	$\tilde{\nu}_{2h}$	$\tilde{\nu}_h$	Work Units	Multigrid Cycles	CPU Time
1	1	–	–	–	–	–	–	1412.00	–	5863.0 sec
2	4	8	–	–	–	–	5	804.00	61	3345.3 sec
3	10	8	18	–	–	8	10	791.25	30	3316.9 sec
4	12	9	8	14	8	9	12	775.78	25	3034.7 sec

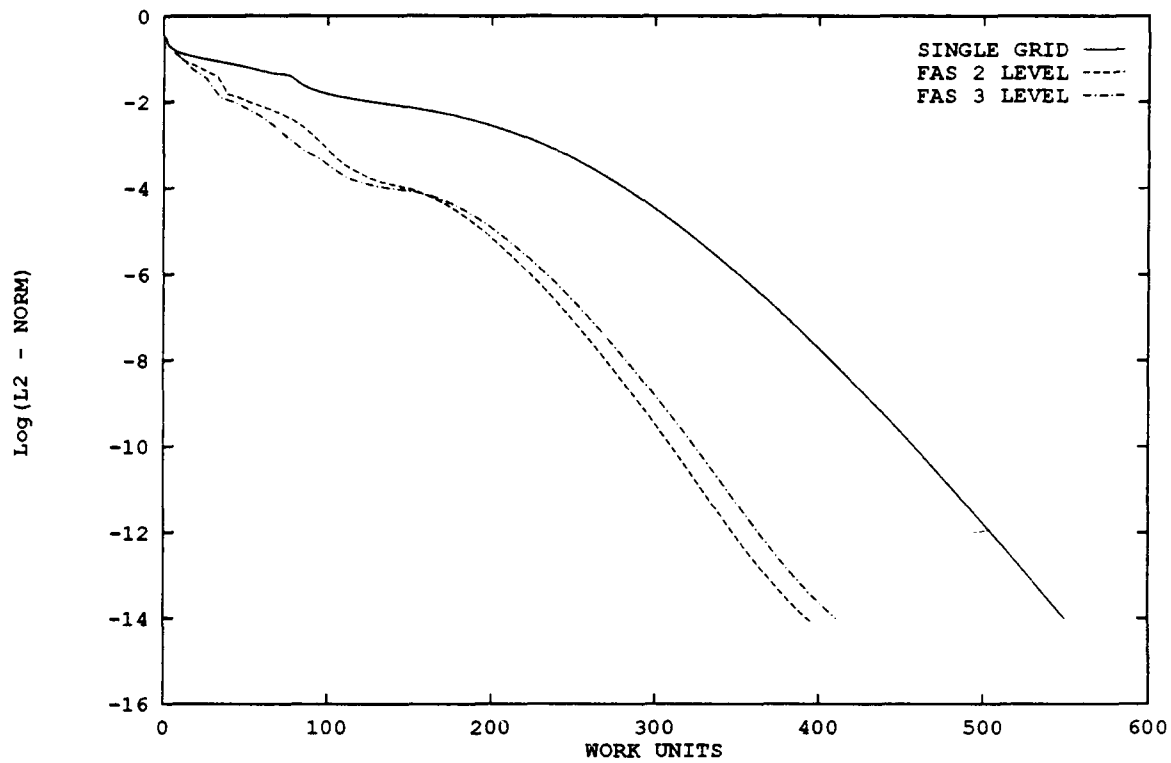Figure 12: 40 X 32 Grid and Two Coarser Levels

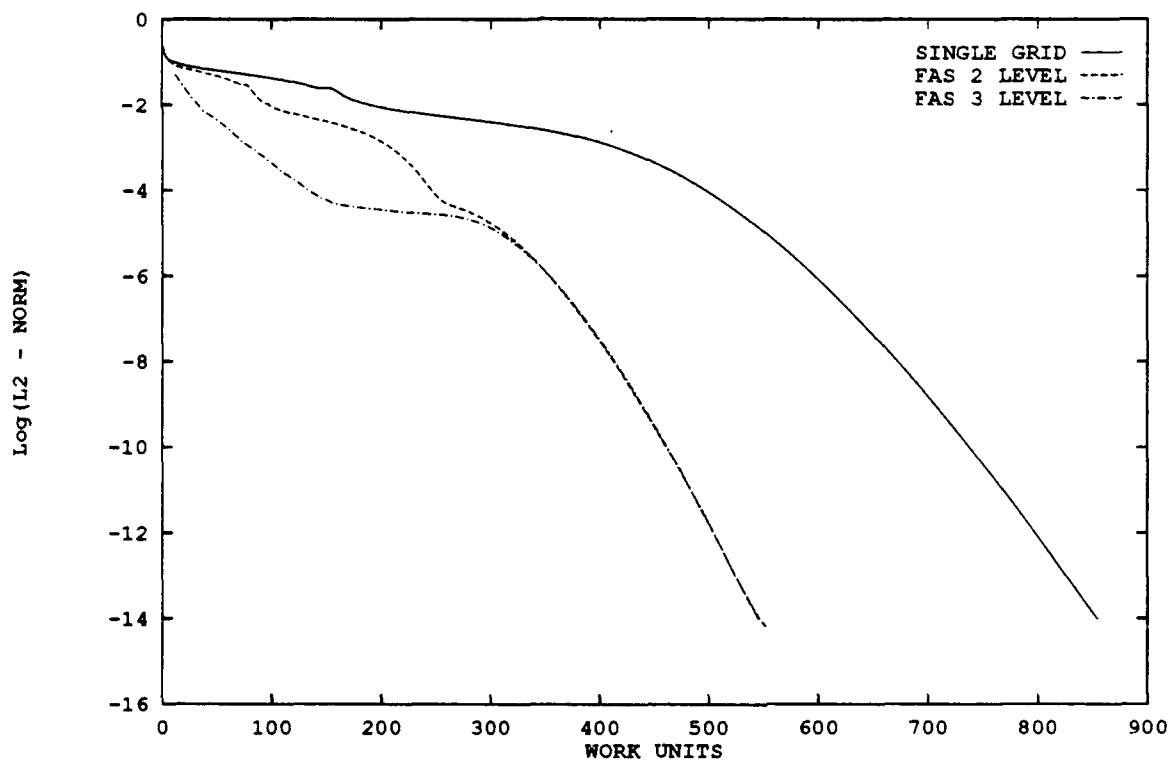Figure 13: FAS Residual Histories on 40 X 32 Grid
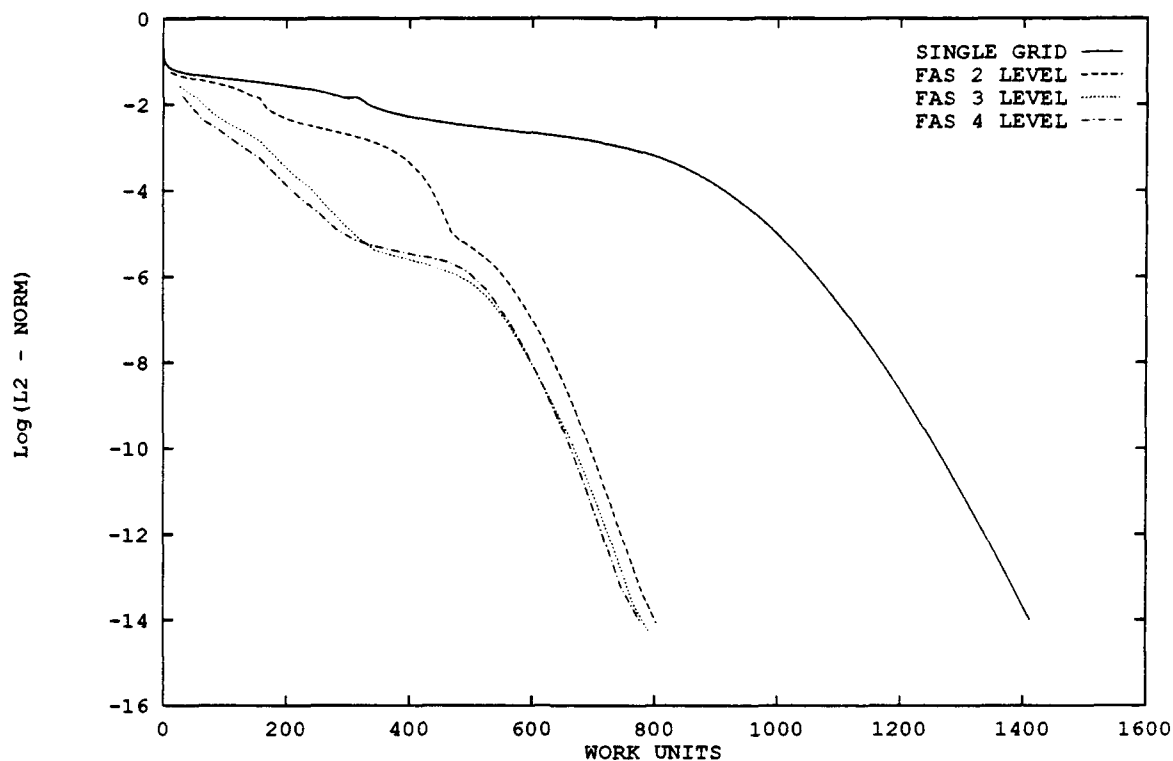

Figure 14: FAS Residual Histories on 80 X 64 Grid

Figure 15: FAS Residual Histories on 160 X 128 Grid

Embedded Grid Calculations

The embedded grid approach is used here to provide an improved solution by inserting local grids over the regions where the shock waves occur. Several variations are attempted in order to predict the theoretical solution as accurately as possible.

5° Ramp Calculations

In order to provide a better understanding the embedded approach is first implemented on a 5° ramp. This eliminates the reflected shock and makes the results easier to interpret. A 16 by 16 local grid is embedded into a 40 by 32 grid as indicated in Figure 16. The cell centered grids are plotted in Figure 17 to show the hole caused by the local grid. Since all boundary points are blanked the overlapping regions appear to be much narrower than they actually are.

Aligning the local grid lines parallel to the shock wave allows the FDS to produce a much more accurate solution than is possible on the global grid. The reason for the improved accuracy is due to the assumption in FDS that the characteristic waves propagate in a direction perpendicular to the grid interfaces [32].

The density contours from this two grid solution are plotted in Figure 18. This solution is much better than could be obtained on the global grid alone and is virtually the same as the theoretical solution (regions I and II in Table 1 and in Figure 5).

