FORMULATION OF BOUNDARY CONDITIONS FOR THE MULTIGRID ACCELERATION OF THE EULER AND NAVIER STOKES EQUATIONS

Thomas Neil Jentink
William J. Usab, Jr.

School of Aeronautics and Astronautics
Purdue University
West Lafayette, IN

Final Technical Report for
NASA Research Grant No. NAG 3-875

Grant Title: Development of a Multi-Grid Acceleration Method for the Solution of the Average-Passage Equations

Principal Investigator: W. J. Usab, Jr.

Report Period: February 20, 1988 to March 19, 1990

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF SYMBOLS

Symbol	*Description*
a	sonic velocity
ρ	density
E	total energy per unit volume
u	x-component of velocity
v	y-component of velocity
p	pressure
κ	coefficient of thermal conductivity
M_∞	free stream mach number
T	temperature
t	time
τ	shear stress
q	heat transfer
μ	viscosity coefficient
C_p	specific heat at constant pressure
P_r	Prandtl number
γ	ratio of specific heats
J	Jacobian of general coordinate transformation
x,y	cartesian coordinate system
ξ , η	general non-orthogonal coordinate system
α_i	Runge-Kutta coefficients
CFL	Courant-Friedrichs-Lewy stability condition

ABSTRACT

Jentink, Thomas Neil. MSAA, Purdue University, August, 1989. Formulation Of Boundary Conditions For The Multigrid Acceleration Of The Euler And Navier Stokes Equations. Major Professor: William J. Usab, Jr.

An explicit, Multigrid algorithm has been written to solve the Euler and Navier Stokes equations with special consideration given to the coarse mesh boundary conditions. These are formulated in a manner consistent with the interior solution, utilizing forcing terms to prevent coarse-mesh truncation error from affecting the fine-mesh solution. A 4-Stage Hybrid Runge-Kutta Scheme is used to advance the solution in time, and Multigrid convergence is further enhanced by using local time-stepping and implicit residual smoothing. Details of the algorithm are presented along with a description of Jameson's standard Multigrid method and a new approach to formulating the Multigrid equations. This approach utilizes a general filtering operator to derive the coarse-mesh equations in partial differential form, and is reduced to Jameson's Multigrid scheme by specifying a particular discrete filter. The correct boundary transfer operator is formulated from this filter, and forcing terms are then derived for the coarse-mesh boundary conditions. Results are presented for inviscid channel flow of subsonic, transonic, and supersonic speeds over a circular-arc bump, viscous flat plat flow, viscous flow over a circular-arc bump, and inviscid and viscous flow over a VKI gas turbine rotor blade. These results will show the importance of the correct

implementation of the coarse-mesh boundary conditions by comparison of convergence levels with and without the boundary forcing terms that were derived.

CHAPTER 1 - INTRODUCTION

Advances in computer technology over the past few years , coupled with the rising cost of experimentation has resulted in the increased role of computational methods in the design process. The improved algorithm development occurring along with these advances has made it possible to surpass the level of approximation provided by potential solvers. The higher order Euler and Navier-Stokes equations can now be solved for a number of more realistic problems covering a wide range of complexity.[1–4] Finite-volume, explicit time-stepping schemes have been used quite extensively in solving these problems for both the Euler and Navier-Stokes equations.[1–9] Explicit schemes are generally favored over implicit schemes due to their ease of implementation, minimum number of computations, and low storage requirements. Although explicit schemes have been successful, their main disadvantage over implicit schemes is their stability limit. Whereas an implicit scheme is by theory unconditionally stable and therefore able to use very large time-steps, an explicit scheme can be quite limited by the maximum allowable time-step that may be used. In order to retain the advantages of an explicit scheme, while still reaching solutions efficiently, a large amount of research has focused on improving the performance of explicit time-stepping schemes.

As part of this effort, a class of hybrid Runge-Kutta time-stepping schemes was developed by Jameson.[6] Runge-Kutta schemes are usually chosen for their high level of time accuracy, but in the interest of steady-state solutions, Jameson neglected time accuracy and tuned the Runge-Kutta coefficients to provide the largest possible time step and the best error damping characteristics. To further enhance the convergence of steady-state solutions, three standard acceleration methods are also used:

— Local time-stepping

— Implicit residual smoothing

— Multiple-Grid and Multigrid Acceleration

The time-step for a given computational cell is proportional to its size. Consequently, the convergence of a scheme may be severly limited by mesh geometry rather than the actual physics of the problem if very small mesh spacings are present, since the smallest time step is the limiting factor in global time-stepping (required for time-accurate problems). For steady-state solutions, since time-accuracy is not required, the solution in each cell may be marched in time by its local time-step, greatly increasing convergence rates.

Implicit residual smoothing is a method introduced by Jameson[6] for the Euler equations in which the stability limit (CFL number) may be increased to a large degree by replacing the residual at each point by a implicit average based on the residuals in the rest of the domain. This smoothing, used in conjunction with Jameson's hybrid Runge-Kutta schemes can increase the maximum allowable CFL number by three times

over that given by the stability limit.[6]

Multigrid is an acceleration tool first developed by Achi Brandt[10,11] to increase convergence rates for elliptic-type problems. In elliptic problems, high frequency errors are eliminated quickly by the relaxation scheme. The limiting factor on convergence is the low frequency errors that remain. In order to more rapidly eliminate these low frequency errors, Brandt utilized a sequence of meshes made up of successively larger cells. Relaxation sweeps are made on the initial, finest mesh, effectively damping the high frequency errors (wavelengths on the order of the mesh spacings). Then, the solution is transferred by an appropriate interpolation scheme, called a transfer operator, to the next coarser mesh, where the relaxation sweeps are again performed. Now, since the cells are larger, the relaxation scheme removes error frequencies that are lower (longer wavelengths) with respect to the fine mesh solution. When the solution is transferred back to the fine mesh, a large amount of the low frequency errors have been eliminated by the relaxation on the coarse mesh levels. On the coarse levels, forcing terms (source terms added to the coarse-mesh equations) are utilized to prevent the coarse-mesh accuracy from affecting the level of accuracy of the final fine-mesh solution. Brandt's Multigrid scheme is labelled the Full Approximation Storage (FAS) scheme due to the fact that the full solution from the finest mesh is transferred to the coarser mesh levels.

The first application of this type of acceleration scheme for the hyperbolic Euler equations was introduced by Ni.[12] His Multiple-Grid acceleration scheme differs from Brandt's FAS Multigrid scheme in that only changes in the fine-mesh solution,

versus the full solution, are transferred to the coarse mesh levels.

Jameson[6] was the first to apply Brandt's FAS Multigrid scheme to the Euler equations, and it is this scheme upon which much of the present work is based. The success of Multigrid acceleration for the Euler equations is based on the premise that wave propagation, in addition to error damping, determines convergence rates. As the problem is solved on increasingly coarser mesh levels, larger time-steps may be used, thus propagating the errors out of the domain at a faster rate, and speeding convergence. Used in conjunction with Jameson's hybrid Runge-Kutta schemes, high frequency error damping, characteristic of these schemes, is also used. The time-stepping schemes are tailored to provide good high frequency damping characteristics, which, when applied on the coarse mesh levels, help to eradicate the low frequency errors.

Multigrid Acceleration has proven to be a robust and reliable tool for the Euler equations,[2,3,6,12–16] with recent advances also being made for the Navier-Stokes equations.[13–20] However, improvements are still required, especially for the Navier-Stokes equations. The addition of the shear-stress terms and the high mesh stretching which accompany these problems tend to decrease the efficiency of Multigrid. The theory behind Multigrid for the Euler and Navier-Stokes equations has not reached as high a level of development as it has for elliptic problems. It essentially lacks a general, theoretical approach, and many times tends to be problem dependent.

An area which to date has not been explored to any large degree is the effect of the coarse-mesh boundary conditions on convergence. Enforcing the boundary conditions on the coarse-mesh levels with coarse-mesh accuracy will affect the fine-mesh solution

due to the higher truncation error on the coarse levels. For interior points, forcing terms are added to the equations for this very reason, and it is the present hypothesis that similar terms are required for the boundary conditions.

It is therefore the objective of this work to obtain a correct, consistent formulation of the coarse-mesh boundary conditions and to determine the effect of these boundary conditions on the convergence of both inviscid and viscous problems. As an initial step in obtaining these conditions, a new, general approach to formulating the Multigrid equations is given. The equations on the coarse mesh levels are viewed as a filtered sub-set of the fine mesh equations, since certain information is, in effect, filtered out on the coarse levels due to the lack of mesh resolution. In this context, a filtering operator is first defined. The coarse mesh equations in partial differential form are then derived by filtering the original partial differential equations one or more times. The specification of a discrete filter then gives the procedure for transferring the fine-mesh solution to the coarse mesh. The coarse-mesh equations in discrete form are constructed through a finite-volume approximation of the filtered coarse-mesh equations. Although this method may be used to obtain any number of coarse-mesh discretizations based on how the filter is defined, one choice reduces the present formulation to the standard Jameson Multigrid scheme. Further, the present analysis leads to the correct formulation of boundary conditions on the coarse levels, without the coarse-mesh truncation error affecting the fine-mesh solution.

The work presented in this thesis is divided into the following chapters. In Chapter 2, the governing equations are presented. The 2-dimensional Euler, and full, laminar,

Navier-Stokes equations are described along with their non-dimensionalization with respect to freestream conditions. The equations are then given in a general, non-orthogonal coordinate system. Chapter 3 gives the important aspects of the numerical method. Details of the finite-volume cell-centered spatial discretization, 4-stage hybrid Runge-Kutta time-stepping scheme, and blended 2nd and 4th difference artificial dissipation model are described. In Chapter 4, the boundary conditions for inviscid and viscous flow are given. Both a characteristic variable formulation and a Riemann Invariant formulation are given for the far-field boundaries. Chapter 5 presents the methods that are used to accelerate convergence. Jameson's Multigrid scheme is presented, and implicit residual smoothing for 2-dimensional problems is described. The general Multigrid formulation followed by the derivation of the boundary transfer operator and the coarse-mesh boundary conditions is given in Chapter 6. In Chapter 7, results are presented and discussed for inviscid channel flow over a bump for subsonic, transonic, and supersonic speeds, flow over a viscous, flat plate, viscous subsonic channel flow over a bump, and inviscid and viscous flow over a VKI gas turbine rotor blade. Chapter 8 summarizes the work that was performed and presents the conclusions that were drawn.

CHAPTER 2 - GOVERNING EQUATIONS

2.1 Navier Stokes Equations

The two-dimensional, unsteady, compressible Navier Stokes equations may be written in conservative form as follows:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} \tag{2.1}$$

where :

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix} \quad F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{Bmatrix} \quad G = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{Bmatrix} \tag{2.2}$$

$$R = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{Bmatrix} \qquad S = \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{Bmatrix}$$

and where, based on Stokes Hypothesis:

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \qquad \tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \tag{2.3}$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \qquad q_x = -\kappa\frac{\partial T}{\partial x} \qquad q_y = -\kappa\frac{\partial T}{\partial y}$$

with density, ρ , cartesian velocity components, u , v , total energy per unit volume, E , pressure, p , temperature, T , viscosity coefficient, μ , and the coefficient of

thermal conductivity, κ. Pressure is defined by the equation of state for an ideal gas :

$$p = \rho RT = (\gamma - 1)\left[E - \frac{1}{2}\rho(u^2 + v^2)\right] \tag{2.4}$$

where γ is the ratio of specific heats. Assuming laminar flow, the equations are closed by using Sutherland's Viscosity Law relating the viscosity to the local temperature:[21]

$$\mu = \frac{c_1 T^{\frac{3}{2}}}{T + c_2} \tag{2.5}$$

where for air and moderate temperatures:

$$c_1 = 1.458\text{x}10^{-6}\ \text{kg/(m sec °K}^{1/2}\text{)} \qquad c_2 = 110.4\ \text{°K}$$

The coefficient of thermal conductivity is expressed in terms of the local viscosity and the Prandtl Number P_r :

$$\kappa = \frac{C_p \mu}{P_r} \tag{2.6}$$

where C_p is the specific heat at constant pressure.

In this work, the governing equations are non-dimensionalized with respect to a reference length, L, and freestream conditions, ρ_∞ , V_∞ , T_∞ , and μ_∞ . :

$$x^* = \frac{x}{L} \qquad y^* = \frac{y}{L} \qquad u^* = \frac{u}{V_\infty} \qquad v^* = \frac{v}{V_\infty} \qquad t^* = \frac{t\,V_\infty}{L}$$

$$\rho^* = \frac{\rho}{\rho_\infty} \qquad p^* = \frac{p}{\rho_\infty V_\infty^2} \qquad T^* = \frac{T}{T_\infty} \qquad E^* = \frac{E}{\rho_\infty V_\infty^2} \qquad \mu^* = \frac{\mu}{\mu_\infty}$$

Where * denotes non-dimensional quantities. Under this non-dimensionalization, the Euler equations retain their original form and a constant, $1/\mathrm{Re}_L$, appears before the viscous terms in the Navier-Stokes equations.

$$\frac{\partial U^*}{\partial t^*} + \frac{\partial F^*}{\partial x^*} + \frac{\partial G^*}{\partial y^*} = \frac{1}{\mathrm{Re}_L}\left[\frac{\partial R^*}{\partial x^*} + \frac{\partial S^*}{\partial y^*}\right] \tag{2.7}$$

where Re_L is the reference Reynolds number defined as:

$$\mathrm{Re}_L = \frac{\rho_\infty u_\infty L}{\mu_\infty} \tag{2.8}$$

The non-dimensional temperature is derived from the equation of state (Eq. 2.4) :

$$T^* = \frac{\gamma M_\infty^2 p^*}{\rho^*} \tag{2.9}$$

And the non-dimensional viscosity and coefficient of thermal conductivity may be derived as:

$$\mu^* = T^{*\frac{3}{2}}\left[\frac{1 + c_2/T_\infty}{T + c_2/T_\infty}\right] \qquad \kappa^* = \frac{\mu^*}{P_r\,(\gamma - 1)\,M_\infty^2} \tag{2.10}$$

where T_∞ is a free-stream reference temperature. The equations for the non-dimensional pressure and other thermodynamic variables remain the same as their dimensional counterparts. For convenience, the * is dropped from the notation at this point, and all variables can be assumed to be non-dimensional.

In order to perform computations over general mesh topologies, the above cartesian (x,y) system of equations can be transformed to a non-orthogonal system (ξ , η),

where:

$$\xi = \xi(x,y) \qquad \eta = \eta(x,y)$$

Equation (2.1) becomes:

$$\frac{\partial}{\partial t}\left[\frac{U}{J}\right] + \left[\frac{\partial}{\partial \xi}\left[\frac{\xi_x F + \xi_y G}{J}\right] + \frac{\partial}{\partial \eta}\left[\frac{\eta_x F + \eta_y G}{J}\right]\right] = \tag{2.11}$$

$$= \frac{1}{Re_L}\left[\frac{\partial}{\partial \xi}\left[\frac{\xi_x R + \xi_y S}{J}\right] + \frac{\partial}{\partial \eta}\left[\frac{\eta_x R + \eta_y S}{J}\right]\right]$$

where J is the Jacobian of the transformation and is given by:

$$\frac{1}{J} = x_\xi\, y_\eta - x_\eta\, y_\xi \tag{2.12}$$

Given the metric relationships:

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ y_\xi & y_\eta \end{bmatrix}^{-1} \tag{2.13}$$

the Navier Stokes equations in general coordinates become:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = \frac{1}{Re_L}\left[\frac{\partial \hat{R}}{\partial \xi} + \frac{\partial \hat{S}}{\partial \eta}\right] \tag{2.14}$$

Where:

$$\begin{aligned} \hat{U} &= U/J \\ \hat{F} &= (\xi_x F + \xi_y G)/J = y_\eta F - x_\eta G \\ \hat{G} &= (\eta_x F + \eta_y G)/J = x_\xi G - y_\xi F \\ \hat{R} &= (\xi_x R + \xi_y S)/J = y_\eta R - x_\eta S \\ \hat{S} &= (\eta_x R + \eta_y S)/J = x_\xi S - y_\xi R \end{aligned} \tag{2.15}$$

CHAPTER 3 - NUMERICAL METHOD

3.1 Finite Volume Formulation

A discrete form of Eq. (2.14) is found using a control volume formulation in which the solution values are stored at the cell centers. First, Eq. (2.14) is put into integral form for the computational control volume, $\hat{V}$:

$$\frac{\partial}{\partial t}\iint_{\hat{v}} \hat{U}\, d\hat{V} + \iint_{\hat{v}} (\hat{F}_\xi + \hat{G}_\eta)\, d\hat{V} = \iint_{\hat{v}} (\hat{R}_\xi + \hat{S}_\eta)\, d\hat{V} \tag{3.1}$$

Then, integration over the volume, $\Delta\xi\Delta\eta$, gives :

$$\begin{aligned} \frac{\partial}{\partial t}\iint_{\hat{v}} \hat{U}\, d\hat{V} + \int [\, \hat{F}(\xi + \Delta\xi) - \hat{F}(\xi)\,]\, d\eta + \int [\, \hat{G}(\eta + \Delta\eta) - \hat{G}(\eta)\,]\, d\xi = \\ = \int [\, \hat{R}(\xi + \Delta\xi) - \hat{R}(\xi)\,]\, d\eta + \int [\, \hat{S}(\eta + \Delta\eta) - \hat{S}(\eta)\,]\, d\xi \end{aligned} \tag{3.2}$$

Applying the Mean Value Theorem to the control volume in Figure 3.1, $\hat{F}$ and $\hat{G}$ for each side of the control volume may be given by their values at each particular cell face center. Eq. (3.2) becomes:

$$\begin{aligned} \frac{\partial}{\partial t}\iint_{\hat{v}} \hat{U}\, d\hat{V} + [\, \hat{F}_{i+\frac{1}{2},j} - \hat{F}_{i-\frac{1}{2},j}\,]\, \Delta\eta + [\, \hat{G}_{i,j+\frac{1}{2}} - \hat{G}_{i,j-\frac{1}{2}}\,]\, \Delta\xi = \\ = [\, \hat{R}_{i+\frac{1}{2},j} - \hat{R}_{i-\frac{1}{2},j}\,]\, \Delta\eta + [\, \hat{S}_{i,j+\frac{1}{2}} - \hat{S}_{i,j-\frac{1}{2}}\,]\, \Delta\xi \end{aligned} \tag{3.3}$$

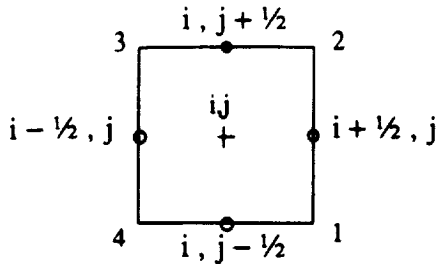


Figure 3.1. Cell Centered Control Volume Notation

Integrating the first term in Eq. (3.3) produces an ordinary derivative in time for the solution U at each cell, (i,j):

$$\frac{\partial}{\partial t}\int\int_{\hat{v}} \hat{U}d\hat{V} = \frac{\partial}{\partial t}\int\int U(J^{-1}\,d\xi d\eta) = \left[\Delta V \frac{dU}{dt}\right]_{i,j} \tag{3.4}$$

where ΔV is the cell area, which is defined in this work by the cross-product rule as :

$$\Delta V = -1/2\,[\,(x_2 - x_4)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_4)\,] \tag{3.5}$$

The Jacobian of the general coordinate transformation describes the ratio of the cell areas between the (ξ, η) coordinate system and the (x, y) coordinate system. Since the cell area in the transformed computational domain is equal to 1 $(\Delta\xi = \Delta\eta = 1)$, ΔV can also be defined as the inverse of the Jacobian :

$$\Delta V = \frac{1}{J} = x_\xi\, y_\eta - y_\xi\, x_\eta \tag{3.6}$$

The second term in Eq. (3.3) may be expressed in F and G as:

$$\begin{aligned}[\,\hat{F}_{i+\frac{1}{2},j} - \hat{F}_{i-\frac{1}{2},j}\,]\,\Delta\eta &= [(\,y_\eta F - x_\eta G\,)_{i+\frac{1}{2},j} - (\,y_\eta F - x_\eta G\,)_{i-\frac{1}{2},j}]\,\Delta\eta = \\ &= (\,F\Delta y - G\Delta x\,)_{i+\frac{1}{2},j} - (\,F\Delta y - G\Delta x\,)_{i-\frac{1}{2},j}\end{aligned} \tag{3.7}$$

Treating the remaining terms of Eq. (3.3) in the same manner produces a spatially

discrete system of ordinary differential equations for the cell shown in Figure 3.1 :

$$\left[\frac{dU}{dt}\right]_{i,j} = \frac{-1}{\Delta V_{i,j}} [\,(F\Delta y - G\Delta x)_{i+\frac{1}{2},j} - (F\Delta y - G\Delta x)_{i-\frac{1}{2},j} + \quad (3.8)$$
$$+ (G\Delta x - F\Delta y)_{i,j+\frac{1}{2}} - (G\Delta x - F\Delta y)_{i,j-\frac{1}{2}} +$$
$$- (R\Delta y - S\Delta x)_{i+\frac{1}{2},j} + (R\Delta y - S\Delta x)_{i-\frac{1}{2},j} +$$
$$- (S\Delta x - R\Delta y)_{i,j+\frac{1}{2}} + (S\Delta x - R\Delta y)_{i,j-\frac{1}{2}}\,]$$

where:

$$\begin{aligned} \Delta x_{i+\frac{1}{2},j} &= x_2 - x_1 & \Delta y_{i+\frac{1}{2},j} &= y_2 - y_1 \\ \Delta x_{i,j+\frac{1}{2}} &= x_2 - x_3 & \Delta y_{i,j+\frac{1}{2}} &= y_2 - y_3 \\ \Delta x_{i-\frac{1}{2},j} &= x_3 - x_4 & \Delta y_{i-\frac{1}{2},j} &= y_3 - y_4 \\ \Delta x_{i,j-\frac{1}{2}} &= x_1 - x_4 & \Delta y_{i,j-\frac{1}{2}} &= y_1 - y_4 \end{aligned} \quad (3.9)$$

In the present work, the convective terms, $F_{i+\frac{1}{2},j}$ and $G_{i+\frac{1}{2},j}$, are obtained from the average of the conservative variables, U, existing in the two cells adjacent to the face. For linear problems, this is equivalent to averaging F and G. However, since F and G are non-linear functions of U, Turkel[7] claims that averaging U instead helps to couple the even and odd points throughout the domain. To save on the number of computations that must be performed at each point, pressure is stored at the cell centers and averaged to the face.

$$F_{i+\frac{1}{2},j} = \begin{Bmatrix} (\rho u)_{i+\frac{1}{2},j} \\ \left[\frac{(\rho u)^2}{\rho}\right]_{i+\frac{1}{2},j} + p_{i+\frac{1}{2},j} \\ \left[\frac{(\rho u)(\rho v)}{\rho}\right]_{i+\frac{1}{2},j} \\ E_{i+\frac{1}{2},j} \end{Bmatrix} \quad G_{i+\frac{1}{2},j} = \begin{Bmatrix} (\rho v)_{i+\frac{1}{2},j} \\ \left[\frac{(\rho u)(\rho v)}{\rho}\right]_{i+\frac{1}{2},j} \\ \left[\frac{(\rho v)^2}{\rho}\right]_{i+\frac{1}{2},j} + p_{i+\frac{1}{2},j} \\ E_{i+\frac{1}{2},j} \end{Bmatrix} \quad (3.10)$$

where

$$(\rho)_{i+\frac{1}{2},j} = \frac{1}{2}\left[(\rho)_{i+1,j} + (\rho)_{i,j}\right] \tag{3.11}$$

$$(\rho u)_{i+\frac{1}{2},j} = \frac{1}{2}\left[(\rho u)_{i+1,j} + (\rho u)_{i,j}\right]$$

$$(\rho v)_{i+\frac{1}{2},j} = \frac{1}{2}\left[(\rho v)_{i+1,j} + (\rho v)_{i,j}\right]$$

$$(E)_{i+\frac{1}{2},j} = \frac{1}{2}\left[(E)_{i+1,j} + (E)_{i,j}\right]$$

$$(p)_{i+\frac{1}{2},j} = \frac{1}{2}\left[(p)_{i+1,j} + (p)_{i,j}\right]$$

Quantities at the other faces are defined in the similar manner.

The evaluation of the viscous fluxes for each cell requires that the first differences of u, v, and T be defined at the center of each cell face. Spatial discretization of these derivatives for the point $(i+\frac{1}{2}, j)$ is performed by integrating over a surface bounded by the two adjacent cell centers, (i,j) and (i+1,j), and the endpoints of their dividing face,[17] shown in Figure 3.2.

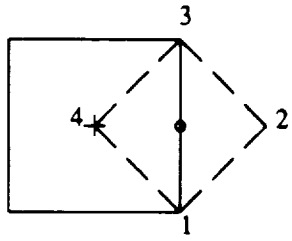


Figure 3.2. Control Volume for Evaluation of Shear Stress Derivatives at face (i+1/2,j)

For this control volume formulation, the following relationship may be derived for any variable, f :

$$\iint_V f_x \, dV = \int_{\partial V} f \, dy \qquad (3.12)$$
$$\iint_V f_y \, dV = -\int_{\partial V} f \, dx$$

Where V and ∂V are the surface area and boundary, respectively, of the control volume in Figure 3.2. The line integrals are approximated by trapezoidal integration, which gives a discrete representation for the first derivatives of f:

$$\frac{\partial f}{\partial x} = \frac{1}{2\,\Delta V} \sum_{k=1}^{4} (f_{k+1} + f_k) (y_{k+1} - y_k) \qquad (3.13)$$
$$\frac{\partial f}{\partial y} = -\frac{1}{2\,\Delta V} \sum_{k=1}^{4} (f_{k+1} + f_k) (x_{k+1} - x_k)$$

Where

$$f_5 = f_1 \quad x_5 = x_1 \quad y_5 = y_1$$

Using this method, the shear stress terms can be evaluated in their cartesian form without the need for a general coordinate transformation. When evaluating these derivatives at boundaries, such as the (i,j-1/2) face of the first interior cells, the center of the image cell must have specified coordinates for the above finite-volume formulation. To avoid difficulties of defining these coordinates, such as in areas of high curvature, a different approach is utilized at the boundaries. The viscous control volume of the (i,j-1/2) face of a cell along the boundary is collapsed to a triangular volume, shown in Figure 3.3, in which the first derivatives required at the boundary are obtained directly by integration over the truncated control volume. This results in a 1st-order accurate evaluation of the boundary shear stress terms, which is essentially equivalent to performing one-sided differencing there.

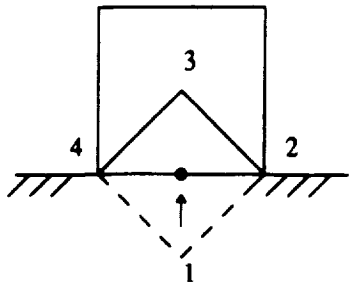


Figure 3.3. Control Volume For Boundary Shear Stress Evaluation

The finite-volume, cell centered formulation for the Navier-Stokes equations defined in this section is equivalent to a central difference scheme, and is 2nd order accurate for sufficiently smooth meshes.[6]

3.2 Time Stepping Scheme

The time-marching algorithm used in this research is a 4-stage Runge-Kutta scheme. Typically, multi-stage time-stepping schemes such as these are chosen for their high order of time accuracy. However, since the goal of this research is steady-state computations, time accuracy is not a requirement.

This 4-stage scheme is one of a class of hybrid multi-stage schemes all of which were developed by Jameson[6] specifically for their damping and stability characteristics. The coefficients of the scheme are tailored to give the maximum allowable time step and to provide optimum damping of the high frequency error modes. The attenuation of these modes is essential for the success of rapid steady-state convergence.

The semi-discrete system of ordinary differential equations in Eq. (3.3) may be

rewritten as

$$\frac{dU}{dt} = \frac{-1}{\Delta V}[\, C\,(U) + V\,(U) - D\,(U)] \tag{3.14}$$

Where C (U) is the convective flux,

$$\begin{aligned} C\,(U) = {} & (F\Delta y - G\Delta x)_{i+\frac{1}{2},\,j} - (F\Delta y - G\Delta x)_{i-\frac{1}{2},\,j} + \\ & + (G\Delta x - F\Delta y)_{i,\,j+\frac{1}{2}} - (G\Delta x - F\Delta y)_{i,\,j-\frac{1}{2}} \end{aligned} \tag{3.15}$$

V (U) is the viscous flux,

$$\begin{aligned} V\,(U) = {} & -(R\Delta y - S\Delta x)_{i+\frac{1}{2},\,j} + (R\Delta y - S\Delta x)_{i-\frac{1}{2},\,j} - \\ & - (S\Delta x - R\Delta y)_{i,\,j+\frac{1}{2}} + (S\Delta x - R\Delta y)_{i,\,j-\frac{1}{2}} \end{aligned} \tag{3.16}$$

and D (U) is the artificial dissipation which will be defined later in section 3.3.

The 4-stage Runge Kutta scheme used to complete the discretization of Eq. (3.11) is implemented in the following manner :

$$\begin{aligned} U^{(0)} &= U^{n} \\ U^{(1)} &= U^{(0)} - \frac{\alpha_1\,\Delta t}{\Delta V}\left[\, C\,(U^{(0)}) + V\,(U^{(0)}) - D\,(U^{(0)})\right] \\ U^{(2)} &= U^{(0)} - \frac{\alpha_2\,\Delta t}{\Delta V}\left[\, C\,(U^{(1)}) + V\,(U^{(0)}) - D\,(U^{(0)})\right] \\ U^{(3)} &= U^{(0)} - \frac{\alpha_3\,\Delta t}{\Delta V}\left[\, C\,(U^{(2)}) + V\,(U^{(0)}) - D\,(U^{(0)})\right] \\ U^{(4)} &= U^{(0)} - \frac{\alpha_4\,\Delta t}{\Delta V}\left[\, C\,(U^{(3)}) + V\,(U^{(0)}) - D\,(U^{(0)})\right] \\ U^{n+1} &= U^{(4)} \end{aligned} \tag{3.17}$$

The superscripts in parentheses refer to the particular stages of the scheme, and where α_i are the coefficients designed to give optimum stability and damping to this

scheme.[6]

$$\alpha_1 = 1/4 \quad \alpha_2 = 1/3 \quad \alpha_3 = 1/2 \quad \alpha_4 = 1$$

Note that the viscous fluxes and the artificial dissipation are evaluated once and held constant throughout the time step. Other Runge-Kutta schemes may possess better damping characteristics or allow larger time steps, but the single evaluation of the artificial dissipation in this 4-stage scheme produces substantial savings in computational time.

This 4-stage Runge-Kutta scheme has been used with good success in many steady-state computations,[6][8][13][14][7][22] and it's high frequency damping characteristics make it ideal when used in conjunction with multigrid acceleration.[6]

3.2.1 Stability Criteria

Turkel[7] derives the stability limit for the Euler Equations based on the maximum eigenvalues of the convective Jacobian matrices. Ignoring the effect of artificial dissipation on the equations, the maximum allowable time step for a general, multi-stage scheme is given there as :

$$\Delta t \leq \frac{\Delta V\,(CFL)}{|uy_\eta - vx_\eta| + |vx_\xi - uy_\xi| + (x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2 + 2|x_\xi x_\eta + y_\xi y_\eta|)^{1/2} a} \qquad (3.18)$$

An approximation of which is :

$$\Delta t \leq \frac{\Delta V\,(CFL)}{\lambda_\xi + \lambda_\eta} \qquad (3.19)$$

Where a is the speed of sound, ΔV is the cell area, and CFL is the Courant number for

stability based on a Fourier analysis of the 1-D model equation :

$$u_t + u_x + \mu\Delta x^3 u_{xxxx} = 0 \tag{3.20}$$

λ_ξ and λ_η are the maximum local wave-speeds in the ξ and η directions, respectively, and are defined as :

$$\lambda_\xi = \left(|uy_\eta - vx_\eta| + a\sqrt{y_\eta^2 + x_\eta^2} \right) \tag{3.21}$$
$$\lambda_\eta = \left(|vx_\xi - uy_\xi| + a\sqrt{y_\xi^2 + x_\xi^2} \right)$$

The maximum CFL number that may be used is dependent on the type of hybrid Runge-Kutta scheme that is used. Factors affecting this stability condition are the Runge-Kutta coefficients, α_i, the number of evaluations of the artificial dissipation, the number of Runge-Kutta stages, and the amount of smoothing, μ . The maximum CFL number for an m-stage scheme in which the coefficients are optimized for the largest time step is[6] :

$$\text{CFL} \leq m-1 \tag{3.22}$$

For the present 4-stage scheme with a single evaluation of the artificial dissipation, Eq. (3.17), with $\mu = 1/32$, gives the condition :

$$\text{CFL} \leq 2.6 \tag{3.23}$$

For the Navier Stokes equations, diffusive effects must be considered in the stability analysis as well as the convective effects. In Reference [18], the Navier-Stokes time step is given as :

$$\Delta t = CFL \left[\frac{\Delta t_c \, \Delta t_v}{\Delta t_c + \Delta t_v} \right] \tag{3.24}$$

where Δt_c is the convective time step given in Eq. (3.15), and Δt_v is the viscous time step :

$$\Delta t_v = K_v \frac{\Delta V}{\lambda_v} \tag{3.25}$$

K_v is a specified empirical constant that weights the importance of the viscous terms over the convective terms and is given as 0.25.[18] λ_v is the sum of the maximum spectral radii in the ξ and η directions of the viscous operator in the Navier-Stokes equations :

$$\lambda_v = \lambda_{v_\xi} + \lambda_{v_\eta} \tag{3.26}$$

Where

$$\lambda_{v_\xi} = \frac{1}{Re_L \Delta V} \left[\frac{\gamma \mu}{P_r \rho} \Delta l^2 + \frac{\mu}{3\rho} \Delta l \Delta m \right] \tag{3.27}$$

$$\lambda_{v_\eta} = \frac{1}{Re_L \Delta V} \left[\frac{\gamma \mu}{P_r \rho} \Delta m^2 + \frac{\mu}{3\rho} \Delta l \Delta m \right]$$

and where Δl and Δm are the lengths of the cell in the ξ and η directions, respectively:

$$\Delta l = \sqrt{x_\xi^2 + y_\xi^2} \tag{3.28}$$

$$\Delta m = \sqrt{x_\eta^2 + y_\eta^2}$$

For the viscous cases considered in the present work, the maximum CFL number used with the above viscous time step criteria was approximately the same as those for the inviscid cases.

3.3 Artificial Dissipation

Central difference schemes require the addition of artificial dissipation to damp the high frequency error modes that occur as a result of the odd-even point decoupling. Artificial dissipation is also required for shock capturing and to damp oscillations in other high pressure gradient regions, such as stagnation points. Even though the viscous terms in the Navier-Stokes equations provide physical dissipation and would be capable of resolving the structure of a shock, mesh spacing in the region would need to be on the order of the molecular mean-free-path. This is not practical from a computational standpoint, and therefore artificial dissipation is still required to capture shocks. Also, artificial dissipation is required to damp instabilities that may occur in regions dominated by the convective terms.

In the present work, the dissipation model is based on that introduced by Jameson, Schmidt, and Turkel.[5] It employs modifications by Swanson and Turkel,[22] and Martinelli,[23] to improve accuracy and to increase convergence rates for viscous solutions. It is a blended 2nd and 4th difference adaptive dissipation scheme that provides a base level of 3rd order dissipation throughout smooth regions of the flow, and decreases to 1st order in the vicinity of shocks and other high pressure gradient regions.

The conservative form of the equations is maintained by evaluating the dissipation at each cell center by summing the dissipative fluxes in each coordinate direction. The basic form of the dissipation, $D_{i,j}$, for each cell is:

$$D_{i,j} = \left(d_\xi^2 - d_\xi^4 + d_\eta^2 - d_\eta^4 \right) U_{i,j} \tag{3.29}$$

Where $U_{i,j}$ is the vector containing the conservative variables, and d^2 and d^4 are the 2nd and 4th adaptive difference operators. For cell i,j, shown in Figure 3.1, these are defined as :

$$\begin{aligned}
d_\xi^2 U &= \nabla_\xi \left\{ \left(\bar{\lambda}_{i+\frac{1}{2},j} \; \varepsilon^{(2)}_{i+\frac{1}{2},j} \right) \Delta_\xi U_{i,j} \right\} \\
d_\xi^4 U &= \nabla_\xi \left\{ \left(\bar{\lambda}_{i+\frac{1}{2},j} \; \varepsilon^{(4)}_{i+\frac{1}{2},j} \right) \Delta_\xi \nabla_\xi \Delta_\xi U_{i,j} \right\} \\
d_\eta^2 U &= \nabla_\eta \left\{ \left(\bar{\lambda}_{i,j+\frac{1}{2}} \; \varepsilon^{(2)}_{i,j+\frac{1}{2}} \right) \Delta_\eta U_{i,j} \right\} \\
d_\eta^4 U &= \nabla_\eta \left\{ \left(\bar{\lambda}_{i,j+\frac{1}{2}} \; \varepsilon^{(4)}_{i,j+\frac{1}{2}} \right) \Delta_\eta \nabla_\eta \Delta_\eta U_{i,j} \right\}
\end{aligned} \tag{3.30}$$

Where Δ and ∇ are the standard forward and backward finite differences, respectively.

$$\begin{aligned}
\Delta_\xi U &= U_{i+1,j} - U_{i,j} \qquad & \nabla_\xi U &= U_{i,j} - U_{i-1,j} \\
\Delta_\eta U &= U_{i,j+1} - U_{i,j} \qquad & \nabla_\eta U &= U_{i,j} - U_{i,j-1}
\end{aligned} \tag{3.31}$$

$\bar{\lambda}_{i+\frac{1}{2},j}$ and $\bar{\lambda}_{i,j+\frac{1}{2}}$ are local variable scaling factors averaged to the cell face

$$\bar{\lambda}_{i+\frac{1}{2},j} = \frac{1}{2}\left(\bar{\lambda}_{i+1,j} + \bar{\lambda}_{i,j}\right) \qquad (3.32)$$
$$\bar{\lambda}_{i,j+\frac{1}{2}} = \frac{1}{2}\left(\bar{\lambda}_{i,j+1} + \bar{\lambda}_{i,j}\right)$$

In the original model,

$$\bar{\lambda}_{i,j} = \frac{\Delta V}{\Delta t^*} \qquad (3.33)$$

where Δt^* is the inviscid time step due to a unit CFL number, or,

$$\bar{\lambda}_{i,j} = (\lambda_\xi)_{i,j} + (\lambda_\eta)_{i,j} \qquad (3.34)$$

where λ_ξ and λ_η are the maximum local wave-speeds in the ξ and η directions defined in Eq. (3.21). The adaptive coefficients, $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ control the blending of the 2nd and 4th differences in the dissipative operator :

$$\varepsilon^{(2)}_{i+\frac{1}{2},j} = K^{(2)} \max\left(\nu_{i-1,j}, \nu_{i,j}, \nu_{i+1,j}\right)_\xi \qquad (3.35)$$
$$\varepsilon^{(4)}_{i+\frac{1}{2},j} = \max\left\{0, \left[K^{(4)} - \varepsilon^{(2)}_{i+\frac{1}{2},j}\right]\right\}$$

$$\varepsilon^{(2)}_{i,j+\frac{1}{2}} = K^{(2)} \max\left(\nu_{i,j-1}, \nu_{i,j}, \nu_{i,j+1}\right)_\eta$$
$$\varepsilon^{(4)}_{i,j+\frac{1}{2}} = \max\left\{0, \left[K^{(4)} - \varepsilon^{(2)}_{i,j+\frac{1}{2}}\right]\right\}$$

Where ν is the norm of the centered 2nd difference of pressure used to locate large pressure gradients and turn on the second difference dissipation.

$$(\nu_{i,j})_\xi = \left| \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{P_{i+1,j} + 2P_{i,j} + P_{i-1,j}} \right| \qquad (3.36)$$

$$(\nu_{i,j})_\eta = \left| \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{P_{i,j+1} + 2P_{i,j} + P_{i,j-1}} \right|$$

When the second differences are strong, the fourth differences are turned off by $\varepsilon^{(4)}$ to sharply resolve shocks. $K^{(2)}$ and $K^{(4)}$ are user specified constants which typically hold values of:

$$K^{(2)} = \frac{1}{4} \text{ to } \frac{1}{2}$$

$$K^{(4)} = \frac{1}{256} \text{ to } \frac{1}{50}$$

The values that are used are based on the best convergence rate that may be obtained (highest possible $K^{(4)}$) while still maintaining accurate solutions, and on the ability to cleanly capture shocks without too much smearing ($K^{(2)}$).

