NASA Technical Paper 2953

1990

An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows

Peter A. Gnoffo
Langley Research Center
Hampton, Virginia

NASA
National Aeronautics and Space Administration
Office of Management
Scientific and Technical Information Division

Contents

Symbols

$\mathbf{A}$	Jacobian matrix of $\mathbf{g}$ with respect to $\mathbf{q}$
a	speed of sound, nondimensionalized by V_∞
a'	geometric weighting parameter defined by equation (3.2a)
$\mathbf{B}$	Jacobian matrix of $\mathbf{h}$ with respect to $\mathbf{q}$
b'	geometric weighting parameter defined by equation (3.2b)
$\mathbf{C}$	matrix defined by equation (6.11) used in point-implicit algorithm
C_H	heat-transfer coefficient
C_N	Courant number defined in appendix D
C_f	skin friction coefficient
C_m	pitching-moment coefficient
C_p	pressure coefficient
D	geometric weighting parameter defined by equation (3.2c)
E	total energy per unit mass, nondimensionalized by V_∞^2
e	energy per unit mass, nondimensionalized by V_∞^2
$\vec{\mathbf{f}}$	three-dimensional flux vector in Cartesian coordinates
$\mathbf{g}$	inviscid component of flux vector relative to cell wall
H	total enthalpy per unit mass, nondimensionalized by V_∞^2
$\mathbf{h}$	viscous component of flux vector relative to cell wall
$\mathbf{I}$	identity matrix
I, J, K	indices of cell centers in ξ, η, ζ directions
i, j, k	indices of cell walls in ξ, η, ζ directions
$\vec{\mathrm{i}}$, $\vec{\mathrm{j}}$, $\vec{\mathrm{k}}$	unit vectors in x, y, z directions
L_{ref}	reference length
L_2	error norm defined by equation (7.2)
l	dummy index for i, j, or k
$\vec{\mathrm{l}}$	three-dimensional unit vector tangent to computational cell wall
$\mathrm{l}_x, \mathrm{l}_y, \mathrm{l}_z$	components of $\vec{\mathrm{l}}$ in x, y, z directions
M	Mach number
$\mathbf{M}$	point-implicit Jacobian matrix
$\mathbf{M}'$	point-implicit Jacobian matrix including relaxation factors
$\vec{\mathrm{m}}$	three-dimensional unit vector tangent to computational cell wall
$\mathrm{m}_x, \mathrm{m}_y, \mathrm{m}_z$	components of $\vec{\mathrm{m}}$ in x, y, z directions
N_c	cell Reynolds number defined by equation (7.1)
$\vec{\mathrm{n}}$	three-dimensional unit vector normal to computational cell wall

n_x, n_y, n_z	components of $\vec{n}$ in x, y, z directions
Pr	Prandtl number
p	pressure, nondimensionalized by $\rho_\infty V_\infty^2$
q	heating rate
$\mathbf{q}$	vector of conserved variables
$\mathbf{R}$	matrix of eigenvectors of $\mathbf{A}$
Re	Reynolds number, $\frac{\rho_\infty V_\infty L_{\text{ref}}}{\mu_\infty}$
$\mathbf{r}$	residual defined by equation (5.2b)
$\vec{r}$	three-dimensional position vector in Cartesian space
$\mathbf{s}$	approximation to gradient of characteristic variables across cell face
s	dummy variable for x, y, or z
T	temperature
t	time, nondimensionalized by $\frac{L_{\text{ref}}}{V_\infty}$
U	velocity component normal to computational cell wall, nondimensionalized by V_∞
$\vec{u}$	three-dimensional velocity vector in Cartesian space, nondimensionalized by V_∞
u,v,w	velocity components in x, y, z directions, nondimensionalized by V_∞
V	velocity component tangent to computational cell wall in $\vec{l}$ direction, nondimensionalized by V_∞
V_∞	free-stream velocity
W	velocity component tangent to computational cell wall in $\vec{m}$ direction, nondimensionalized by V_∞
x, y, z	Cartesian coordinates, nondimensionalized by L_{ref}
α	kinetic energy per unit mass defined by equation (B7) or angle of attack, deg
β	$= \gamma - 1$
γ	ratio of specific heats
δ	forward difference in time, difference between successive iterations
ϵ	eigenvalue limiter defined by equations (6.3) and (6.4)
ζ	computational coordinate
η	computational coordinate
θ	integer parameter controlling first- or second-order dissipation
$\mathbf{\Lambda}$	diagonal matrix of eigenvalues of $\mathbf{A}$
λ	eigenvalue of $\mathbf{A}$
$\tilde{\lambda}$	restricted eigenvalue of $\mathbf{A}$ defined by equation (6.1)
μ	viscosity, nondimensionalized by μ_∞