Figure 19 shows the residual histories for the global grid alone and for the two grid case. The residual for the embedded grid is obtained by using Eq. 50 on each grid. For blanked points no contribution is made to the residual. This plot shows that the cost of the embedded grid calculation (340.4 seconds) is greater than for the single grid calculation (96.2 seconds); however, the solution on the embedded system is so superior to that of the single grid that this cost comparison is virtually meaningless. For the single grid calculation the dissipation of the first order FDS causes the width of the predicted shock to be slightly wider than the width of the local grid in Figure 16 at the back plane. The increased accuracy of this embedded solution becomes apparent when comparing the width of the smeared shock to that of the shock predicted in Figure 18.
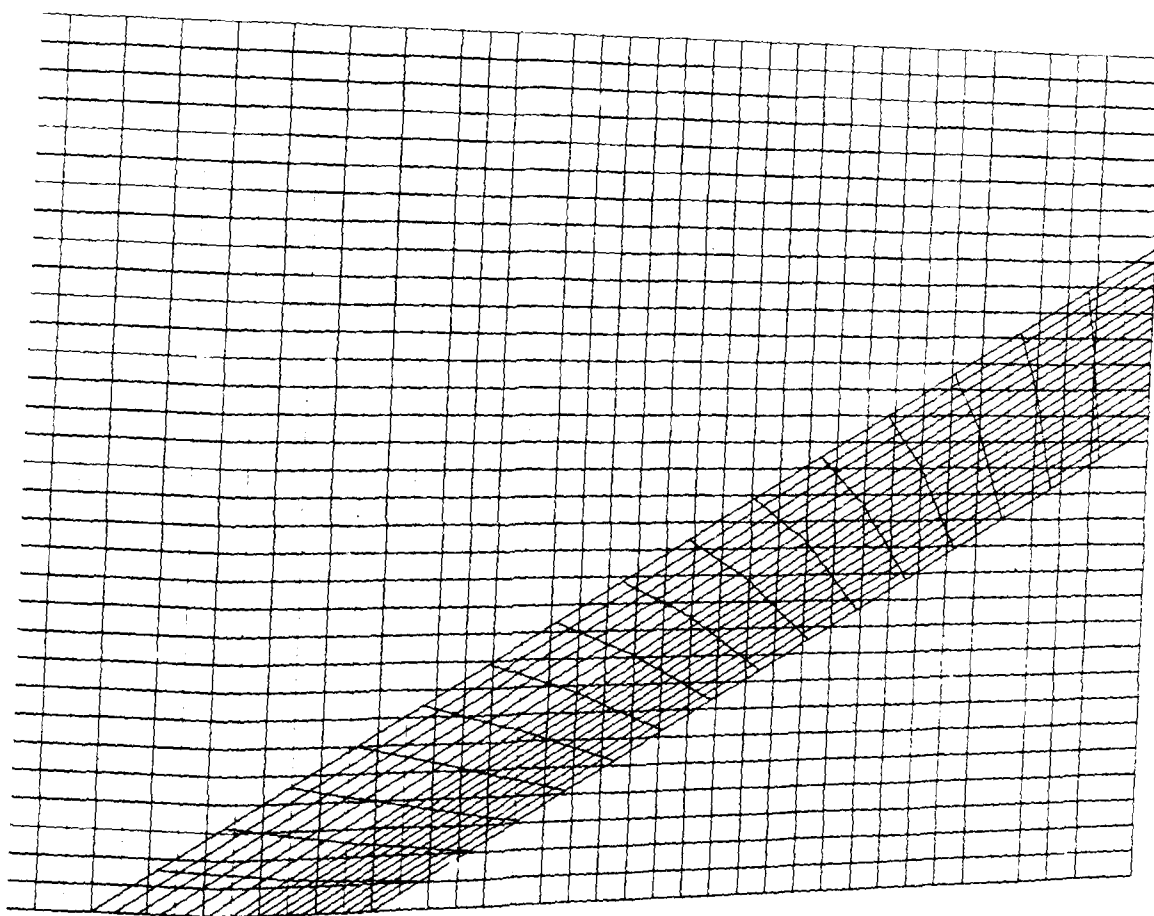
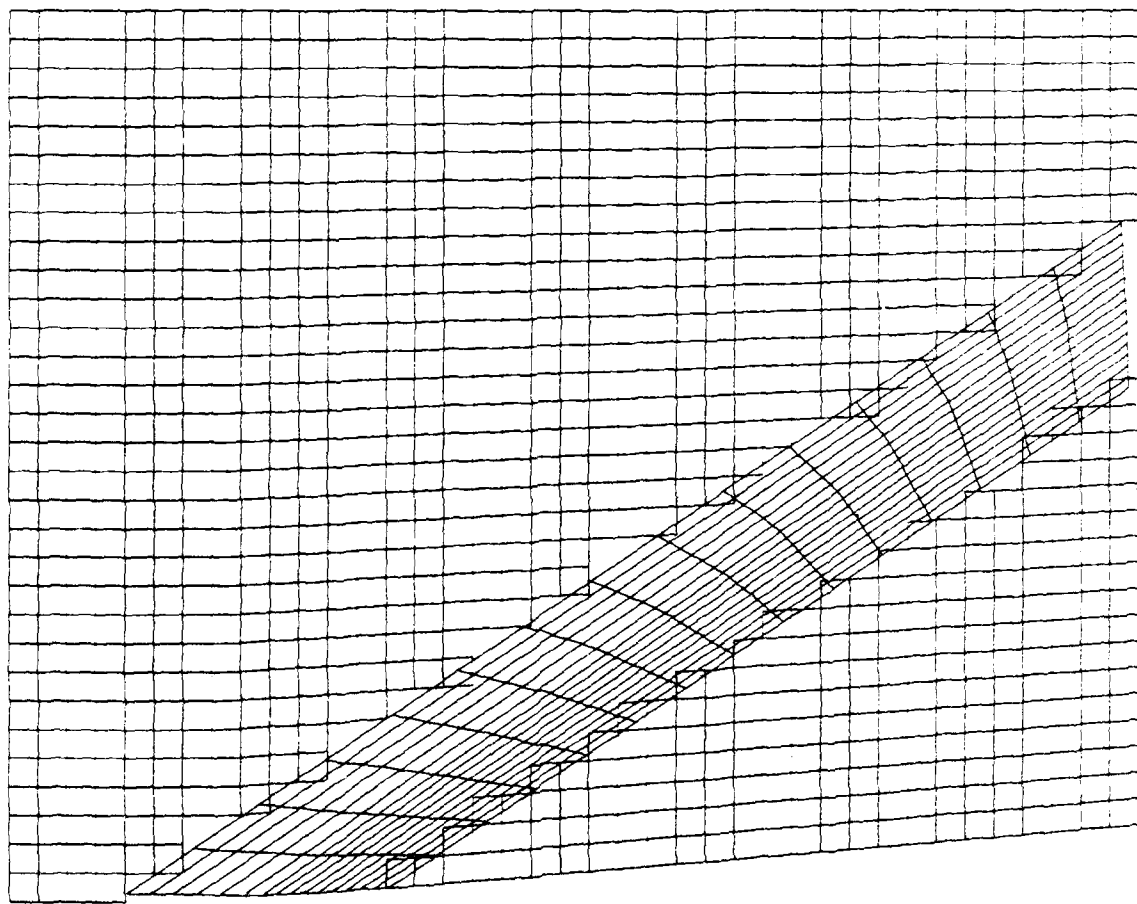Figure 16: Embedded Grids for the 5° Ramp
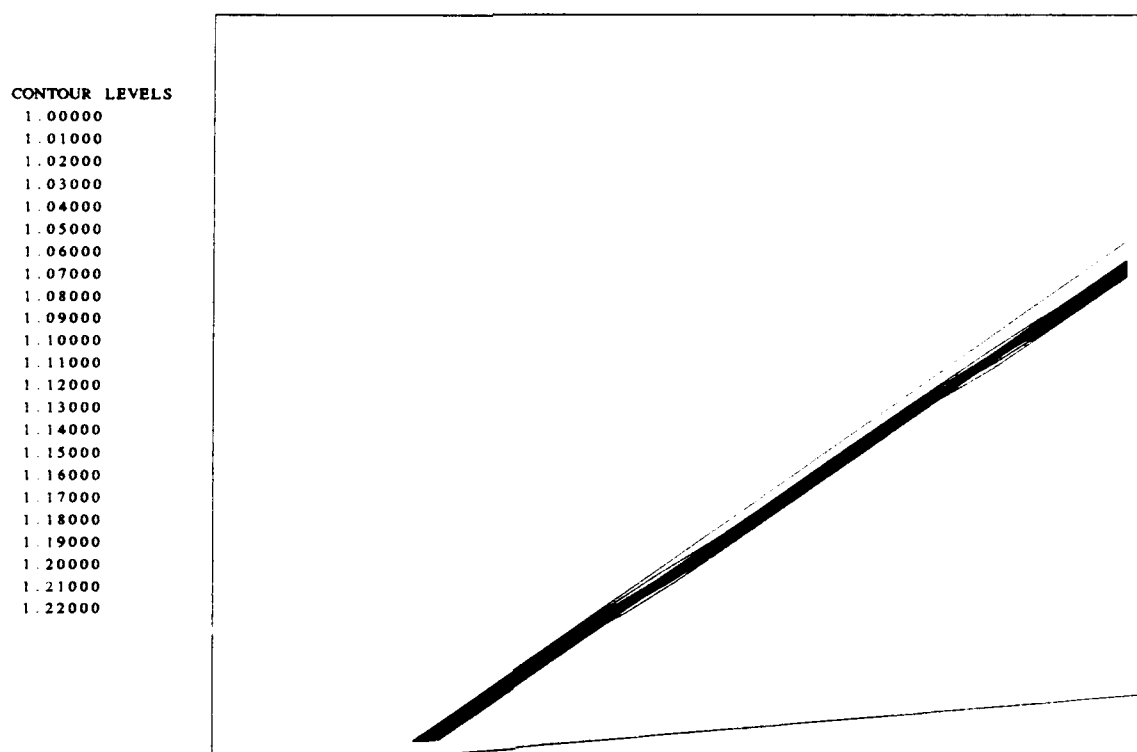
Figure 17: Blanking for the 5° Ramp

CONTOUR LEVELS
1.00000
1.01000
1.02000
1.03000
1.04000
1.05000
1.06000
1.07000
1.08000
1.09000
1.10000
1.11000
1.12000
1.13000
1.14000
1.15000
1.16000
1.17000
1.18000
1.19000
1.20000
1.21000
1.22000

Figure 18: Density Contours for the 5° Ramp Embedded Solution
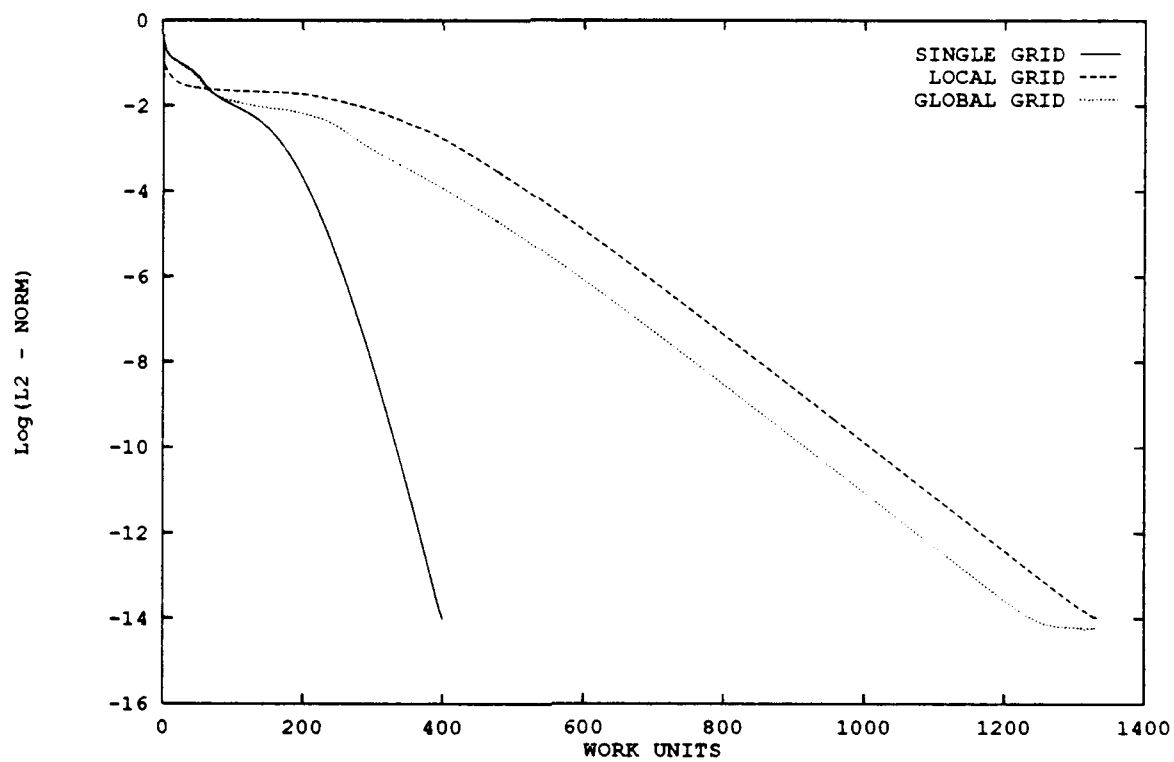
Figure 19: Residual Histories for the 5° Ramp Embedded Calculation

5° Ramp Near a Flat Plate Calculations

Two grid case

Figure 20 shows the embedded grid configuration with the lines of the local grid again parallel to the shock. The hole created in the global grid can be seen in Figure 21.

Density contours of the solution obtained on this grid system are shown in Figure 22. Notice how much more accurately the initial shock is captured than in the single grid solution (compare to Figure 7). The reflected shock is not resolved due to the alignment of the local grid lines.

Comparing the residual histories plotted in Figure 23 with those in Figure 11 for the 40 by 32 single grid calculation shows that the number of iterations is more than double that of the single grid at machine zero. A large part of this is due to the introduction of the artificial boundaries into the domain (the front and back boundaries of the local grid). The explicit communication between grids at these boundaries hinders the development of the solution.