The difference operators in Eq. (3.19) are applied in two steps. A sweep is made through the domain in the ξ direction, taking centered first differences of U at each cell face for d^2, and 3rd differences at each face for d^4. Then another sweep is made, taking centered differences of the 1st and 3rd differences, yielding the desired 2nd and 4th differences, respectively, at cell centers. The same procedure is repeated in the η direction.

3.3.1 Modifications

Based on an analysis of Eq. (3.25), Swanson and Turkel[22] determined that the 4th difference operator produces dispersion as well as the required dissipation. If the 4th

differences are applied as a sequence of two 2nd differences instead of a 1st and a 3rd difference, the operator d^4 will produce only dissipative terms.

$$D_\xi^4 U = \left[\nabla_\xi \Delta_\xi\right] \left\{ \left[\bar{\lambda}_{i,j}\ \varepsilon^{(4)}_{i,j}\right] \nabla_\xi \Delta_\xi U_{i,j} \right\} \tag{3.37}$$

$$D_\eta^4 U = \left[\nabla_\eta \Delta_\eta\right] \left\{ \left[\bar{\lambda}_{i,j}\ \varepsilon^{(4)}_{i,j}\right] \nabla_\eta \Delta_\eta U_{i,j} \right\}$$

Note that $\varepsilon^{(4)}$ and $\bar{\lambda}$ are now evaluated at the cell centers, not at the faces.

The second area of modification is in the evaluation of the variable scaling factor, $\bar{\lambda}$. In viscous calculations, severe mesh stretching can result from the need to resolve the boundary layer. Cell aspect ratios may vary from $\Delta l/\Delta m \gg 1$ near a solid boundary, to 1 in the far-field. This large variation slows convergence, and diminishes the accuracy of steady state solutions. Also, these problems increase in magnitude for multigrid calculations due to the large difference of the high-frequency modes in the two coordinate directions.[22]

To overcome these difficulties, a new $\bar{\lambda}$ based on cell aspect ratio was introduced by Martinelli[23] It is essentially a combination of the scaling factor in the original model with an anisotropic scaling factor suggested by some researchers.[13][24] In the ξ direction,

$$(\bar{\lambda}_{i,j})_\xi = \lambda_\xi \left[1 + \frac{\lambda_\eta}{\lambda_\xi}^{\alpha} \right] \tag{3.38}$$

And in the η direction,

$$(\bar{\lambda}_{i,j})_\eta = \lambda_\eta \left[1 + \frac{\lambda_\xi}{\lambda_\eta}^{\alpha} \right] \tag{3.39}$$

Where

$$0 \le \alpha \le 1 \tag{3.40}$$

If α is 1, then $\bar{\lambda}$ reduces to the scaling in the original dissipation model given in Eq. (3.30). If α is 0, $\bar{\lambda}$ reduces to scaling in one direction only at each given face. This scaling is not recommended as it may decrease convergence rates and create problems for multigrid applications[22]. A value of α that has produced good results[22][18] is

$$\alpha = 2/3$$

For the cases in the present work, solution convergence did not seem to change significantly with changing α.

In the present cell-centered scheme, image cells are included around the physical domain to allow the same algorithm to be used for all interior cells. In order to determine the 4th difference dissipation terms in cells adjacent to boundaries as in Figure 3.4, 2nd differences must be given at the image cells.

$$(\nabla\Delta\nabla\Delta U)_2 = (\nabla\Delta U)_3 - 2(\nabla\Delta U)_2 + (\nabla\Delta U)_1 \tag{3.41}$$

Where ∇ and Δ are the difference operators given in Eq. (3.27), and where :

$$(\nabla\Delta U)_1 = U_2 - 2U_1 + U_0 \tag{3.42}$$

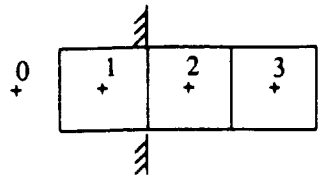


Figure 3.4. Cell Notation for Boundary Dissipative Flux

Since these differences can not actually be calculated (U_0 is not defined) they must be assigned in a manner that maintains conservation form and doesn't introduce instability at the boundaries.

Swanson and Turkel[22] provide a detailed analysis of various treatments of these terms near boundaries. For far-field boundaries, the standard approach is to set the 2nd differences in the image cells to zero.

$$(\nabla\Delta U)_1 = 0 \tag{3.43}$$

A typical treatment at solid boundaries is to set the surface dissipative flux to zero

$$(\nabla\Delta U)_1 = (\nabla\Delta U)_2 \tag{3.44}$$

and then either :

$$1)\ (\nabla\Delta U)_2 = 0 \tag{3.45}$$

or

$$2)\ (\nabla\Delta\nabla\Delta U)_3 = 0$$
$$(\nabla\Delta U)_3 = (\nabla\Delta U)_2 = 0$$

Option 2 results in zero dissipation normal to the boundary for the first two interior cells, 2 and 3.

In the present work, both methods were used with equal success, and without noticeable differences in the results.

CHAPTER 4 - BOUNDARY CONDITIONS

4.1 Inflow/Outflow Boundaries

For the channel flow cases, where the outflow conditions are essentially freestream, a characteristic formulation of the boundary conditions is used. For the turbine cascade case, in order to explicitly satisfy constant total temperature, total pressure, and flow angle, a Riemann Invariant formulation is used at the upstream boundary, while a characteristic formulation is retained at the outflow boundary.

4.1.1 Characteristic Formulation

This approach utilizes a transformed system of equations in which the boundary cells are updated by specified and extrapolated characteristic variables. Rewriting the Euler equations in a local coordinate frame that is normal and tangential to the boundary gives :

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial n} + \frac{\partial G}{\partial s} = 0 \tag{4.1}$$

Where n and s denote the normal and tangential directions. At these boundaries, it is assumed that variations of the flow conditions in the tangential direction are negligible, and so Eq. (4.1) can be written as :

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial n} = 0 \tag{4.2}$$

The above one-dimensional system of coupled differential equations can be linearized and transformed to an uncoupled linear system of partial differential equations:

$$\frac{\partial W}{\partial t} + \lambda \frac{\partial W}{\partial n} = 0 \tag{4.3}$$

where W is the vector of characteristic variables, and λ is the diagonal matrix of eigenvalues. Denoting q_n and q_s as the normal and tangential velocities at the boundary, where the normal is pointing into the solution domain, λ and W are defined as :

$$\lambda = \begin{bmatrix} q_n & 0 & 0 & 0 \\ 0 & q_n & 0 & 0 \\ 0 & 0 & q_n+a & 0 \\ 0 & 0 & 0 & q_n-a \end{bmatrix} \qquad W = \begin{bmatrix} \rho - p/\bar{a}^2 \\ q_s \\ (1/\sqrt{2})(q_n + p/\bar{\rho}\,\bar{a}) \\ (1/\sqrt{2})(-q_n + p/\bar{\rho}\,\bar{a}) \end{bmatrix} \tag{4.4}$$

The barred terms represent variables evaluated at the linearized state, which in this work are taken as the values from the previous Runge-Kutta stage.

Along the characteristics defined by slope $dt/dn = 1/\lambda_i$, W_i is constant and Eq. (4.3) reduces to an ordinary differential equation. If the slope of the characteristic, $1/\lambda_i$, is such that it originates from outside the computational domain, W_i must be specified. If the characteristic originates from within the computational domain, the value of W_i at the boundary for the new time level, $t + \Delta t$ is found by tracing the characteristic back to its intersection point with the previous time level, t. Evaluating W_i at this intersection point will give the new value of W_i since W_i is constant along

these characteristic lines defined by $dt/dn = 1/\lambda_i$. However, in steady-state computations, it is common practice to approximate the value of W_i which lies along the characteristic by the value contained in the first interior cell (Figure 4.1).

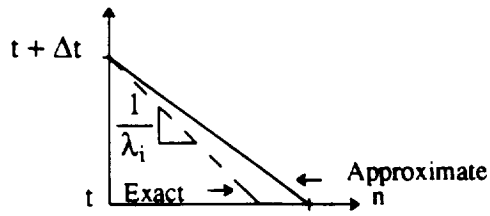


Figure 4.1. Approximation To W_i For The Characteristics Originating Within The Domain

The boundary conditions are enforced at the boundary, or cell face, instead of the image cell (Figure 4.2) to accommodate a consistent formulation of the boundary forcing terms that will be given later in Chapter 6. The conservation variable in the image cells are then updated by linear extrapolation from the interior as:

$$U_1 = 2U_b - U_2 \tag{4.5}$$

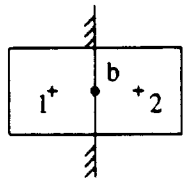


Figure 4.2. Boundary Cell Notation

For subsonic inflow, and an inward pointing normal, n, λ_1 , λ_2 , and λ_3 are

positive, meaning the characteristics originate from outside the domain (see Figure 4.3). Therefore, W_1 , W_2 , and W_3 must be specified, and W_4 must be extrapolated from the interior. The following equations result:

$$\rho_b - \frac{p_b}{\bar{a}^2} = \rho_\infty - \frac{p_\infty}{\bar{a}^2} \tag{4.6}$$

$$q_{s_b} = q_{s_\infty}$$

$$q_{n_b} + \frac{p_b}{\bar{\rho}\,\bar{a}} = q_{n_\infty} + \frac{p_\infty}{\bar{\rho}\,\bar{a}}$$

$$-q_{n_b} + \frac{p_b}{\bar{\rho}\,\bar{a}} = -q_{n_{ex}} + \frac{p_{ex}}{\bar{\rho}\,\bar{a}}$$

where the subscripts b, ∞, and ex refer to the boundary, freestream, and extrapolated values, respectively. Solving Eq. (4.6) for the primative variables at the boundary gives:

$$q_{s_b} = q_{s_\infty} \tag{4.7}$$

$$p_b = \frac{1}{2}[\, p_{ex} + p_\infty + \bar{\rho}\,\bar{a}\,(\, q_{n_\infty} - q_{n_{ex}}\,)]$$

$$\rho_b = \rho_\infty + (\, p_b - p_\infty\,)/\,\bar{a}^2$$

$$q_{n_b} = q_{n_\infty} + (\, p_\infty - p_b\,)/\,\bar{\rho}\,\bar{a}$$

For subsonic outflow, with an inward pointing normal, n, only λ_3 is positive (Figure 4.3), and so W_3 is the only characteristic variable that must be specified. Following the same procedure as above, the updated values are given by :

$$
\begin{aligned}
q_{s_b} &= q_{s_{ex}} \\
p_b &= \frac{1}{2}[\, p_{out} + p_{ex} + \bar{\rho}\,\bar{a}\,(\, q_{n_{out}} - q_{n_{ex}}\,)] \\
\rho_b &= \rho_{ex} + (\, p_b - p_{ex}\,)/\,\bar{a}^2 \\
q_{n_b} &= q_{n_{ex}} + (\, p_b - p_{ex}\,)/\,\bar{\rho}\,\bar{a}
\end{aligned}
\tag{4.8}
$$

where the subscript out refers to the exit conditions.

For the inviscid channel flow cases, p_{out} and $q_{n_{out}}$ are assumed to be at freestream conditions. For viscous problems, where the boundary layer (or viscous wake) intersects the outflow boundary, the characteristic relations are incorrect due to the non-isentropic conditions existing there. For these cases, the exit pressure is specified and the other variable are extrapolated.

For supersonic inflow, all four λ_i are positive (Figure 4.3), and therefore all four W_i must be specified, which reduces to specifying the primative variables:

$$
\begin{aligned}
q_{s_b} &= q_{s_\infty} \\
p_b &= p_\infty \\
\rho_b &= \rho_\infty \\
q_{n_b} &= q_{n_\infty}
\end{aligned}
\tag{4.9}
$$

For supersonic outflow (Figure 4.3), all λ_i are negative, and so all W_i must be extrapolated:

$$
\begin{aligned}
q_{s_b} &= q_{ex} \\
p_b &= p_{ex} \\
\rho_b &= \rho_{ex} \\
q_{n_b} &= q_{n_{ex}}
\end{aligned}
\tag{4.10}
$$

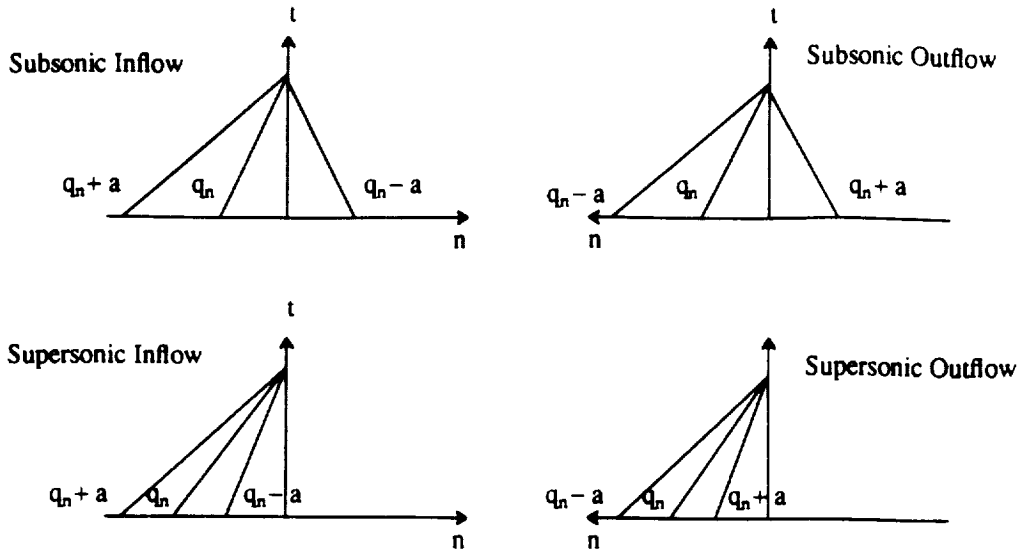


Figure 4.3. Characteristics At Far-Field Boundaries

4.1.2 Riemann Invariant Formulation

In this formulation, an upstream-running Riemann invariant is used in conjunction with specified total pressure, total temperature, and flow angle to solve for the updated inlet conditions.[25] The Riemann invariant, R^-, is based on total velocity, Q, and is approximated by extrapolation from the first interior cell :

$$R^- = Q_{ex} - \frac{2a_{ex}}{\gamma - 1} \tag{4.11}$$

Where a is the local speed of sound

$$a = \left[\frac{\gamma P}{\rho}\right]^{1/2} \tag{4.12}$$

Computing R^- in this fashion represents extrapolation along streamlines versus normal to the boundary. Based on isentropic relations and the specified total temperature at the

inlet, T_{o_∞} , the inlet speed of sound can also be given by :

$$a_b = \left[(\gamma - 1) C_p T_{o_\infty} - Q_1^2 \left(\frac{\gamma - 1}{2} \right) \right]^{1/2} \tag{4.13}$$

Where C_p is the non-dimensional specific heat

$$C_p = \frac{1}{M_\infty^2 (\gamma - 1)} \tag{4.14}$$

and Q_b is the unknown inlet total velocity. Using Eq. (4.12) and Eq. (4.14), a quadratic in Q_b results. Being an inflow boundary, the negative root is non-physical, and therefore Q_b is given by the positive root:

$$Q_b = \frac{(\gamma - 1) R^- + \left[4(\gamma + 1) C_p T_{o_\infty} - 2(\gamma - 1)(R^-)^2 \right]}{\gamma + 1} \tag{4.15}$$

Pressure and density at the inlet are then calculated using isentropic relations and the specified inlet total pressure :

$$p_b = p_{o_\infty} \left[1 + \frac{(\gamma - 1)}{2} \left(\frac{Q_b}{a_b} \right)^2 \right]^{-\gamma/(\gamma - 1)} \tag{4.16}$$

$$\rho_b = \rho_{o_\infty} \left(\frac{p_b}{p_{o_\infty}} \right)^{1/\gamma}$$

The cartesian velocity components are then given by Q_b and the specified inlet flow angle, β_1 :

$$u = Q_b \cos\beta_1$$
$$v = Q_b \sin\beta_1 \tag{4.17}$$

At the outflow, the characteristic formulation (section 4.1.1) is used with a specified exit pressure. The exit pressure in the turbine case is based on a given exit isentropic mach number $M_{out_{is}}$, assuming that $p_{o_{out}} = p_{o_\infty}$:

$$p_{out} = p_{o_\infty} \left[1 + \frac{(\gamma - 1)}{2} M_{out_{is}}^2 \right]^{-\gamma/(\gamma-1)} \tag{4.18}$$

4.2 Solid Wall Boundaries

For a solid wall in viscous flow, the physical boundary conditions required to close the system of governing equations are zero velocity and either specified surface temperature or heat flux:

$$Q = 0$$
$$T = T_{wall} \quad \text{or} \quad \frac{\partial T}{\partial n} = \left(\frac{\partial T}{\partial n} \right)_{wall} \tag{4.19}$$

where Q is the total velocity and n is the normal to the surface. The physical boundary conditions required for inviscid flow is flow tangency at the surface (zero flux through the surface):

$$Q \cdot \vec{n} = 0 \tag{4.20}$$

For a numerical scheme, however, other boundary conditions are required to close the discrete equations, and these must be specified in a manner consistent with the physics that are occurring at the boundaries. The implementation of these numerical boundary conditions at a solid wall is given below for inviscid and viscous surfaces.

For a cell-centered, finite volume scheme, the only contribution to the momentum equations at a solid wall is the pressure. The pressure may be obtained by extrapolation from the interior based on the pressure derivative normal to the boundary[5]:

$$\frac{\partial p}{\partial \eta} = \frac{(x_\xi x_\eta + y_\xi y_\eta)\frac{\partial p}{\partial \xi} + \rho(y_\eta u - x_\eta v)(x_{\xi\xi} v - y_{\xi\xi} u)}{(x_\xi^2 + y_\xi^2)} \tag{4.21}$$

giving

$$p_b = p_2 - \frac{\partial p}{\partial \eta} \tag{4.22}$$

Where the subscript b denotes the boundary point and the subscript 2 denotes the first interior cell as before (Figure 4.2). For inviscid flow, flow tangency is specified by setting

$$\begin{aligned} q_{n_b} &= 0 \\ q_{s_b} &= q_{s_2} \end{aligned} \tag{4.23}$$

where q_n and q_s are the normal and tangential velocity components. Density is obtained by isentropic relations based on the assumption that the entropy gradient normal to the boundary is zero:

$$\rho_b = \rho_2 \left[\frac{p_b}{p_2} \right]^\gamma \tag{4.24}$$

For the viscous cases, the normal pressure derivative at the boundary is assumed to be zero in the present work. A no-slip condition is specified for the velocity :

$$q_{n_b} = 0 \qquad (4.25)$$
$$q_{s_b} = 0$$

and a specified wall temperature, T_B, is used to define the density :

$$\rho_b = \frac{\gamma M_\infty^2 p_b}{T_b} \qquad (4.26)$$

For the flat plate problem, an adiabatic surface is specified:

$$\left[\frac{\partial T}{\partial n} \right]_b = 0 \rightarrow T_b = T_2 \qquad (4.27)$$

For the viscous bump and turbine problems, the temperature at the boundary is set equal to the freestream total temperature:

$$T_b = T_{o_\infty} \qquad (4.28)$$

CHAPTER 5 - CONVERGENCE ACCELERATION

5.1 Multigrid Method

Multigrid was developed by Brandt [10][11] to increase convergence rates for the solution of steady, elliptic-type problems, and was later applied to the Euler equations by Jameson.[6] Jameson used Multigrid directly with his hybrid Runge-Kutta schemes on all mesh levels, solving the same fine-mesh equations with the addition of forcing terms to maintain fine-mesh accuracy. In this chapter, a brief discussion of Brandt's FAS scheme for steady problems will be given. Then his notation will be used to describe Jameson's procedure for applying Multigrid to the time-dependent Euler equations.

A general, non-linear steady equation is given in Brandt's notation as :

$$L_h U_h = F_h \tag{5.1}$$

where h denotes relative mesh spacing, L is the discrete spatial operator, and F is a forcing term which is usually zero on the finest mesh level. The corresponding coarse-grid equation is :

$$L_{2h} U_{2h} = F_{2h} \tag{5.2}$$

where the forcing term is based on the transferred fine-mesh residual ($F_h - L_h U_h$):

$$F_{2h} = I_h^{2h} (F_h - L_h U_h) + L_{2h} (I_h^{2h} U_h) \tag{5.3}$$

I_h^{2h} is the fine-to-coarse transfer operator (interpolation scheme). Changes in the solution are transferred back to the fine mesh to update the solution :

$$U_{h_{new}} = U_{h_{old}} + I_{2h}^{h} \Delta U_{2h} \tag{5.4}$$

where

$$\Delta U_{2h} = U_{2h} - I_h^{2h} U_{h_{old}} \tag{5.5}$$

and where I_{2h}^{h} is the coarse-to-fine transfer operator. This transfer introduces high frequency errors into the solution on the fine grid, and therefore, it is important that the relaxation scheme (or time-stepping algorithm for unsteady equations) possess good high- frequency damping characteristics.[13]

For the unsteady Navier-Stokes equations using Runge-Kutta time-stepping, Jameson's Multigrid equations are of the form:

$$U_{2h}^{(k)} = U_{2h}^{(0)} - \alpha\Delta t(L_{2h} U_{2h}^{(k-1)} - F_{2h}) \tag{5.6}$$

where the coarse-mesh forcing term is :

$$F_{2h} = I_h^{2h} (F_h - L_h U_h^{(4)}) + L_{2h} (I_h^{2h} U_h^{(4)})$$

with $F_h = 0$. The quantities in parentheses refer to the stage of the Runge-Kutta scheme, and L represents the discrete spatial operator given in Eq. (3.11) :

$$L_{2h} U_{2h} = \frac{1}{\Delta V} \left[C(U_{2h}) + V(U_{2h}) - D(U_{2h}) \right] \tag{5.7}$$

The forcing term, formulated in the above fashion, is necessary in order that the fine-

mesh solution will not be affected by the coarse-mesh truncation error. If the fine mesh converges ($F_h - L_h U_h = 0$), the coarse mesh converges also. Looking at the 1st Runge-Kutta stage on the coarse mesh:

$$\begin{aligned} U_{2h}^{(1)} &= U_{2h}^{(0)} - \alpha\Delta t(L_{2h}U_{2h}^{(0)} - F_{2h}) \\ &= U_{2h}^{(0)} - \alpha\Delta t[L_{2h}U_{2h}^{(0)} - I_h^{2h} (F_h - L_h U_h^{(4)}) - L_{2h}(I_h^{2h} U_h^{(4)})] \\ &= U_{2h}^{(0)} - \alpha\Delta t[- I_h^{2h} (F_h - L_h U_h^{(4)})] \\ &= U_{2h}^{(0)} \end{aligned} \tag{5.8}$$

Or, in other words, $\partial U / \partial t = 0$ on the coarse mesh.

For Jameson's cell-centered scheme, the transfer operator I_h^{2h} is a volume weighted average over the 4 fine-mesh cells that make up a given coarse-mesh cell (Figure 5.1).

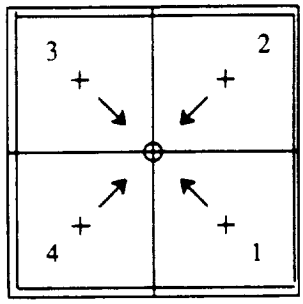


Figure 5.1. Fine-to-coarse Transfer

The transfer of the solution and the residual from the fine mesh are therefore given as:

Solution transfer:

$$I_h^{2h} U_h = \frac{\sum_{i=1}^{4} U_{h_i} \Delta V_{h_i}}{\sum_{i=1}^{4} \Delta V_{h_i}} \tag{5.9}$$

Residual transfer:

$$I_h^{2h}[F_h - L_h U_h] = \frac{\sum_{i=1}^{4} \Delta V_i \left[\frac{-1}{\Delta V_i}[C(U_h) + V(U_h) - D(U_h)] \right]}{\sum_{i=1}^{4} \Delta V_i} \tag{5.10}$$

$$= \frac{\sum_{i=1}^{4} Flux_i}{\sum_{i=1}^{4} \Delta V_i} = \sum_{i=1}^{4} Residual_i$$

This residual transfer approximates the integral equations on the coarse mesh cell ($2\Delta\xi\Delta\eta$) with fine mesh accuracy, and maintains their conservative property.

Bi-linear interpolation is used to transfer the coarse mesh solution changes back to the fine mesh (Figure 5.2) :

$$\Delta U_{i,j} = \frac{9}{16} \Delta U_1 + \frac{3}{16} [\Delta U_2 + \Delta U_4] + \frac{1}{16} \Delta U_3 \tag{5.11}$$

$$\Delta U_{i+1,j} = \frac{9}{16} \Delta U_2 + \frac{3}{16} [\Delta U_1 + \Delta U_3] + \frac{1}{16} \Delta U_4$$

$$\Delta U_{i,j+1} = \frac{9}{16} \Delta U_4 + \frac{3}{16} [\Delta U_1 + \Delta U_3] + \frac{1}{16} \Delta U_2$$

$$\Delta U_{i+1,j+1} = \frac{9}{16} \Delta U_3 + \frac{3}{16} [\Delta U_4 + \Delta U_2] + \frac{1}{16} \Delta U_1$$

Where the subscripts i ,j denote the indices for the fine-mesh cells and 1-4 denote the coarse-mesh cells, and where ΔU is the coarse-mesh solution change given in Eq. (5.5).

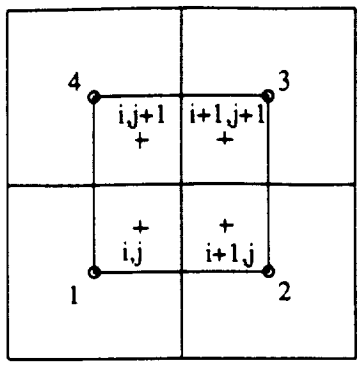


Figure 5.2. Coarse-to-fine Transfer

The cycling process employed in this work is referred to as a simple Saw-Tooth cycle (Figure 5.3). One Runge-Kutta integration is performed on each mesh level until the coarsest mesh is reached, and then the solution changes are transferred back to the finest mesh without any integration steps being performed on the way down. This may not be the optimum cycling strategy, but it has been found to be computationally efficient for a number of flow calculations.[6][13] [7] [26]

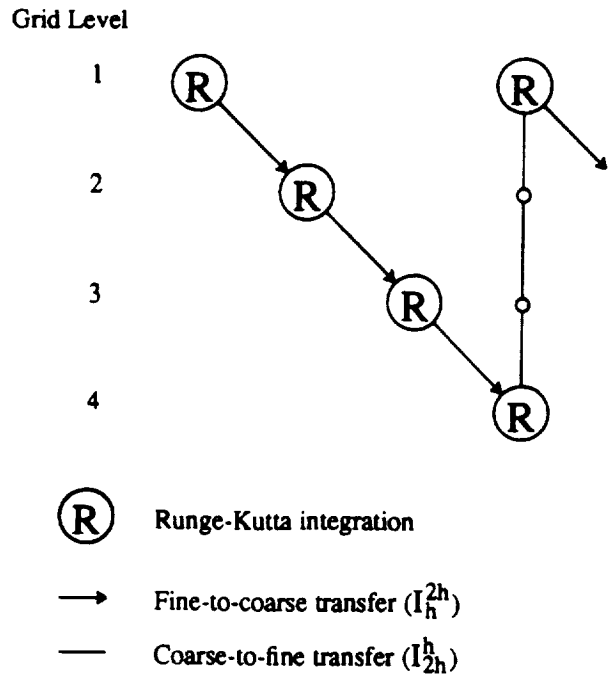


Figure 5.3. Saw-Tooth Multigrid Procedure

5.2 Implicit Residual Smoothing

The concept of implicit residual smoothing for Runge-Kutta schemes was introduced by Jameson[6] to permit the use of larger time steps and therefore increase steady-state convergence rates. The residuals at each point, $R_{i,j}$, are replaced by smoothed residuals, $\bar{R}_{i,j}$ by implicitly solving the equation:

$$\bar{R}_{i,j} = R_{i,j} + \varepsilon_\xi(\nabla_\xi\Delta_\xi)\bar{R}_{i,j} + \varepsilon_\eta(\nabla_\eta\Delta_\eta)\bar{R}_{i,j} \qquad (5.12)$$

where ∇ and Δ are the first forward and first backward differences, respectively, that were defined in Chapter 3, and ε_ξ and ε_η are smoothing coefficients in each coordinate direction. This could be thought of as solving the partial differential equation:

$$\frac{\partial R}{\partial t} = \frac{\varepsilon_\xi}{\Delta t}\left[\frac{\partial^2 R}{\partial \xi^2}\right] + \frac{\varepsilon_\eta}{\Delta t}\left[\frac{\partial^2 R}{\partial \eta^2}\right] \tag{5.13}$$

which is similar in form to the 2-D Heat Equation, and possesses the same properties. This equation (Eq. 5.13) therefore has the effect of smoothing the residual at each point based on all the residuals in the entire domain. The global influence on each residual is determined by ε, a "diffusivity" term. Discretized implicitly, Eq. (5.13) is the same as Eq. (5.12). Rearranging Eq. (5.12) gives:

$$(1 - \varepsilon_\xi \nabla_\xi \Delta_\xi - \varepsilon_\eta \nabla_\eta \Delta_\eta)\bar{R}_{i,j} = R_{i,j} \tag{5.14}$$

of which a factored approximation is :

$$(1 - \varepsilon_\xi \nabla_\xi \Delta_\xi)(1 - \varepsilon_\eta \nabla_\eta \Delta_\eta)\bar{R}_{i,j} = R_{i,j} \tag{5.15}$$

In the present work, Eq. (5.15) is applied as a combination of one-dimensional smoothing steps :

$$\begin{aligned} (1 - \varepsilon_\xi \nabla_\xi \Delta_\xi)R^*_{i,j} &= R_{i,j} \\ (1 - \varepsilon_\eta \nabla_\eta \Delta_\eta)\bar{R}_{i,j} &= R^*_{i,j} \end{aligned} \tag{5.16}$$

Where R^* are the intermediate smoothed residuals after the sweep in the ξ direction.

Neglecting the effect of artificial dissipation, Jameson[6] showed that stability can be maintained for any CFL number so long as ε satisfies the requirement :

$$\varepsilon \geq \frac{1}{4}\left[\frac{(CFL)^2}{(CFL^*)^2} - 1\right] \tag{5.17}$$

where CFL^* is the stability limit for the unsmoothed scheme. However, using CFL

numbers that are too large may severely decrease high frequency error damping and reduce convergence rates.[7] Also, with non-linear equations containing artificial dissipation, other factors may enter in and create instabilities. Jameson[6] suggests that optimum convergence rates are usually obtained for

$$CFL = 3(CFL^*) \tag{5.18}$$

For even-stage hybrid Runge-Kutta schemes, stability is maintained when the smoothing is applied during every even-numbered stage, while for an odd-stage scheme, it is applied during odd-numbered stages.

Locally varying ε_ξ and ε_η may be used at each point to optimize the smoothing for highly stretched meshes.[18][19] However, in this work, good results were obtained by using constant coefficient values of : $\varepsilon_\xi = \varepsilon_\eta = 2.0$

CHAPTER 6 - MULTIGRID BOUNDARY CONDITION FORMULATION

In Multigrid applications, a topic that has received little attention is the effect of the coarse-mesh boundary conditions and boundary transfer operators on the accuracy and convergence of the solution. In fact, the procedure for dealing with these issues is rarely mentioned, and if so, not in a very detailed fashion. Merely enforcing the fine-mesh boundary conditions on the coarse mesh levels will introduce coarse-mesh truncation error into the fine-mesh solution. Therefore, a consistent formulation is required for the boundary conditions just as it is for the interior Multigrid equations.

In this section a new way of viewing the Multigrid acceleration process will be introduced in order to derive a general approach to obtaining the coarse-mesh Multigrid equations. This, in turn, will lead to a consistent formulation of the coarse-mesh boundary conditions. Rather than formulating the Multigrid acceleration scheme directly from the discrete problem, the governing partial differential equations used in the Multigrid acceleration scheme will first be constructed. This is accomplished by introducing a filter operator which is applied one or more times to the original system of partial differential equations. The resulting series of filtered partial differential equations may then be discretized leading to a consistent series of coarse mesh equations to be solved in the Multigrid acceleration scheme. This approach is motivated by the observation that both the coarse mesh solutions and the coarse mesh

discrete equations solved in a Multigrid acceleration scheme are in fact filtered approximations to the corresponding fine mesh solution and discrete equations, although they are rarely viewed in this manner. The advantage of the present approach is that the filtering process is clearly defined through the choice of the filter operator. Further, since in the present formulation a general filter is initially assumed, a wide class of different multigrid discrete formulations are possible by simply changing the actual filter that is used. Finally, through a clear definition of the filtering process between mesh levels, the proper formulation of the corresponding coarse mesh boundary conditions follows directly.

In the discussion which follows the filtered form of the governing equations will be derived through the introduction of a general filter operator. It will then be shown that with one possible choice of this filter the corresponding discrete equations reduce to the FAS Multi-grid scheme proposed by Jameson. The remainder of this chapter will then center on the proper formulation of Multigrid boundary conditions, with special attention directed toward boundary conditions for Jameson's Multi-grid scheme.

6.1 Filtered Partial Differential Equations

The Euler and Navier-Stokes equations may be written in the following general form for a non-orthogonal coordinate system:

$$\frac{\partial}{\partial t}\left[\frac{U}{J}\right] + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = S \tag{6.1}$$

Where $\hat{F}$ and $\hat{G}$ represent the combined convective and viscous terms, and where S is a

source term, which is zero for the unfiltered equations. Consider a general spatial filter operator defined as follows:

$$L(f) = \bar{f}(\xi,\eta) = \frac{1}{\lambda(\xi,\eta)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \bar{f}(\bar{\xi},\bar{\eta})\, H(\bar{\xi},\bar{\eta})\, W(\bar{\xi}-\xi,\bar{\eta}-\eta)\, d\bar{\xi}d\bar{\eta} \tag{6.2}$$

where :

$$\begin{aligned} H(\xi,\eta) &= 1 \ \text{ inside the domain} \\ &= 0 \ \text{ outside the domain} \\ W(\xi,\eta) &\equiv \text{Weighting Function} \end{aligned} \tag{6.3}$$

and :

$$\lambda(\xi,\eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\bar{\xi},\bar{\eta})\, W(\bar{\xi}-\xi,\bar{\eta}-\eta)\, d\bar{\xi}d\bar{\eta} \tag{6.4}$$

Apply this filter to Eq. (6.1),

$$L\left[\frac{\partial}{\partial t}\left(\frac{U}{J}\right) + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta}\right] = L\left[S\right] \tag{6.5}$$

The first term may be integrated as:

$$\mathcal{L}\left[\frac{\partial}{\partial t}(J^{-1}U)\right] = \frac{1}{\lambda}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{\partial}{\partial t}(J^{-1}U)\,H(\bar{\xi},\bar{\eta})\,W(\bar{\xi}-\xi,\bar{\eta}-\bar{\eta}-\eta)\,d\bar{\xi}d\bar{\eta} \tag{6.6}$$

$$= \frac{1}{\lambda}\frac{\partial}{\partial t}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(J^{-1}U)\,H(\bar{\xi},\bar{\eta})\,W(\bar{\xi}-\xi,\bar{\eta}-\bar{\eta}-\eta)\,d\bar{\xi}d\bar{\eta}\right]$$

$$= \frac{1}{\lambda}\frac{\partial}{\partial t}(\lambda\overline{J^{-1}U})$$

$$= \frac{\partial}{\partial t}(\overline{J^{-1}U})$$

The filtered form of Eq. (6.1) becomes :

$$\frac{\partial}{\partial t}(\overline{J^{-1}U}) + \mathcal{L}\left[\frac{\partial\hat{F}}{\partial\xi} + \frac{\partial\hat{G}}{\partial\eta}\right] = \mathcal{L}(S) \tag{6.7}$$

The goal is now to formulate a discrete approximation to this equation which will be solved on the first coarse mesh level. Unfortunately, the present equation gives a time-dependent equation for $\overline{J^{-1}U}$ and not the filtered solution $\bar{U}$. It is important to note that in general $\overline{J^{-1}U} \neq \overline{J^{-1}}\,\overline{U}$. Therefore, we will now define the following "average" solution, $\tilde{U}$:

$$\tilde{U} = \frac{\overline{J^{-1}U}}{\overline{J^{-1}}} \tag{6.8}$$

In terms of $\tilde{U}$ eq. (6.7) may be written as:

$$\frac{\partial}{\partial t}(\overline{J^{-1}}\,\tilde{U}) = \mathcal{L}\left[S - \frac{\partial\hat{F}}{\partial\xi} - \frac{\partial\hat{G}}{\partial\eta}\right] \tag{6.9}$$

In order to put Eq. (6.9) into the same form as the original equation, Eq. (6.1), we define filtered approximations to $\hat{F}_\xi$ and $\hat{G}_\eta$ as follows:

$$\tilde{F}_\xi = \hat{F}(\tilde{U})_\xi \qquad \tilde{G}_\eta = \hat{G}(\tilde{U})_\eta \tag{6.10}$$

Then adding these filtered spatial derivatives to each side of Eq. (6.9), the filtered equation takes on the same form as the original governing equations (Eq. (6.1)):

$$\frac{\partial}{\partial t}(\overline{J^{-1}}\tilde{U}) + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = \tilde{S} \tag{6.11}$$

where

$$\tilde{S} = L\left[S - \frac{\partial \hat{F}}{\partial \xi} - \frac{\partial \hat{G}}{\partial \eta} \right] + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta}$$

Note that the source term which appears in the filtered equation can not be simplified any further since:

$$L\left[\frac{\partial \hat{F}(U)}{\partial \xi} \right] \neq \frac{\partial \hat{F}(\tilde{U})}{\partial \xi} \tag{6.12}$$
$$L\left[\frac{\partial \hat{G}(U)}{\partial \eta} \right] \neq \frac{\partial \hat{G}(\tilde{U})}{\partial \eta}$$

because $\hat{F}$ and $\hat{G}$ are non-linear functions of U and the metrics of the coordinate transformation, and $\tilde{U} \neq U$.