The solution on this system of grids requires 425.3 CPU seconds to converge to machine zero. This is more than three times the cost of the single grid calculation. This increase reflects the cost of making additional calculations on the local grid, the cost of interpolations between grids, and the cost of running more iterations due to the slower convergence.
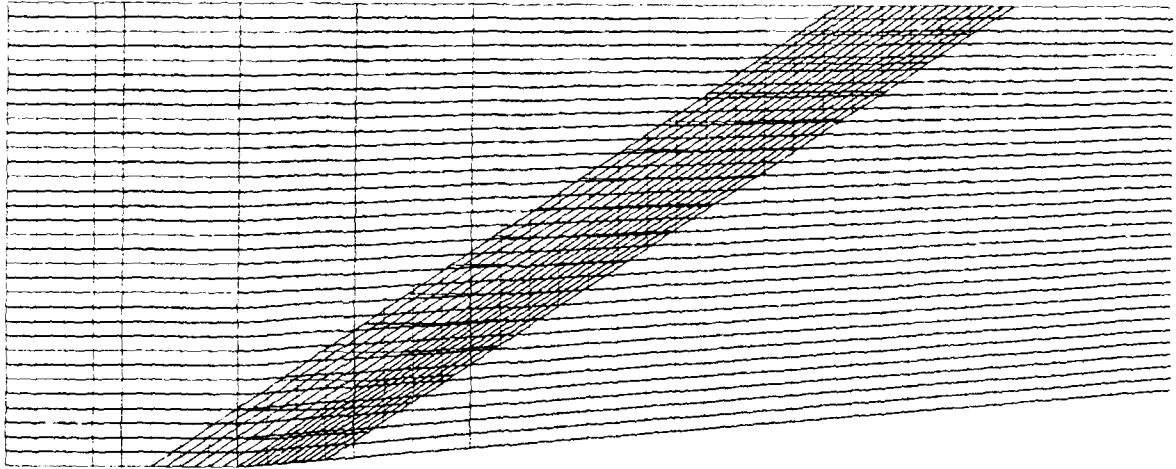
Figure 20: Two Embedded Grids for the 5° Ramp Near a Flat Plate

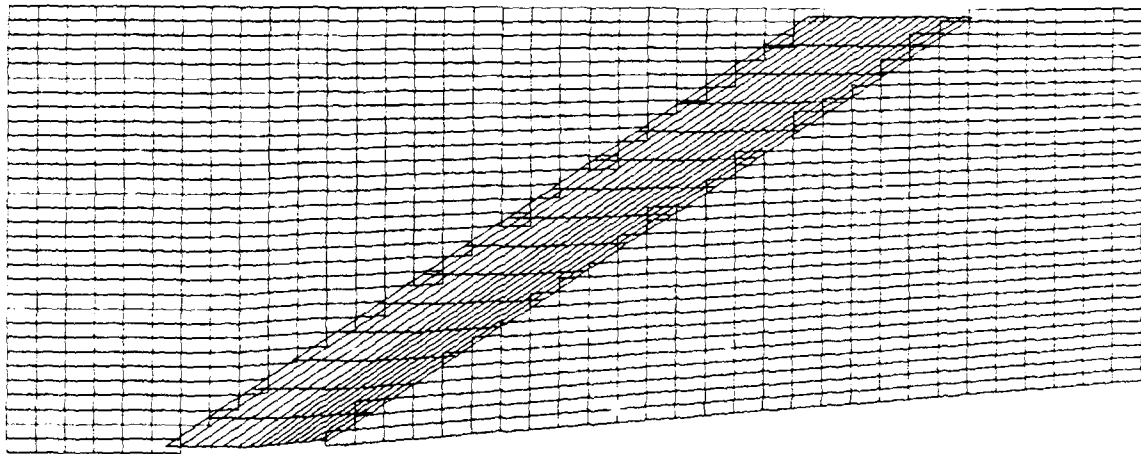
Figure 21: Two Grid Blanking for the 5° Ramp Near a Flat Plate

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
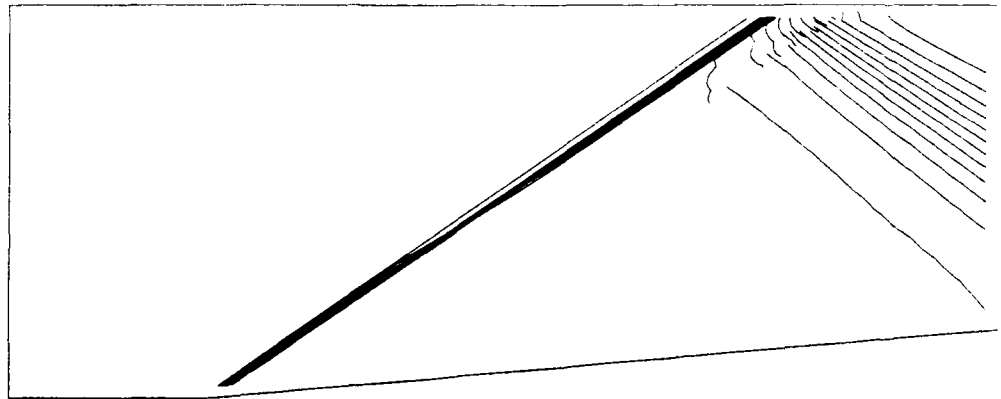1.40000
1.42000
1.44000
1.46000
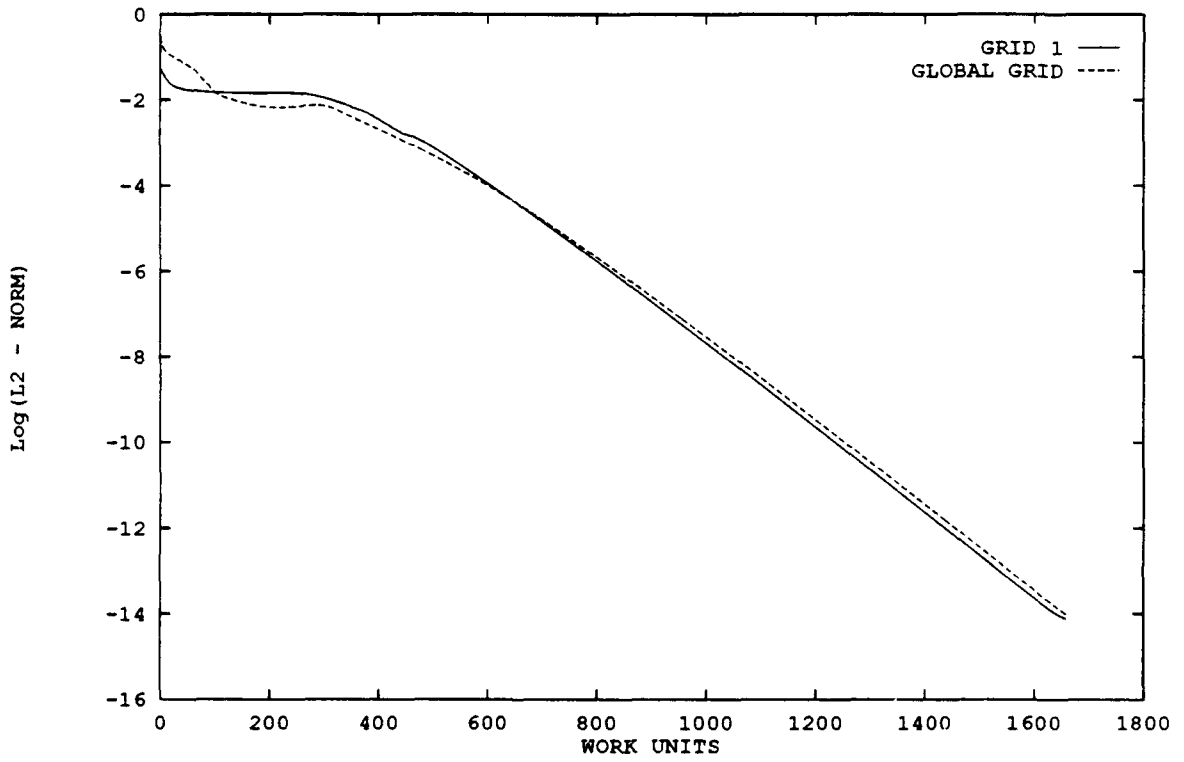
Figure 22: Two Embedded Grid Density Contours

Figure 23: Two Embedded Grid Residual Histories

Three grid case

Based on the results of the two grid calculations a third grid that is aligned with the reflected shock is inserted as indicated in Figure 24. The plot of the cell centered grids in Figure 25 shows the holes for this grid system. In this case the local grid from the previous section (referred to as grid 1) punches holes in both the new local grid (grid 2) and in the global grid while grid 2 only punches a hole in the global grid.

The solution obtained with this grid arrangement is shown in Figure 26. The solution remains good for the initial shock but the reflected shock is still not sufficiently resolved. Since the reflected shock is not resolved on grid 1 it cannot feed good values to grid 2 hence grid 2 is incapable of capturing the reflected shock accurately.

Residual histories for this calculation are in Figure 27. Once again the cost is increased by the addition of the embedded grid. This grid system requires 514.9 CPU seconds to reach machine zero.

Figure 24: Three Embedded Grids for the 5° Ramp Near a Flat Plate

Figure 25: Three Grid Blanking for the 5° Ramp Near a Flat Plate

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
1.40000
1.42000
1.44000
1.46000
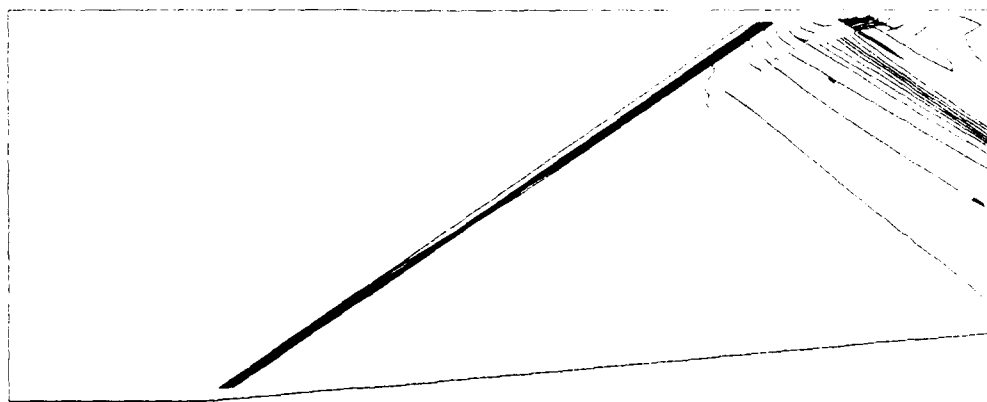
Figure 26: Three Embedded Grid Density Contours
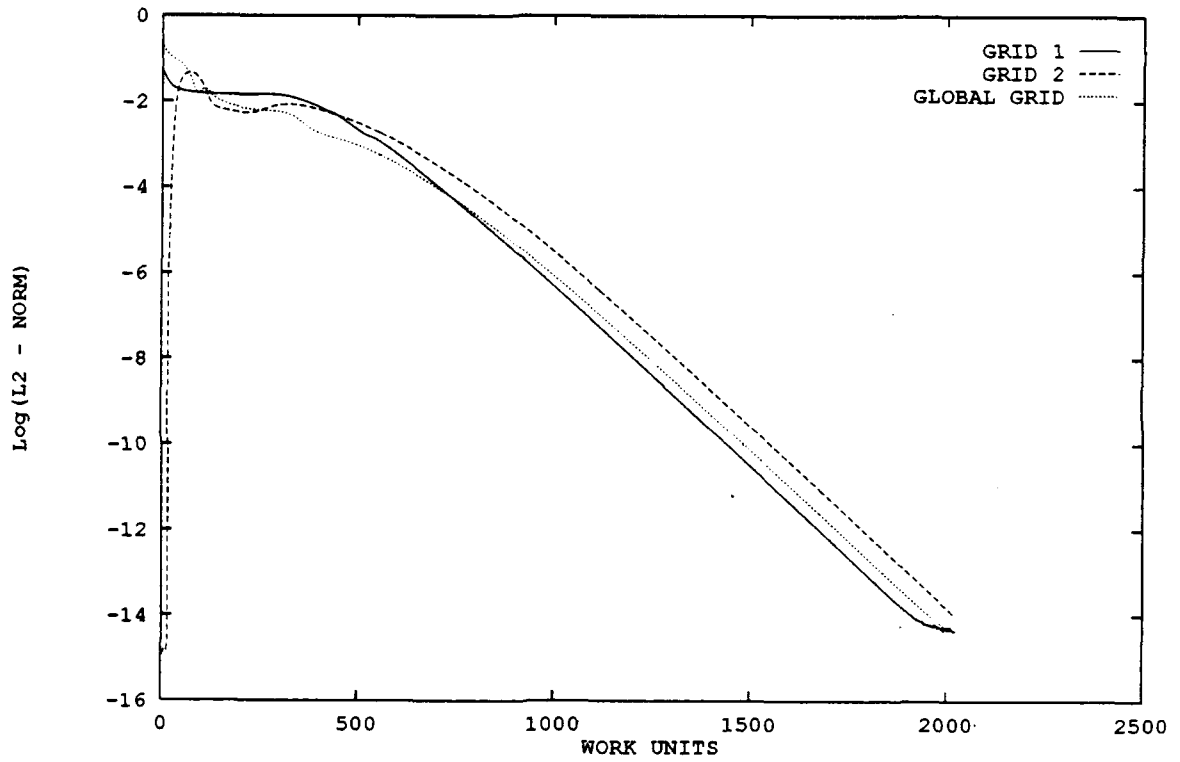
Figure 27: Three Embedded Grid Residual Histories

Four grid case

The addition of a third local grid (grid 3) is made in hopes of accurately repre-
senting both shocks near the reflection point. Grid 3 is a semicircle with its center at
the shock reflection point as indicated in Figure 28. The radial lines in this grid are
arranged such that one grid line will coincide with the initial shock and another line
with the reflected shock. In using this grid a singularity is introduced at the shock
reflection point. When the FDS scheme is applied to this grid the fluxes at this point
are set to zero to prevent flow through the solid wall. This new grid punches holes in
the three previous grids as shown in Figure 29.

Density contours of the solution obtained for this system are plotted in Figure 30.
This new solution is very accurate and is almost indistinguishable from the theoretical
solution.

The residual history for this grid system is plotted in Figure 31 and the CPU
time to reach this point is 865.0 seconds. Although this embedded system requires
many times the amount of work needed for the single grid, the solution obtained here
is far superior to that in Figure 7. Actually, this embedded solution is better than
the solution generated on the 160 by 128 single grid and costs less to produce. The
iterations in Figure 11 must be multiplied by 16 to compare with the embedded grid
work units (this is due to one embedded work unit being equal to one iteration on
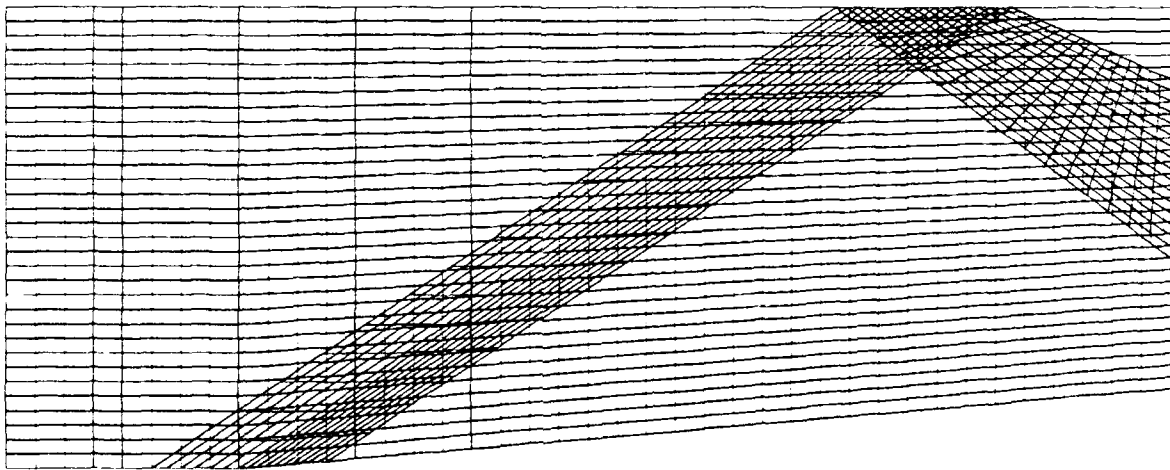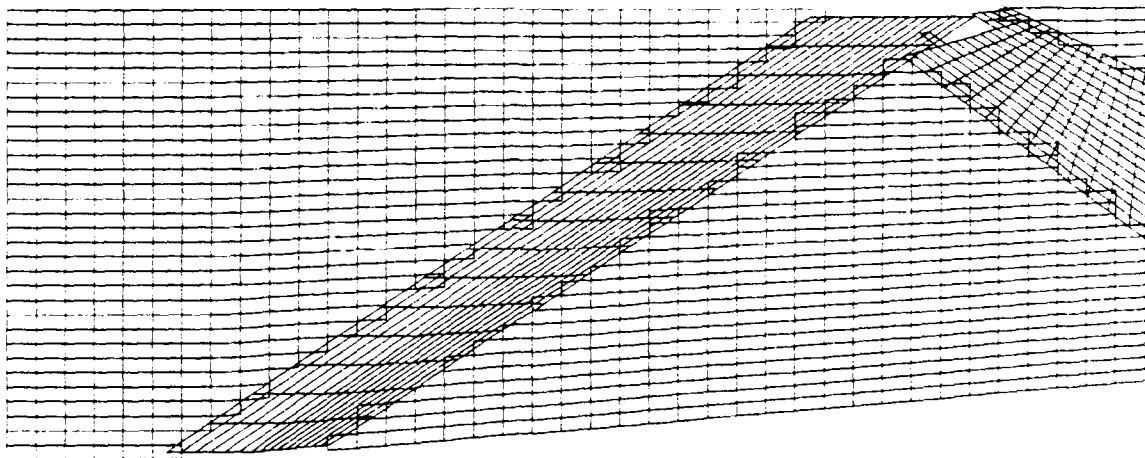the 40 by 32 grid).
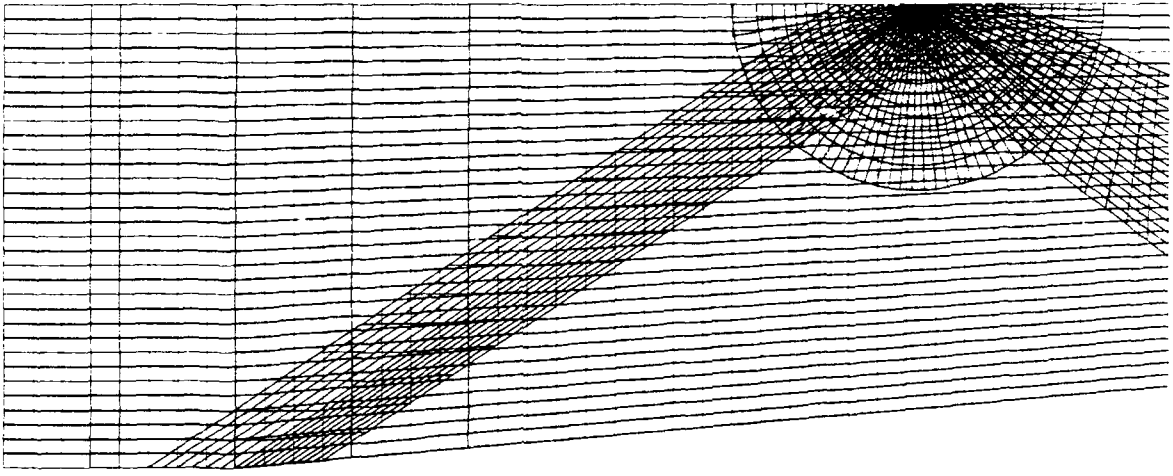
Figure 28: Four Embedded Grids for the 5° Ramp Near a Flat Plate

Figure 29: Four Grid Blanking for the 5° Ramp Near a Flat Plate

CONTOUR LEVELS

1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
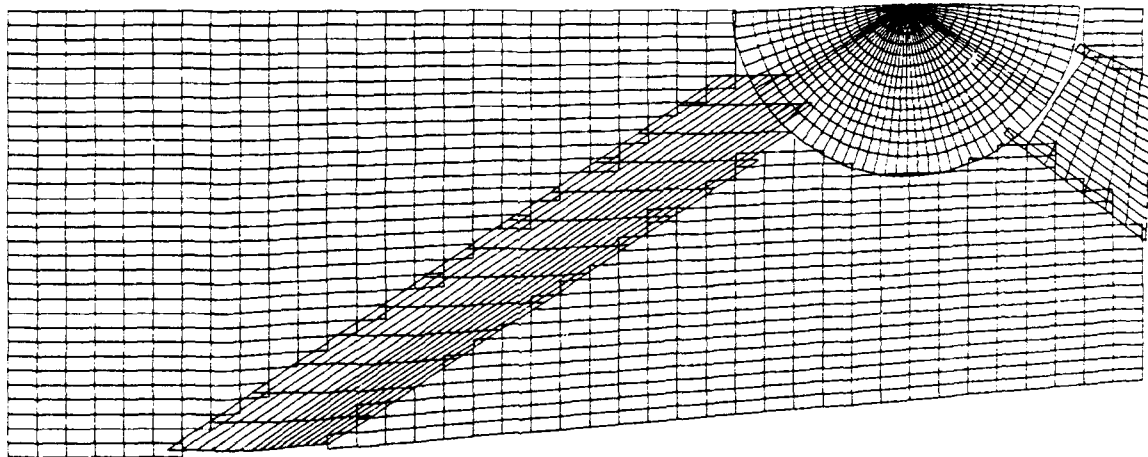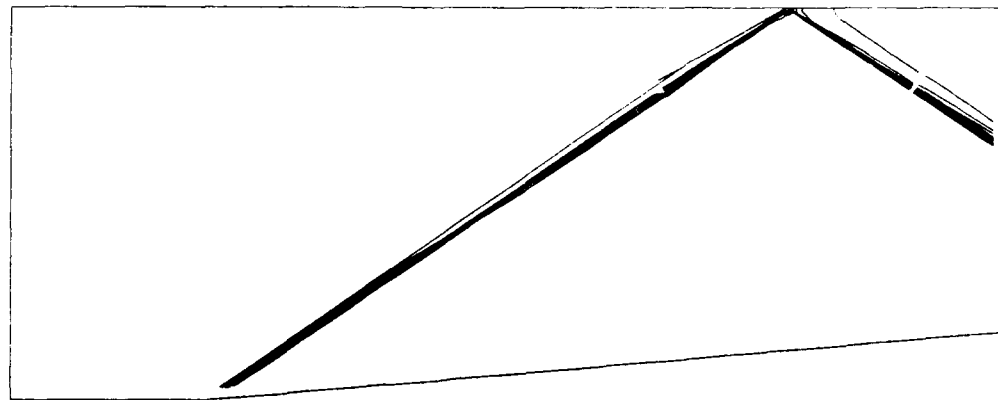1.38000
1.40000
1.42000
1.44000
1.46000
1.48000

Figure 30: Four Embedded Grid Density Contours
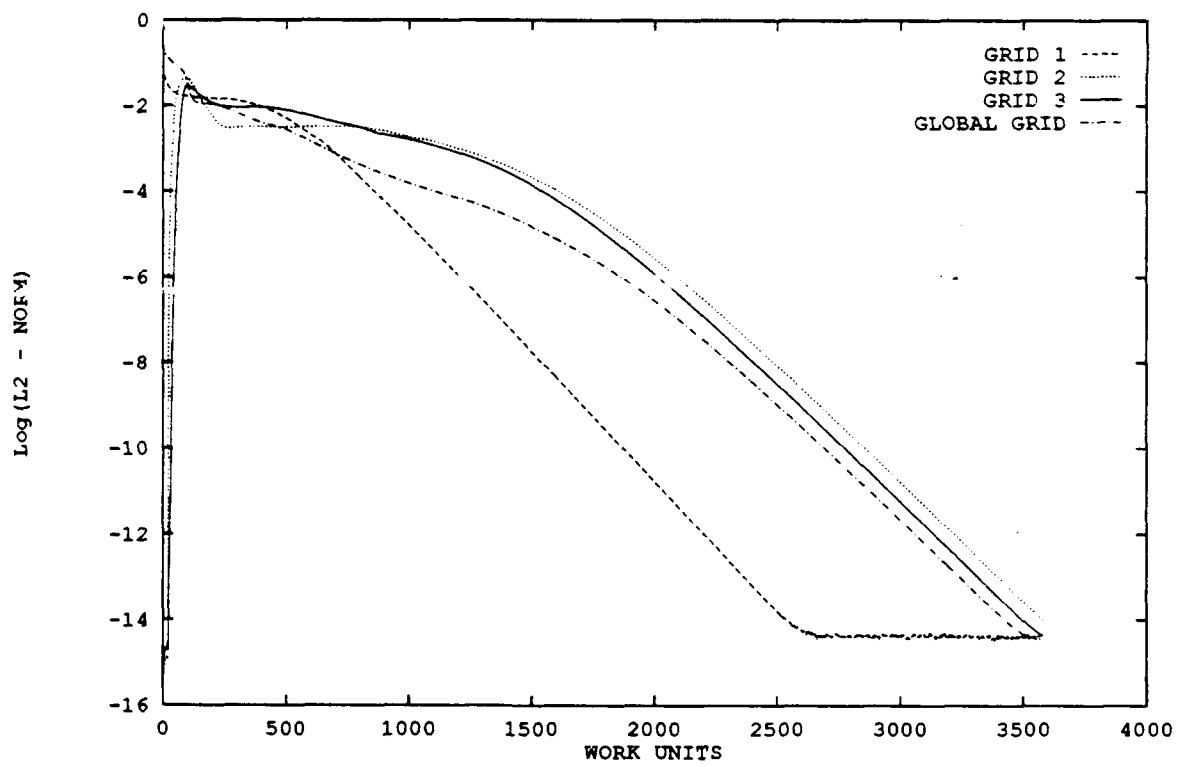
Figure 31: Four Embedded Grid Residual Histories

Nonaligned Multigrid Calculations

The goal of this section is to combine the improved solution obtained in the embedded approach with the improved convergence of the multigrid in order to show the feasibility of the nonaligned multigrid approach.