If we define a discrete approximation to the filter operator (i.e., the fine to coarse mesh transfer operator in Multi-grid terminology), Eq. (6.11) is the differential equation which corresponds to the coarse mesh discrete equation solved in Multigrid acceleration schemes To illustrate this point consider the following discrete filter operator. Assume a uniform mesh in computational space with $\Delta\xi = \Delta\eta = 1$ and a weighting function, W, defined as having unit influence over the four fine-mesh cells

surrounding a given mesh point, and zero influence elsewhere (Figure 6.1):

$$W(\bar{\xi}-\xi,\bar{\eta}-\eta) = 1 \quad \text{if} \quad -1 \le \bar{\xi}-\xi \le 1 \text{ and } -1 \le \bar{\eta}-\eta \le 1 \qquad (6.13)$$
$$= 0 \quad \text{if} \quad |\bar{\xi}-\xi| > 1 \text{ or } |\bar{\eta}-\eta| > 1$$

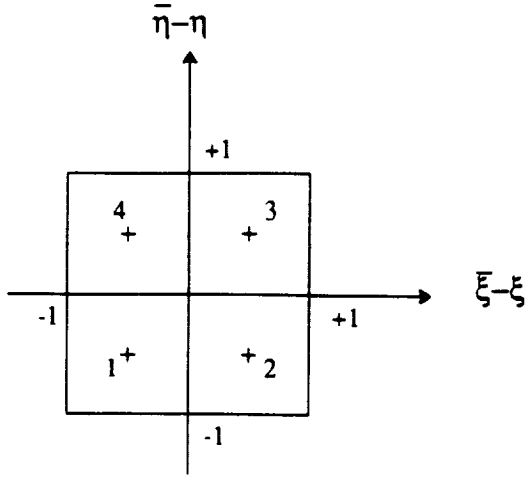


Figure 6.1. Weighting Function Description

The corresponding "average" solution, $\tilde{U}$, is defined as:

$$\tilde{U} = \frac{\overline{J^{-1}U}}{\overline{J^{-1}}} = \frac{L(J^{-1}U)}{L(J^{-1})} \qquad (6.14)$$

Inside the domain (H = 1), with W as defined above:

$$\overline{J^{-1}U} = L(J^{-1}U) = \frac{1}{\lambda}\int_{-1}^{1}\int_{-1}^{1} J^{-1}U\, d\bar{\xi}d\bar{\eta} = \frac{1}{\lambda}\int_{-1}^{1}\int_{-1}^{1} U(J^{-1}\, d\bar{\xi}d\bar{\eta}) = \frac{1}{\lambda}\int_{-1}^{1}\int_{-1}^{1} U\, dV \qquad (6.15)$$
$$= \frac{1}{\lambda}\sum_{i=1}^{4}\Delta V_i U_i$$

In the last step we have used the mean-value theorem with U_i being the cell centered

value of the solution on the fine mesh. In a similar fashion :

$$\overline{J^{-1}} = \mathbb{L}(J^{-1}) = \frac{1}{\lambda} \int_{-1}^{1} \int_{-1}^{1} (J^{-1}\, d\bar{\xi} d\bar{\eta}) = \frac{1}{\lambda} \sum_{i=1}^{4} \Delta V_i \tag{6.16}$$

and $\tilde{U}$ is therefore given as:

$$\tilde{U} = \frac{\sum_{i=1}^{4} \Delta V_i U_i}{\sum_{i=1}^{4} \Delta V_i} \equiv U_{2h} = I_h^{2h}\, U_h \tag{6.17}$$

The "average" solution $\tilde{U}$ is exactly the same as the transferred solution, U^{2h}, used in Jameson's FAS Multi-grid formulation (see Eq. (5.9)). Similarly:

$$\mathbb{L}\left[S - \frac{\partial \hat{F}}{\partial \xi} - \frac{\partial \hat{G}}{\partial \eta} \right] = \frac{1}{\lambda} \int_{-1}^{1} \int_{-1}^{1} \left[S - \frac{\partial \hat{F}}{\partial \xi} - \frac{\partial \hat{G}}{\partial \eta} \right] d\bar{\xi} d\bar{\eta} \tag{6.18}$$

$$= \frac{1}{\sum_{i=1}^{4} \Delta V_i} \sum_{i=1}^{4} (\text{Flux})_i$$

$$\equiv I_h^{2h} (F_h - L_h U_h)$$

Which is the same as the transferred fine mesh residual as defined by Jameson (see Eq. (5.10)).

In summary, Eq. (6.11) is the differential counterpart to the discrete coarse mesh equations used in Multi-grid acceleration schemes. Expressed in this form, there is a direct relationship between the filter operator and the resulting filtered governing equations. While the present discussion details a single application of the filtering process, if the process is repeated, filtering over increasing length scales, higher and

higher levels of the filtered equation can be formulated. Further, each level can be cast in the same form as that of eq. (6.11). From each new level of filtered equation the corresponding discrete coarse mesh Multigrid equation can be determined. Based on this general derivation, it is clear that Jameson's Multi-grid acceleration scheme is one of a large number of Multi-grid formulations which can be constructed from these equations.

6.2 Coarse-Grid Boundary Conditions

The boundary conditions on the coarse mesh levels are applied at the cell faces, in the same manner as the fine-mesh boundary conditions, with the addition of forcing terms to allow them to be solved with fine-mesh accuracy. These boundary forcing terms will be derived in a similar fashion to the interior forcing terms, using the above general Multigrid approach to determine the correct way to transfer the boundary values to the coarse-meshes.

At a boundary, as in Figure 6.2, the function H is zero for the image cells and 1 for the interior cells. The weighting function, W, is the same function as for the interior, given in Eq. (6.12).

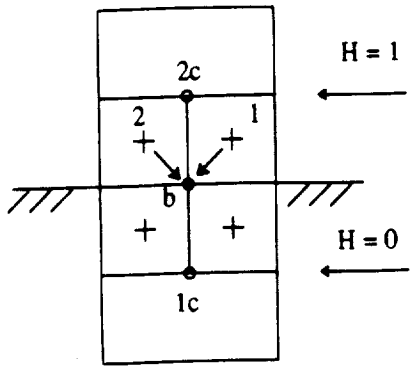


Figure 6.2. Coarse-Mesh Boundary Transfer Notation

Applying the filter, $\mathbb{L}$, to the fine-mesh cells adjacent to the boundary, and integrating as before, the transferred boundary value on the coarse mesh, $\tilde{U}_b$, is defined as:

$$\tilde{U}_b = \frac{\sum_{i=1}^{2} \Delta V_i U_i}{\sum_{i=1}^{2} \Delta V_i} \tag{6.19}$$

The coarse-mesh boundary conditions are now formulated in terms of the transferred solution using the same discrete Multigrid equations as for the interior points:

$$L_{2h}\tilde{U}_b = F_{2h} = I_h^{2h}(F_h - L_h U_h) + L_{2h}(I_h^{2h} U_h) \tag{6.20}$$

Since the boundary conditions on each level are satisfied exactly, the quantity, $I_h^{2h}(F_h - L_h U_h)$ is always zero, and therefore, the forcing term is :

$$F_{2h_b} = L_{2h}(I_h^{2h} U_h) \tag{6.21}$$

which is a function of the transferred boundary value, $\tilde{U}_b$, and the transferred interior solution at point 2c (Figure 6.2). As an example, take the inviscid flow tangency

condition:

$$q_{s_b} = q_{s_{2c}} \quad ; \quad q_{n_b} = 0 \tag{6.22}$$

For the tangential velocity, q_{s_b}, the boundary condition on the fine mesh in Multigrid notation is:

$$L_h U_h = F_h \tag{6.23}$$

where

$$L_h U_h = q_{s_b} - q_{s_{2c}}$$
$$F_h = 0$$

For the coarse mesh levels:

$$L_{2h} U_{2h} = F_{2h} \tag{6.24}$$

where

$$L_{2h} U_{2h} = (q_{s_b}^{(k)} - q_{s_{2c}}^{(k)})_{2h}$$
$$F_{2h} = q_{s_b}^{(0)} - q_{s_{2c}}^{(0)}$$

and where the quantities in parentheses denote the Runge-Kutta stage, with zero representing the initially transferred quantities. The updated boundary value at each Runge-Kutta stage is therefore given as:

$$q_{s_b}^{(k)} = q_{s_{2c}}^{(k)} + [q_{s_b}^{(0)} - q_{s_{2c}}^{(0)}] \tag{6.25}$$

The conservation variables in the image cells are then updated by linear extrapolation from the interior based on the updated value of $\tilde{U}_b$:

$$\tilde{U}_{1c} = 2\tilde{U}_b - \tilde{U}_{2c} \tag{6.26}$$

For the Navier-Stokes equations, the forcing term at a solid wall essentially adds a

partial slip condition as the mesh becomes too coarse to fully resolve the boundary layer. In the limit, as the mesh becomes very coarse, this is equivalent to applying the full inviscid slip condition at the wall.

CHAPTER 7 - RESULTS

The numerical algorithm described in the previous sections has been applied to a variety of both inviscid and viscous 2-D flow computations in order to determine the effect of the Multigrid boundary conditions on convergence. Solutions in each case are given to validate the accuracy of the present code, and then convergence histories for applications with and without the coarse-mesh boundary forcing terms are shown. It is stressed here that all Multigrid and single-grid solutions were the same. References to forcing terms refer to those implemented in the boundary cells (Chapter 6) and should not be confused with the forcing terms for the interior points given in chapter 5. The number of grids specified refers to the number of mesh levels used in the Multigrid calculation, with 1 being the finest mesh.

Convergence is measured in terms of the average residuals of the conservation variables which have been normalized at each point by the local time-step. One Multigrid Cycle is defined as a complete sequencing through all mesh levels to the coarsest mesh and then back to the finest mesh again. For single-grid solutions, this is the same as one time-step, or one iteration.

Computations were performed on a Gould NP-1 mainframe computer, and on an Ardent Titan graphics workstation. CPU times on the Titan with a vectorized code were equivalent to those on the NP-1 with an unvectorized version. All CPU times are

given in minutes.

7.1 Inviscid Channel Flow

Initial code validation was performed for subsonic, transonic, and supersonic flow in a channel with a circular-arc bump on the lower wall. This is a fairly common problem, used in other works as an initial validation tool.[12][13][15][27] A 129x33 algebraically stretched mesh is used for all 3 cases with minimum increments : $\Delta x = \Delta y = .015625$. The subsonic and transonic cases use a 10% bump, and the supersonic case uses a 4% bump, both of which are shown in Figure 7.1. The channel is 3 chords long and 1 chord high.

7.1.1 Subsonic Case

Figure 7.2 shows the solutions obtained for the $M_\infty = .5$ case. The spikes in the total pressure loss at the leading and trailing edge (Figure 7.2b) are due to the artificial dissipation in the numerical scheme activated by the stagnation point pressure gradients, and to the discontinuity in the grid metrics there. The trailing edge pressure loss is convected downstream and slightly affects the symmetry of the solution.

Figure 7.3 presents the convergence histories for the subsonic case with and without forcing terms for 3, 4, and 5 grid levels. From these plots, it can be seen that the fine-grid convergence level of a Multigrid calculation is affected by errors introduced as a result of the incorrect formulation of the boundary conditions on the coarse levels. Note that with forcing terms applied, all 3 solutions reach the same convergence level as the fine-mesh. As the number of grid-levels increases, the coarse-mesh truncation error

grows (without boundary forcing terms) due to the increasing mesh coarseness. This is summarized in Figure 7.4 for the 3,4, and 5 level solutions without forcing terms. The forcing terms therefore permit any number of grids to be used (as permitted by the initial mesh resolution of the finest level) while still maintaining fine-mesh accuracy.

Figure 7.5 shows the efficiency gains realized in the Multigrid calculations for various numbers of grids. The single-grid convergence rate is shown and compared to 2,3,4, and 5-grid rates for Multigrid cycles, and CPU time. CPU time is important in Multigrid calculations, since no actual saving are realized by Multigrid acceleration if the number of cycles required for convergence, though fewer, take longer to perform. Note that the 3,4, and 5 grid solutions use the same amount of CPU time to reach a converged solution. This shows that the additional work required above 2 levels is negligible for the present Multigrid algorithm, illustrating its efficiency.

7.1.2 Transonic Case

Figure 7.6 shows the solutions for the supercritical, $M_\infty = .675$ case. A small region of locally supersonic flow develops over the bump, with a shock occurring over 3 grid points at approximately 70% of the chord. The Mach number upstream of the shock is 1.44, and 5% total pressure loss is generated and convected downstream along streamlines. For a normal shock of 1.44, shock tables give a total pressure loss of 5.24%, which agrees very well with the calculated loss.

Figure 7.7 shows the convergence histories of the transonic case for solutions with and without boundary forcing terms. The same trends exist here as were realized for the subsonic case, with the untreated boundary conditions on the coarse-levels

introducing coarse-mesh truncation error into the fine-mesh solution.

Figure 7.8 shows the difference between the single grid and 5-grid convergence rates compared to both Multigrid cycles and CPU time. The gains realized by Multigrid are somewhat less than for the subsonic case, but this can be attributed to the increased convergence rate of the single-grid solution.

7.1.3 Supersonic Case

Figure 7.9 presents the results for the $M_\infty = 1.4$ channel flow case. Oblique shocks form at both the leading and trailing edge and expand as they propagate downstream and towards the upper boundary. The leading edge shock angle is such that a standing normal shock exists at the upper boundary where the leading edge shock impinges. The trailing edge shock is somewhat thicker due to the effect of the expansion waves generated over the bump. As the leading edge shock is reflected from the upper wall, it is weakened as it passes through the trailing edge shock and then intersects the lower wall before being weakly reflected.

Figure 7.10 shows the single grid convergence rates for the supersonic case compared to the 5-grid computations. Applying or not applying the boundary forcing terms made no difference in this case. The single grid solution converges very quickly, while the Multigrid solution is somewhat slower than in the previous cases, resulting in a large decrease in the overall Multigrid efficiency. This is due to the increased flow speed.

As the overall speed of a flow increases, single-grid convergence rates tend to

increase, and Multigrid efficiency drops. This can be shown by a comparison of the convergence rates, with and without Multigrid acceleration, of the above 3 cases, shown in Figure 7.11. The convergence rates of the single-grid solutions increase with Mach number, while the Multigrid convergence rates remain approximately the same. Therefore, the Multigrid efficiency gains are limited by the fine-mesh.

7.2 Viscous Flat Plate

As a first step in validating the present code for the solution of the full Navier-Stokes equations, laminar viscous flow was computed over a flat plate. A 65x33 algebraically stretched mesh was used for these computations, shown in Figure 7.12, with minimum increments of $\Delta x_{min} = .01$ and $\Delta y_{min} = .0025$. The leading edge of the plate begins one unit in from the inlet boundary and extends the length of the domain.

Flow tangency is enforced along the lower wall upstream of the plate. At the upper wall, the normal velocity is extrapolated from the interior, and the pressure, tangential velocity, and density are specified as having freestream values. The inflow boundary conditions are the same as those used in the inviscid channel flow solutions. However, due to the non-isentropic nature of the boundary layer that passes through the outflow boundary, pressure is specified and the other variables are extrapolated from inside the domain. For this calculation, the exit pressure is specified as freestream. A no slip condition is enforced along the flat plate, with pressure and density obtained by specified adiabatic conditions at the wall, $(\partial T/ \partial \eta)_{wall} = 0$, and zero pressure change normal to the wall, $(\partial P/ \partial \eta)_{wall} = 0$.

Figure 7.13 shows the solutions for $M_\infty = .5$ and Reynolds Number: $Re_x)_{x=1} = 8000$. The skin friction coefficient and boundary layer profiles are compared to results obtained from a 2-D compressible boundary layer calculation[28] utilizing 99 points normal to the wall. The velocity and temperature are normalized by freestream values and plotted verses the boundary layer parameter, $\eta = y/x\sqrt{Re_x}$, where Re_x is based on distance from the leading edge of the plate.

Figure 7.14 presents the convergence histories for this case with and without boundary forcing terms. As in the inviscid computations, the level of error becomes higher as the number of grids increases. Note that with the forcing terms applied, the same convergence level is reached for both the 4 and 5 grid calculations. For this viscous calculation, the error introduced by the coarse-mesh boundary conditions without forcing terms seems to be greater than for the previous inviscid cases. This can be attributed to the presence of the boundary layer. As the grid levels become coarser, boundary layer resolution is not possible and so the no-slip boundary condition on these levels is physically incorrect. The forcing terms essentially add a partial slip condition as the first mesh line becomes farther away from the boundary, and in the limit of the coarsest possible mesh, this is equivalent to specifying flow tangency along the wall.

However, merely specifying flow tangency on the coarse mesh levels without the forcing terms still introduces error into the fine-mesh solution, as can be seen in Figure 7.15a. When the forcing terms are utilized, all coarse-mesh truncation error is effectively eliminated. In Figure 7.15b, this flow tangency condition on the coarse mesh levels with forcing terms is compared to the no-slip condition with forcing terms,

and the flow tangency convergence is somewhat faster. This may may be due to an over-relaxation at the boundaries resulting in increased steady-state convergence.

Figure 7.16 shows the gains in efficiency realized by Multigrid solutions for the flat plate case. They are not as high as those obtained in the previous inviscid cases which is typical for viscous solutions. The boundary forcing terms, although reducing the error for the 5-grid solution, did not improve the convergence rate as was initially hoped.

7.3 Viscous Circular Arc Bump

Subsonic, laminar flow through a channel with a 5% circular-arc bump on the lower wall is computed. [15][27][29] A 65x33 mesh is used in these calculations and is shown in Figure 7.17. The minimum Δx and Δy are the same as for the flat plate mesh, and the stretching is applied at both the leading and the trailing edge.

Flow tangency is enforced both along the upper wall and the lower wall upstream of the leading edge. A no-slip condition is applied over the bump and downstream of the trailing edge. This case may also be referred to as a 10% bi-circular arc cascade with a sting mounted at the trailing edge and extending downstream.[15] The normal momentum equation (Eq. 4.20) is used to determine the pressure, and density is obtained from the wall temperature, specified for this case as the freestream total temperature: $T_{wall} = T_{o_\infty}$. The inflow and outflow boundary conditions are the same as for the flat plate case.

Figure 7.18 gives the solutions for $M_\infty = .5$ and a Reynolds Number (based on

chord) of 8000. Boundary layer separation occurs at 81% of the chord, with the separation bubble extending downstream to 120% chord, where the flow re-attaches. Skin friction and static pressure results agree very well with those given by Kallinderis.[29] The velocity profiles agree fairly well with those reported by Rhie.[27] However, a slightly thicker boundary layer is predicted by Rhie, possibly due to higher levels of artificial dissipation, and the discrepancy increases with distance over the bump. The separation bubble occurring at the trailing edge may be seen in the contour plots and velocity vector plots.

Figure 7.19 shows the Multigrid results for this case with and without boundary forcing terms for 3, 4, and 5 grids. The increase in error with number of grids is greater than that for the flat plate case, possibly due to more complex flow phenomena and the additional stretching at the trailing edge.

Figure 7.20 summarizes the increased effect of the coarse-mesh truncation error (without forcing terms) on the fine-mesh as the number of mesh levels increases. The effect of applying flow tangency on the coarse grids for this viscous case is shown, with and without forcing terms, in Figure 7.21a. The results are similar to the corresponding flat plate case (Figure 7.15a), although there is no apparent increase in convergence rate for the flow tangency condition with forcing terms (Figure 7.21b).

Figure 7.22 presents the efficiency of the Multigrid calculations compared to both Multigrid cycles and CPU time with the single-grid convergence. The increase in efficiency measured by CPU time is less than that for Multigrid cycles due to the additional work required in calculating the viscous stresses.

7.4 VKI Gas Turbine Rotor Blade

In the interest of applying the present algorithm to a more realistic problem, inviscid and viscous calculations are performed here for a VKI gas turbine rotor blade.[30] The common geometric parameters for cascade calculations are defined in Figure 7.23, and for this case are specified as:

$$\frac{g}{c} = .697 \quad \gamma = 33.35 \quad \beta_1 = 24^\circ$$

Where g/c is the gap-to-chord ratio, γ is the blade stagger angle, and β_1 is the inlet flow angle.

The present computations were performed for $M_1 = .19$ and $M_{2_{is}} = 1.0$, where $M_{2_{is}}$ is the specified outlet isentropic Mach number which sets the ratio of exit static pressure to upstream total pressure (see Eq. 4.19) This ratio is .53 for the current value of $M_{2_{is}}$. The flow is turned 96 degrees to a final steady-state outlet flow angle of approximately 72 degrees.

7.4.1 Inviscid Case

Figure 7.24 shows the 73x17 C-mesh used for the inviscid calculations. At the inlet, Riemann Invariant boundary conditions are used (see chapter 4) with specified total pressure, total temperature, and flow angle. The outflow boundary conditions use characteristic variables with a specified exit pressure. Periodicity is enforced along the upper and lower boundaries, and flow tangency is enforced on the blade.

Figure 7.25 gives the inviscid results. All surface quantities are plotted along the blade's coordinate system, not the axial distance. The local isentropic Mach number is

given as :

$$M_{is} = \left[\frac{2}{(\gamma - 1)} \left[\frac{1}{(p/p_{o_1})^{(\gamma-1)/\gamma}} - 1 \right] \right]^{1/2} \tag{7.1}$$

Comparisons are made with blade-to-blade H-mesh calculations performed by Arts[30] which used a corrected viscosity scheme, and with experimental results obtained by VKI.

The flow is accelerated on the suction surface until a weak shock forms. The present calculations have smeared this region, and have underpredicted the flow expansion, possibly due to the coarseness of the mesh and the type of artificial dissipation that is used. The leading edge pressure gradients are very high, activating the 2nd order dissipative terms and creating large total pressure losses. Large total pressure losses are also incurred at the trailing edge due to the wedge-type truncation verses a smooth, blunt rounding of the trailing edge. In order to realize minimum total pressure loss for inviscid cascade computations, it is essential that the normal momentum equation, tangential momentum equation, and flow tangency all be satisfied simultaneously at each point along the blade.[9] However, this method was not used in the present work. The weak suction surface shock and supersonic bubble may be seen in the Mach contours at about 50% chord.

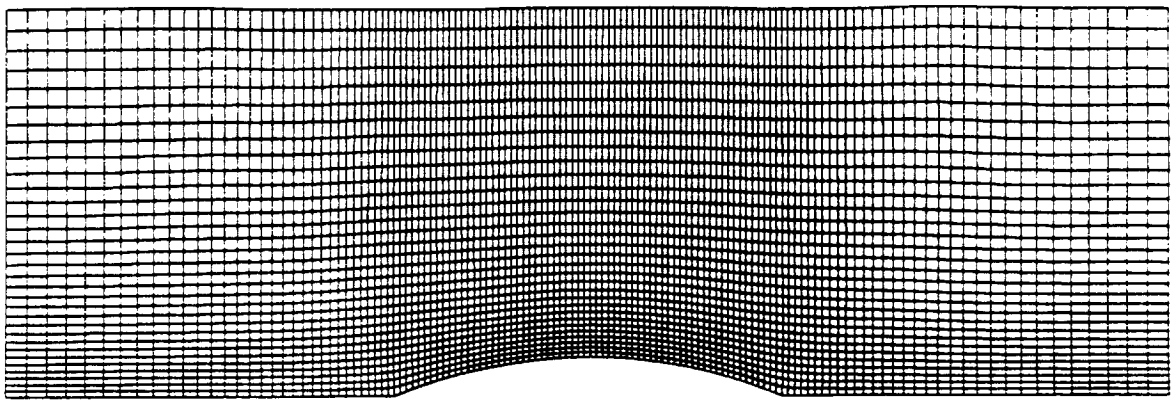
Figure 7.26 shows the convergence histories for the inviscid case. Since the coarse-mesh truncation error is quite low for only 3 levels, the boundary forcing terms had no effect on the convergence levels for this case. Due to a large portion of high-

speed flow, the single-grid convergence was quite rapid, and Multigrid therefore did not produce large savings.

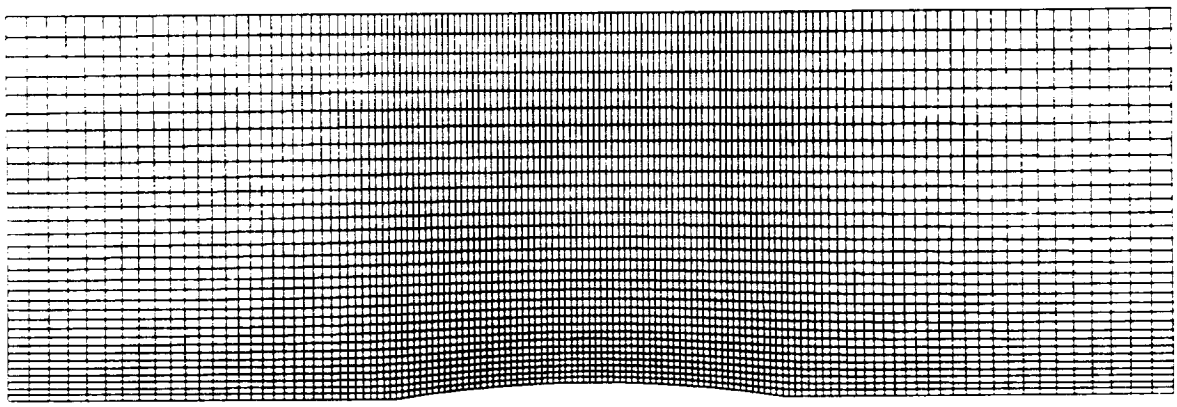
7.4.2 Viscous Case

Figure 7.27 shows the 129x41 C-mesh that is used in the viscous computations. Results for laminar flow at a Reynolds Number of 10,000 are presented in Figure 7.28. The isentropic Mach number compares quite well with Arts'[30] computations and the experimental results.[30] However, the shock on the suction surface remains smeared and the flow upstream of it under-expanded due to viscous smearing which occurs in regions of high adverse pressure gradients. A large portion of low-speed flow exists near the stagnation point, seen by the delayed onset of boundary layer development on the pressure surface (Contour Plot). Laminar separation can be seen occurring at about 40% of the blade chord on the suction surface (velocity vectors).

Figure 7.29 shows the convergence histories for this case. 1 and 3 grid convergence rates are plotted versus Multigrid cycles and CPU time. Multigrid produces substantial savings for this case.

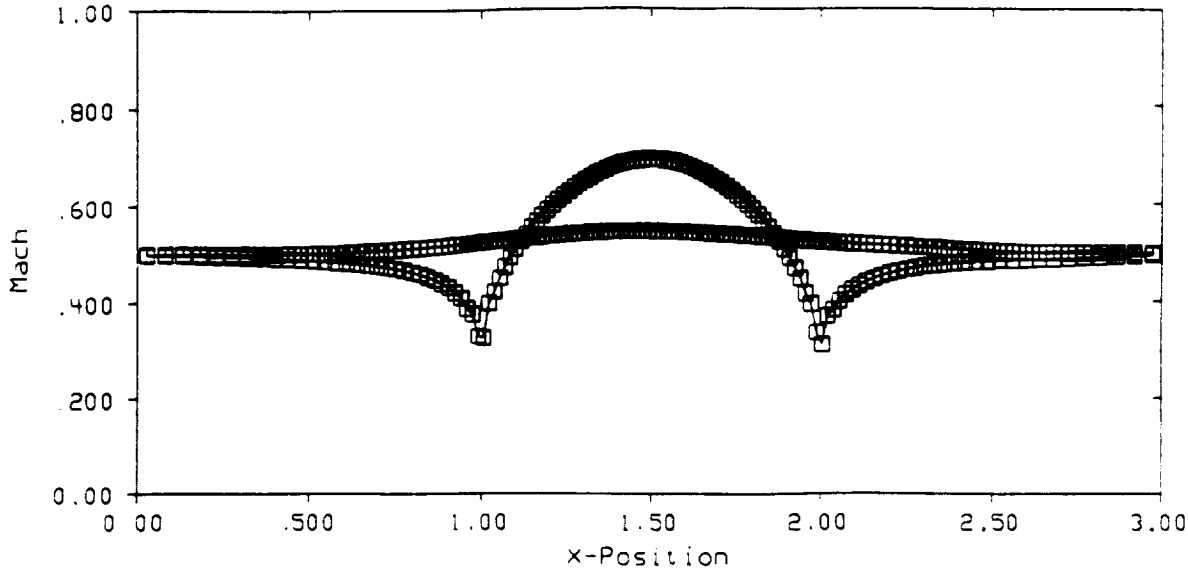


a)

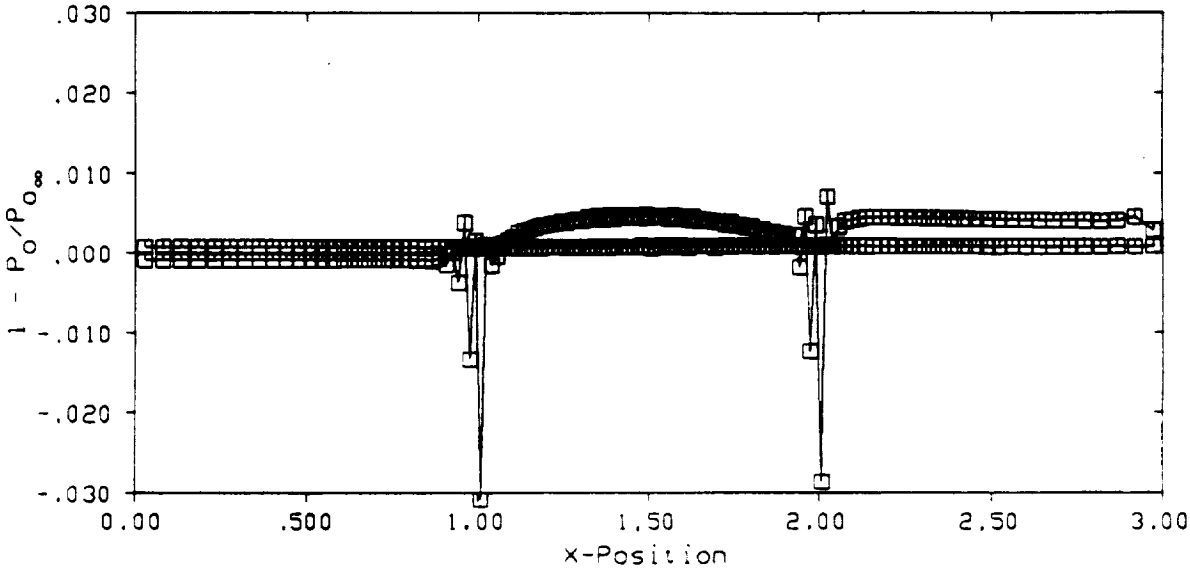


b)

Figure 7.1. 129x33 Computational Grids For Inviscid Channel Flow. (a) 10% Circular Arc Bump (b) 4% Circular Arc Bump

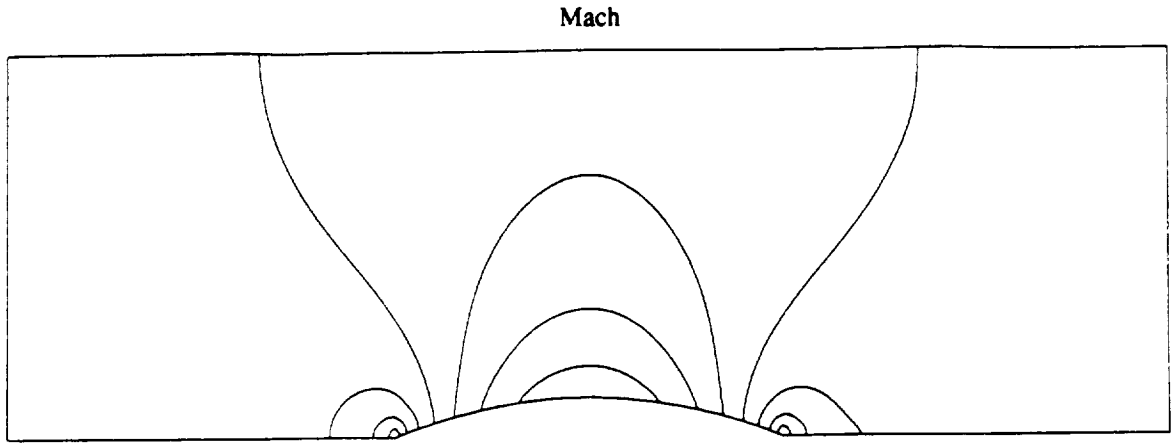


a)

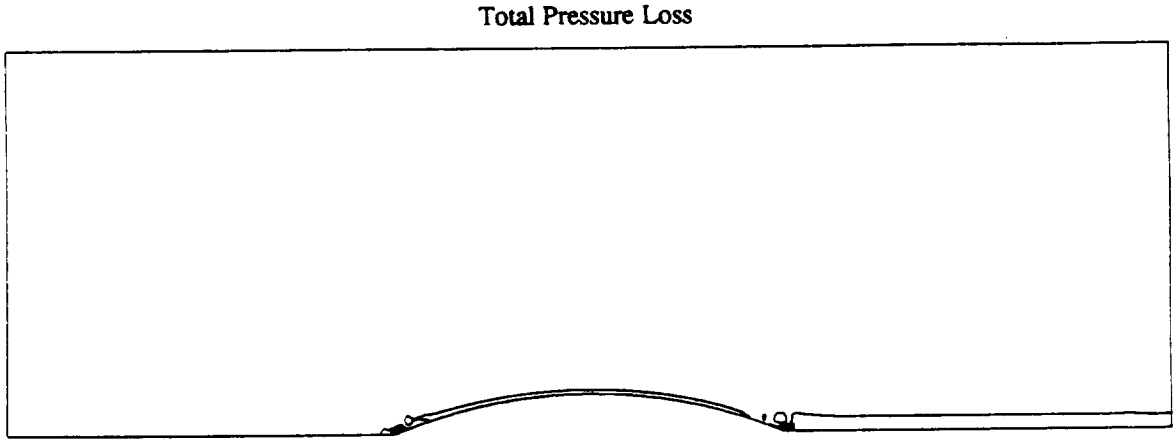


b)

Figure 7.2. $M_\infty = .5$ Inviscid Channel Flow Solution. (a) Surface Mach Number (b) Surface Total Pressure Loss (c) Mach Contours, $\Delta M = .05$ (d) Total Pressure Loss Contours, $\Delta P_o = .01$

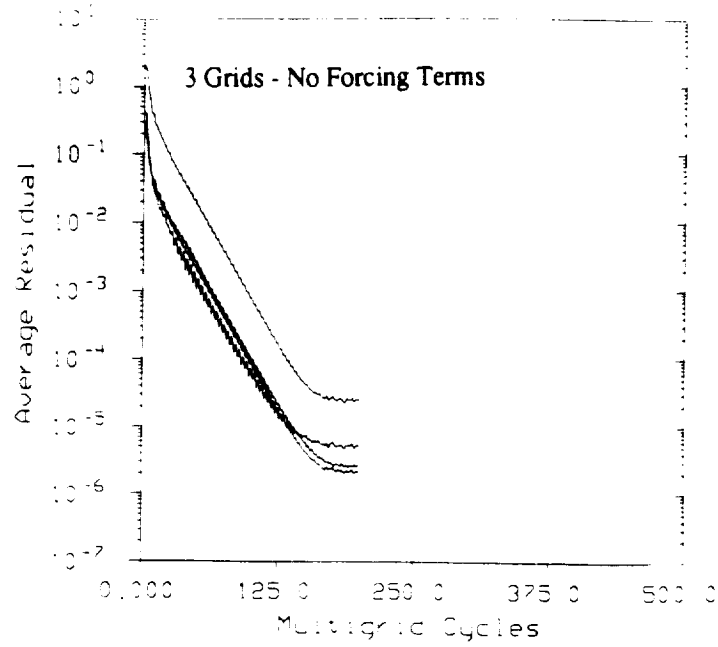


c)

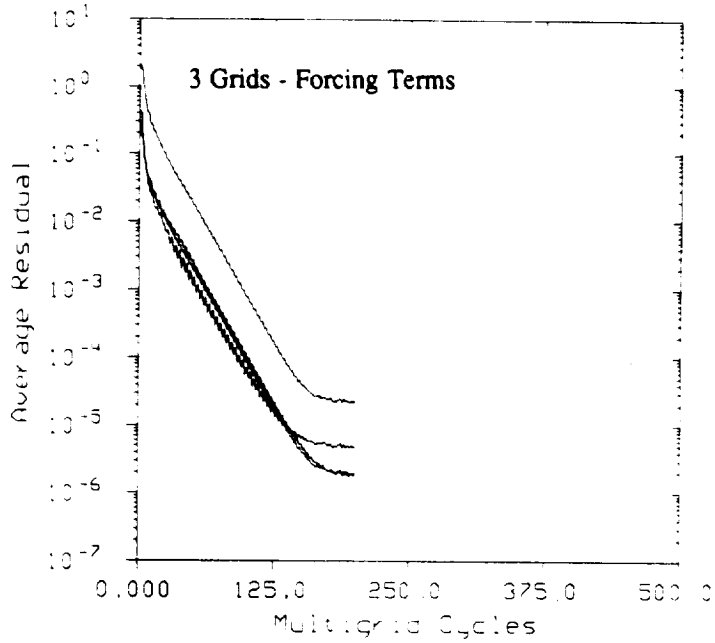


d)

Figure 7.2. Continued.