Restriction and Prolongation Operators

Prior to applying the nonaligned multigrid the restriction and prolongation operators need to be changed as indicated in Chapter 5. In order to determine how well these new operators perform one of the two level FAS calculations was repeated. The same solution was achieved requiring the same number of work units for both sets of operators. Since the same solution was obtainable in the same number of work units it appears that conservation is no more of a problem for the new operators than for the old operators; however, this does not indicate what will happen when the grids are no longer aligned. The new operators required approximately two seconds more than the old operators for the test case (the two level 2-4-2 V-Cycle on the 40 by 32 grid). This slight increase in cost is attributed to the time used to apply the more complicated interpolations and the time required for the "stencil jumping" procedure.

5° Ramp Calculations

Here the NAM procedure is applied to the embedded grid system of Figure 16 to provide accelerated convergence. Designating the local grid as level 1 and the global grid as level 2 the solution in Figure 32 is obtained.

Several calculations were made varying the iteration counts on the two levels in an attempt to achieve optimal convergence. The resulting number of work units required to reach machine zero are listed in Table 6 for five of these runs. This shows the difficulty of finding the best iteration count per level to provide maximum

convergence. This problem was also discussed in the FAS results. Since the severity of this problem increases with the addition of more levels an automated switching scheme is needed.

Eq. 50 is applied to the highest level at each point in the domain to obtain the residuals. This is accomplished by removing any point from the residual calculation that receives information from a higher level. The residual histories for the best case from Table 6, for the embedded grid calculations, and for the single grid calculation are plotted in Figure 33. This shows that the NAM calculations require approximately one third the number of work units required by the embedded grid procedure. For the run plotted, NAM requires 137.6 CPU seconds compared to the 340.4 CPU seconds required by the embedded approach. This is a 59% reduction in CPU time.

Three attempts are made to further increase the convergence rates for this set of grids. The first attempt is to coarsen level 1 to produce a new level 2 and moving the global grid to level 3. The resulting three level NAM procedure requires 524.8 work units and 133.6 CPU seconds to reach machine zero for the best iteration count per level $(\nu_1 = 2, \nu_2 = 0, \nu_3 = 3, \tilde{\nu}_2 = 1, \tilde{\nu}_1 = 2)$. This is not a significant improvement over the two level NAM results. Level 2 in this case acts as a filter to smooth the solution and residual being passed from level 1 to level 3. The same solution is obtained as in the two level calculation.

A second three level NAM procedure is defined by making the local grid level 1, the global grid level 2, and coarsening level 2 to make level 3. This system takes only 469.7 work units requiring 114.3 CPU seconds to converge to machine zero. The iteration count used here is $\nu_1 = 3, \nu_2 = 2, \nu_3 = 2, \tilde{\nu}_2 = 1$, and $\tilde{\nu}_1 = 4$. This system requires only 69.7 more work units (5.2 seconds) than the single grid calculation while producing a far superior solution. This is a 69% reduction in the CPU time required by the embedded grid approach.

CONTOUR LEVELS

1.00000
1.01000
1.02000
1.03000
1.04000
1.05000
1.06000
1.07000
1.08000
1.09000
1.10000
1.11000
1.12000
1.13000
1.14000
1.15000
1.16000
1.17000
1.18000
1.19000
1.20000
1.21000
1.22000
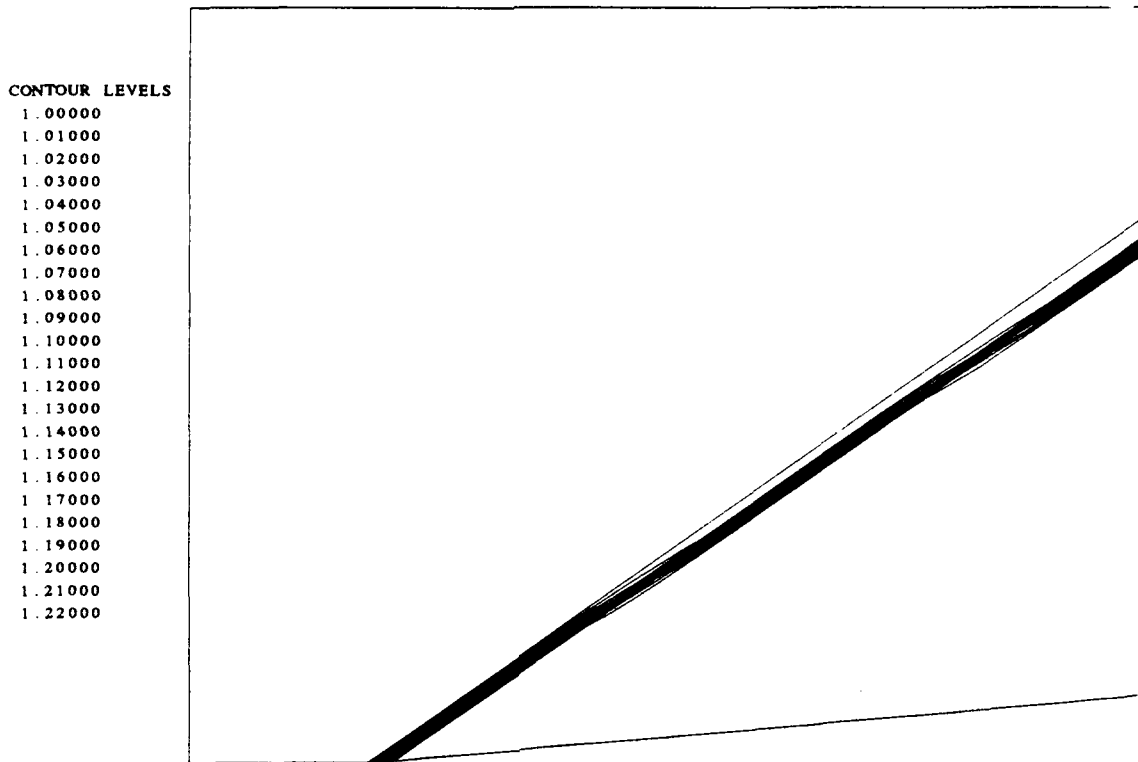
Figure 32: Density Contours for the 5° Ramp NAM Solution

The final attempt to converge faster than the single grid is made by generating a four level NAM system by coarsening level 3 in the first three level procedure discussed. The best case obtainable here was 474.95 work units. This may be caused by the speed at which the solution is being generated on the coarse grid being hindered by the two sets of artificial boundaries (the left and right sides of levels 1 and 2).

These results seem to indicate that the most significant gain in convergence will come from simply implementing NAM on the existing embedded grid system. It also appears to be better to coarsen the global grid to provide additional convergence.

Table 6: NAM Work Units

ν_1	ν_2	$\bar{\nu}_1$	Work Units
1	3	2	563.4
3	2	2	588.4
3	3	3	558.2
3	4	2	571.0
2	3	3	549.6
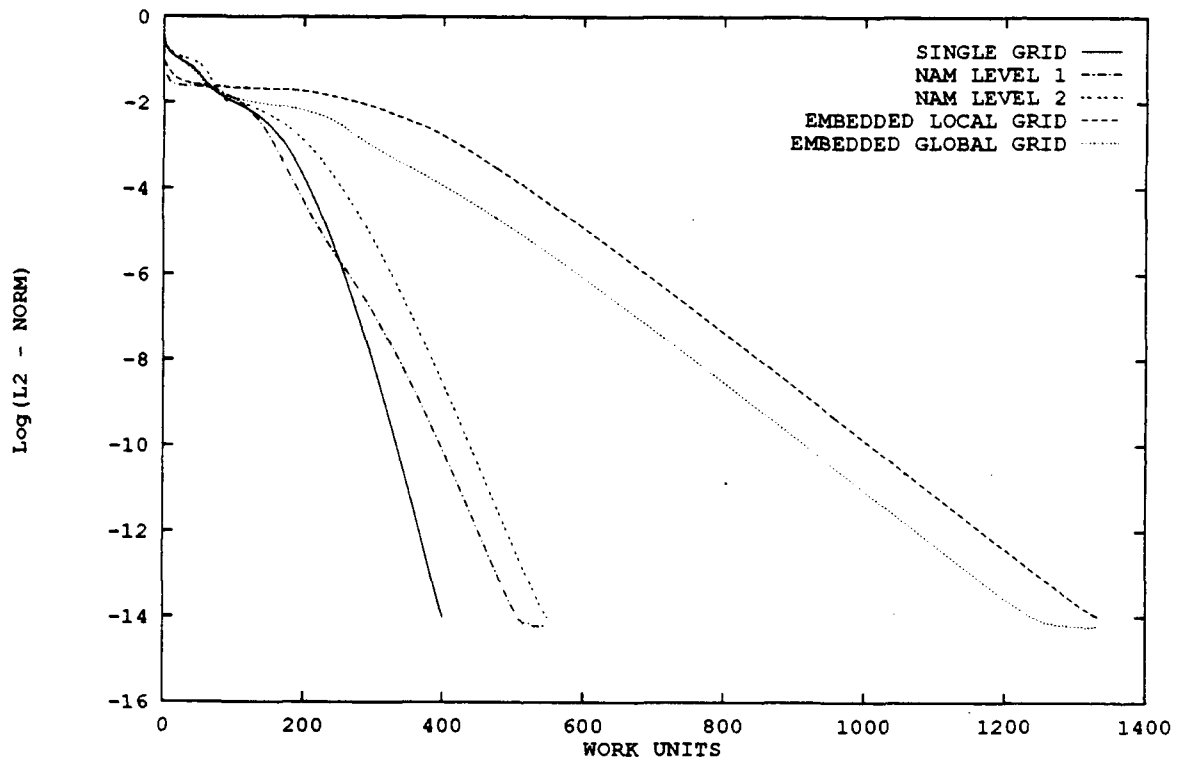
Figure 33: Residual Histories for the 5° Ramp 2 Level NAM Calculation

5° Ramp Near a Flat Plate Calculations

Two grid case

This calculation uses the grids of Figure 20 with the local grid considered to be level 1 and the global grid to be level 2. Density contours of the resulting solution are plotted in Figure 34. This shows the same inability to capture the reflected shock as the embedded solution (Figure 22).

The residual plot in Figure 35 compares the residuals from the two grid embedded calculations with the residuals of the current calculation. This plot shows that NAM requires under half the number of work units needed for the embedded calculation. This could be improved by coarsening the global grid but it is not attempted here since the solution is not that good. The CPU time for the NAM is 174.8 seconds while the embedded approach requires 425.3 seconds. The iteration count for this case was $\nu_1 = 1, \nu_2 = 2$, and $\tilde{\nu}_1 = 1$.

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
1.40000
1.42000
1.44000
1.46000

Figure 34: NAM 2 Level Density Contours
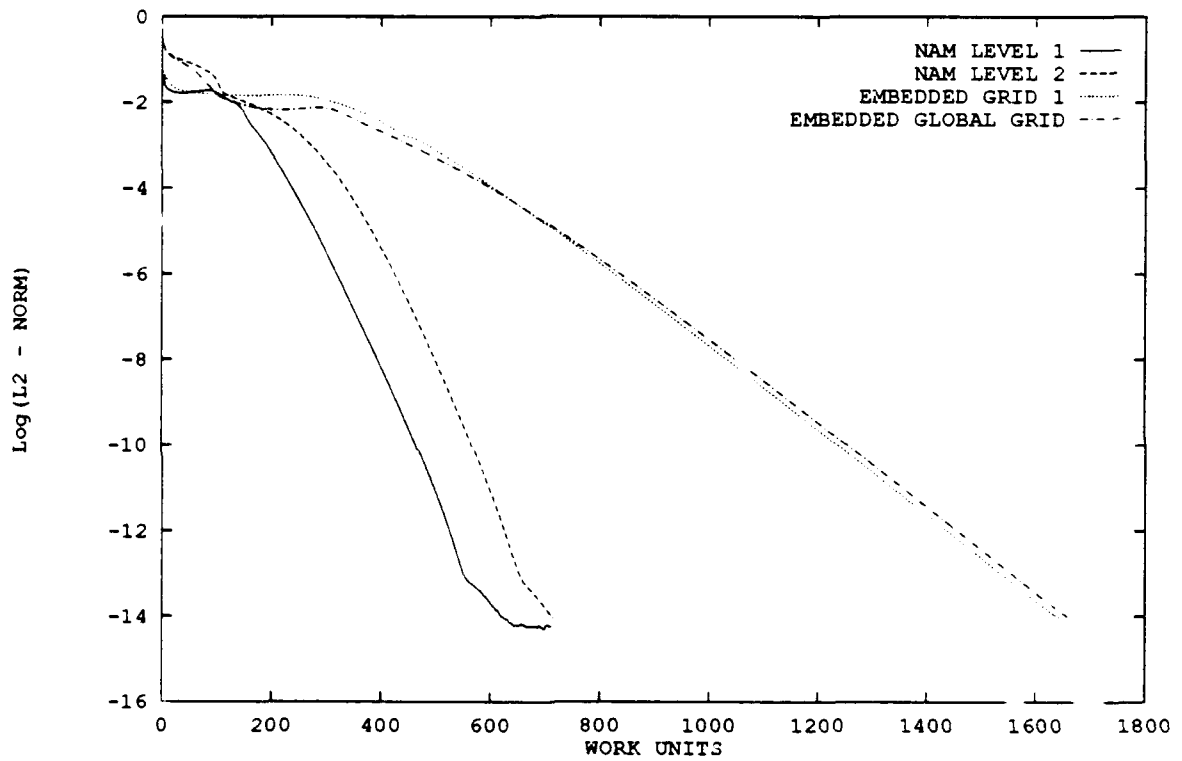
Figure 35: Embedded and NAM 2 Level Residual Histories

Three grid case

In this case the grids of Figure 24 are used. Level 1 contains the local grid covering the initial shock, level 2 is the grid covering the reflected shock, and level 3 is the global grid. Density contours of the resulting solution are plotted in Figure 36. The solution is similar to the embedded results (Figure 26), both are having difficulty capturing the reflected shock accurately. Once again this is due to the inability of the grids to accurately model the physics near the top wall.

The residuals for this calculation and the three grid embedded solution are plotted in Figure 37. This case also requires approximately half the number of work units required for the embedded procedure. NAM requires 259.7 seconds compared to 514.9 for the embedded calculation.

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
1.40000
1.42000
1.44000
1.46000
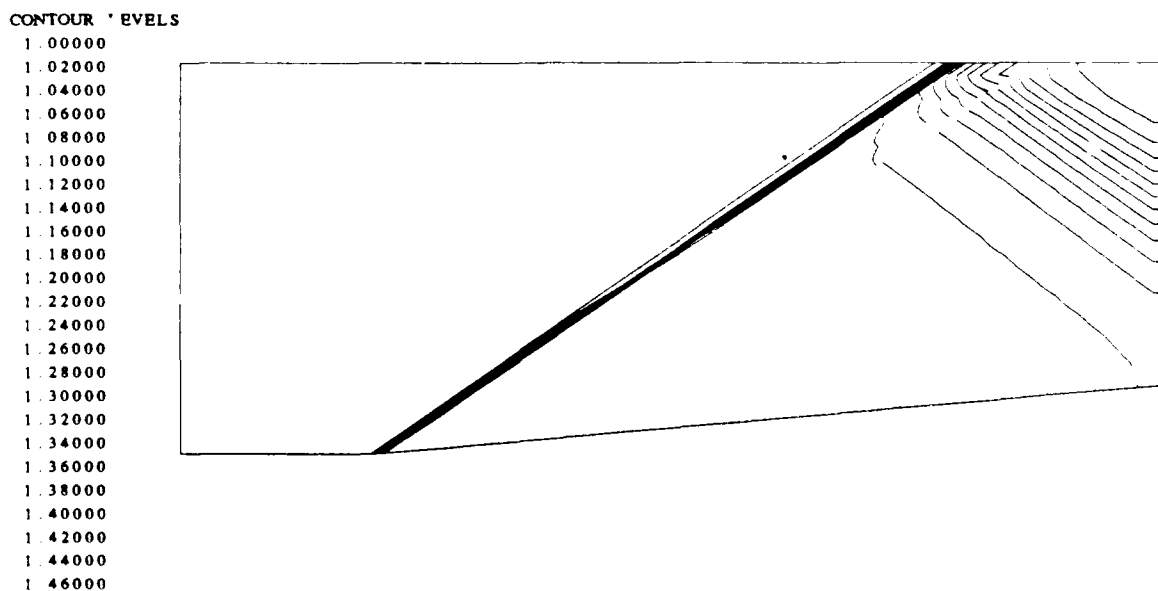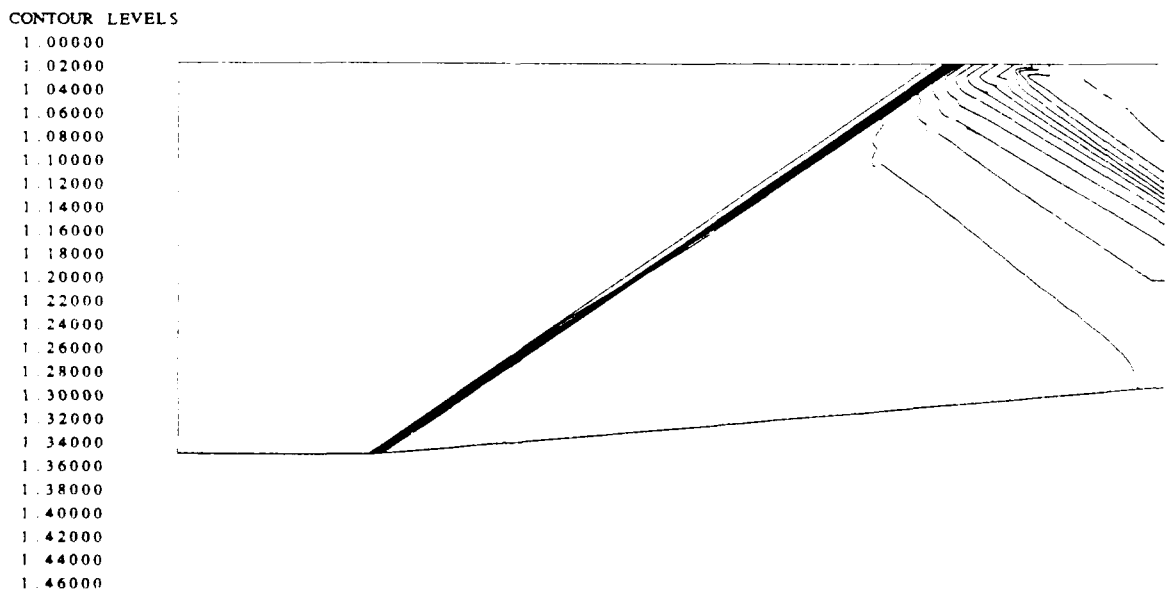
Figure 36: NAM 3 Level Density Contours
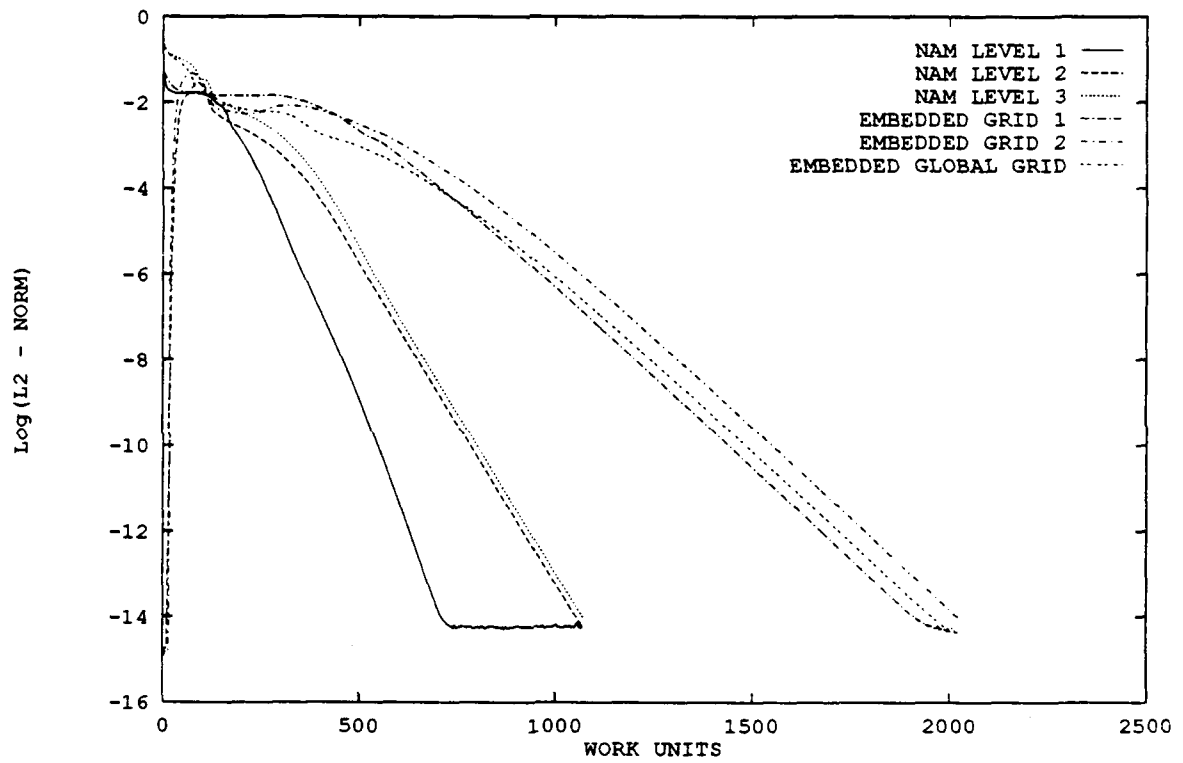
Figure 37: Embedded and NAM 3 Level Residual Histories

Four grid case

For this case the NAM is applied to the four grid system in Figure 28 treating the circular grid as level 1, the grid aligned with the initial shock as level 2, the grid aligned with the reflected shock as level 3 and the global grid is level 4. The four level grid scheduling diagram is shown in Figure 38. The circled numbers indicate the number of iterations performed on each level for a single NAM cycle. Density contours of the resulting solution are plotted in Figure 39. These are similar to that obtained by the embedded system.

Residual histories for this case are plotted in Figure 40. Comparing these to the embedded grid residuals in Figure 31 shows faster convergence is achieved by the NAM. CPU time required for NAM is 712.5 seconds compared to the 865.0 seconds required by the embedded grids.