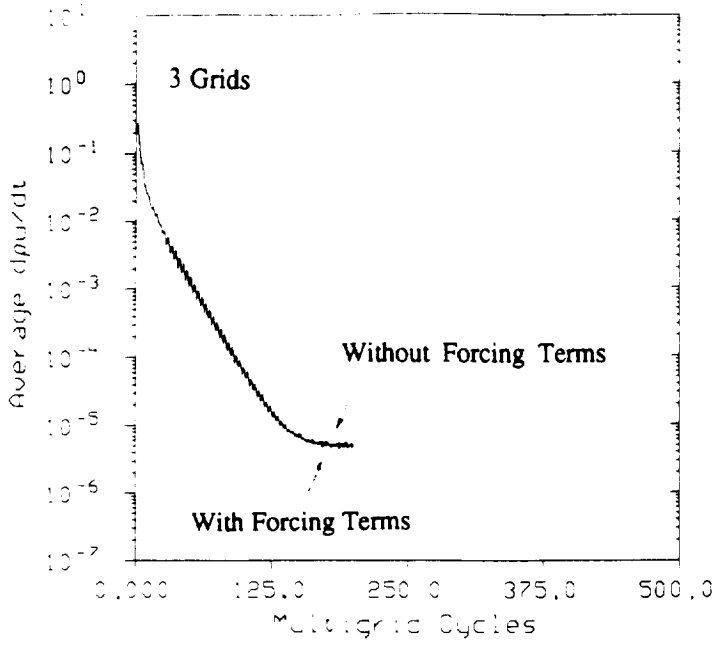


a)

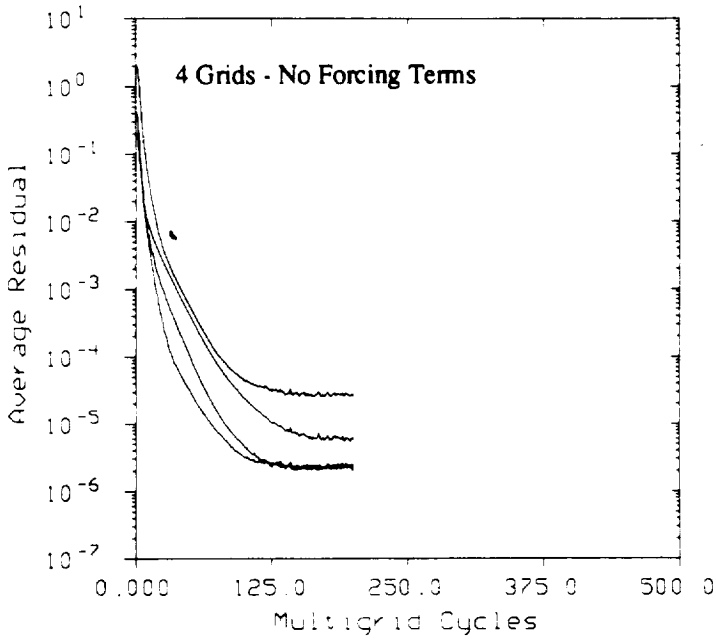


b)

Figure 7.3. Effect of Boundary Forcing Terms on Inviscid $M_\infty = .5$ Convergence. 3 Grids: (a) No Forcing Terms (b) Forcing Terms (c) Summary. 4 Grids: (d) No Forcing Terms (e) Forcing Terms (f) Summary. 5 Grids : (g) No Forcing Terms (h) Forcing Terms (i) Summary

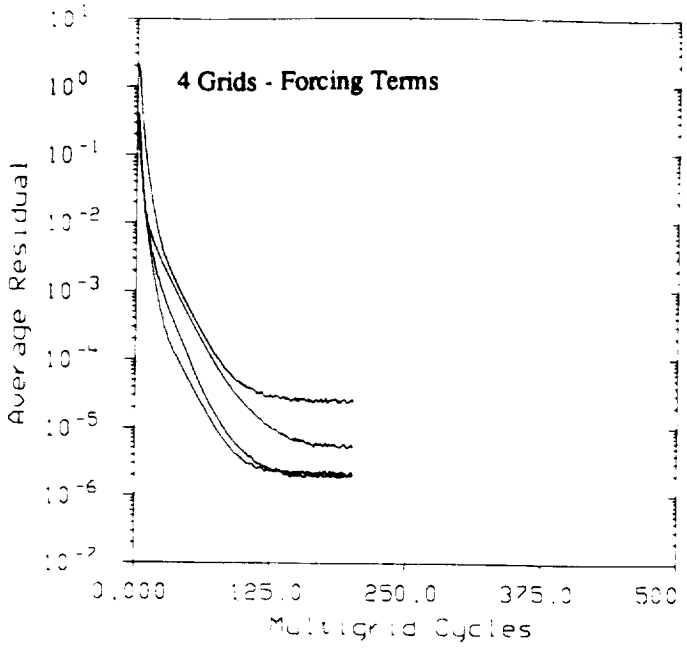


c)

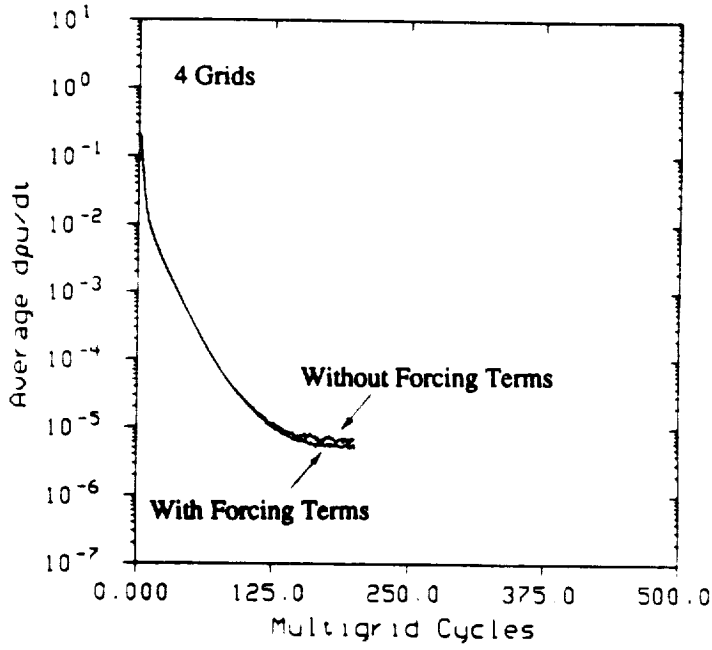


d)

Figure 7.3. Continued.

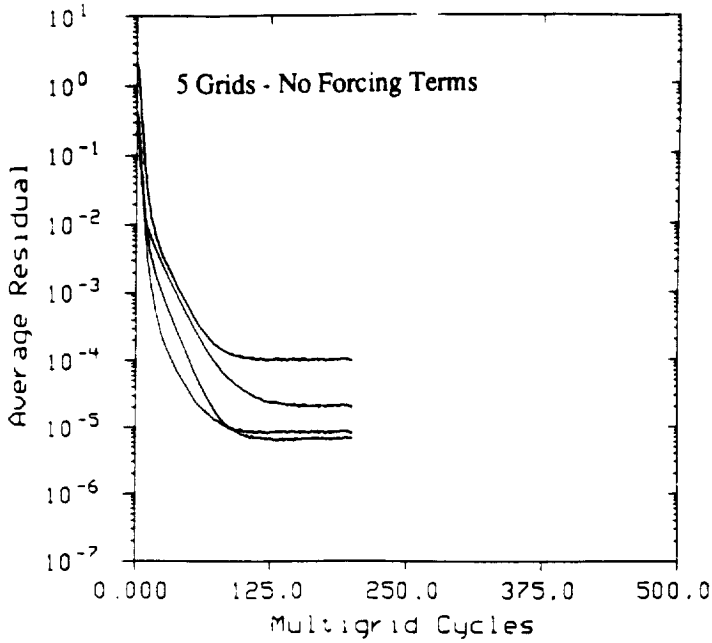


e)

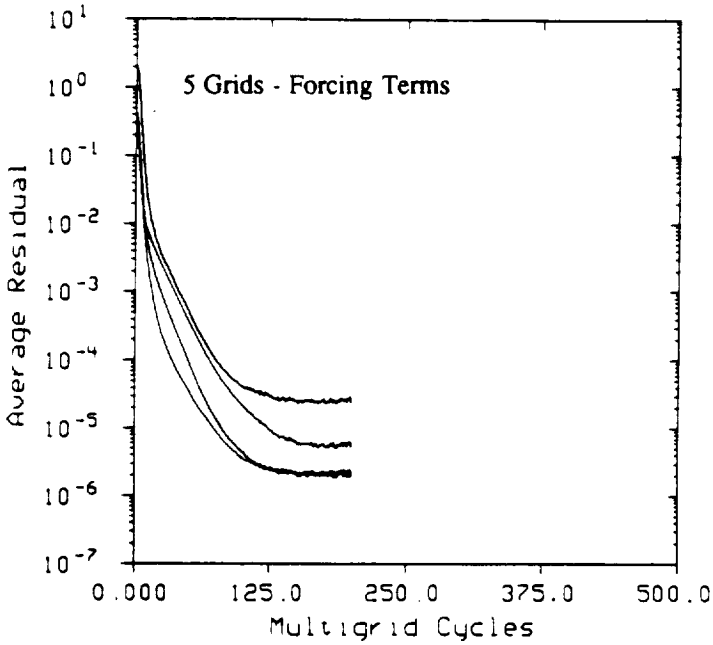


f)

Figure 7.3. Continued.

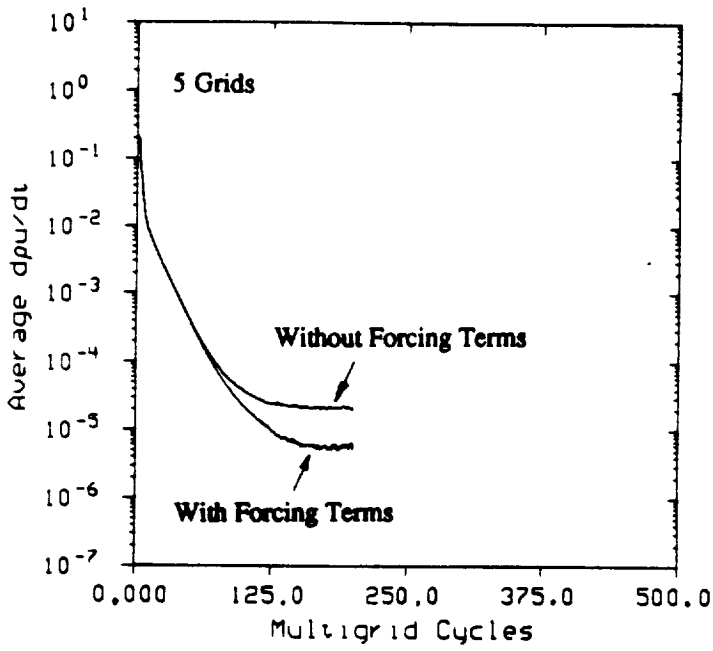


g)



h)

Figure 7.3. Continued.



i)

Figure 7.3. Continued.

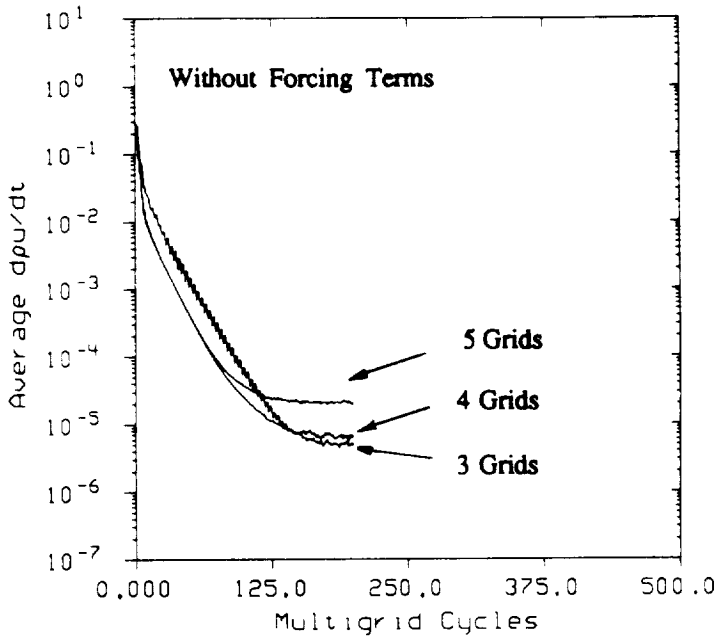
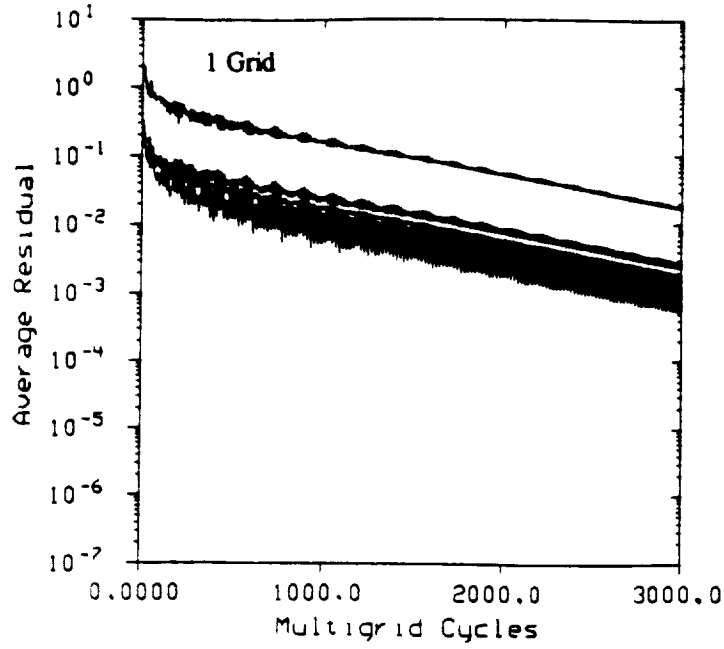
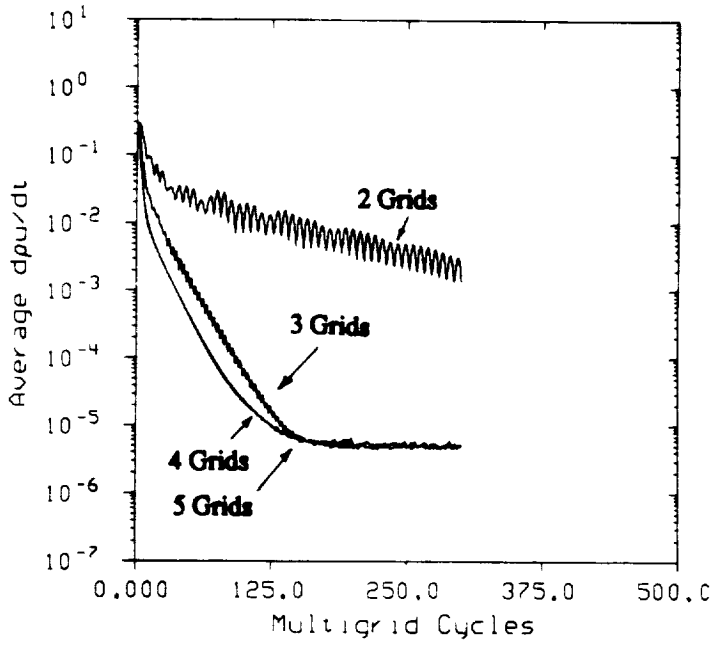


Figure 7.4. Increase In The Influence Of Coarse-Mesh Truncation Error On Fine-Mesh Convergence As The Number Of Grids Increases

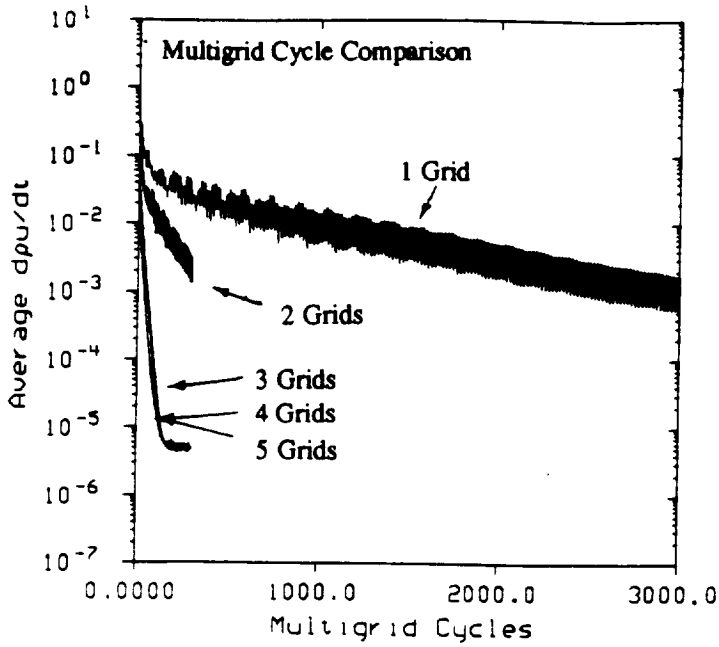


a)

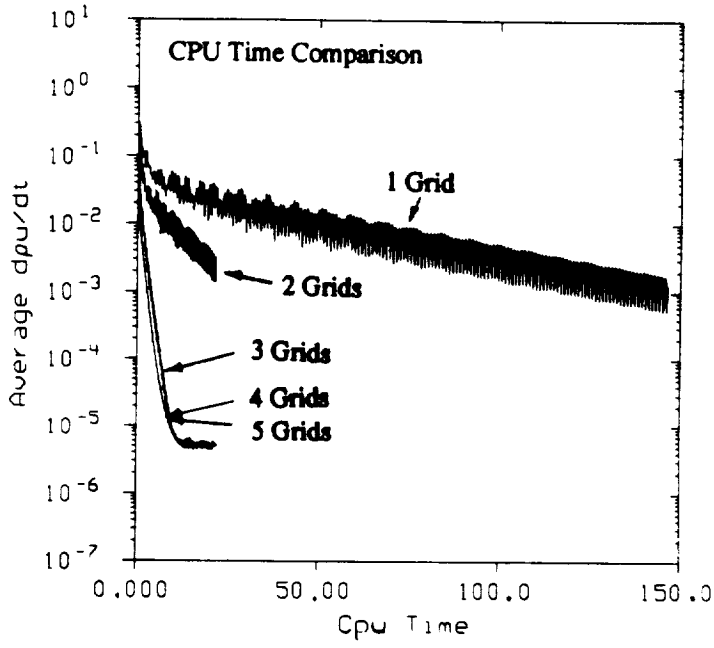


b)

Figure 7.5. Multigrid Efficiency For Inviscid, $M_\infty = .5$ Case. (a) 1 Grid (b) 2-5 Grids (c) 1-5 Grids Multigrid Cycle Comparison (d) 1-5 Grids CPU Time Comparison

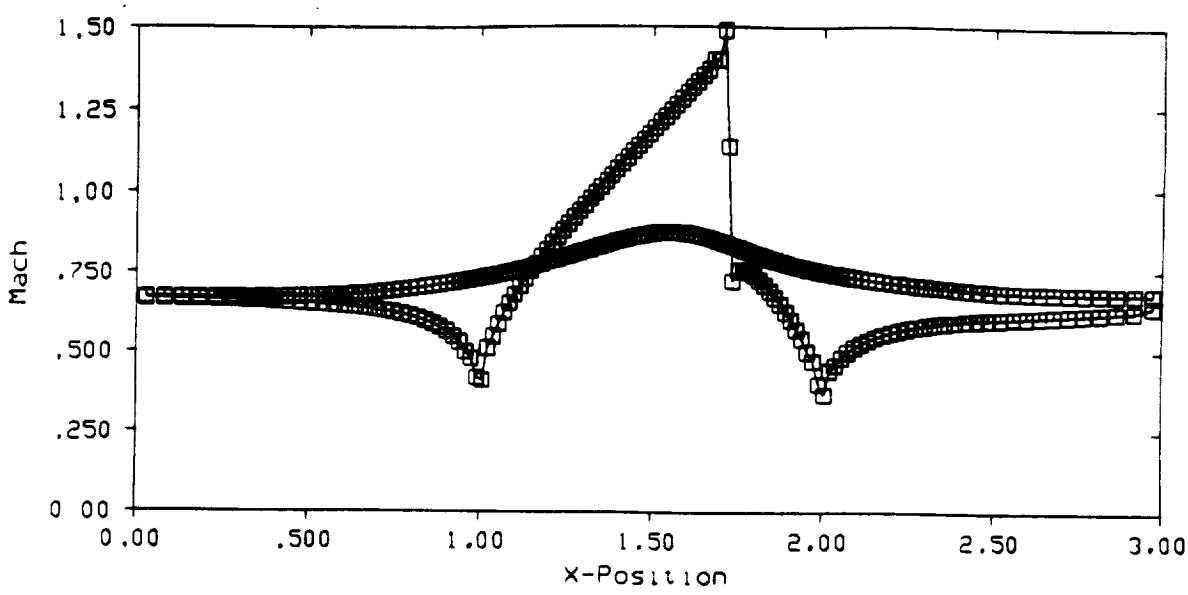


c)

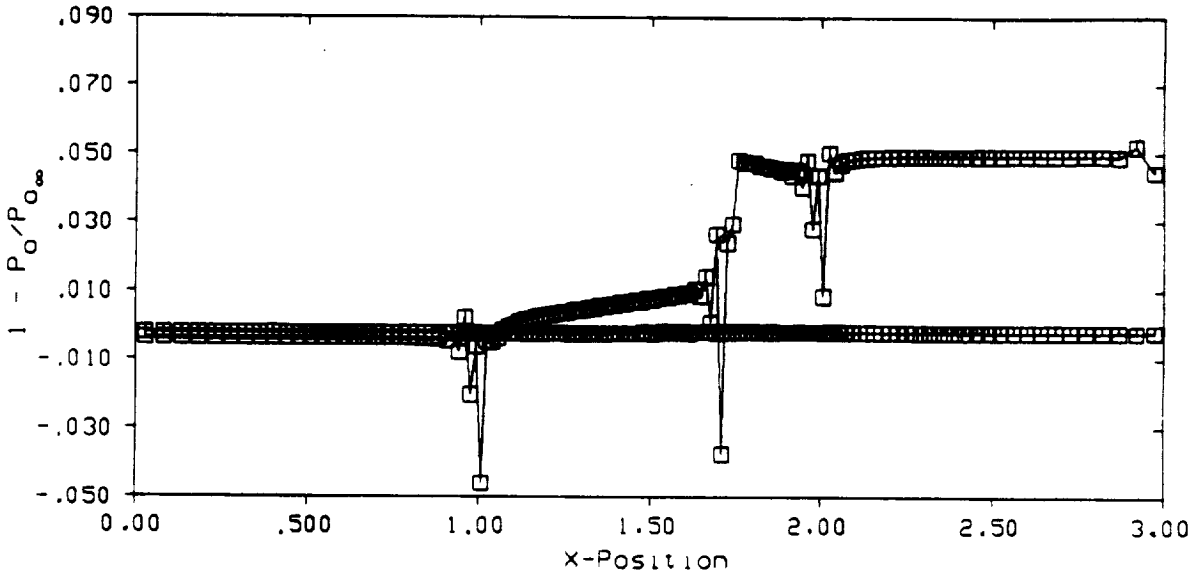


d)

Figure 7.5. Continued.

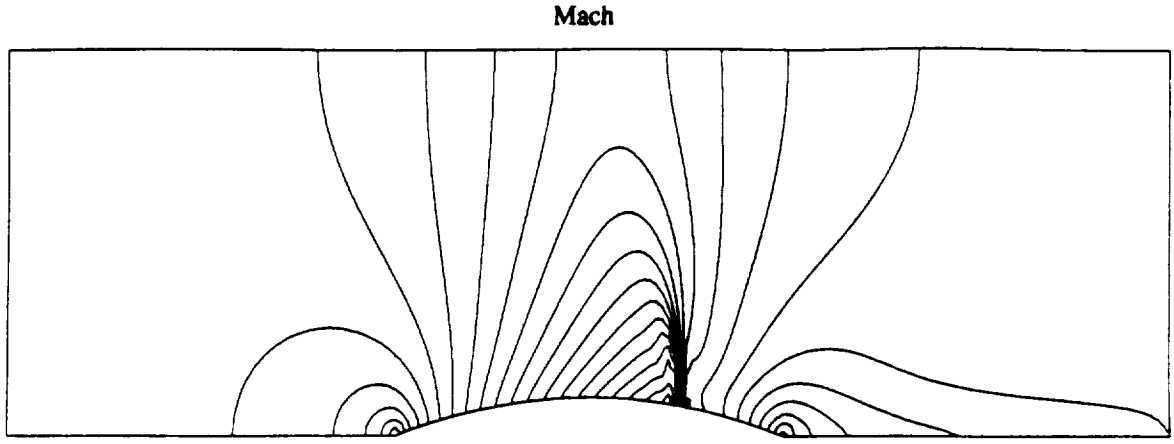


a)

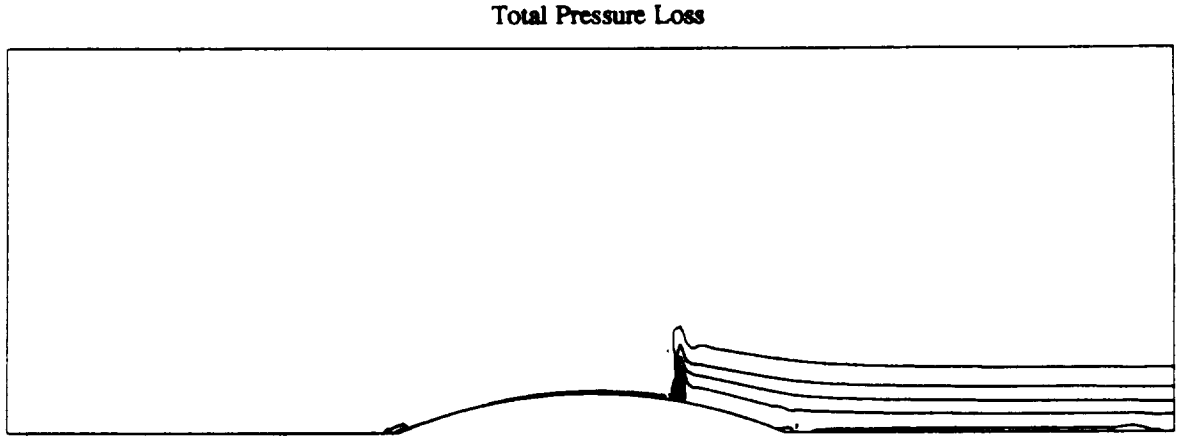


b)

Figure 7.6. $M_\infty = .675$ Inviscid Channel Flow Solution. (a) Surface Mach Number (b) Surface Total Pressure Loss (c) Mach Contours, $\Delta M = .05$ (d) Total Pressure Loss Contours, $\Delta P_o = .01$

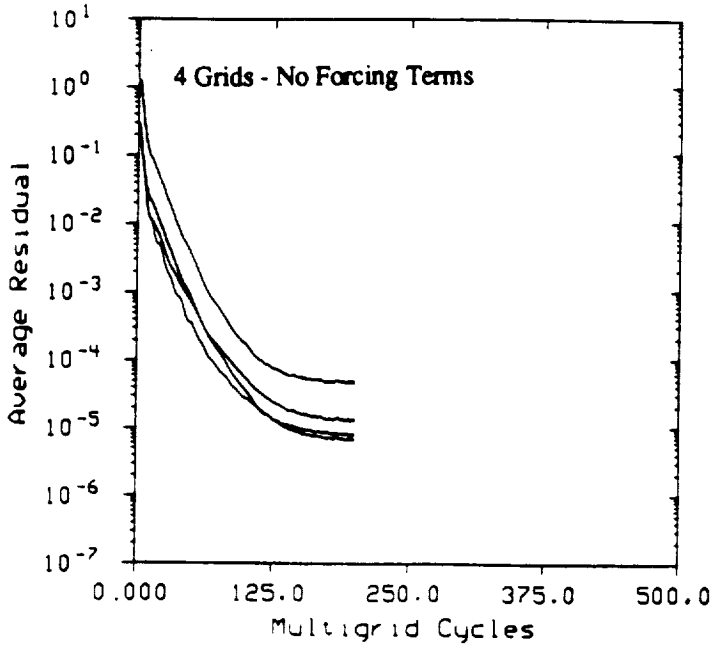


c)

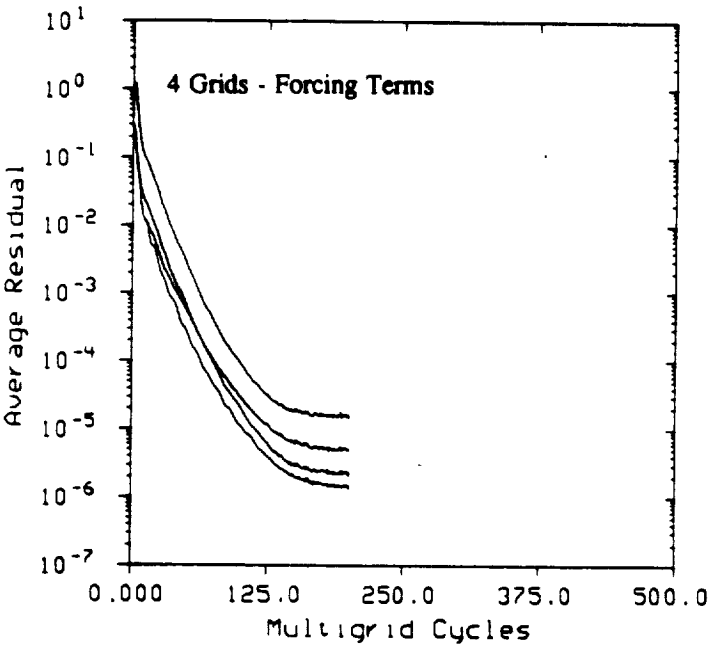


d)

Figure 7.6. Continued.

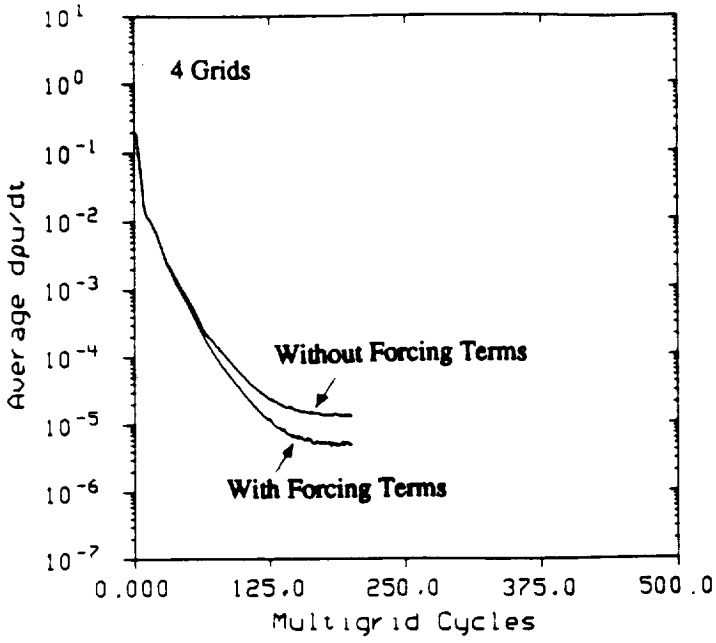


a)

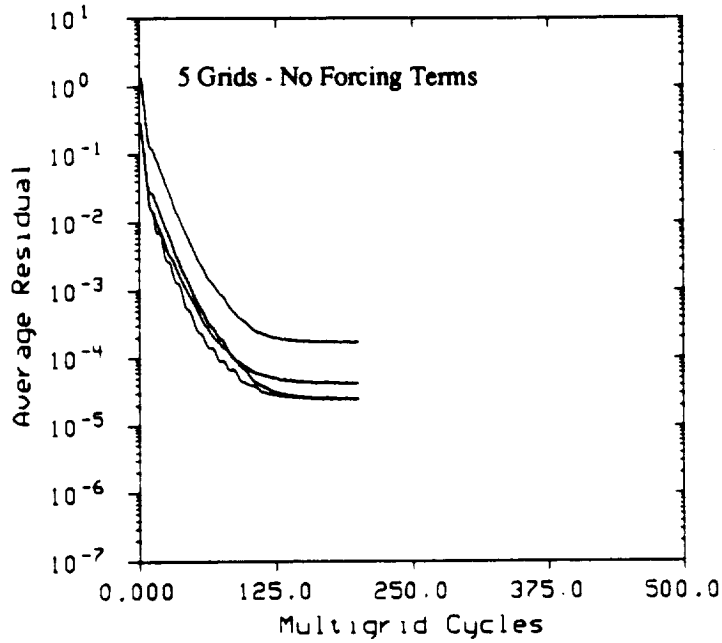


b)

Figure 7.7. Effect of Boundary Forcing Terms on Inviscid $M_\infty = .675$ Convergence. 4 Grids: (a) No Forcing Terms (b) Forcing Terms (c) Summary. 5 Grids: (d) No Forcing Terms (e) Forcing Terms (f) Summary

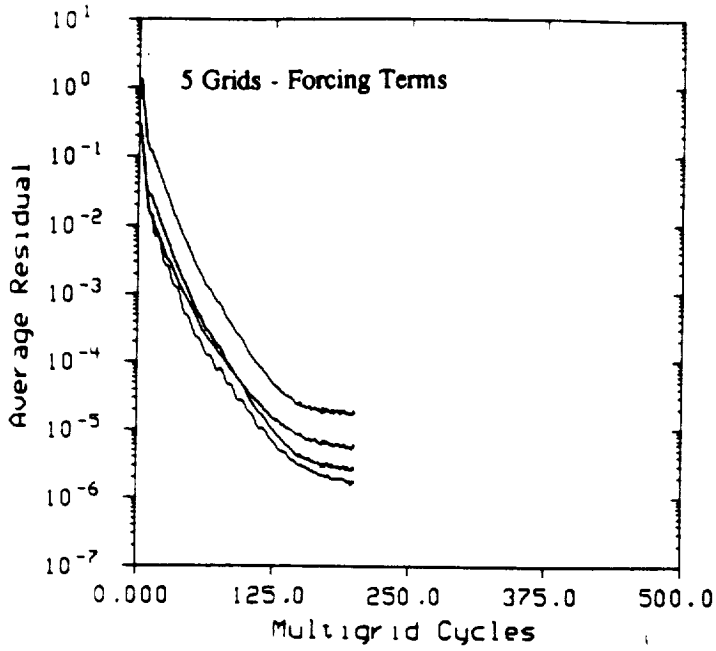


c)

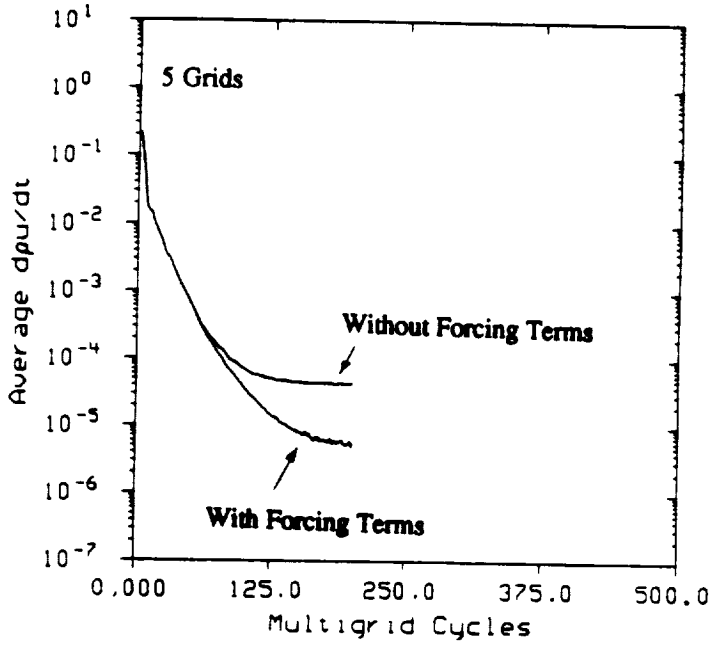


d)

Figure 7.7. Continued.

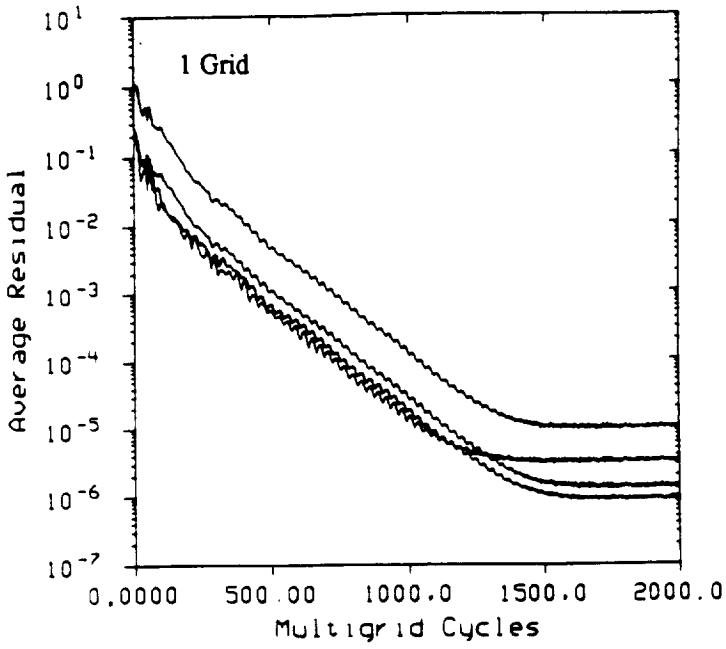


e)

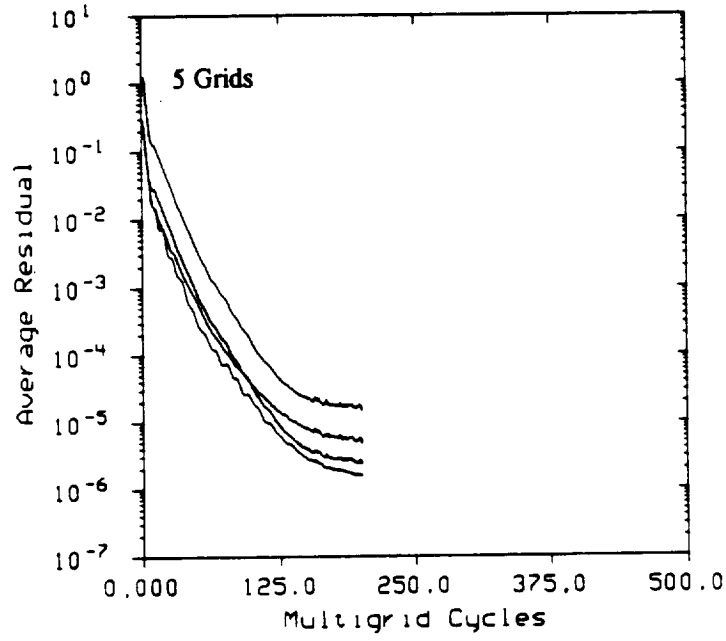


f)

Figure 7.7. Continued.