To further increase the convergence the global grid in this four level system is coarsened to produce a five level system. The grid schedule is as shown in Figure 41 with the iteration count as indicated in the circles. The same solution is obtained in only 623.8 CPU seconds. The residual histories are as indicated in Figure 42. This is 27.9% faster than the equivalent embedded system.
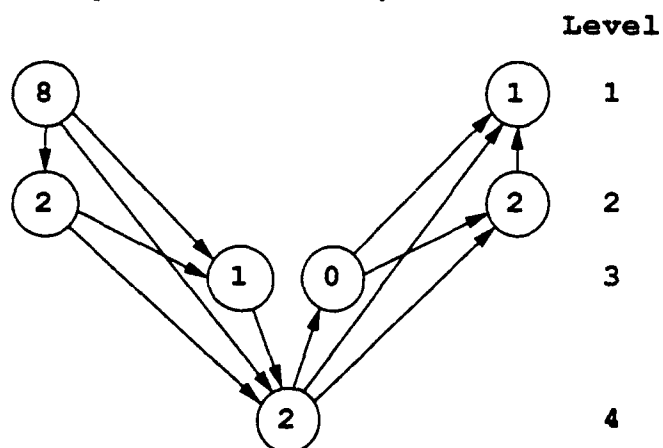
Figure 38: NAM 4 Level Grid Schedule

CONTOUR LEVELS
1.00000
1.02000
1.04000
1.06000
1.08000
1.10000
1.12000
1.14000
1.16000
1.18000
1.20000
1.22000
1.24000
1.26000
1.28000
1.30000
1.32000
1.34000
1.36000
1.38000
1.40000
1.42000
1.44000
1.46000
1.48000
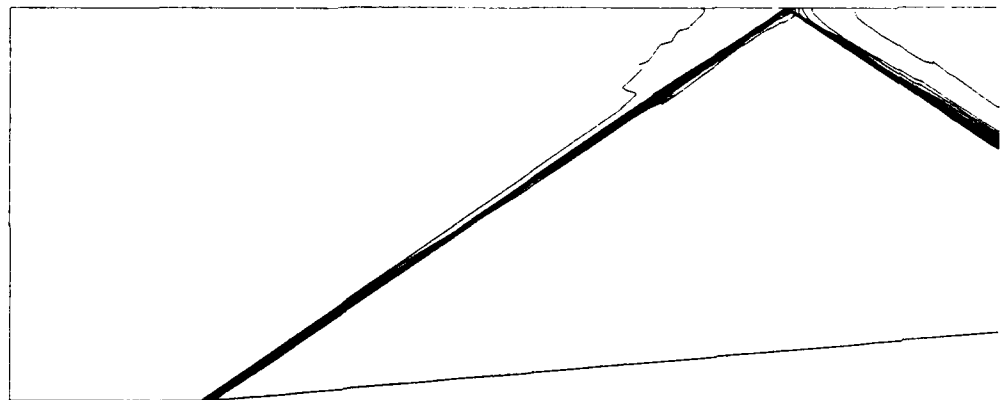
Figure 39: NAM 4 Level Density Contours
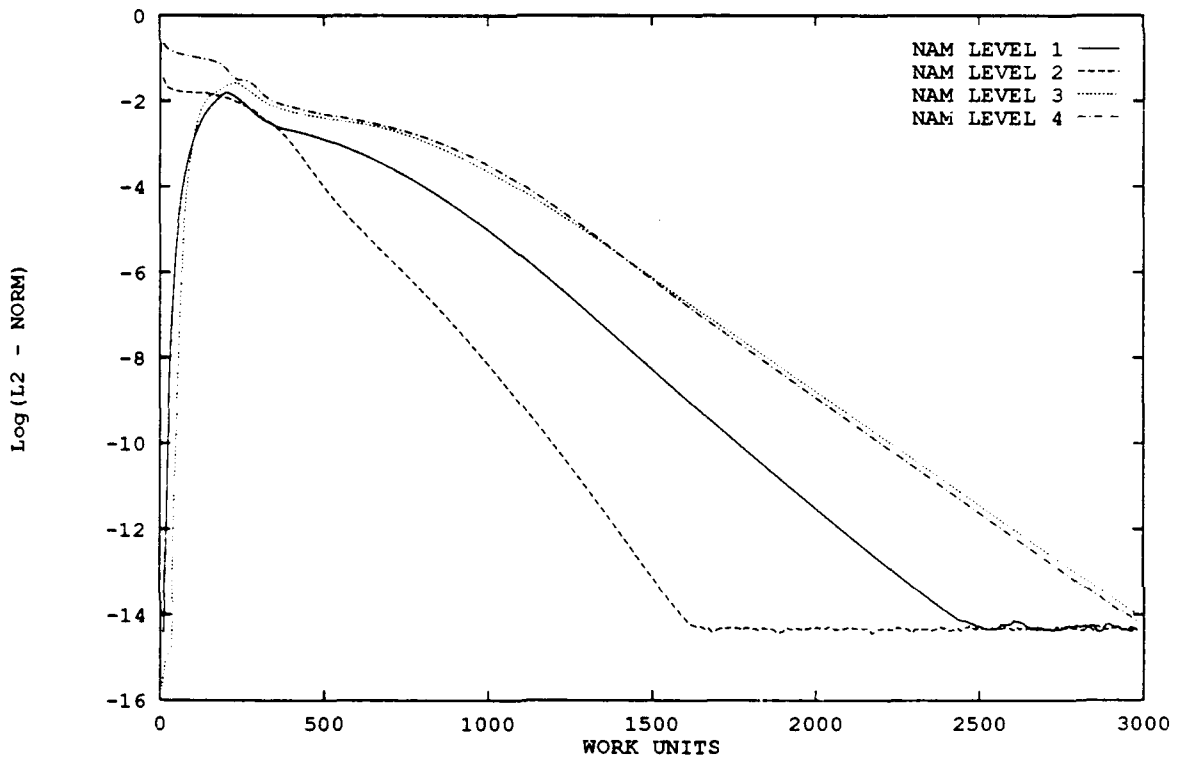
Figure 40: NAM 4 Level Residual Histories
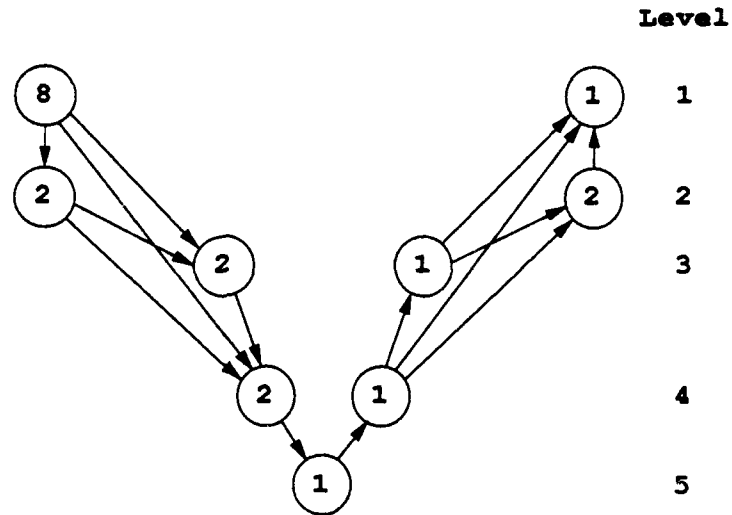
Level

Figure 41: NAM 5 Level Grid Schedule
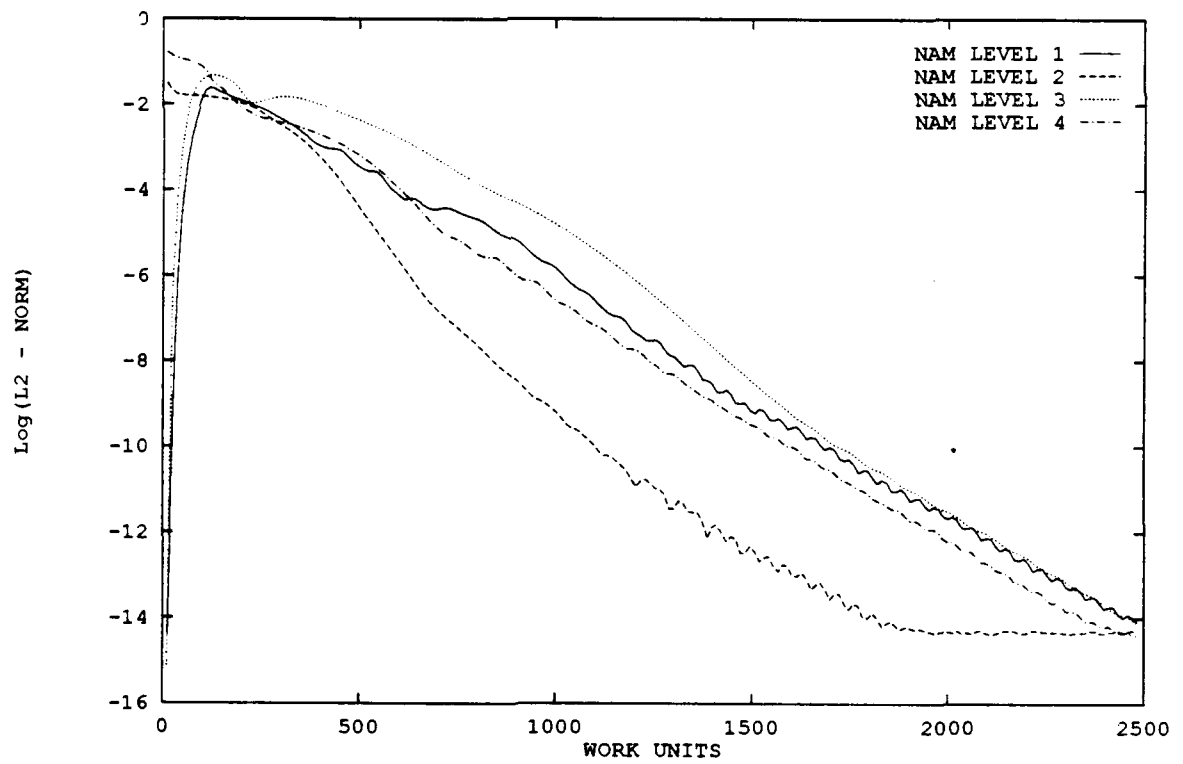
Figure 42: NAM 5 Level Residual Histories

CHAPTER 7
CONCLUSIONS

The explicit first order flux difference splitting (FDS) was used to solve the equations governing inviscid fluid flow on a single grid for the 5° ramp and the 5° ramp near a flat plate. Calculations were made for the Mach 2 case using local time stepping at a Courant number of 0.96. A grid refinement study was also performed on the 5° ramp near a flat plate to show grid dependency of the solution and to show the increasing cost as the grid becomes finer.

The multigrid full approximation scheme (FAS) was applied to this nonlinear problem to provide increased convergence rates and reduced central processing unit (CPU) time requirements. The solution was obtained independent of the number of levels and independent of the number of iterations performed on those levels. The convergence rate was determined to be very sensitive to the number of iterations performed on the various levels.

Several systems of embedded grids were implemented to provide increased accuracy near shock waves. Embedded grids were aligned with the shocks to take advantage of the excellent shock capturing capability of the FDS scheme. Although this method increased accuracy it degraded the convergence rate and increased the required CPU time.

The nonaligned multigrid was introduced to provide increased convergence for the embedded grid systems by treating the individual grids as levels in the multigrid solution procedure. This technique was able to converge 69% faster than the embedded grid procedure for the 5° ramp, and 28% faster for a more complex reflected shock case.

A great deal of work is still required to make the method into a "black box" engineering tool. Some of the topics include automatic determination of the number of iterations to perform on each level, automated grid scheduling for optimal performance, and determining the allowable change in grid spacing between levels. Another very important topic is developing a conservative procedure for communication between levels.

The current follow-on plan is to implement the NAM procedure in a viscous three dimensional higher order implicit computational fluid dynamics (CFD) code currently under development.

APPENDIX
STABILITY ANALYSIS

The first order linear wave equation has the same form as the Euler equations written in matrix form (Eq. 1, Chapter 2).

$$\frac{\partial q}{\partial t} + \frac{\partial f_\xi}{\partial \xi} + \frac{\partial f_\eta}{\partial \eta} = 0 \tag{51}$$

In this case q is the dependent variable and the fluxes are $f_\xi = aq$ and $f_\eta = bq$ where a and b are constants.

Applying the flux difference splitting (FDS) described in Chapter 3 produces the following discretization, which is simply the scalar equivalent of Eq. 19.

$$\frac{q_{i,j}^{n+1} - q_{i,j}^n}{\Delta t} + (\overline{f}_\xi)_{i+1/2,j}^n - (\overline{f}_\xi)_{i-1/2,j}^n + (\overline{f}_\eta)_{i,j+1/2}^n - (\overline{f}_\eta)_{i,j-1/2}^n = 0 \tag{52}$$

Using Eqs. 20, 22, and noting that the left and right eigenvectors are simply reciprocals for the scalar case, the numerical fluxes are reduced to

$$(\overline{f}_\xi)_{i+1/2,j}^n = aq_{i,j}^n + \lambda_\xi^-(q_{i+1,j}^n - q_{i,j}^n) \tag{53a}$$

$$(\overline{f}_\eta)_{i,j+1/2}^n = bq_{i,j}^n + \lambda_\eta^-(q_{i,j+1}^n - q_{i,j}^n). \tag{53b}$$

The actual fluxes f_k evaluated at the discrete points are also represented in terms of the dependent variable. The values for the non-positive eigenvalues depend on a and b as indicated below. The case of a or b equal to zero is not considered since it is trivial for the two dimensional case.