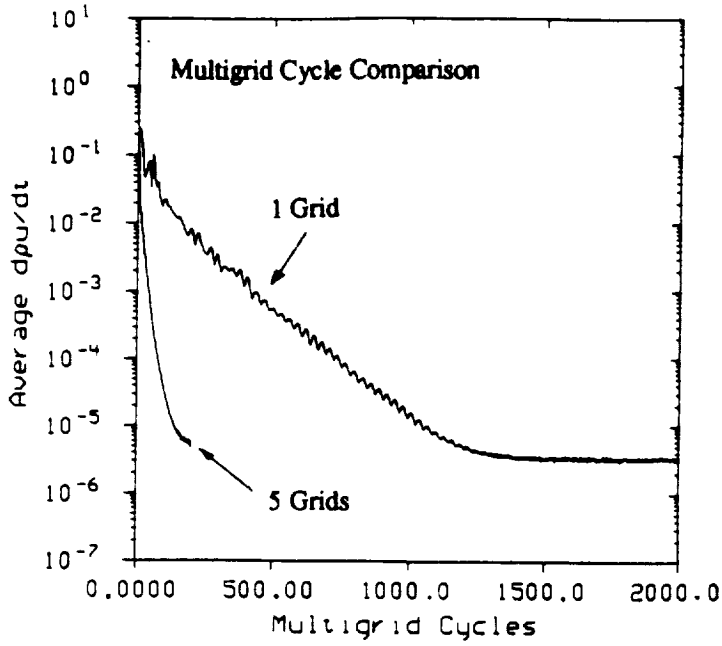


a)

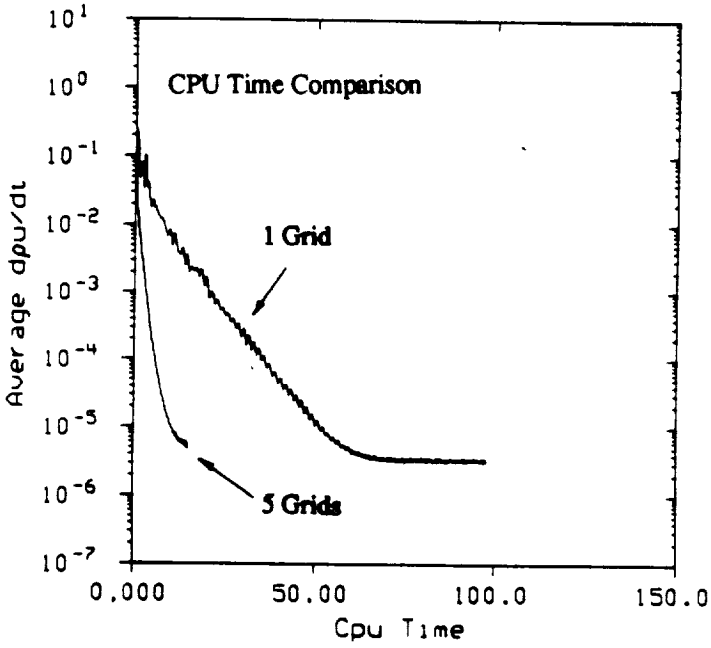


b)

Figure 7.8. Multigrid Efficiency For Inviscid, $M_\infty = .675$ Case. (a) 1 Grid (b) 5 Grids (c) 1 and 5 Grids Multigrid Cycle Comparison (d) 1 and 5 Grids CPU Time Comparison

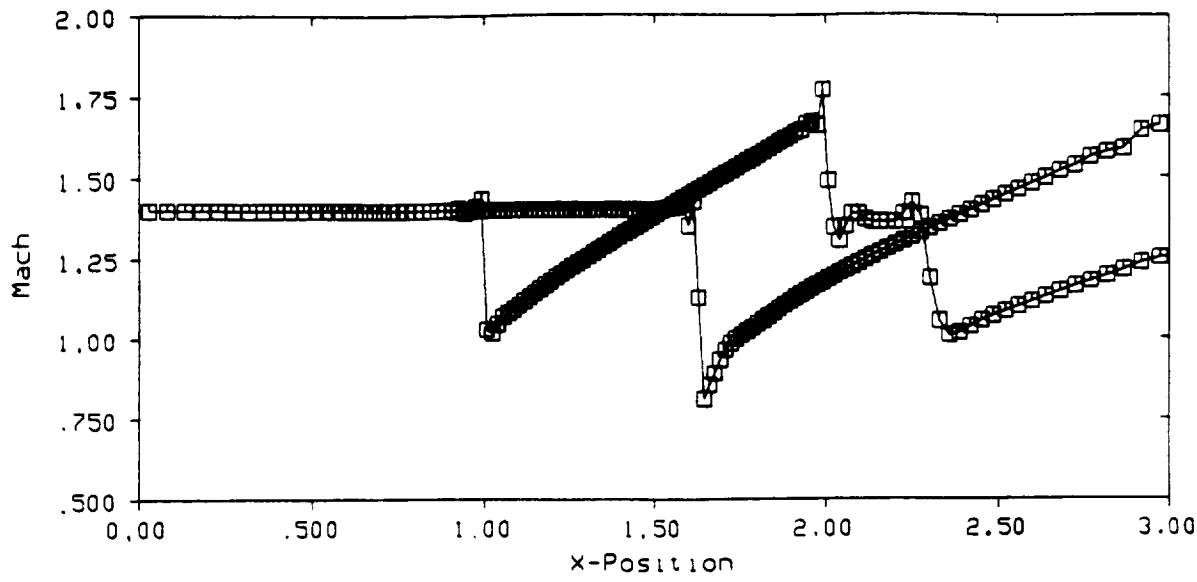


c)

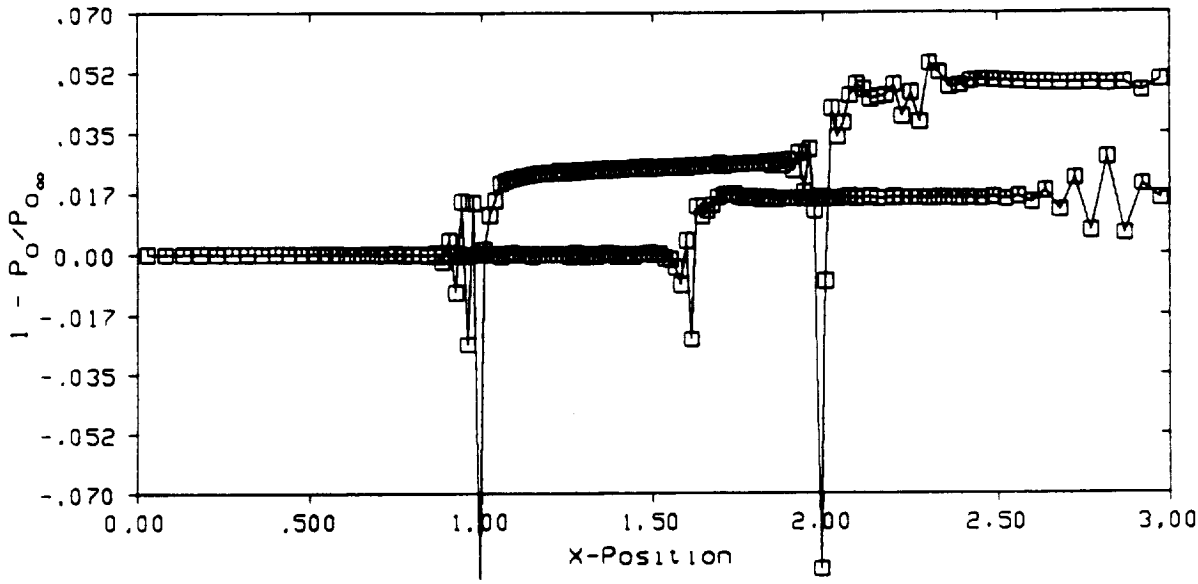


d)

Figure 7.8. Continued.

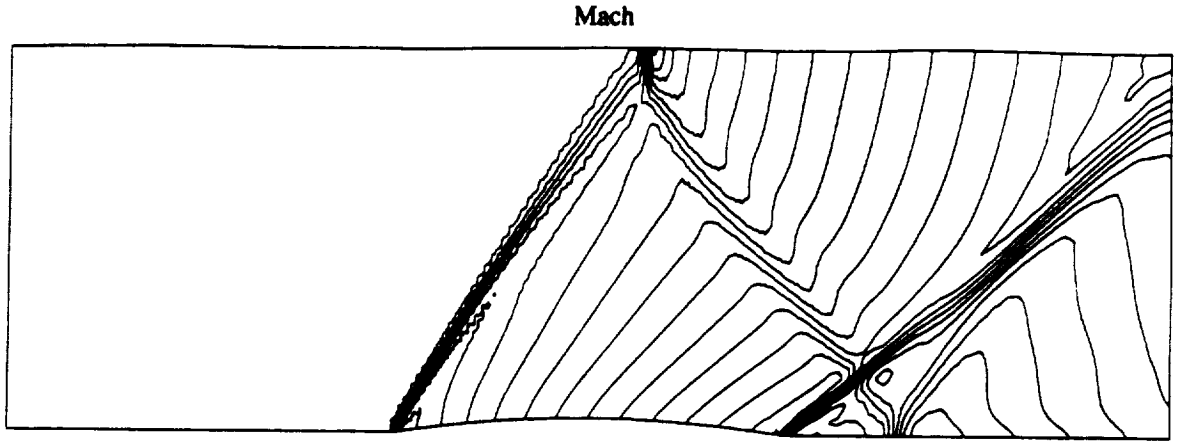


a)

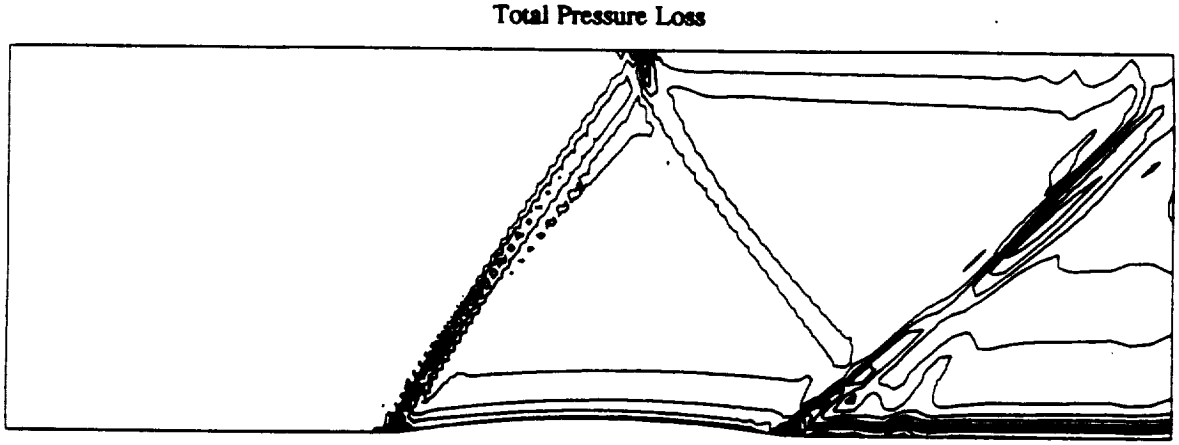


b)

Figure 7.9. $M_\infty = 1.4$ Inviscid Channel Flow Solution. (a) Surface Mach Number (b) Surface Total Pressure Loss (c) Mach Contours, $\Delta M = .053$ (d) Total Pressure Loss Contours, $\Delta P_o = .0052$

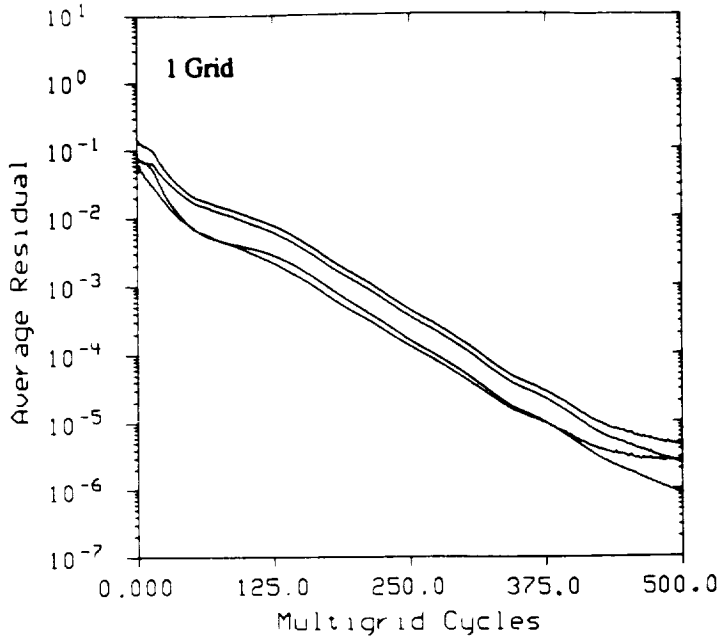


c)

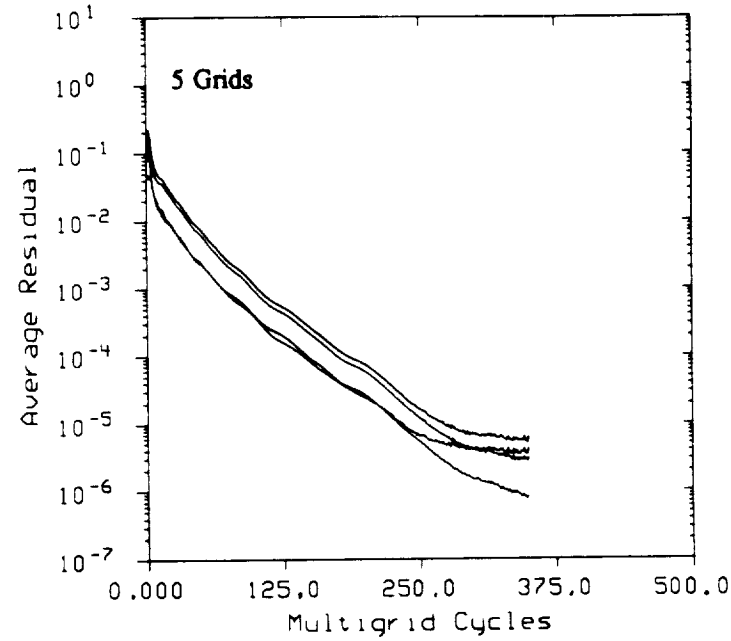


d)

Figure 7.9. Continued.

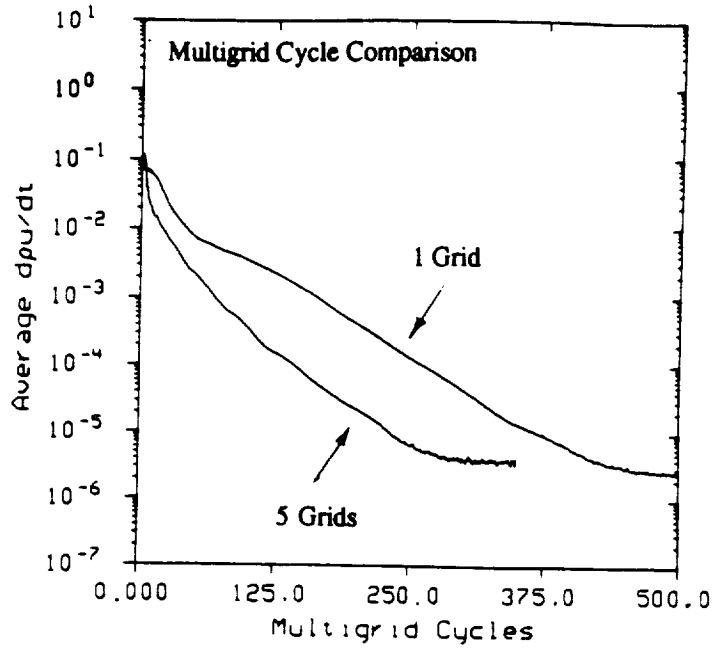


a)

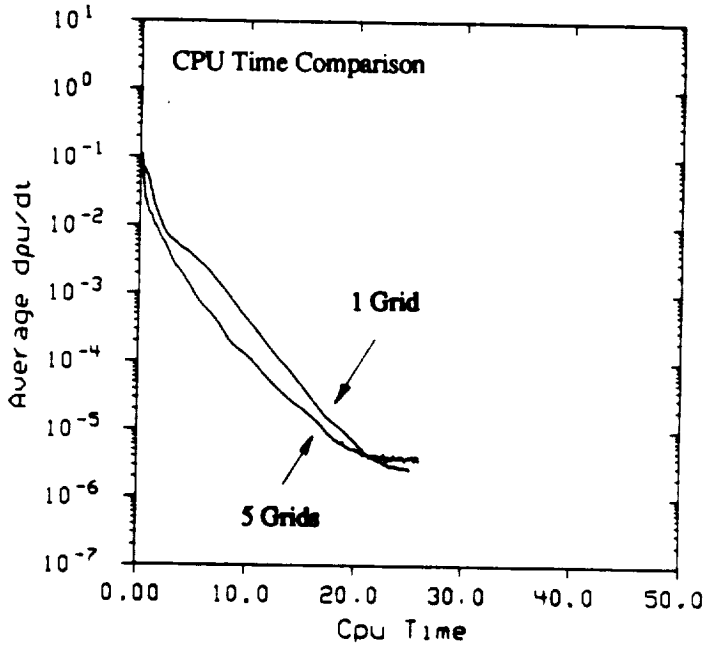


b)

Figure 7.10. Multigrid Efficiency For Inviscid, $M_\infty = 1.4$ Case. (a) 1 Grid (b) 5 Grids (c) 1 and 5 Grids Multigrid Cycle Comparison (d) 1 and 5 Grids CPU Time Comparison



c)



d)

Figure 7.10. Continued.

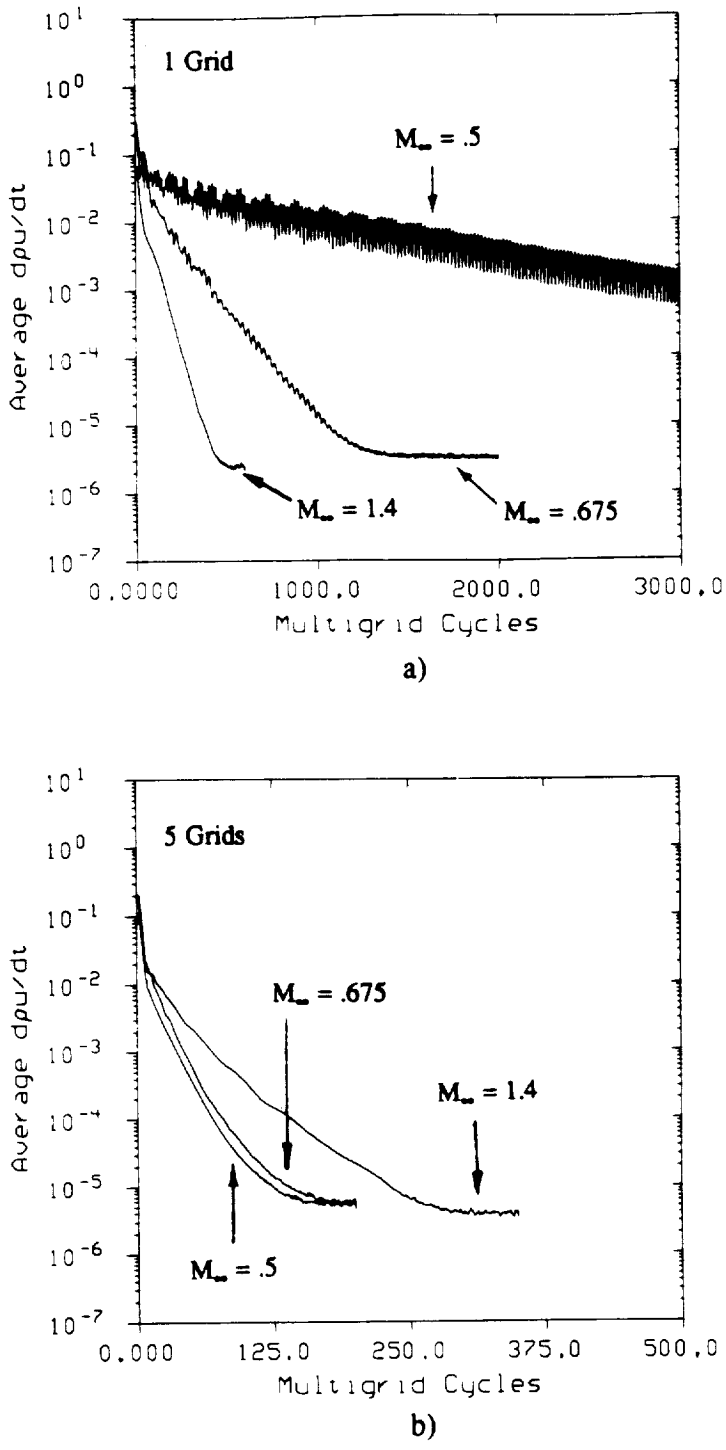


Figure 7.11. Effect Of Mach Number On Convergence For Inviscid Channel Flow. (a) 1 Grid (b) 5 Grids

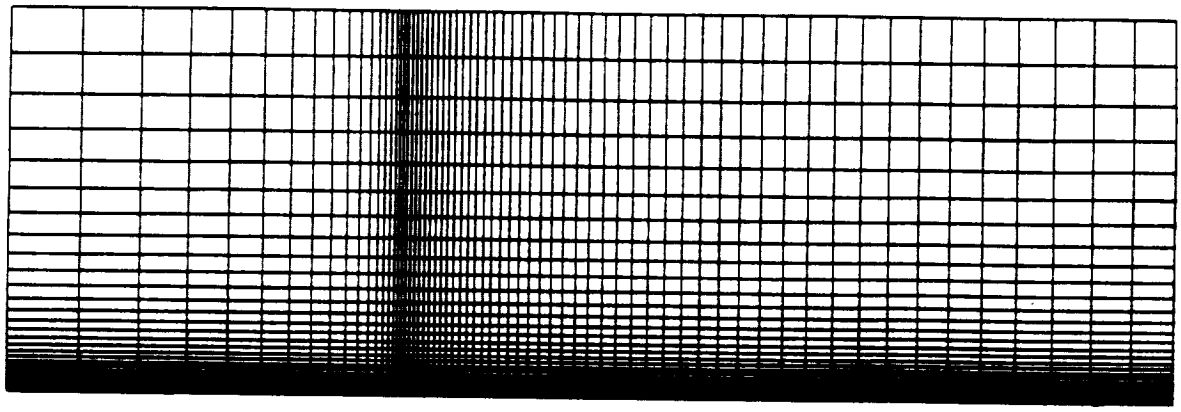
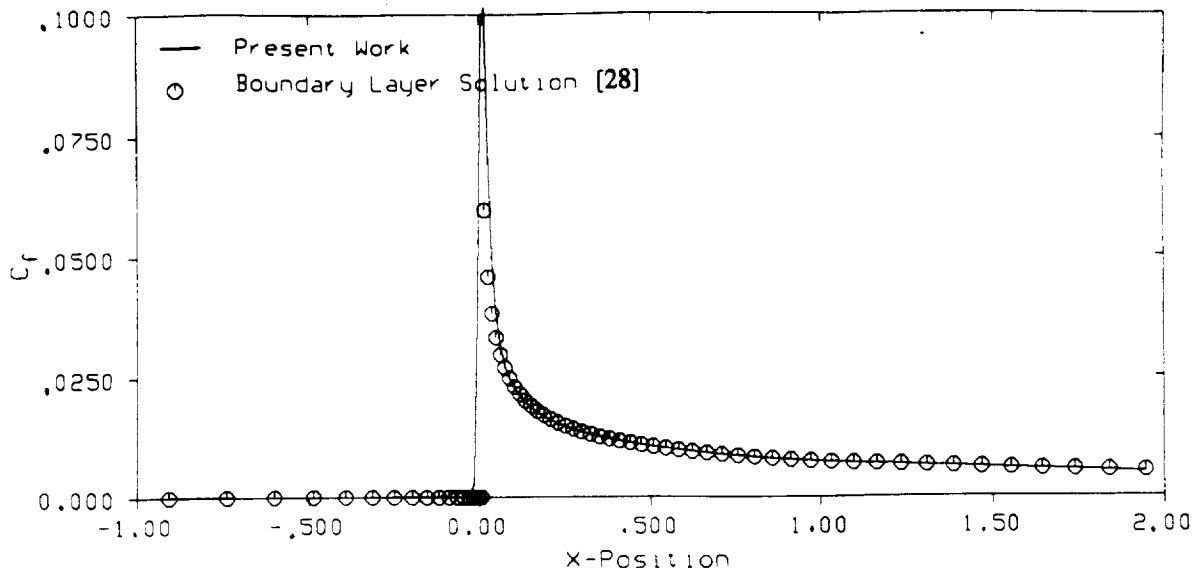
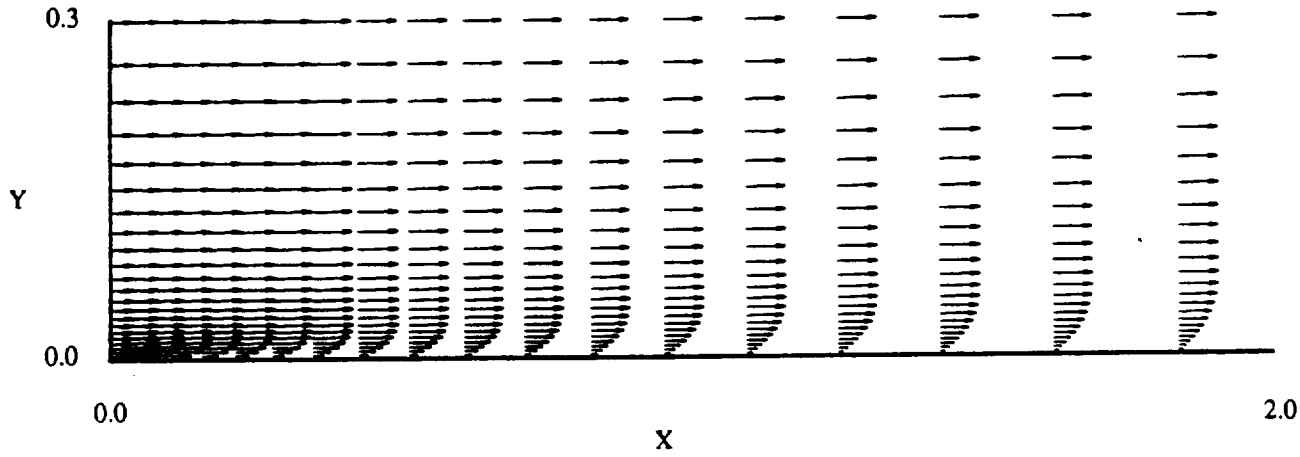


Figure 7.12. 65x33 Flat Plate Computational Mesh

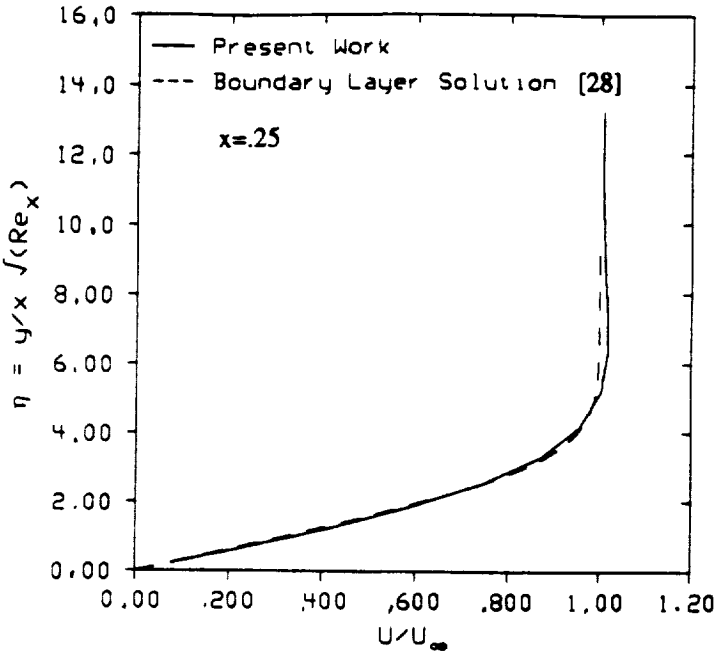


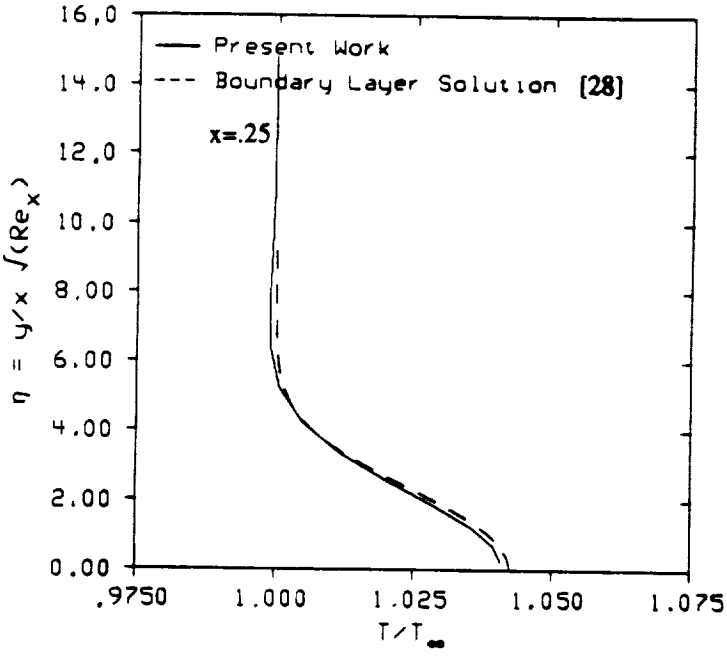
a)



b)

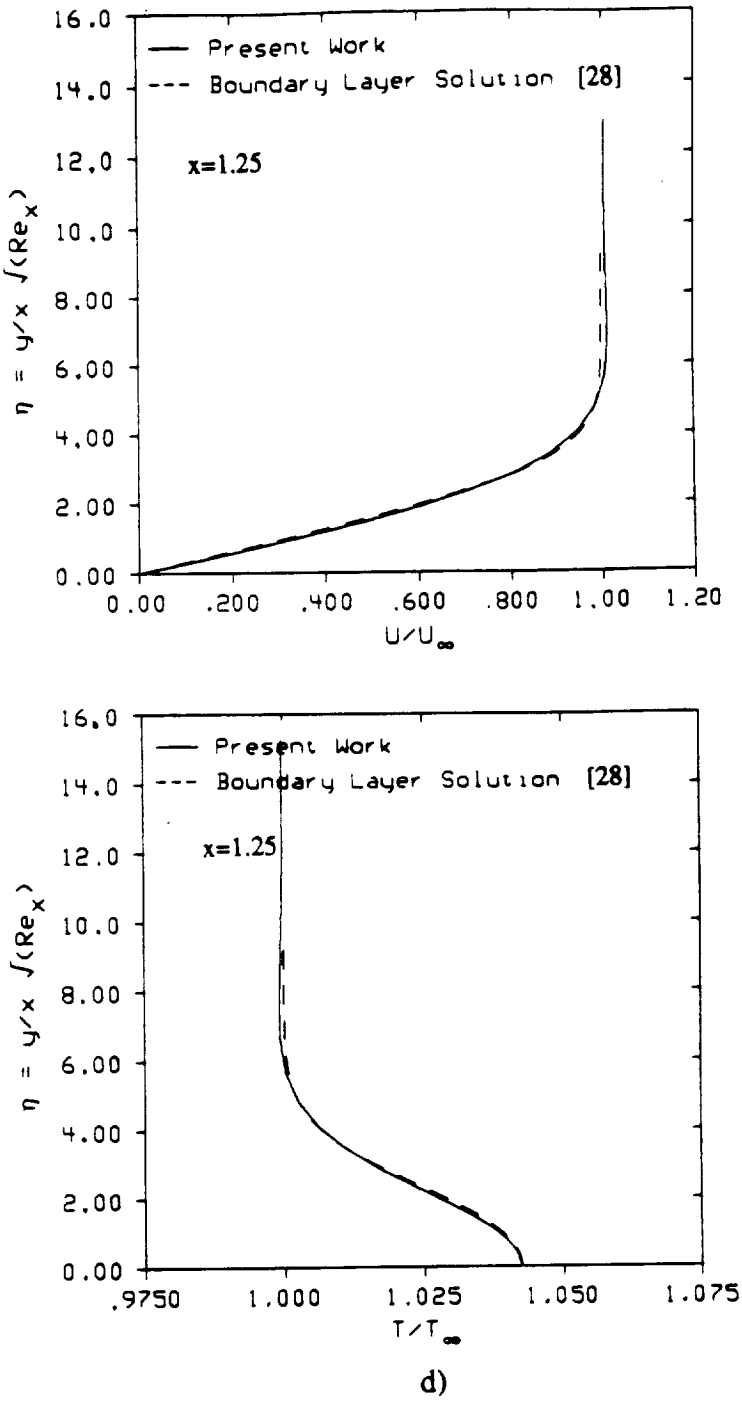
Figure 7.13. Viscous Flat Plate Solution; $M_\infty = .5$, $Re_x)_{x=1} = 8000$. (a) Skin Friction Coefficient (b) Velocity Vectors (c) Velocity and Temperature Profiles at x=.25 (f) Velocity and Temperature Profiles at X=1.25





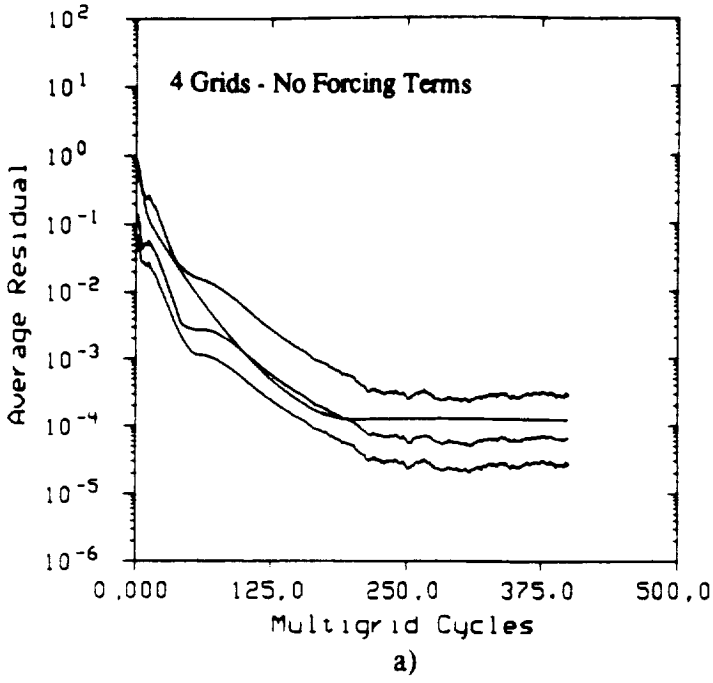
c)

Figure 7.13. Continued.

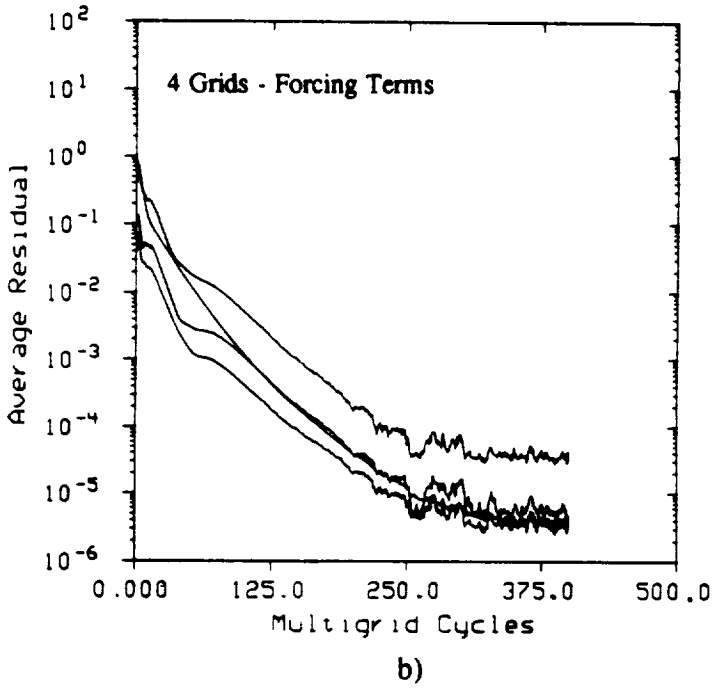


d)

Figure 7.13. Continued.

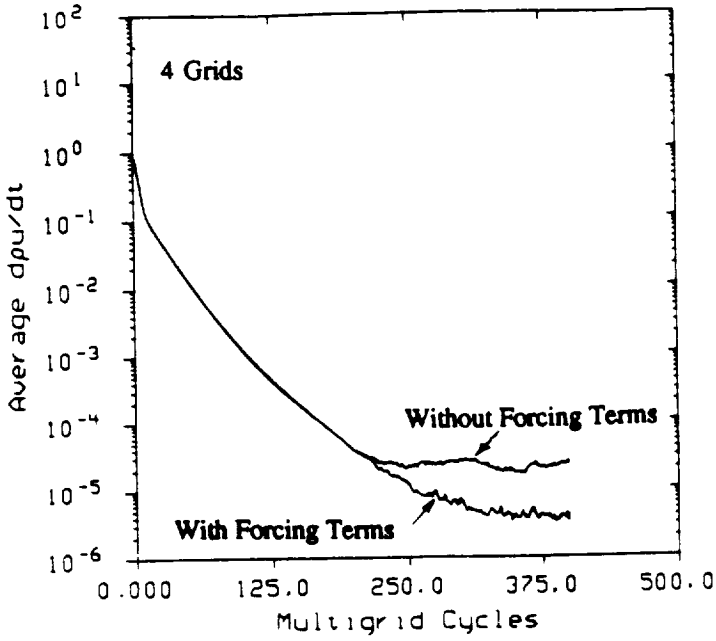


a)

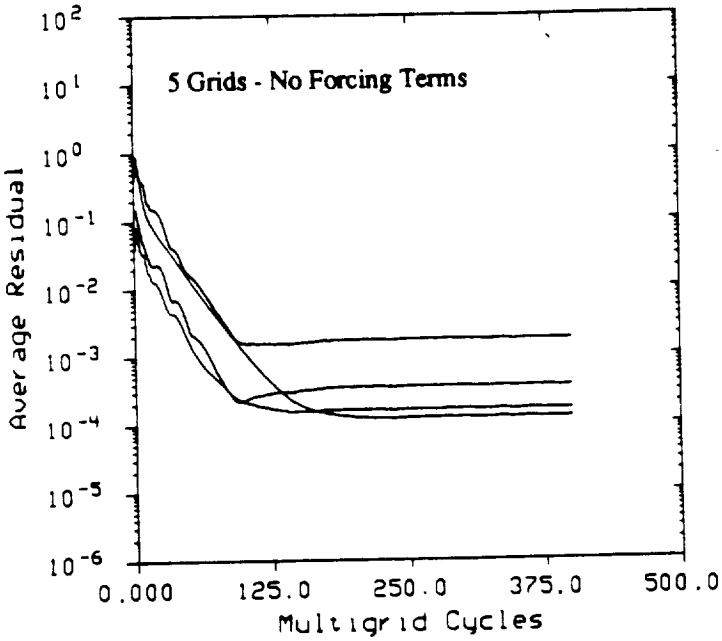


b)

Figure 7.14. Effect of Boundary Forcing Terms on Viscous Flat Plate Convergence. 4 Grids: (a) No Forcing Terms (b) Forcing Terms (c) Summary. 5 Grids: (d) No Forcing Terms (e) Forcing Terms (f) Summary

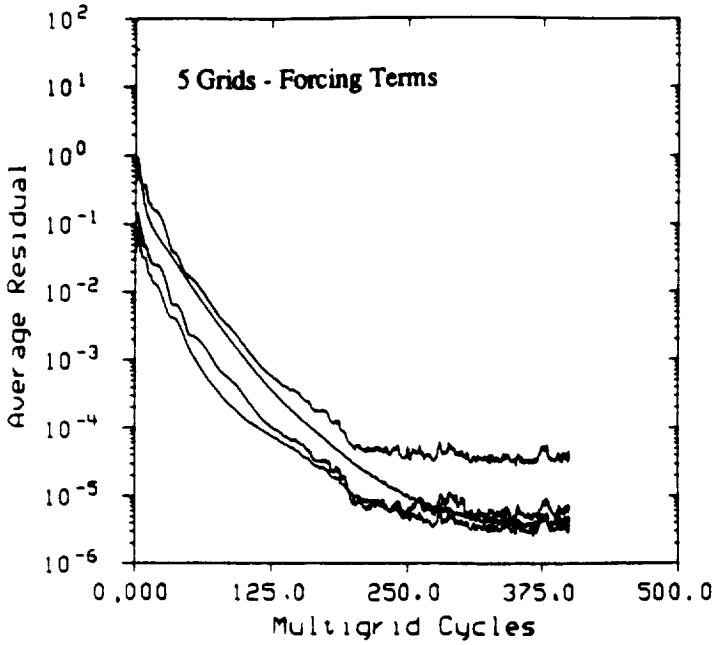


c)

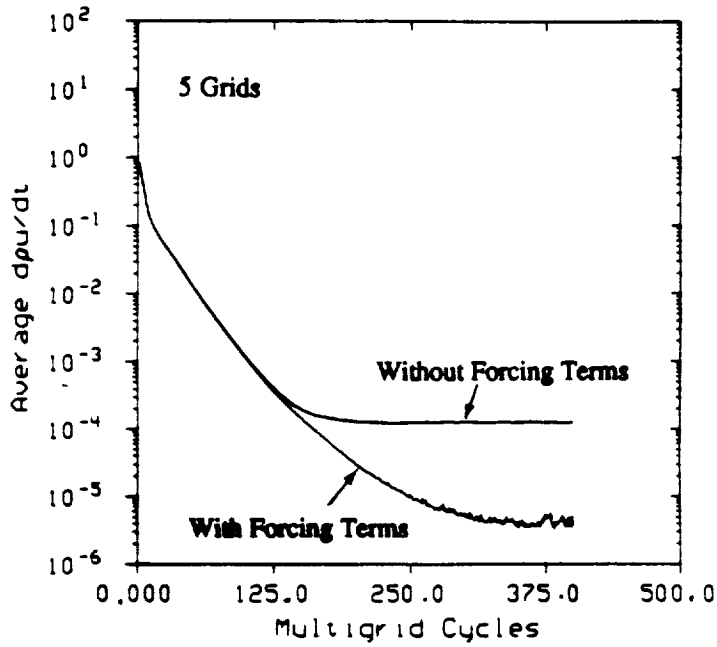


d)

Figure 7.14. Continued.

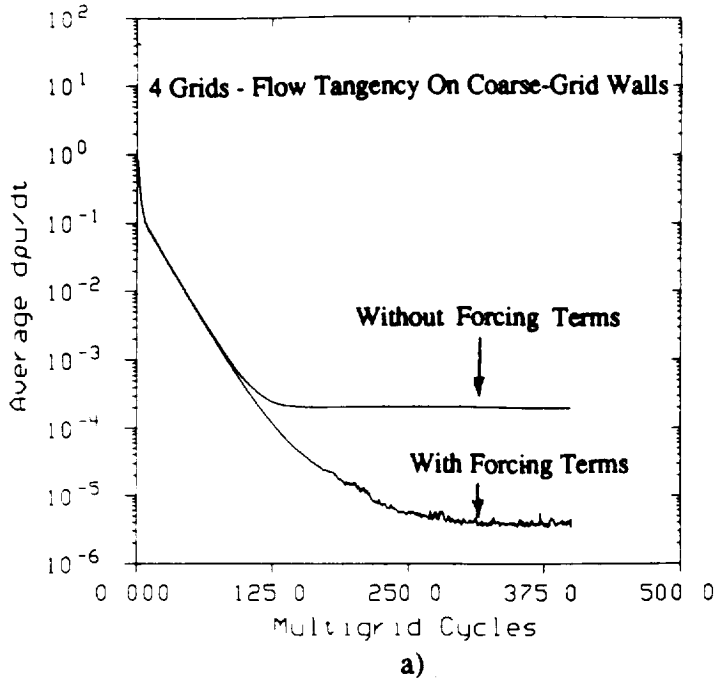


e)

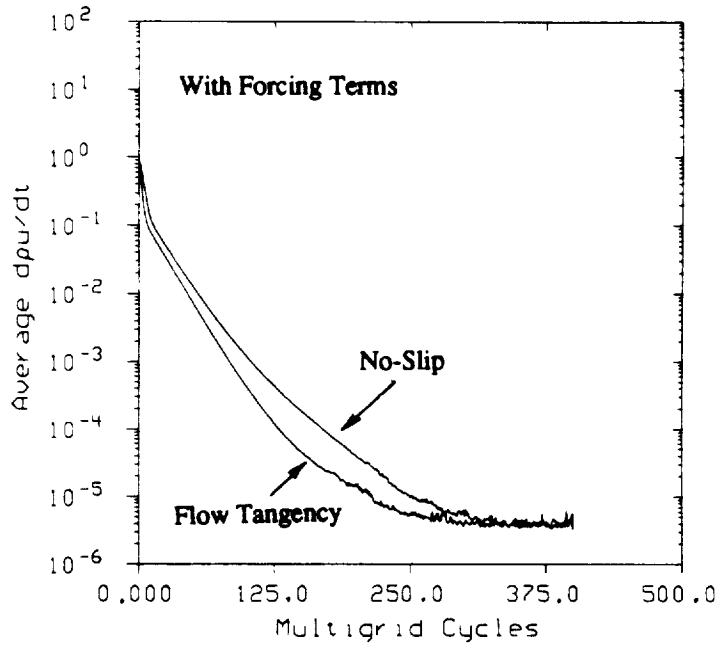


f)

Figure 7.14. Continued.

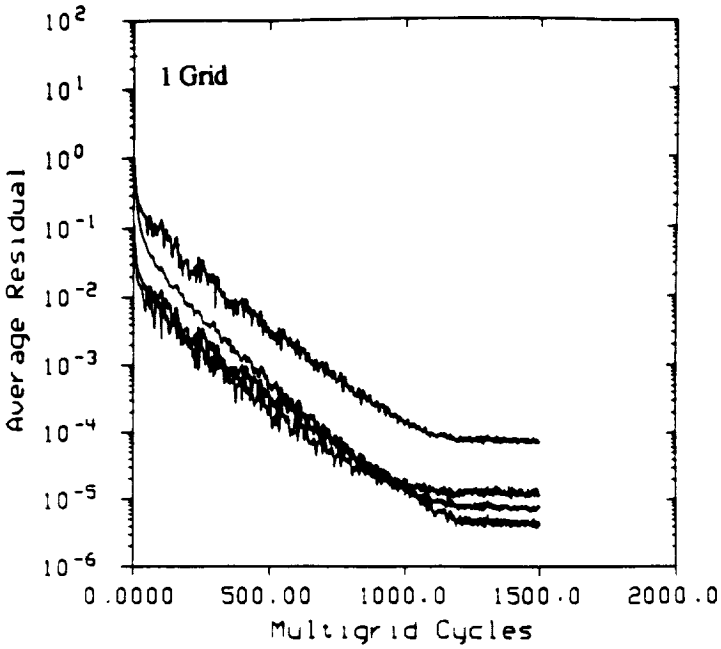


a)

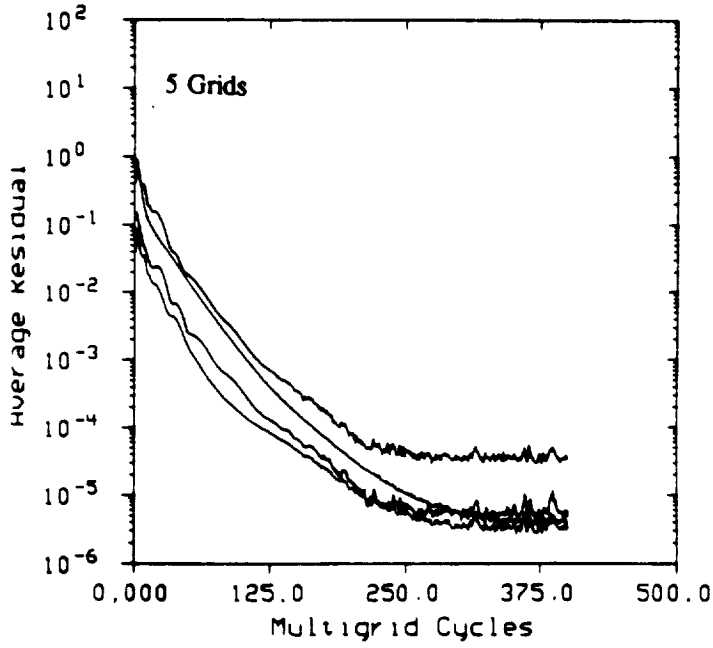


b)

Figure 7.15. Flat Plate Convergence With Flow Tangency Specified On The Coarse Levels. (a) With and Without Forcing Terms (b) Flow Tangency With Forcing Terms versus No-Slip Boundaries With Forcing Terms

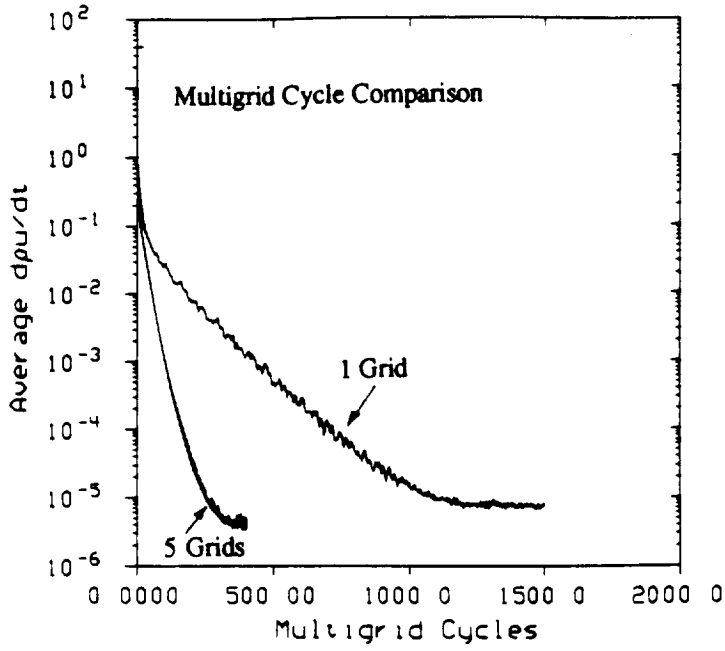


a)

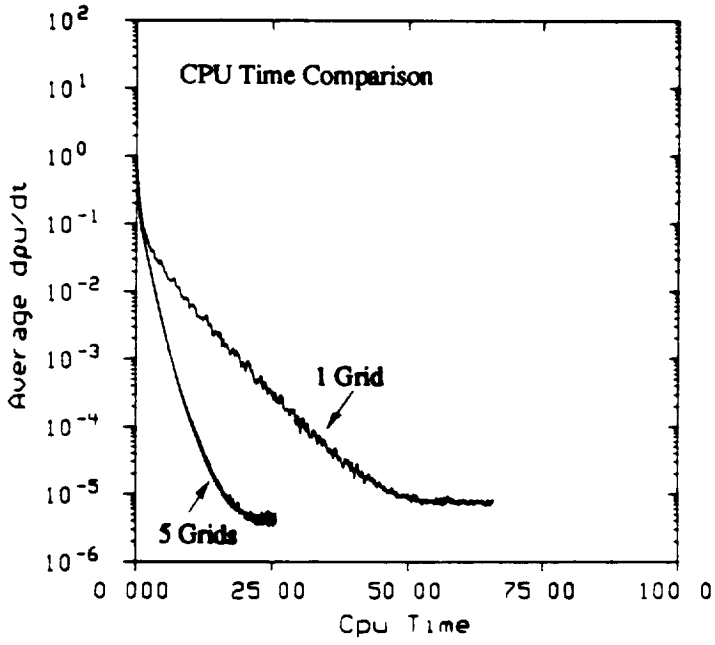


b)

Figure 7.16. Multigrid Efficiency For Viscous Flat Plate. (a) 1 Grid (b) 5 Grids (c) 1 and 5 Grids Multigrid Cycle Comparison (d) 1 and 5 Grids CPU Time Comparison



c)



d)

Figure 7.16. Continued.

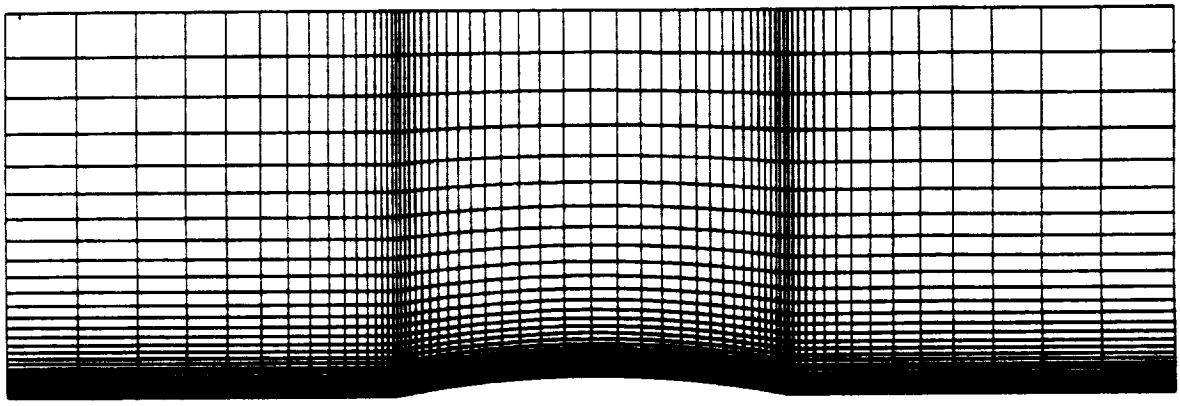
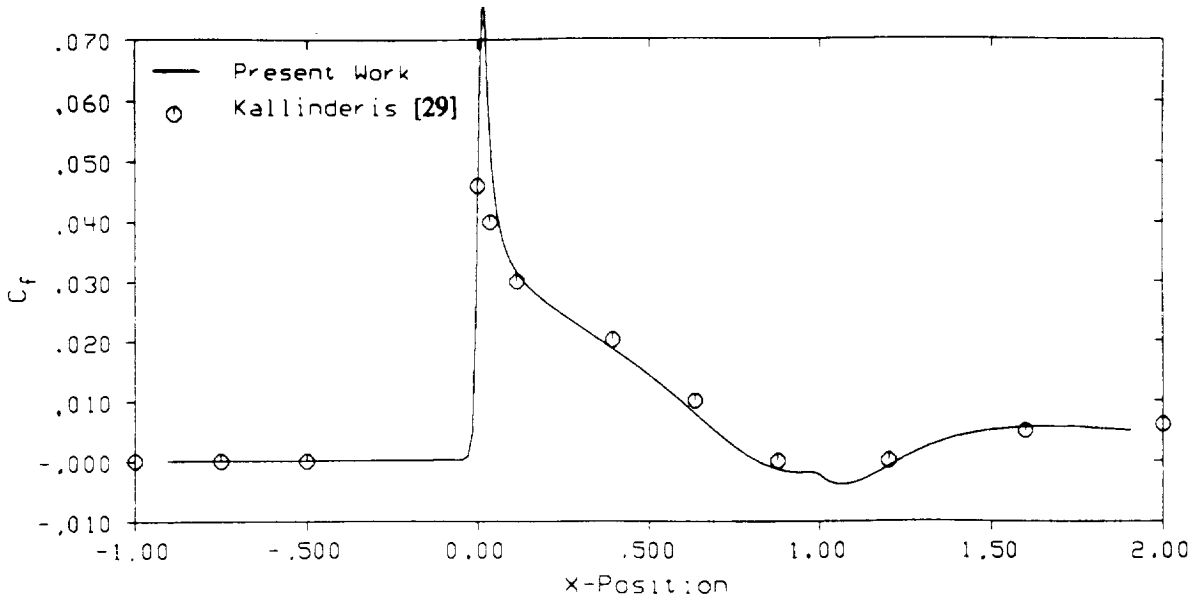
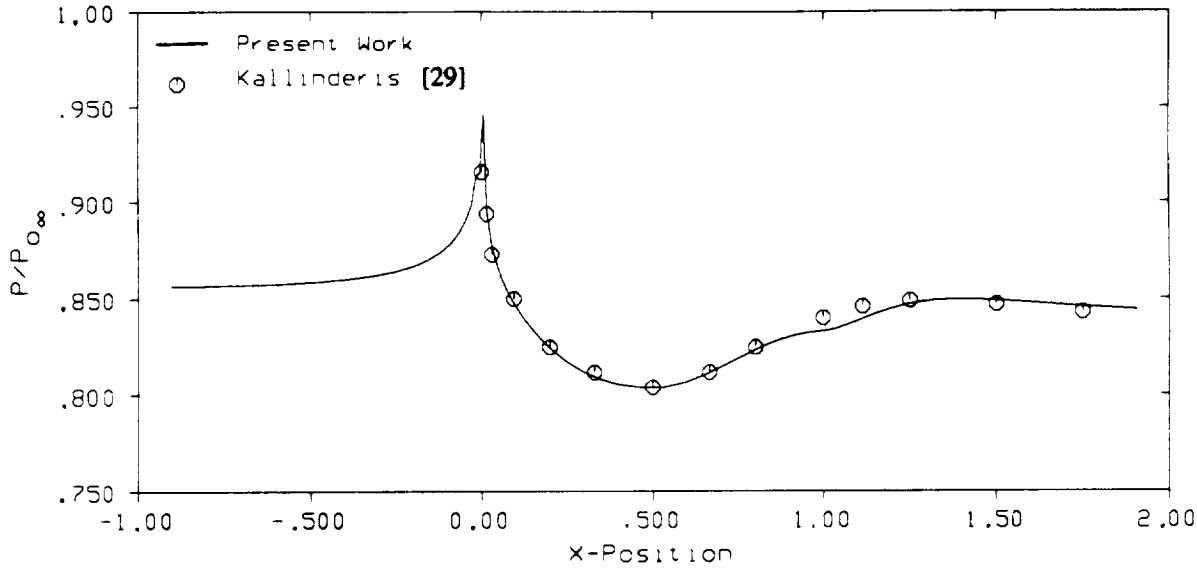


Figure 7.17. 65x33 Computational Mesh for Viscous Flow Over a 5% Circular Arc Bump

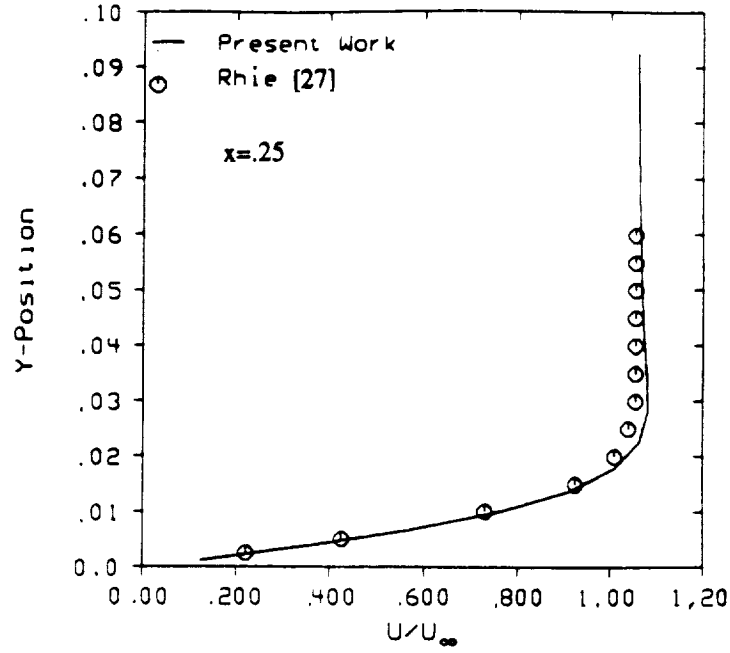


a)

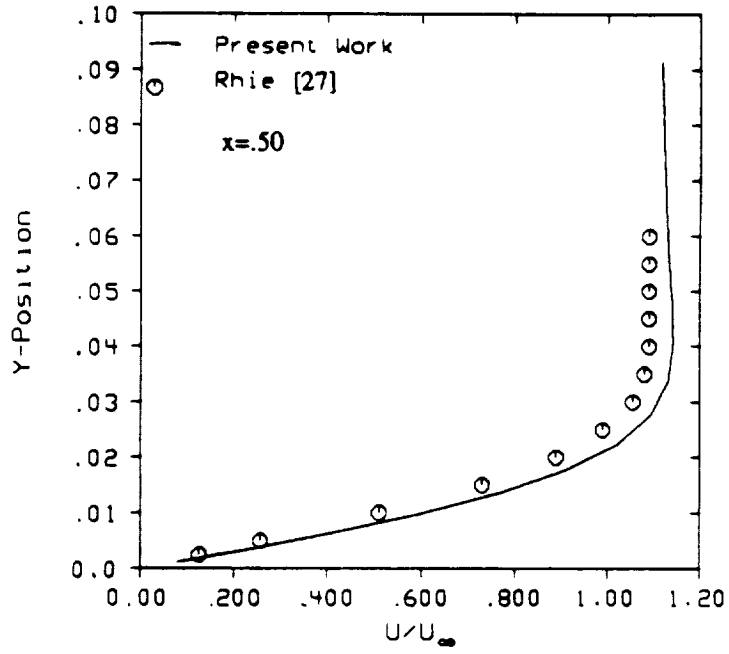


b)

Figure 7.18. Viscous Channel Flow Solutions Over a 5% Circular Arc Bump. (a) Skin Friction Coefficient (b) Normalized Static Pressure Distribution. Velocity Profiles: (c) X=1.25 (d) X=1.50 (e) X=1.75. (f) Mach Contours, $\Delta M = .05$ (g) Total Pressure Loss Contours (h) Trailing Edge Velocity Vectors

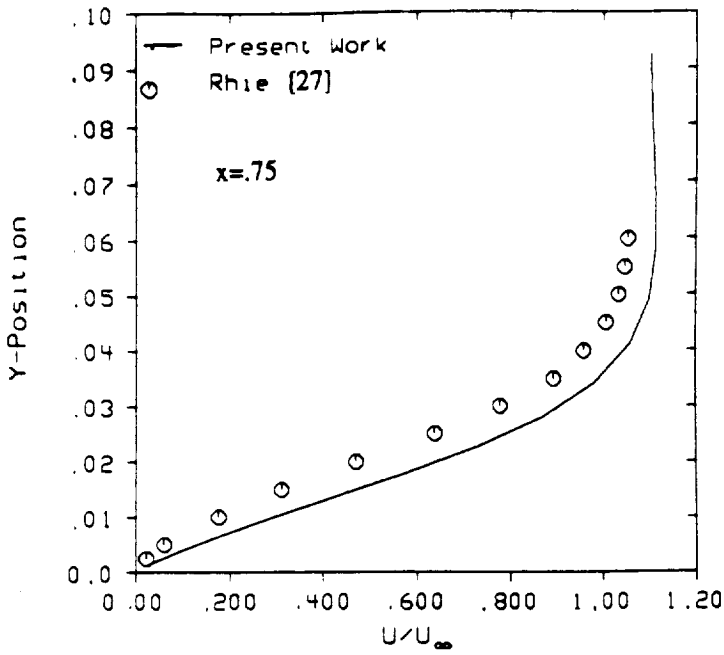


c)

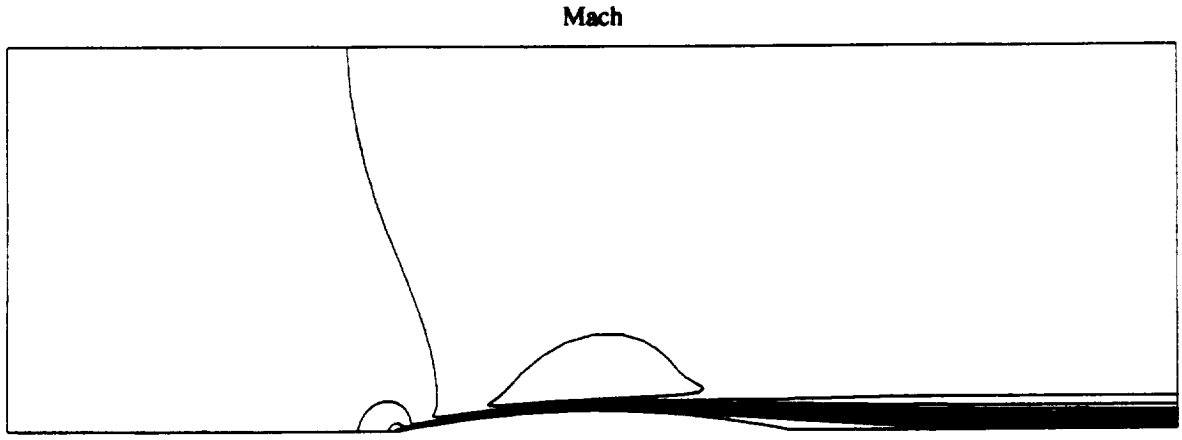


d)

Figure 7.18. Continued.

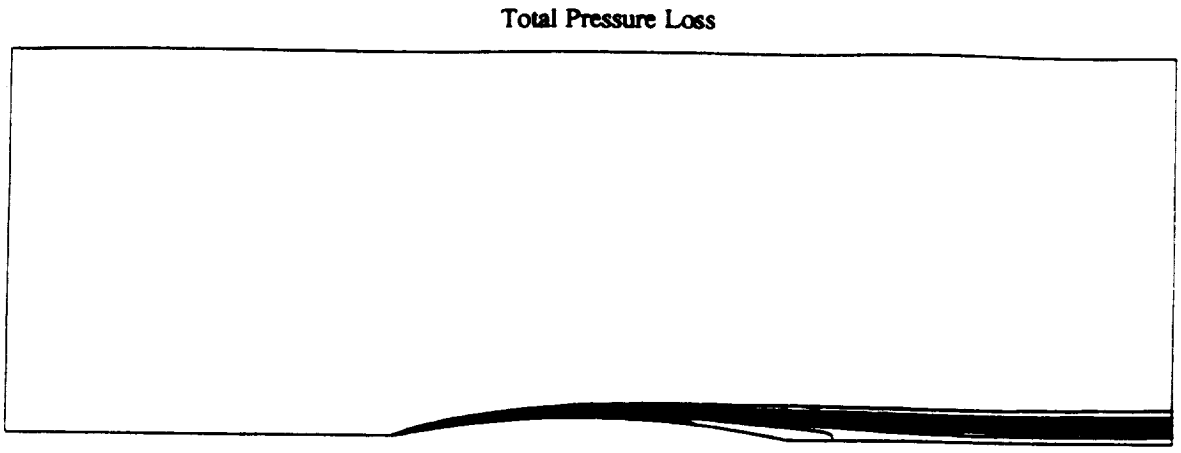


e)

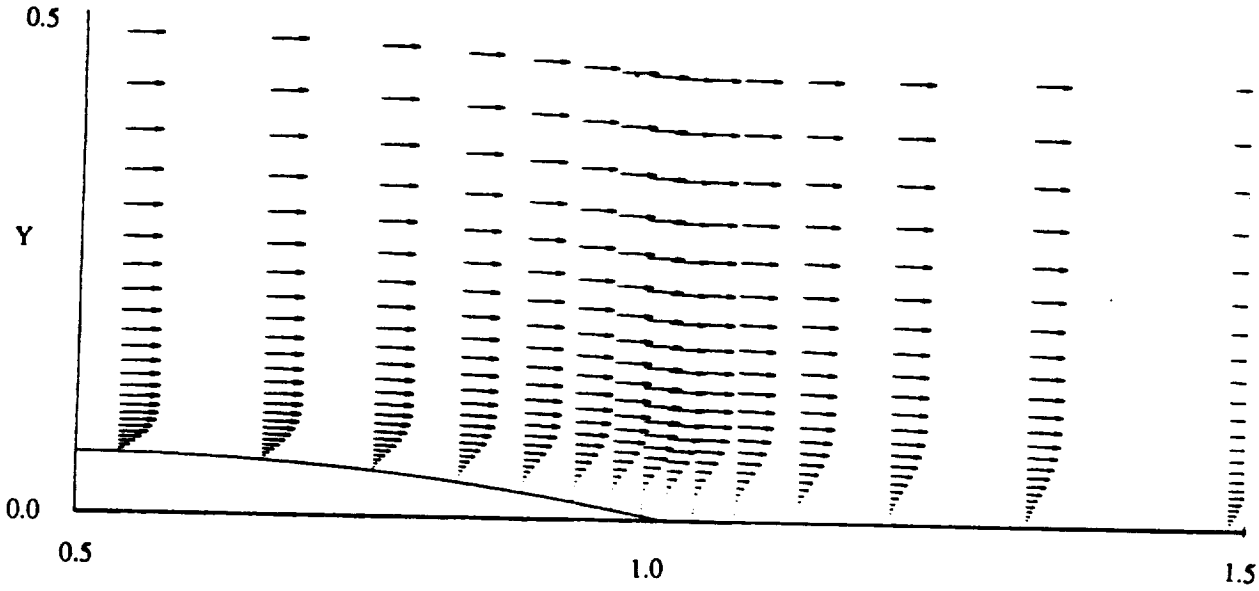


f)

Figure 7.18. Continued.

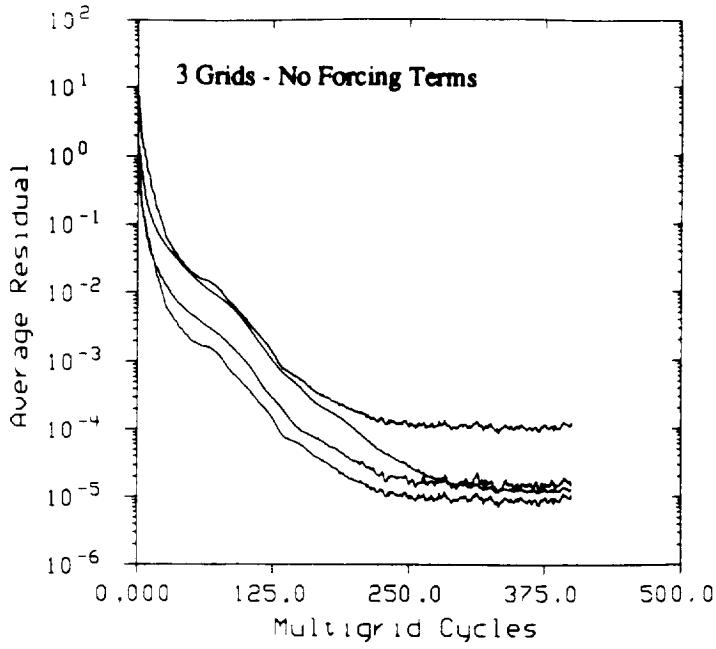


g)

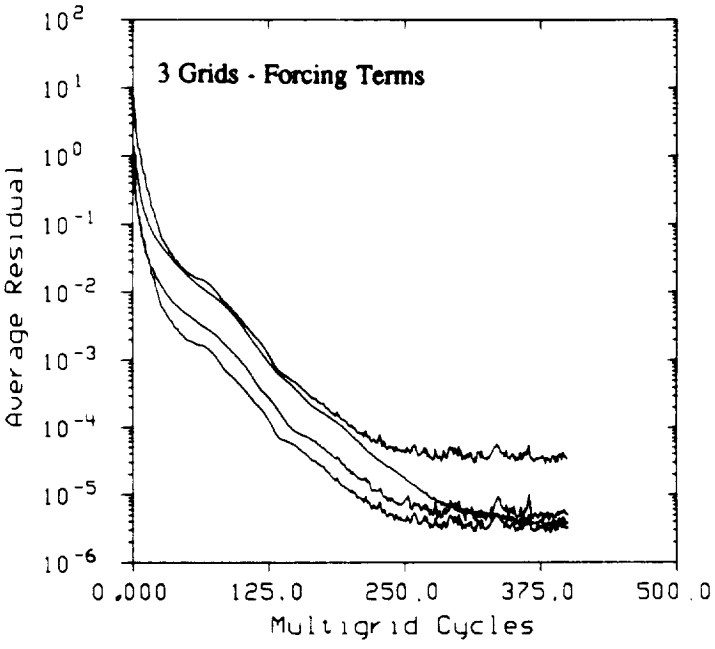


h)

Figure 7.18. Continued.

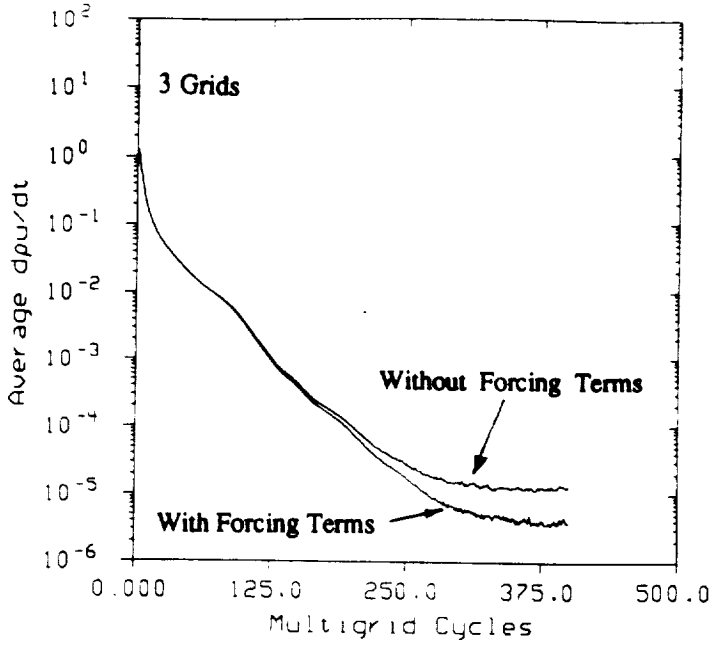


a)

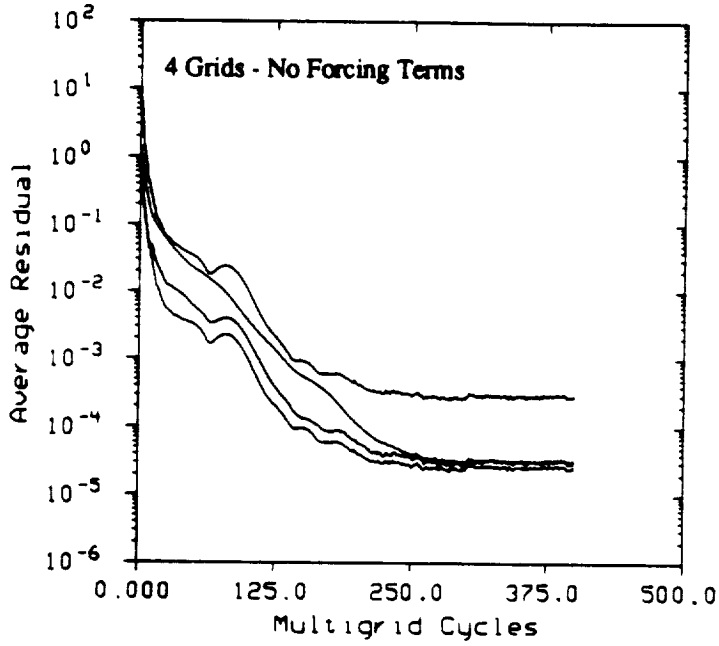


b)

Figure 7.19. Effect of Boundary Forcing Terms on Viscous Circular Arc Convergence. 3 Grids: (a) No Forcing Terms (b) Forcing Terms (c) Summary. 4 Grids: (d) No Forcing Terms (e) Forcing Terms (f) Summary. 5 Grids: (g) No Forcing Terms (h) Forcing Terms (i) Summary

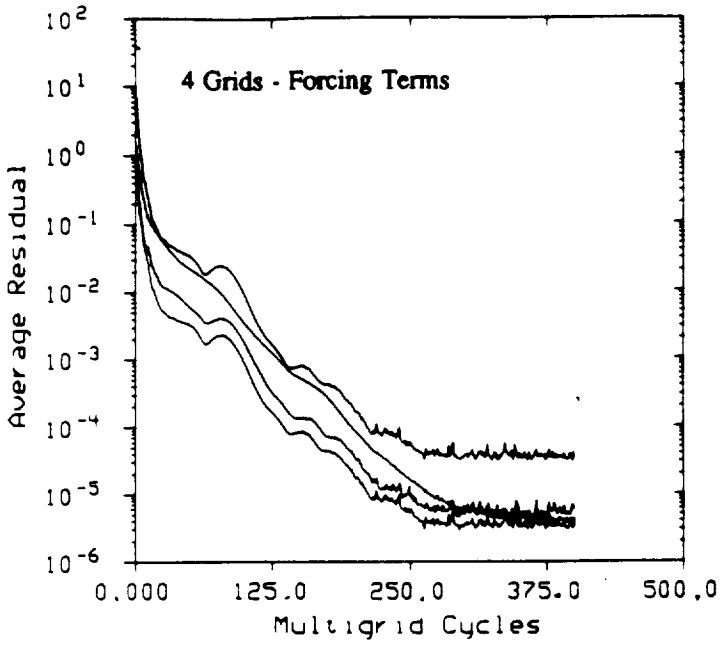


c)

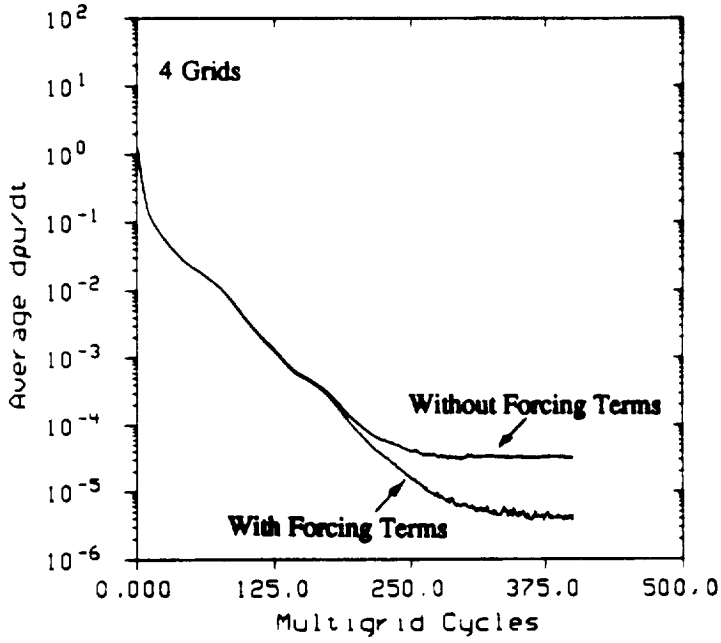


d)

Figure 7.19. Continued.