Case 1   for $a > 0$ and $b > 0 \rightarrow \lambda_\xi^- = 0, \lambda_\eta^- = 0$

Case 2   for $a < 0$ and $b > 0 \rightarrow \lambda_\xi^- = a, \lambda_\eta^- = 0$

Case 3   for $a > 0$ and $b < 0 \rightarrow \lambda_\xi^- = 0, \lambda_\eta^- = b$

Case 4   for $a < 0$ and $b < 0 \rightarrow \lambda_\xi^- = a, \lambda_\eta^- = b$

Substituting the numerical fluxes into Eq. 52 and rearranging yields

$$\begin{aligned} q_{i,j}^{n+1} = q_{i,j}^n &- \Delta t \left[a(q_{i,j}^n - q_{i-1,j}^n) + \lambda_\xi^-(q_{i+1,j}^n - 2q_{i,j}^n + q_{i-1,j}^n) \right] \\ &- \Delta t \left[b(q_{i,j}^n - q_{i,j-1}^n) + \lambda_\xi^-(q_{i,j+1}^n - 2q_{i,j}^n + q_{i,j-1}^n) \right]. \end{aligned} \tag{54}$$

The next step is to apply the von Neumann (or Fourier) stability analysis [1]. This is done by substituting

$$q_{i,j}^n = \zeta^n e^{i\beta i} e^{i\alpha j}. \tag{55}$$

Where $i = \sqrt{-1}$ into Eq. 54. Making the substitution, using trigonometric relations and grouping terms in real and imaginary parts yields

$$\zeta = 1 - 2\Delta t \left[(a - 2\lambda_\xi^-) \sin^2(\beta/2) + (b - 2\lambda_\eta^-) \sin^2(\alpha/2) \right] - i\Delta t \left[a \sin\beta + b \sin\alpha \right]. \tag{56}$$

Once the equation is in this form require $|\zeta| \leq 1$ to obtain the stability condition on the time step size. Applying this restriction and solving for Δt.

$$\Delta t \leq \frac{4 \left[(a - 2\lambda_\xi^-) \sin^2(\beta/2) + (b - 2\lambda_\eta^-) \sin^2(\alpha/2) \right]}{4 \left[(a - 2\lambda_\xi^-) \sin^2(\beta/2) + (b - 2\lambda_\eta^-) \sin^2(\alpha/2) \right]^2 + \left[a \sin\beta + b \sin\alpha \right]^2} \tag{57}$$

The four possible variations on the eigenvalues listed previously are now applied to this equation and the most restrictive values for α and β are sought.

Case 1:   Eq. 57 becomes

$$\Delta t \leq \frac{4 \left(a \sin^2(\beta/2) + b \sin^2(\alpha/2) \right)}{4 \left(a \sin^2(\beta/2) + b \sin^2(\alpha/2) \right)^2 + \left(a \sin\beta + b \sin\alpha \right)^2}. \tag{58}$$

The minimum value of the right hand side of this equation occurs at $\alpha = \beta = \pi$ causing the stability condition to become

$$\Delta t \leq \frac{1}{a + b}. \tag{59}$$

Case 2:   Eq. 57 becomes

$$\Delta t \leq \frac{4 \left(-a \sin^2(\beta/2) + b \sin^2(\alpha/2) \right)}{4 \left(-a \sin^2(\beta/2) + b \sin^2(\alpha/2) \right)^2 + \left(a \sin\beta + b \sin\alpha \right)^2}. \tag{60}$$

The minimum value of the right hand side of this equation occurs at $\alpha = \beta = \pi$ causing the stability condition to become

$$\Delta t \leq \frac{1}{b - a}. \tag{61}$$

Case 3:   Eq. 57 becomes

$$\Delta t \leq \frac{4\left(a\sin^2(\beta/2) - b\sin^2(\alpha/2)\right)}{4\left(a\sin^2(\beta/2) - b\sin^2(\alpha/2)\right)^2 + (a\sin\beta + b\sin\alpha)^2}. \qquad (62)$$

The minimum value of the right hand side of this equation occurs at $\alpha = \beta = \pi$ causing the stability condition to become

$$\Delta t \leq \frac{1}{a-b}. \qquad (63)$$

Case 4:   Eq. 57 becomes

$$\Delta t \leq \frac{4\left(-a\sin^2(\beta/2) - b\sin^2(\alpha/2)\right)}{4\left(-a\sin^2(\beta/2) - b\sin^2(\alpha/2)\right)^2 + (a\sin\beta + b\sin\alpha)^2}. \qquad (64)$$

The minimum value of the right hand side of this equation occurs at $\alpha = \beta = \pi$ causing the stability condition to become

$$\Delta t \leq \frac{1}{-a-b}. \qquad (65)$$

The stability condition in general for FDS algorithm applied to the two dimensional wave equation can be written as

$$\Delta t \leq \frac{1}{|a| + |b|}. \qquad (66)$$

It should be noted that this analysis assumes a linear difference equation with periodic boundary conditions. The standard practice is to ignore these assumptions which usually provides good results. To apply this condition to the Euler equations the maximum eigenvalues (Eq. 14, Chapter 2) of the flux Jacobian matrix are substituted into the stability condition for the wave equation.

$$\Delta t \leq \frac{1}{|\theta_\xi| + c\sqrt{\xi_x^2 + \xi_y^2} + |\theta_\eta| + c\sqrt{\eta_x^2 + \eta_y^2}} \qquad (67)$$

REFERENCES

[1] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, 1984.

[2] Anderson, W. Kyle, *Implicit Multigrid Algorithms for the Three-Dimensional Flux Split Euler Equations*, Ph.D. Dissertation, Mississippi State University, Starkville, August 1986.

[3] Baysal, O., Fouladi, K., and Lessard, V. R., "Multigrid and Upwind Viscous Flow Solver on Three-Dimensional Overlapped and Embedded Grids," AIAA Journal, Vol. 29, No. 6, June 1991, pp. 903-910.

[4] Belk, Dave M., *Unsteady Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids*, Ph.D. Dissertation, Mississippi State University, Starkville, August 1986.

[5] Benek, J. A., Steger, J. L., Dougherty, F. C., and Buning, P. G., "Chimera: A Grid Embedding Technique," AEDC-TR-85-64, December 1985.

[6] Berger, M. J., "On Conservation at Grid Interfaces," SIAM Journal on Numerical Analysis, Vol. 24, No. 5, October 1987, pp. 967-984.

[7] Berger, M. J. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," AIAA Journal, Vol. 23, No. 4, April 1985, pp. 561-568.

[8] Brandt, Achi, "Multilevel Adaptive Computations in Fluid Dynamics," AIAA Journal, Vol. 18, No. 10, October 1980, pp. 1165-1172.

[9] Brandt, Achi, "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," Computational Fluid Dynamics Lecture Series at the von-Karmen Institute for Fluid Dynamics, Rhode-Saint-Gense, March 1984.

[10] Briggs, William L., *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[11] Brown, Jeffrey J., "A Multigrid Mesh-Embedding Technique for Three-Dimensional Transonic Potential Flow Analysis," NASA Conference Publication 2202, October 1981, pp. 131-149.

[12] Dietz, W. E. and Suhs, N. E.,"Pegasus 3.0 User's Manual," AEDC-TR-89-7, August 1989.

[13] Dougherty, F. Carroll, *Development of a Chimera Grid Scheme with Applications to Unsteady Problems*, Ph.D. Dissertation, Stanford University, Palo Alto, California, June 1985.

[14] Dougherty, F. C., Benek, J. A., and Steger, J. L., "On Applications of Chimera Grid Schemes to Store Separation," NASA-TM-88193, October 1985.

[15] Hackbusch, Wolfgang, *Multi-Grid Methods and Applications*, Springer-Verlag Berlin, 1985.

[16] Hänel, D., Schröder, W., and Seider, G., "Multigrid Methods for the Solution of the Compressible Navier-Stokes Equations," Proceedings of the Fourth GAMM-Seminar, Kiel, January 1988, pp. 114-127.

[17] Henshaw, W. D. and Chesshire, G., "Multigrid on Composite Meshes," SIAM Journal on Scientific and Statistical Computation, Vol. 8, No. 6, November 1987, pp. 914-923.

[18] Hessenius, K. A. and Rai, M. M., "Applications of a Conservative Zonal Scheme to Transient and Geometrically Complex Problems," AIAA-84-1532, June 1984.

[19] John, James E. A., *Gas Dynamics*, Second Edition, Allyn and Bacon, Inc., Boston, 1984.

[20] Mavriplis, D. J., "Multigrid Solution of the Two Dimensional Euler Equations on Unstructured Triangular Meshes," AIAA Journal, Vol. 26, No. 7, July 1988, pp. 824-831.

[21] McCormick, Stephen F., *Multilevel Adaptive Methods for Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[22] Meakin, R. and Suhs, N., "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion," AIAA-89-1996, June 1989.

[23] Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA-84-0164, January 1984.

[24] Rai, M. M., "An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculations," AIAA-85-0488, January 1985.

[25] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," Journal of Computational Physics, Vol. 43, 1981 pp. 357-372.

[26] Simpson, L. Bruce, *Unsteady Three-Dimensional Thin-Layer Navier-Stokes Solutions on Dynamic Blocked Grids*, Ph.D. Dissertation, Mississippi State University, Starkville, December 1988.

[27] Swanson, R. C. and Radespiel, R., "Cell Centered and Cell Vertex Multigrid Schemes for the Navier-Stokes Equations," AIAA Journal, Vol. 29, No. 5, May 1991, pp. 697-703.

[28] Thompson, J. F., and Gatlin, B., "Program EAGLE User's Manual: Surface Generation Code," AFATL-TR-88-117, Vol. II, September 1988.

[29] Thompson, J. F., and Gatlin, B., "Program EAGLE User's Manual: Grid Generation Code," AFATL-TR-88-117, Vol. III, September 1988.

[30] Thompson, J. F., Warsi, Z. A. U., and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, North-Holland, New York, 1985.

[31] Whitfield, D. L. and Janus, J. M., "Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting," AIAA-84-1552, June 1984.

[32] Whitfield, D. L., Janus, J. M., and Simpson, L. B., "Implicit Finite Volume High Resolution Wave-Split Scheme for Solving the Unsteady Three-Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids," MSSU-EIRS-ASE-88-2, February 1988.

[33] Yadlin, Yoram and Caughey, D. A., "Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow," AIAA Journal, Vol. 29, No. 5, May 1991, pp. 712-719.

BIOGRAPHICAL SKETCH

Rudy A. Johnson is the oldest of three sons born to Glen and Mary Johnson. Born in Defiance, Ohio, in 1966 he attended Paulding High School and worked part time in his parents' garage. After graduation in 1984 he attended the University of Cincinnati to pursue his undergraduate degree in aerospace engineering. In 1986 he began working as a co-op student in the computational fluid dynamics (CFD) section of what was then the Air Force Armament Laboratory (AFATL/FXA) at Eglin AFB. His primary responsibility was the development of a generalized computer graphics software package for displaying two and three dimensional CFD results. During this time he also co-authored an Air Force technical report (AFATL-TR-88-115).

Upon graduation from college he began working full time in the CFD section (currently designated WL/MNAA) as a junior research scientist/engineer and program manager. He received the Junior Scientist/Engineer of the Year Award" in May 1990 and has received performance awards each year since. He began taking graduate-level classes on a part time basis during the fall semester of 1989 at the University of Florida Graduate Research Center at Eglin AFB. He was accepted as a degree-seeking student in the spring of 1990. In October of 1991 he wed Mary Landers.

DISTRIBUTION
(WL-TR-92-7039)

Defense Technical Info. Center
Attn: DTIC/DDAC   2   Eglin AFB offices:
Cameron Station
Alexandria VA 22304-6145   WL/CA-N   1
                                            WL/MNOI (Scientific and Tech. Info. Facility) 1

AFSAA/SAI   1
Washington DC 20330-5420   HQ USAFE/INATW   1
                                            APO NY 09012-5001

AUL/LSE   1
Maxwell AFB AL 36112-5564   WL/FIES/SURVIAC   1
                                            Wright-Patterson AFB OH 45433-6553

Eglin AFB offices: Wright-Patterson AFB OH 45433-6503

WL/MNSI   1   ASC/ENSTA   1
WL/MNAG   1   ASC/XRH   1
WL/MNM   1
ASC/XRC   1   Wright-Patterson AFB OH 45433-6553
WL/MNAA   4
WL/MNAV   1   WL/FIM   1
WL/MNPX   1   WL/CA-F   1
                                            WL/FIB   1
AFIA/INT   1   WL/FIGX   1
Bolling AFB DC 20332-5000   WL/FIGCC   1

Commander   WL/TXA   1
U. S. Army Missile Command   1   Wright-Patterson AFB OH 45433-6523
Redstone Science Info. Center
Attn: AMSMI-RD-CS-R/Documents   EOARD/LDV   1
Redstone Arsenal AL 35898-5241   Box 14
                                            FPO NY 09510-0200
Commander
Naval Weapons Center (Code 3431) 1   NASA Langley Research Center   1
Attn: Technical Library   Technical Library - MS 185
China Lake CA 93555-6001   Attn: Document Cataloging
                                            Hampton VA 23665-5225
NASA-Ames Research Center   1
Attn: Dr. Tony Strawa, MS 229-3
Moffett Field CA 94035-0000