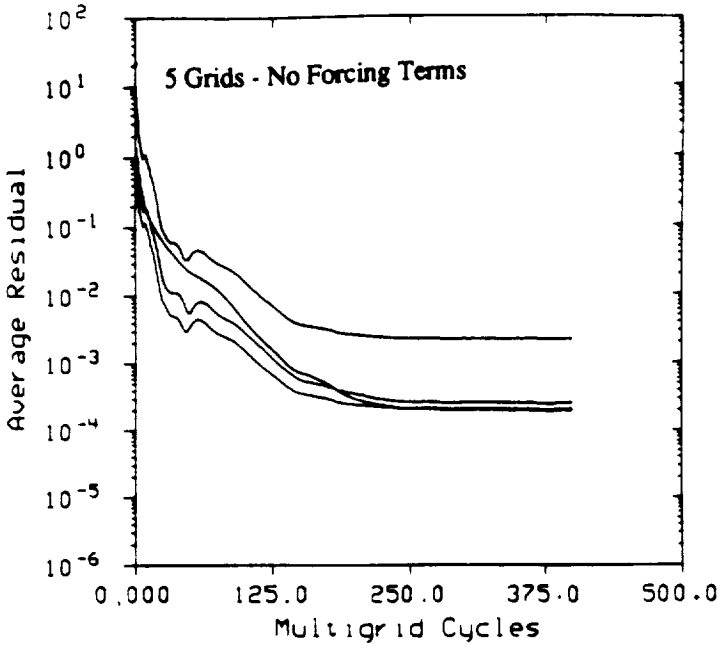


e)

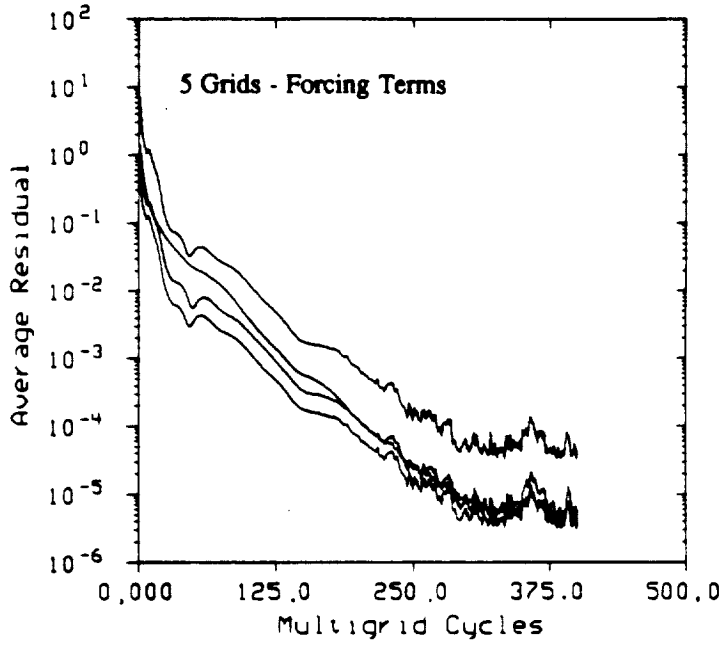


f)

Figure 7.19. Continued.

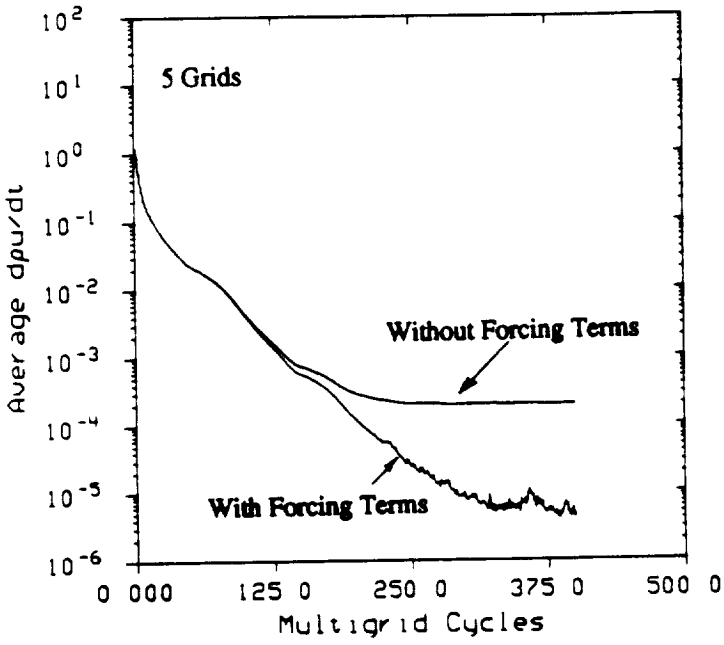


g)



h)

Figure 7.19. Continued.



i)

Figure 7.19. Continued.

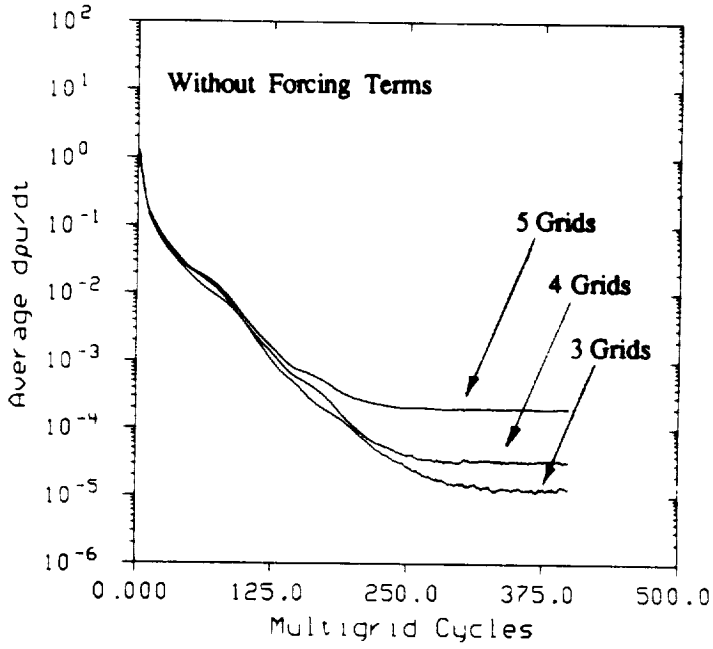
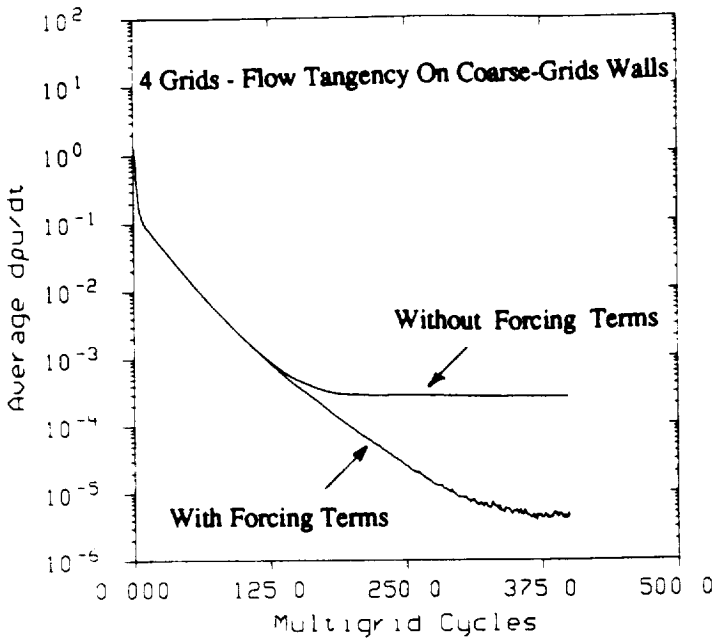
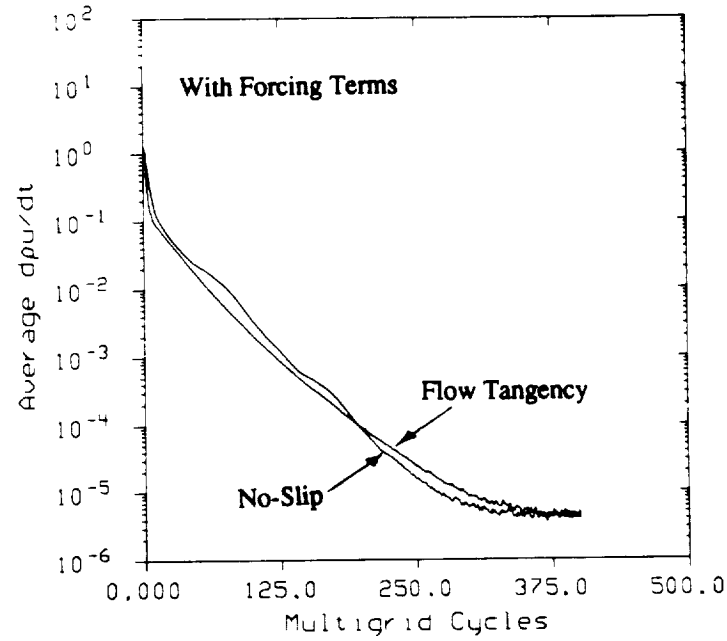


Figure 7.20. Increase In The Influence Of Coarse-Mesh Truncation Error On Fine-Mesh Convergence As The Number Of Grids Increases

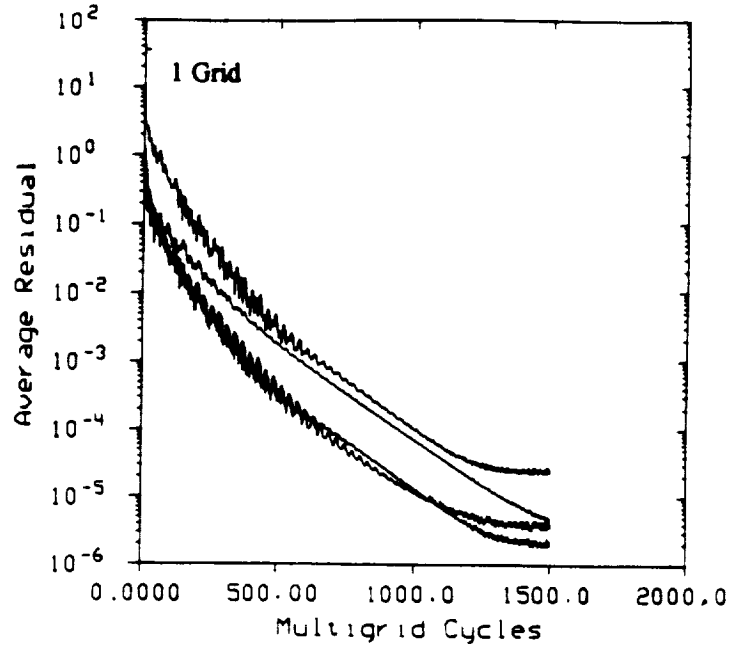


a)

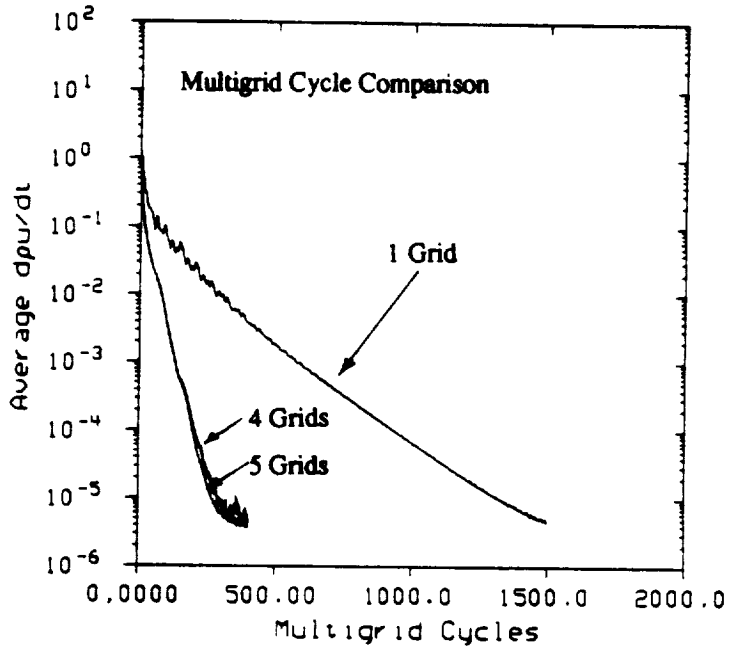


b)

Figure 7.21. Viscous Bump Convergence With Flow Tangency Specified On The Coarse Levels. (a) With and Without Forcing Terms (b) Flow Tangency With Forcing Terms versus No-Slip Boundaries With Forcing Terms

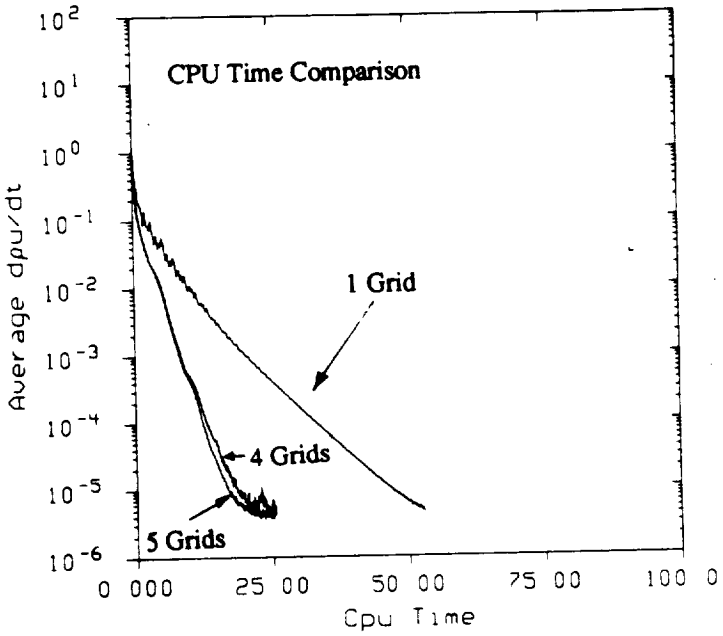


a)



b)

Figure 7.22. Multigrid Efficiency For Viscous Circular Arc. (a) 1 Grid (b) 1, 4 and 5 Grid Multigrid Cycle Comparison (c) 1, 4 and 5 Grid CPU Time Comparison



c)

Figure 7.22. Continued.

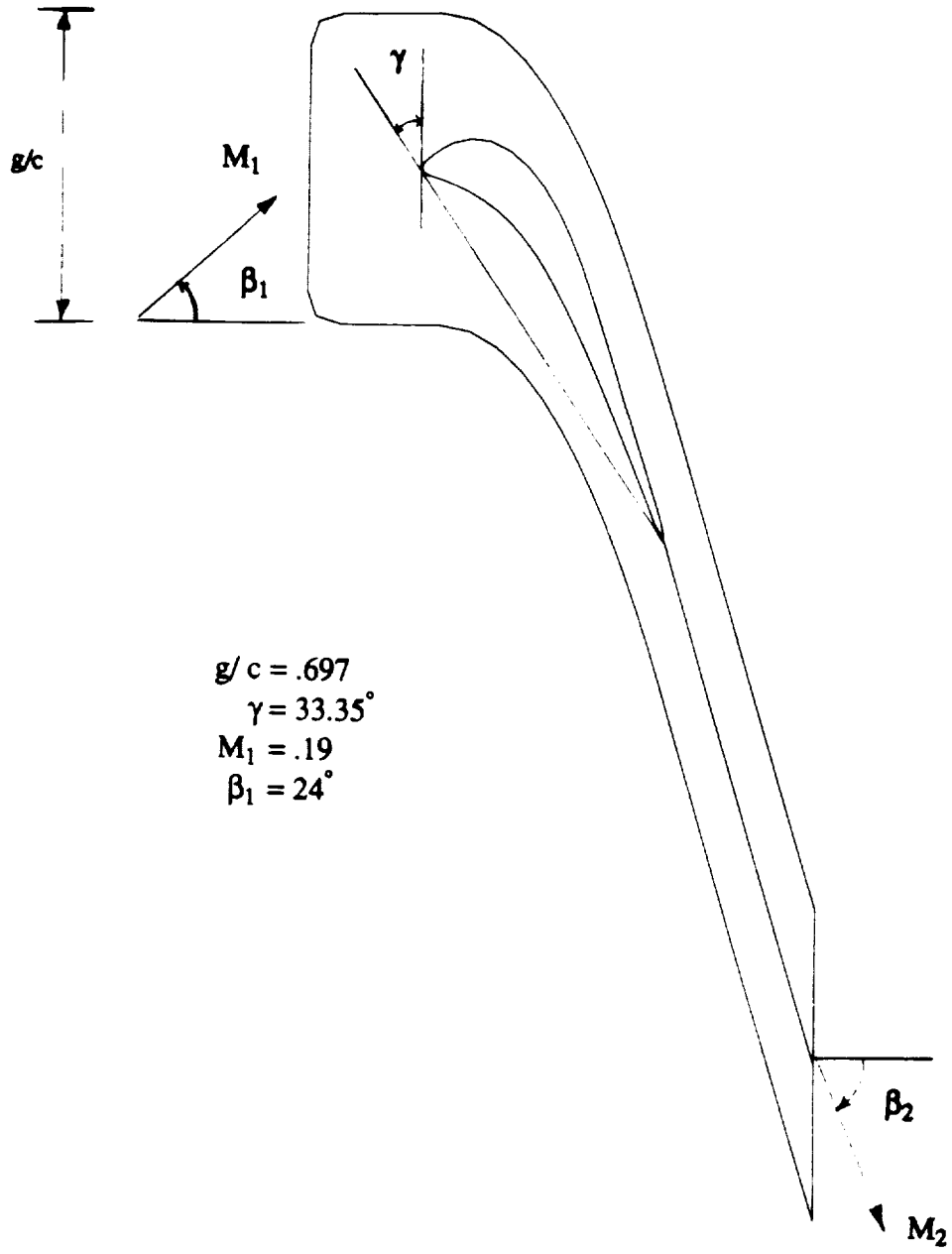
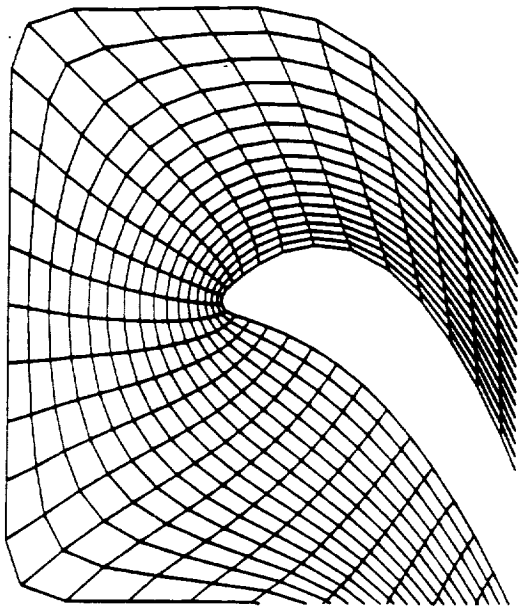


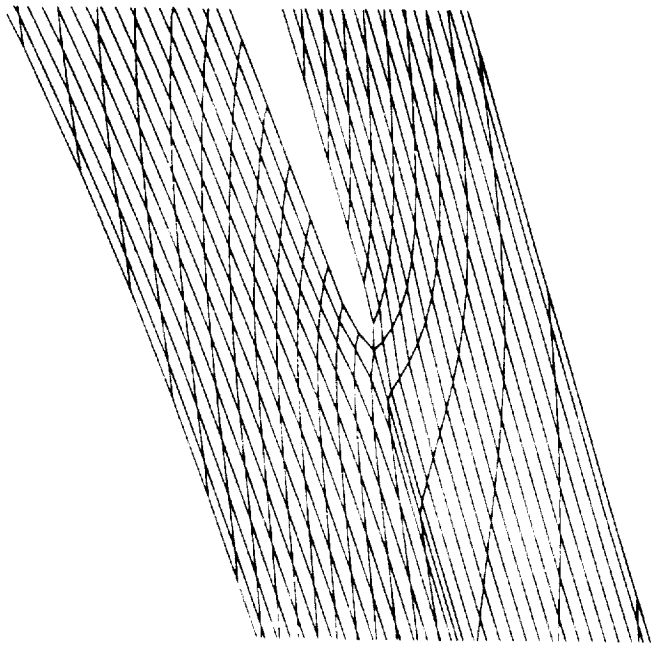
Figure 7.23. Definition of Turbine Cascade Geometric Parameters

a)

Figure 7.24. 73x17 Computational Mesh for Inviscid Turbine Calculations. (a) Global View (b) Leading Edge Detail (c) Trailing Edge Detail

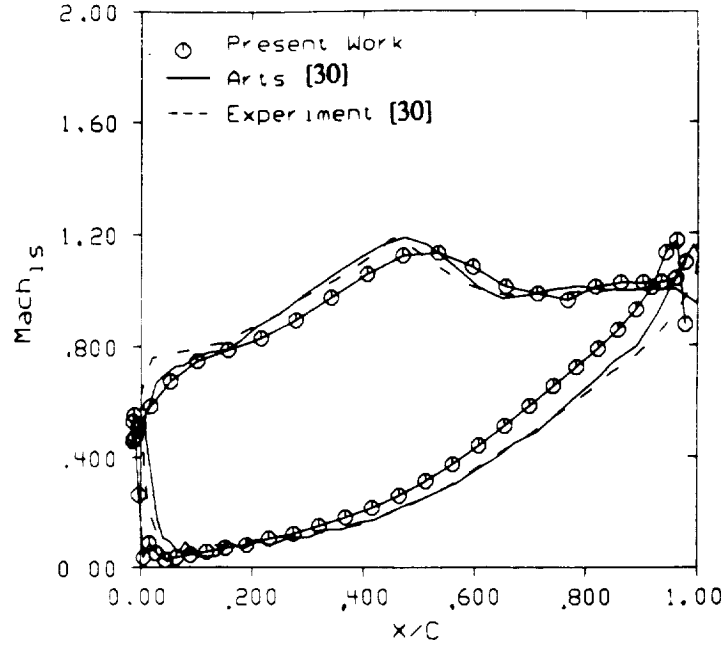


b)

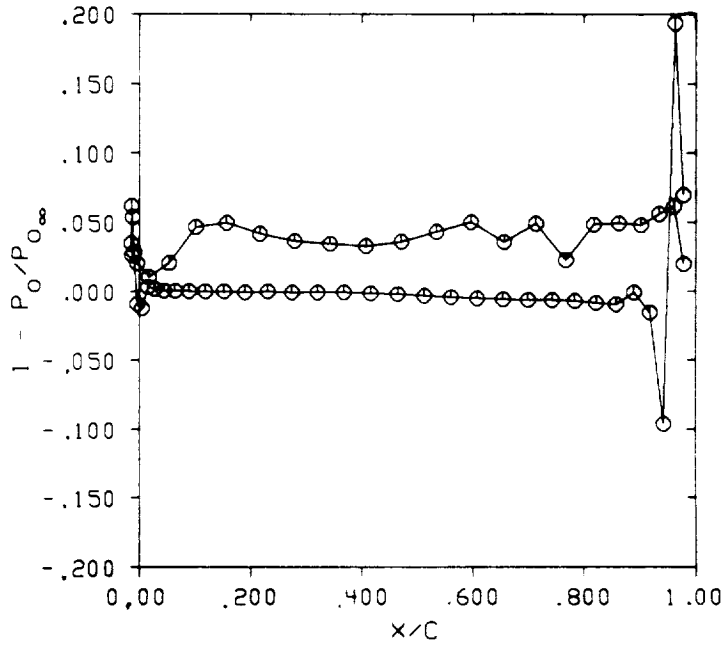


c)

Figure 7.24. Continued.

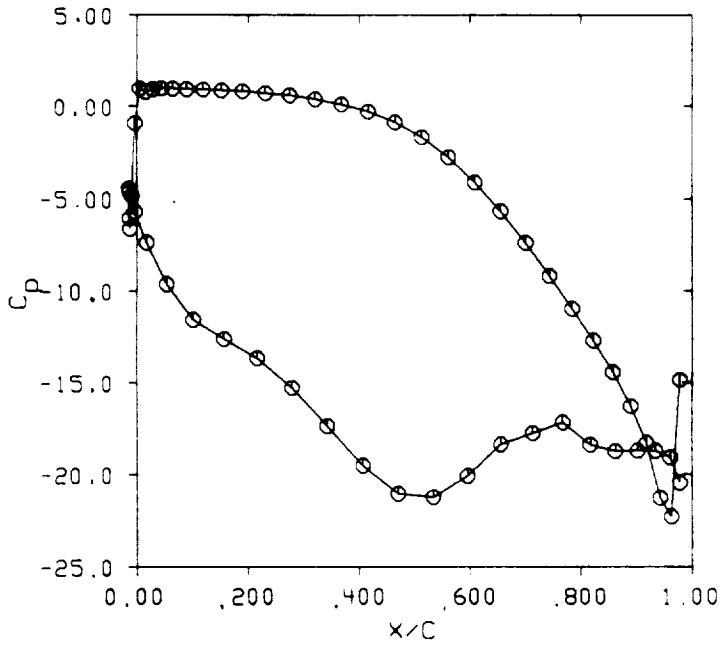


a)



b)

Figure 7.25. Inviscid Turbine Solutions. (a) Blade Isentropic Mach Number (b) Blade Total Pressure Loss (c) Coefficient of Pressure (d) Mach Contours, $\Delta M = .04$ (e) Inflow Velocity Vectors (f) Trailing Edge Velocity Vectors



c)

Figure 7.25. Continued.

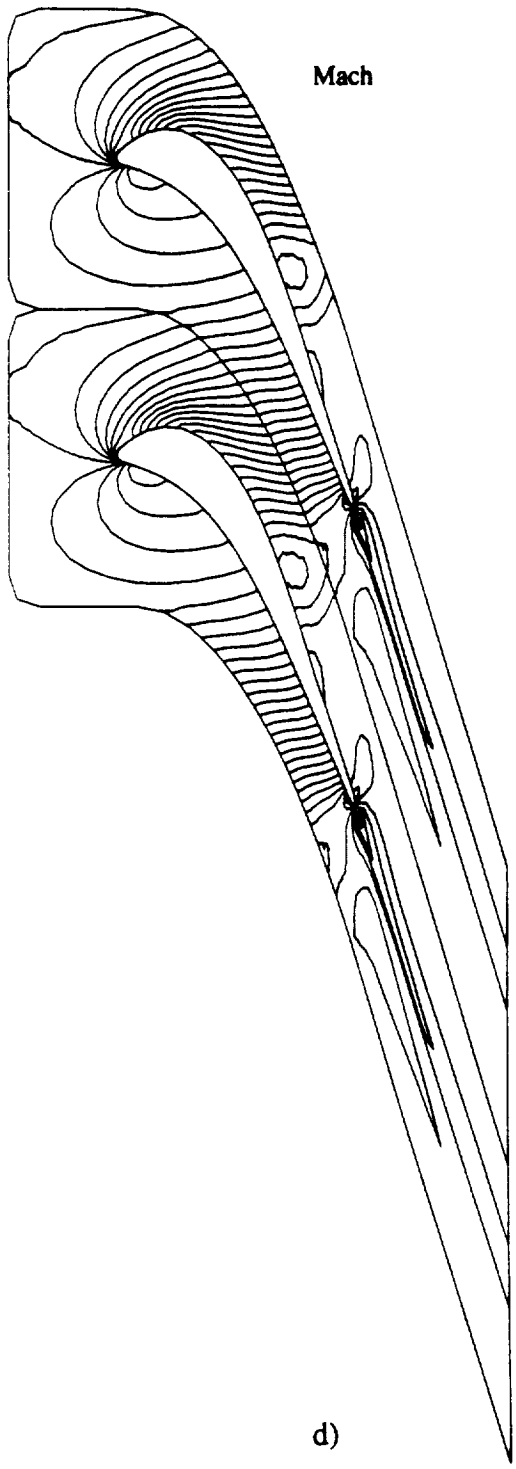
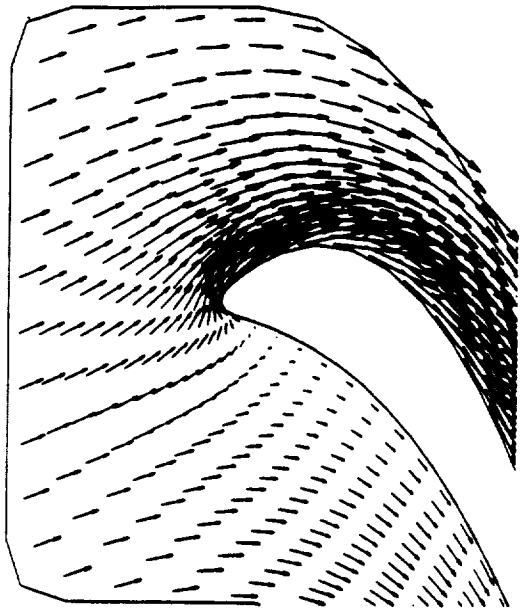
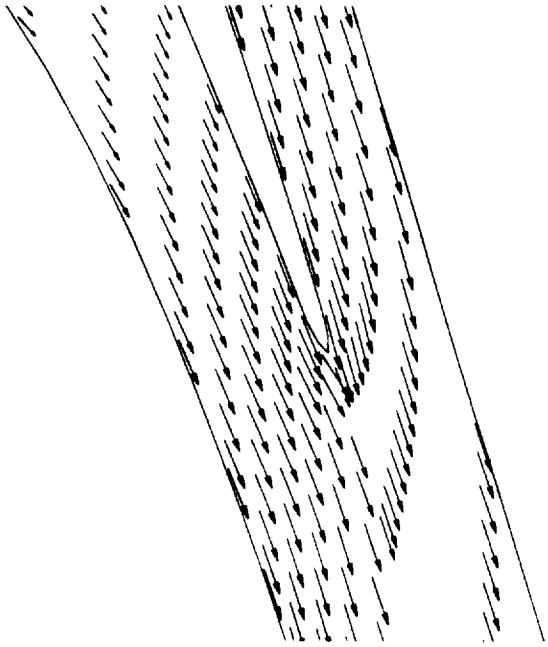


Figure 7.25. Continued.

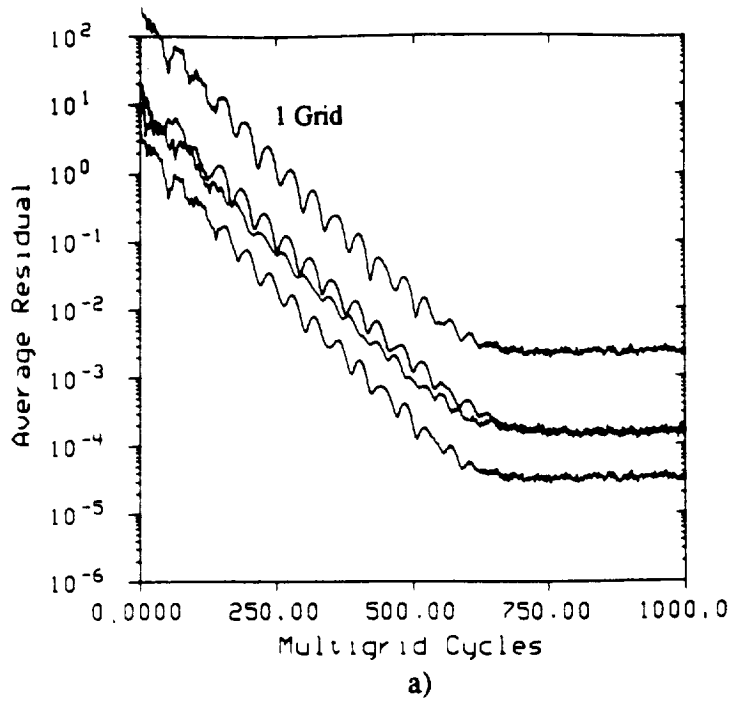


e)

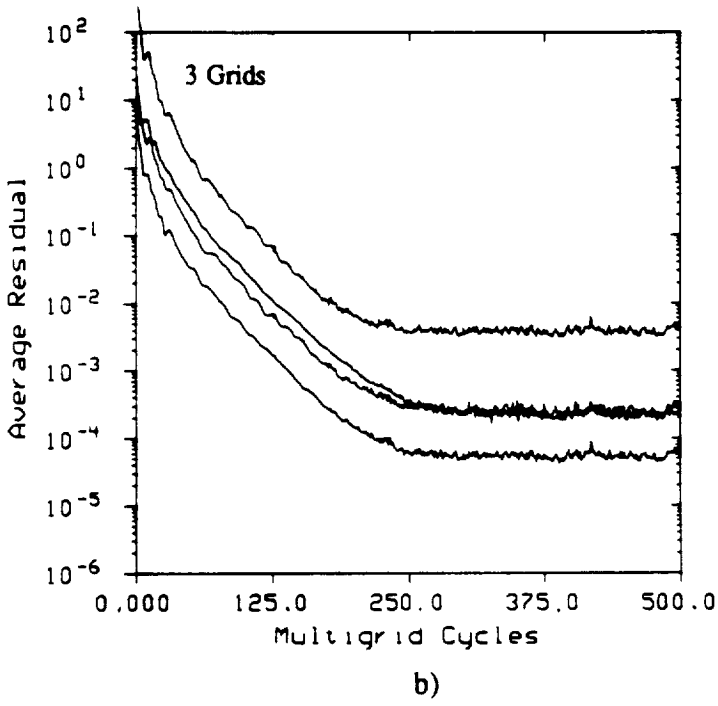


f)

Figure 7.25. Continued.

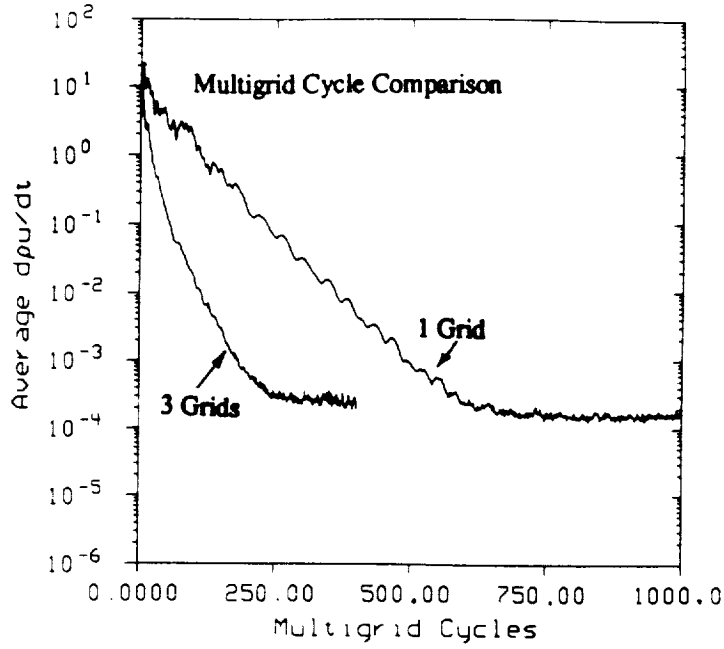


a)

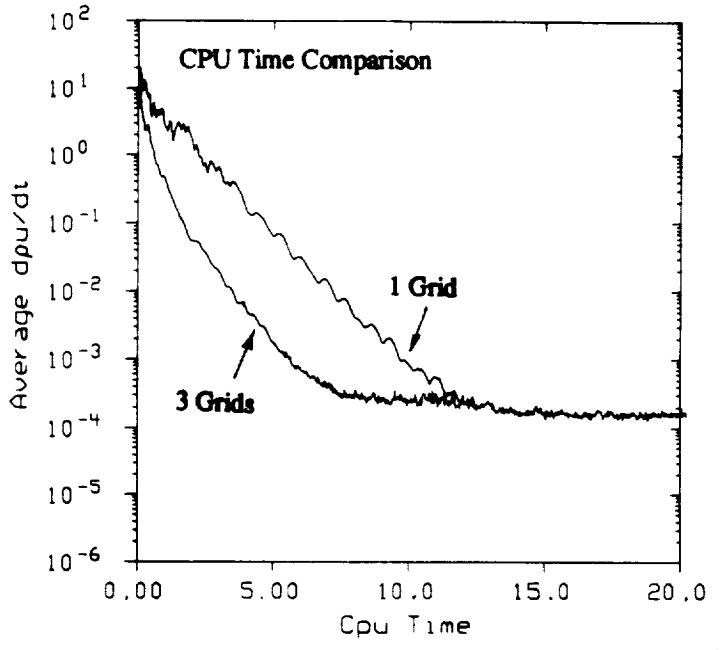


b)

Figure 7.26. Convergence Histories for Inviscid Turbine (a) 1 Grid (b) 3 Grids (c) 1 and 3 Grids Multigrid Cycle Comparison (d) 1 and 3 Grids CPU Time Comparison



c)

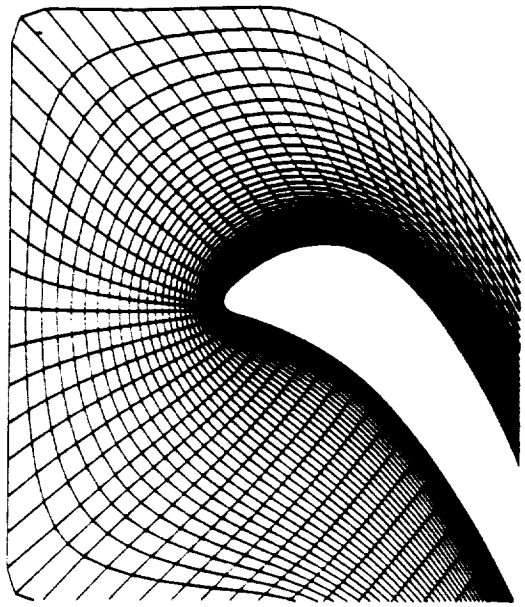


d)

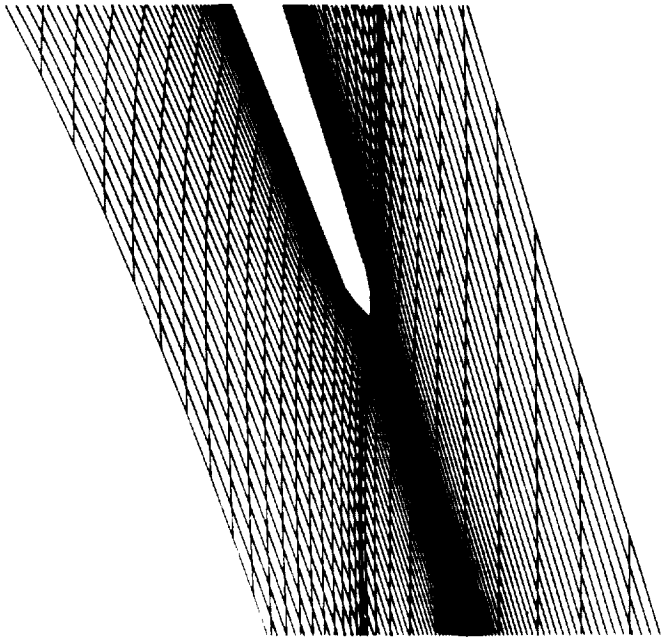
Figure 7.26. Continued.

a)

Figure 7.27. 129x41 Computational Mesh for Viscous Turbine Calculations. (a) Global View (b) Leading Edge Detail (c) Trailing Edge Detail

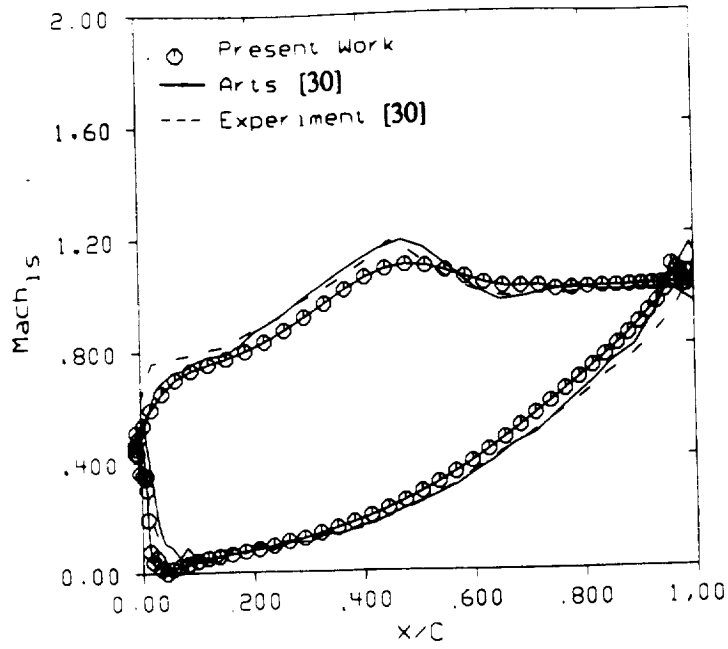


b)

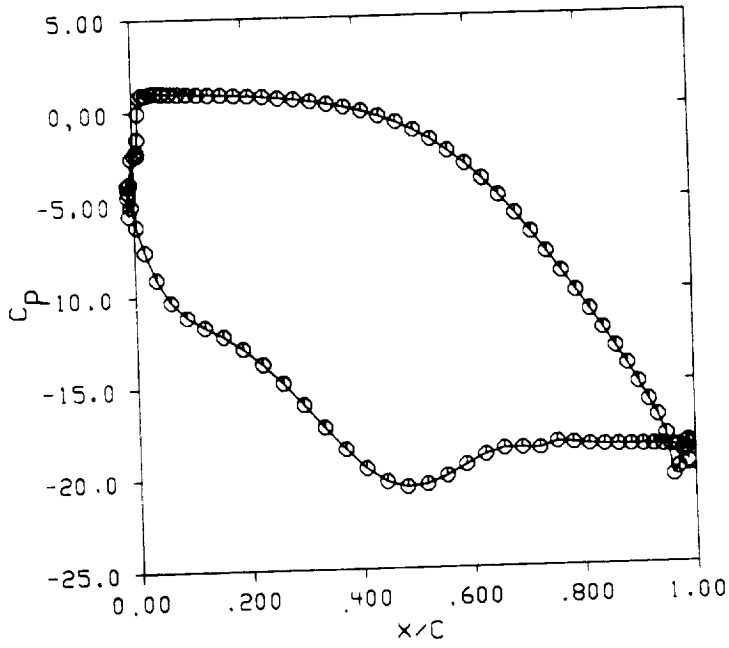


c)

Figure 7.27. Continued.



a)



b)

Figure 7.28. Viscous Turbine Solutions, $Re_L = 10{,}000$. (a) Blade Isentropic Mach Number (b) Blade Coefficient of Pressure (c) Mach Contours, $\Delta M = .04$ (d) Total Pressure Loss Contours (e) Leading Edge Velocity Vectors (f) Separation Point Velocity Vectors (g) Trailing Edge Velocity Vectors

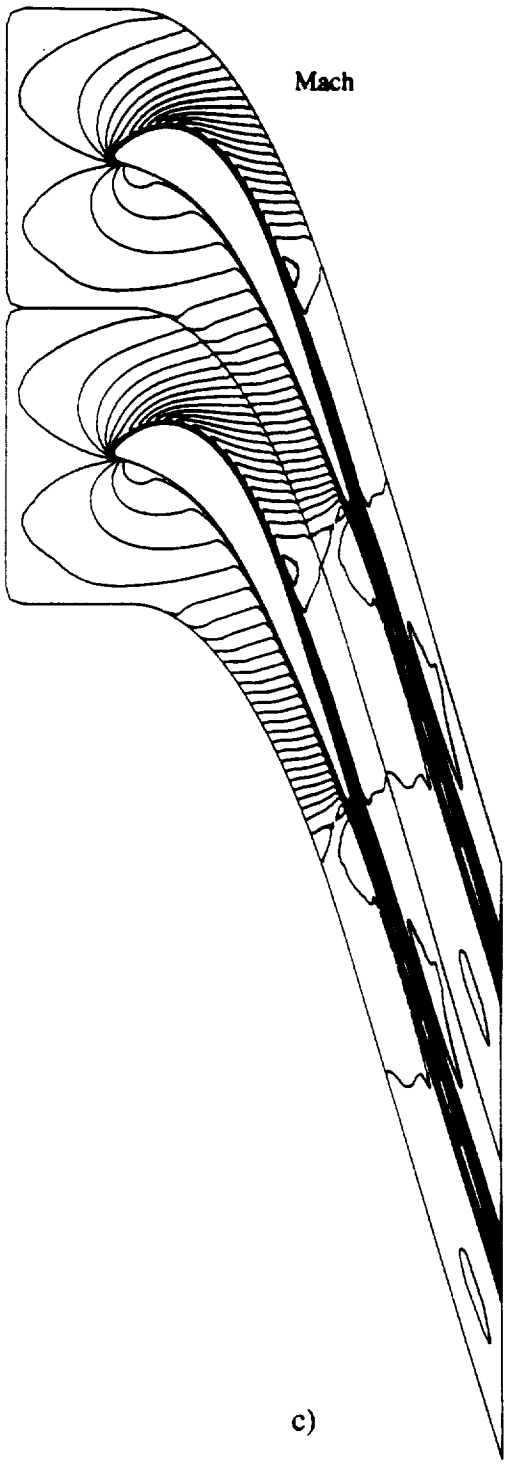
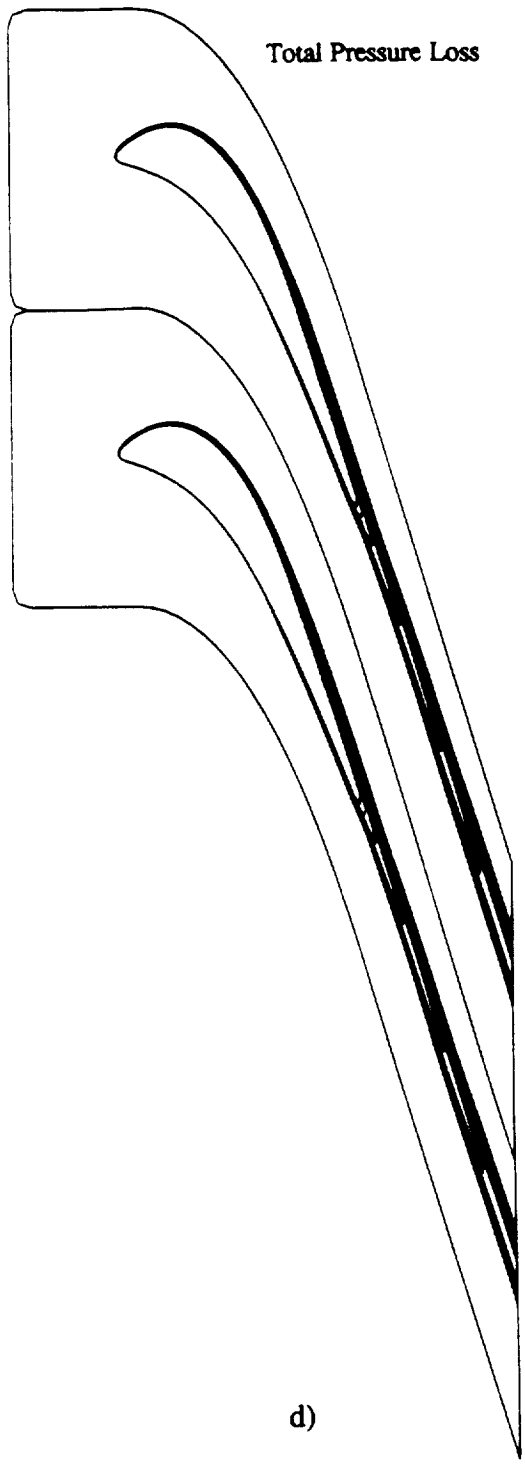
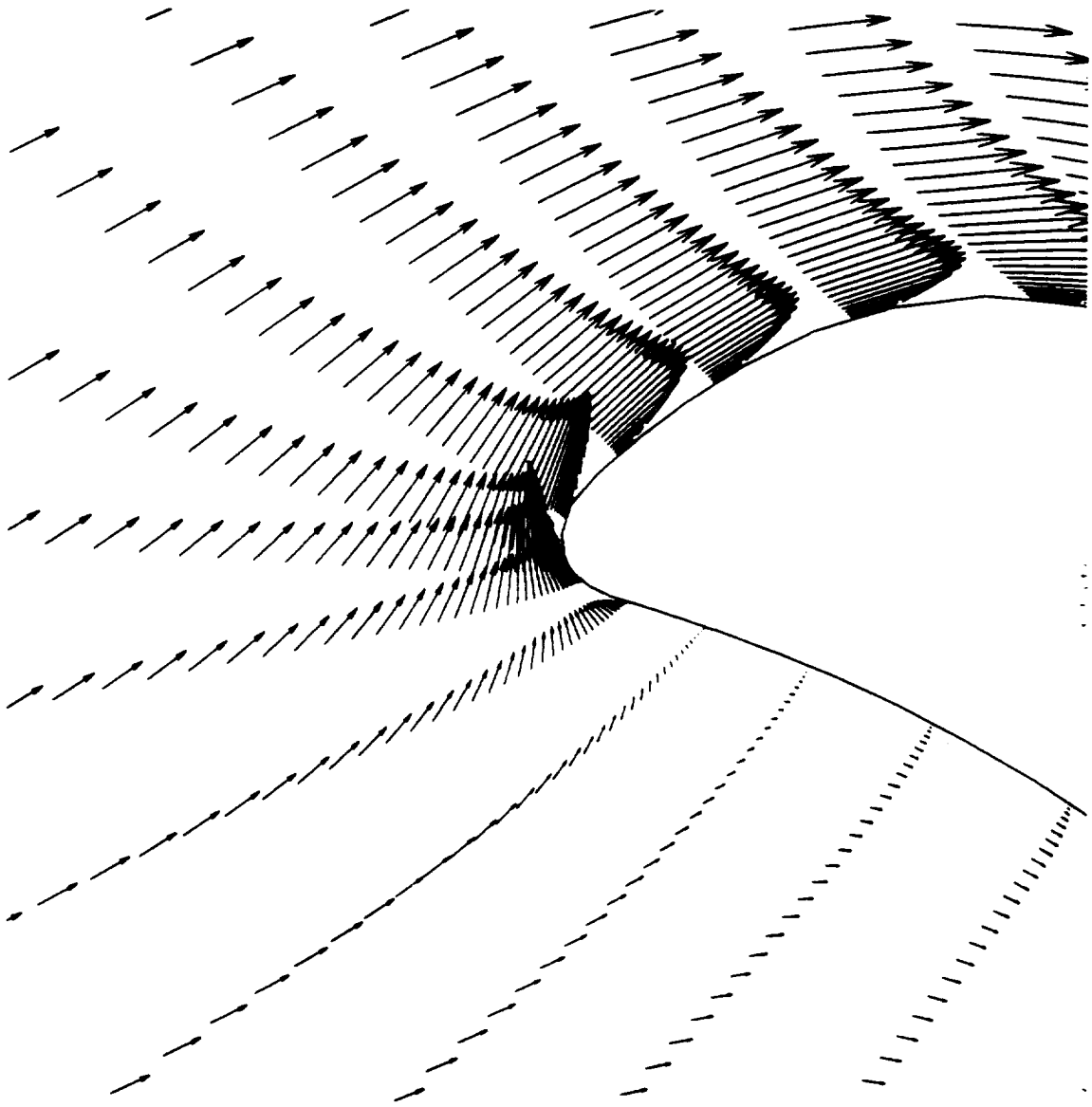


Figure 7.28. Continued.



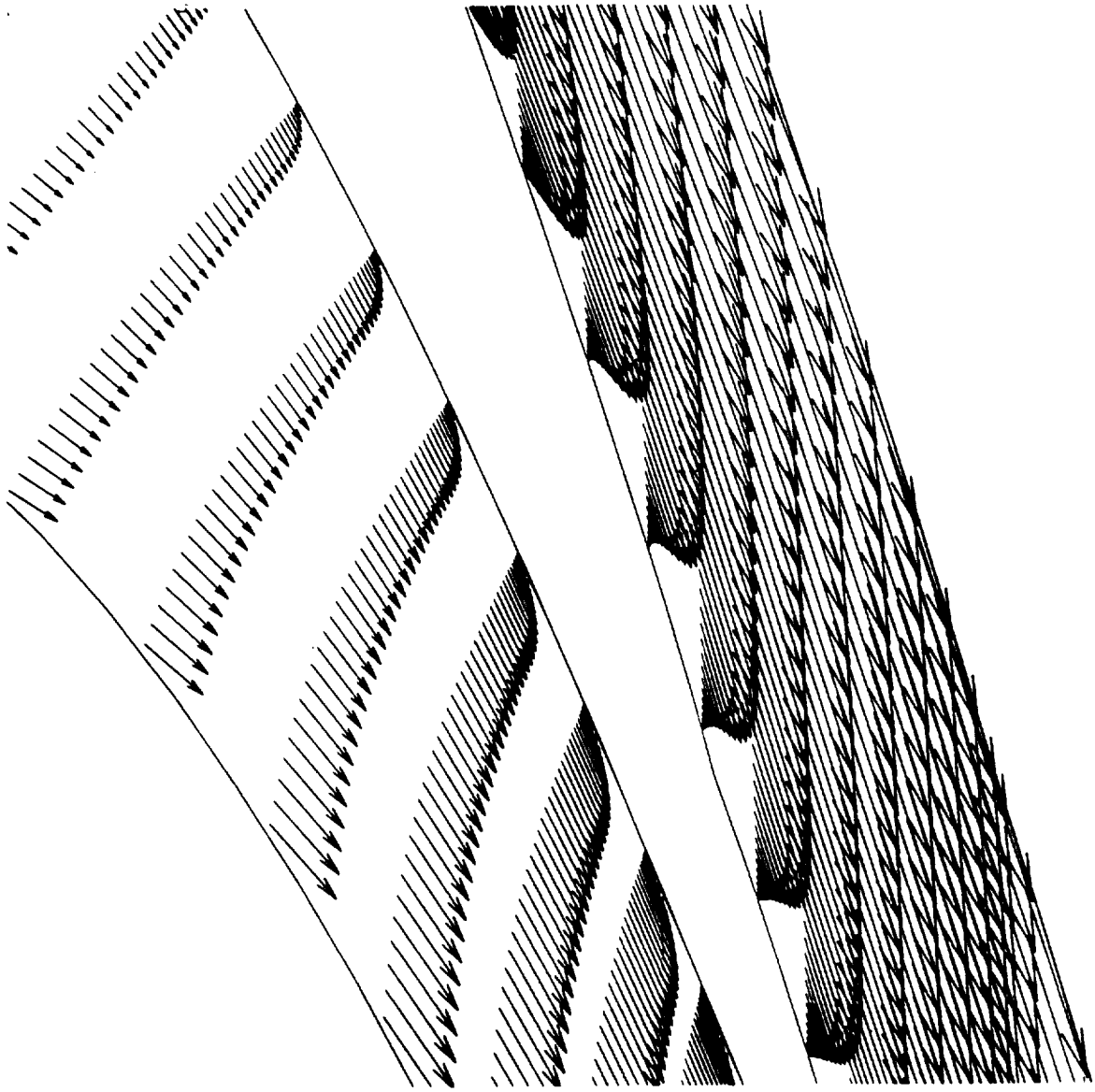
d)

Figure 7.28. Continued.



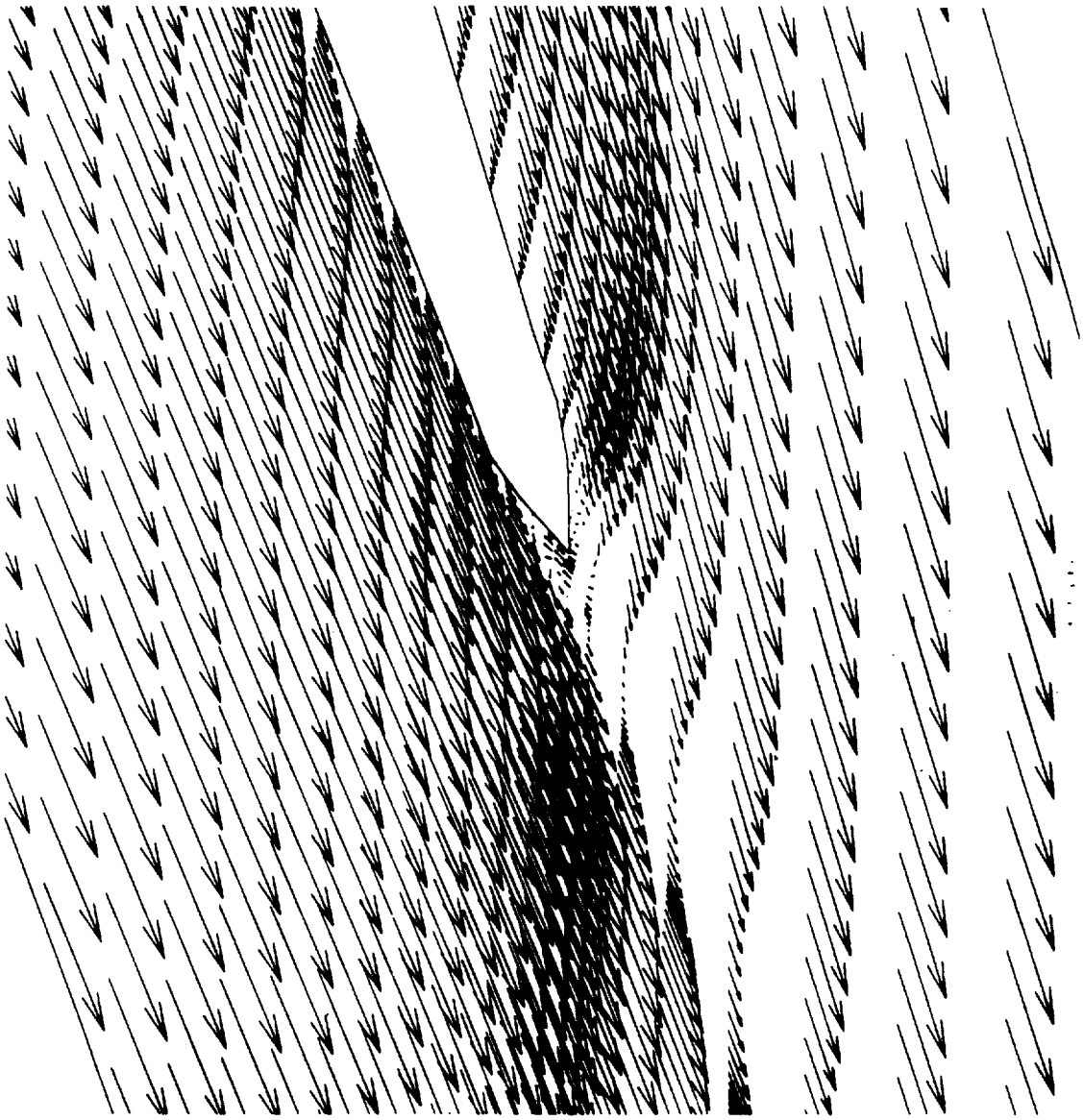
e)

Figure 7.28. Continued.



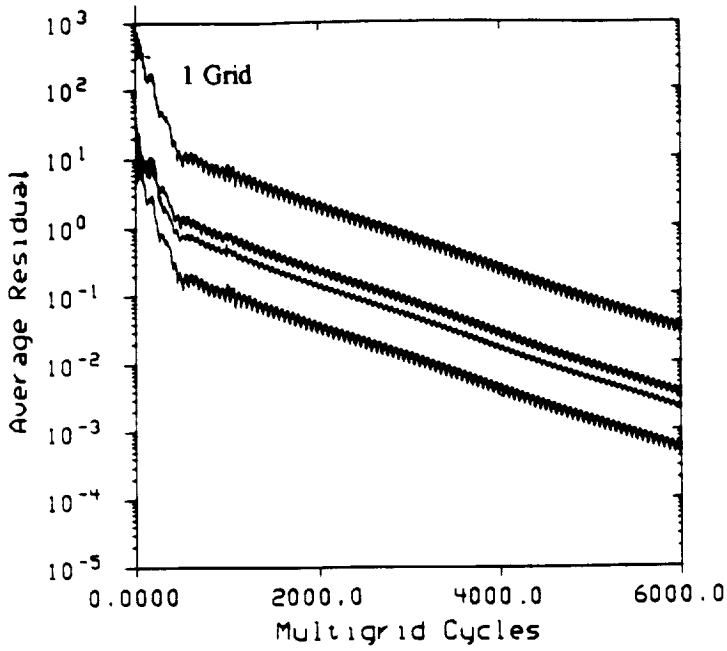
f)

Figure 7.28. Continued.

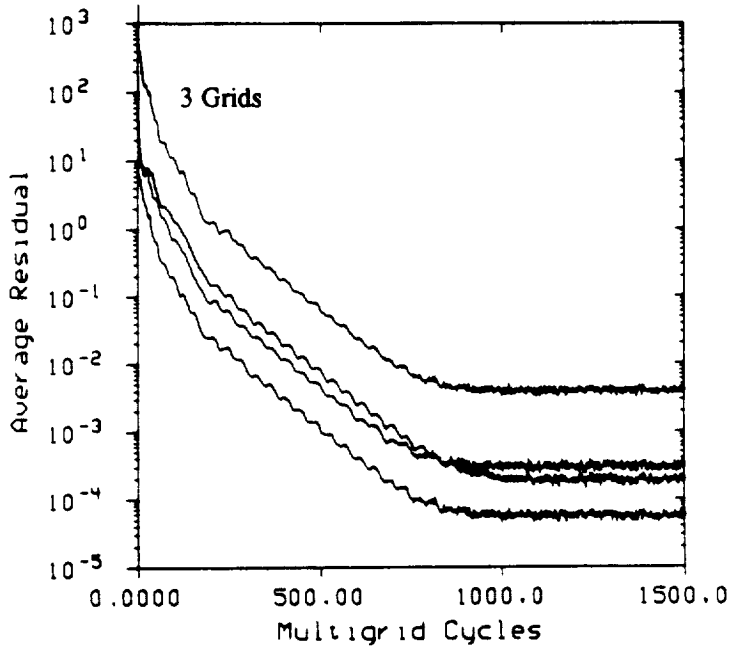


g)

Figure 7.28. Continued.

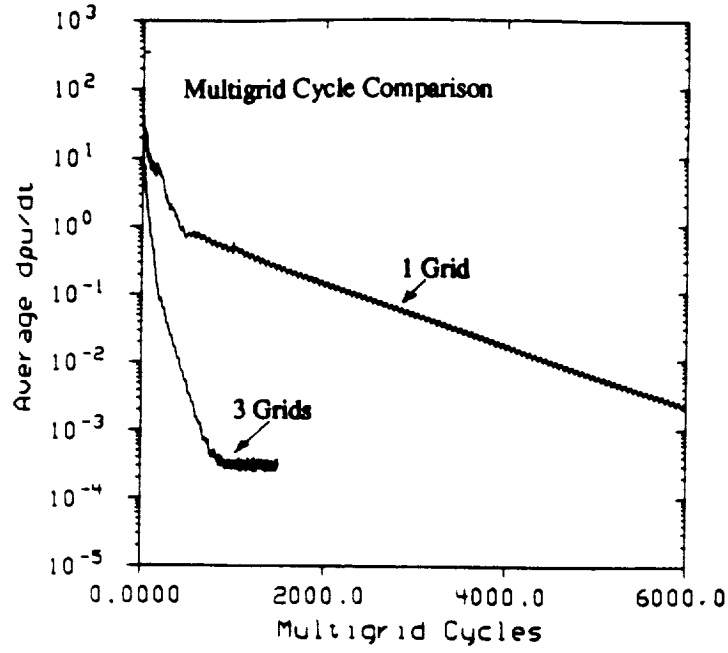


a)

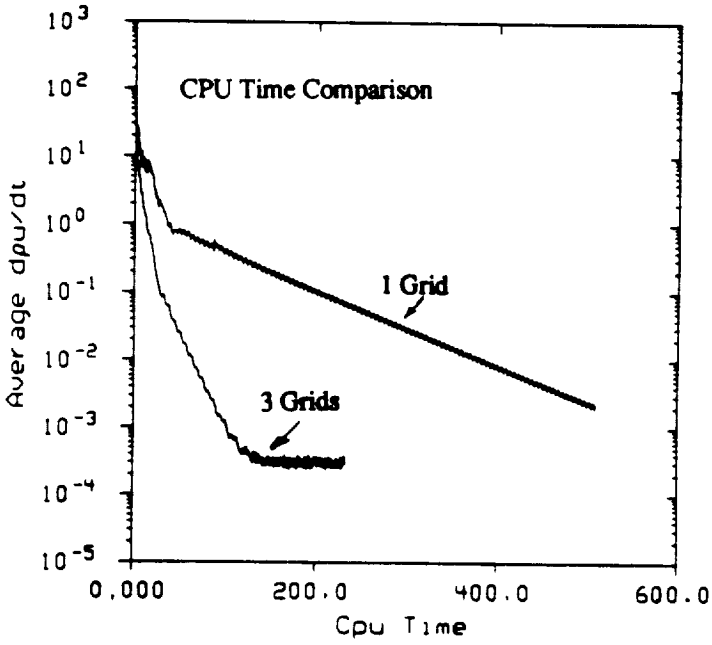


b)

Figure 7.29. Convergence Histories for Viscous Turbine (a) 1 Grid (b) 3 Grids (c) 1 and 3 Grids Multigrid Cycle Comparison (d) 1 and 3 Grids CPU Time Comparison



c)



d)

Figure 7.29. Continued.

CHAPTER 8 - CONCLUSIONS

8.1 Major Contributions and Summary of Results

This work has presented initial research into the effects of coarse-mesh boundary conditions on the convergence of Multigrid Acceleration. An explicit Multigrid algorithm has been written and validated for a variety of both inviscid and viscous flow computations. The basic algorithm has been fashioned after Jameson's finite-volume Multigrid scheme which utilizes explicit, Runge-Kutta time-stepping. Forcing terms have been derived and added to the coarse-mesh boundary conditions which permits them to be solved with fine-mesh accuracy, i.e., without the coarse-mesh truncation error polluting the fine mesh solution.

In order to derive the correct interpolation procedure, or transfer operator, for the solution at boundary points, a new, general approach to formulating the governing equations on the coarse mesh levels has also been presented. In this approach, the equations on the coarse-meshes have been viewed as a filtered sub-set of the fine-mesh equations, and a general filtering operator has been derived which models this filtering process. This approach is based on the fact that certain information is not resolved on the coarse-mesh due to the increased mesh spacing, and, therefore, is "filtered out". By applying the filter to the governing fine-mesh equations, a formal description between these and the coarse-mesh equations is given. Then, by specifying a discrete form of

the filter, the discrete coarse-mesh equations may be formulated. The advantage to this formulation is that any number of Multigrid schemes may be formed by specification of different discrete filters. In this work, the filter was chosen such that Jameson's Multigrid scheme was obtained. Then, the boundary transfer operator and coarse-mesh boundary forcing terms were derived.

In summary, the major contributions that have been presented in this work are:

— A new, general approach to obtaining the coarse-mesh governing equations in partial differential form, where they are formally related to the fine-mesh equations

— Formulation of a boundary transfer operator that is consistent with the interior scheme

— Derivation of forcing terms for the coarse-mesh boundary conditions

— Demonstration of the ability of these forcing terms to allow the coarse-mesh boundary conditions to be applied on any number of mesh levels with fine-mesh accuracy

Flow calculations were performed for inviscid channel flow over a circular-arc bump for subsonic, transonic, and supersonic speeds, a viscous flat plate, viscous subsonic channel flow over a circular-arc bump, and inviscid and viscous flow over a VKI gas turbine rotor blade. Good agreement with other results was seen in all cases, with only a few minor discrepancies occurring. The discrepancies which occurred in the viscous bump velocity profiles (Figure 7.18c-e) and the inviscid turbine isentropic Mach number distribution (Figure 7.25a) seemed to be related to the artificial dissipation model.

Implementing the coarse-mesh boundary conditions without forcing terms introduced coarse-mesh truncation errors into the fine-mesh solutions. This was apparent in all cases except the supersonic bump and the VKI turbine blade. For these

2 cases, the truncation error was apparently lower than the local round-off error. The error introduced into the fine-mesh solution increased as the number of mesh levels increased (Figures 7.4,7.20) due to the larger mesh spacings on the coarsest mesh. With forcing terms, however, the fine-mesh accuracy was retained for any number of mesh levels used. This becomes very important as problems increase in size and more mesh-levels are used. The problems here have used relatively coarse initial meshes with few mesh levels in the Multigrid cycle. In larger applications which don't use forcing terms in the boundary conditions, the number of mesh levels that could possibly be used would be restricted due to the error that is introduced into the fine mesh solution. This would consequently reduce the efficiency of the Multigrid solution. Therefore, the forcing terms, by permitting the maximum number of mesh levels to be used, increase Multigrid efficiency even though they don't actually increase convergence rates.

The forcing terms along the solid boundaries for the viscous problems were stated (Chapter 6) as applying a partial slip condition as the mesh becomes too coarse to accurately resolve the boundary layer. As the mesh spacing scale on the coarsest mesh level exceeds the length scale of the boundary layer, this should be equivalent to enforcing flow tangency along the surface. However, when flow tangency is explicitly applied, it was shown that the forcing terms are still required to produce fine-mesh accuracy (Figures 7.15, 7.21).

The efficiency of Multigrid acceleration was seen to decrease with increasing Mach number for the inviscid channel flow cases. However, this may be attributed to the

increase in efficiency of the single-mesh solutions as Mach Number increases (Figure 7.11a). The Multigrid convergence rates for the 3 cases were approximately the same, with slightly reduced performance in the supersonic case (Figure 7.11b). The efficiency of Multigrid acceleration for the flat plate, viscous bump, and viscous turbine blade problems was greater in terms of Multigrid cycles than in terms of the actual work performed, measured in CPU time (Figures 7.16c,d 7.22c,d 7.29c,d). This is typical of viscous problems, and is due to the additional work required to evaluate the viscous fluxes.

8.2 Future Work

The general Multigrid approach that was formulated may be used to derive any number of discrete Multigrid schemes based on a specified filter, or prescribed method of transferring the solution and residuals. In this work, only one such filter was looked at: the simple 4-cell volume weighted averaging as given in Jameson's scheme. Future work may entail the derivation of various other filters and a study of their effect on the performance of Multigrid acceleration.

Further study could also be performed into the reasons behind the effect of Mach number on convergence, based on the observations made for the inviscid channel flow problem. For the Multigrid solutions to this case, not much difference existed between the convergence rates, although the supersonic Multigrid convergence was the slowest of the three, exactly opposite from the single-mesh convergence rates.

Finally, a more robust dissipation model could be pursued. Some difficulties are realized when performing Euler calculations on very fine, highly stretched meshes,

possibly due to the scaling that is used. Alternative scaling theories could be studied, i.e., based on something other that the maximum spectral radii of the convective Jacobian matrices.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Jameson, A.; Baker, T.J.: Solution Of The Euler Equations For Complex Configurations, AIAA 6th Computational Fluid Dynamics Conference, AIAA, New York, 1983, pp 293-302.

[2] Jameson, A.: Multigrid Solutions Of The Euler Equations For Aircraft Configurations, AIAA Paper No. 84-0093, January 1984.

[3] Jameson, A.: A Vertex Based Multigrid Algorithm For Three Dimensional Compressible Flow Calculations, Numerical Methods For Compressible Flows; Finite Difference, Element And Volume Techniques, ASME Winter Annual Meeting, December 1986.

[4] Jameson, A.: Successes And Challenges In Computational Aerodynamics, AIAA Paper No. 87-1184, January 1984.

[5] Jameson, A.; Schmidt, W.; and Turkel, E.: Numerical Solutions Of The Euler Equations By Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA Paper No. 81-1259, January 1981.

[6] Jameson, A: Numerical Solution Of The Euler Equation For Compressible Inviscid Fluids, Numerical Methods For The Euler Equations, SIAM, 1985.

[7] Turkel, E: Acceleration To A Steady State For The Euler Equations, Numerical Methods For The Euler Equations, SIAM, 1985.

[8] Swanson, R.C.: A Multistage Time-Stepping Scheme For The Navier-Stokes Equations, AIAA Paper No. 85-0035, January 1985.

[9] Thompkins, W.T., Jr.; Tong, S.S.; Busch, R.H.; Usab, W.J., Jr.; and Norton, R.J.G.: Solution Procedures For Accurate Numerical Simulations Of Flow In Turbomachinery Cascades, AIAA Paper No. 83-0257, January 1983.

[10] Brandt, A.: Guide To Multigrid Development, Multigird Methods, Springer-Verlag, 1982.

[11] Brandt, A.: Multi-Level Adaptive Solutions To Boundary-Value Problems, Mathematics Of Computation, Vol. 31, No. 138, April 1977, pp. 333-390.

[12] Ni, R.H.: A Multiple Grid Scheme For Solving The Euler Equations, *AIAA Journal*, Vol. 20, No. 11, November 1982.

[13] Chima, R.V; and Turkel, E: Comparison Of Three Explicit Multigrid Methods For The Euler And Navier-Stokes Equations, AIAA Paper No. 87-0602, January 1987.

[14] Siclari, M.J.; and DelGuidice, P.: A Multigrid Finite Volume Method For Solving The Euler And Navier-Stokes Equatinos For High Speed Flows, AIAA Paper No. 89-0283, January 1989.

[15] Chima, R.V.; and Johnson, G.M.: Efficient Solution Of The Euler And Navier-Stokes Equations With A Vectorized Multiple-Grid Algorithm, AIAA Journal, Vol. 23, No. 1, January 1985.

[16] Chima, R.V.: Analysis Of Inviscid And Viscous Flows In Cascades With An Explicit Multiple-Grid Algorithm, AIAA Paper No. 84-1663, June 1984.

[17] Martinelli, L.; Jameson, A.; and Grasso, F.: A Multigrid Method For The Navier-Stokes Equations, AIAA Paper No. 86-0208, January 1986.

[18] Mavriplis, D.J.; Jameson, A.; and Martinelli, L.: Multigrid Solution Of The Navier-Stokes Equations On Triangular Meshes, AIAA Paper No. 89-0120, January 1989.

[19] Martinelli, L.; and Jameson, A.: Validation Of A Multigrid Method For The Reynolds Averaged Equations, AIAA Paper No. 88-0414, January 1988.

[20] Radespiel, R.; and Swanson, R.: An Investigation Of Cell Centered And Cell Vertex Multigrid Schemes For The Navier-Stokes Equations, AIAA Paper No. 89-0548, January 1989.

[21] Schlichting, H.: Boundary Layer Theory, McGraw-Hill Book Company, New York, 1979.

[22] Swanson, R.C.; and Turkel, E.: Artificial Dissipation And Central Difference Schemes For The Euler And Navier-Stokes Equations, AIAA Paper No. 87-1107, January 1987.

[23] Martinelli, L.: Calculation Of Viscous Flows With Multigrid Methods, Ph.D. Dissertation, MAE Department, Princeton University.

[24] Caughey, D.A.; and Turkel E.: Effects Of Numerical Dissipation On Finite Volume Solutions Of Compressible Flow Problems, AIAA Paper No. 88-0621, January 1988.

[25] Yakota, J.: Private Communication.

[26] Jameson, A.; and Yoon, S.: Multigrid Solutions Of The Euler Equations Using Implicit Schemes, AIAA Paper No. 85-0293, January 1985.

[27] Rhie, C.M. : A Pressure Based Navier-Stokes Solver Using the Multigrid Method, AIAA Paper No. 86-0207, January 1986.

[28] Williams, Marc.: Private Communication, Purdue University, January 1989.

[29] Kallinderis, J.G.; and Baron, J.R.: Adaptation Methods For A New Navier-Stokes Algorithm, AIAA Paper No. 87-1167, January 1987.

[30] Arts, A.: Cascade Flow Calculations Using A Finite Volume Method, Numerical Methods For Flows In Turbomachinery Bladings, Von Karman Institute For Fluid Dynamics, Lecture Series 1982-05, April 1982.