ξ	computational coordinate
ρ	density, nondimensionalized by ρ_∞
σ	area of cell wall, nondimensionalized by L_{ref}^2
$\vec{\sigma}$	directed area of cell wall in three-dimensional space, nondimensionalized by L_{ref}^2
τ	shear stress
ν	dummy variable for u, v, or w
Φ	relaxation factor defined in equations (6.5) and (6.7)
χ	dummy variable for ξ, η, ζ
Ω	cell volume, nondimensionalized by L_{ref}^3
ω	vector of source terms for conserved quantities in cell
Subscripts:	
I, J, K	indices of cell centers in ξ, η, ζ directions
i, j, k	indices of cell boundaries in ξ, η, ζ directions
inv	function of inviscid terms alone
L	dummy index for cell center
l	dummy index for cell boundary
ref	reference value
vis	function of viscous terms alone
∞	free stream
Superscripts:	
min	value returned from min mod function
n	iteration level index

Abstract

An upwind-biased, point-implicit relaxation algorithm for obtaining the numerical solution to the governing equations for three-dimensional, viscous, compressible perfect-gas flows is described. The algorithm is derived by using a finite-volume formulation in which the inviscid components of flux across cell walls are described with Roe's averaging and Harten's entropy fix with second-order corrections based on Yee's symmetric total variation-diminishing scheme. Viscous terms are discretized by using central differences. The relaxation strategy is well suited for computers employing either vector or parallel architectures. It is also well suited to the numerical solution of the governing equations on unstructured grids. Because of the point-implicit relaxation strategy, the algorithm remains stable at large Courant numbers without the necessity of solving large, block tridiagonal systems. Convergence rates and grid refinement studies are conducted for Mach 5 flow through an inlet with a 10° compression ramp and Mach 14 flow over a 15° ramp. Predictions for pressure distributions, surface heating, and aerodynamic coefficients compare well with experimental data for Mach 10 flow over a blunt body.

1. Introduction

The formulation of an algorithm for obtaining the numerical solution of a system of partial differential equations may be divided into two tasks. First, the solution domain is discretized by using a collection of points or cells. Relations between quantities at neighboring cells are defined such that, in the limit as cell size goes to zero (cell number goes to infinity), the numerical relations are consistent with the partial differential equations. Second, a relaxation algorithm must be defined which will drive the residual of the approximation scheme to zero simultaneously at all the cells in the domain. The two tasks are clearly related, but in the parlance of computational fluid dynamics, the first task deals primarily with how the physical approximations are defined on the "right-hand side," and the second task deals with how the solution variables evolve on the "left-hand side." Right- and left-hand sides refer to the position of the terms relative to the equal sign in the definition of the complete algorithm.

When considering the governing partial differential equations for compressible viscous flow, the numerical approximations to the component terms can be conveniently defined as a function of the physically relevant fluxes and stresses acting on cell walls. This treatment constitutes a finite-volume formulation of the governing conservation laws. If properly constructed, it guarantees a conservative formulation in that no extraneous sources of mass, momentum, or energy will be introduced into the system as a result of numerical imbalances of quantities passing from one cell to the next. The numerical approximations to fluxes and stresses across a cell wall completely define the finite-volume formulation of the physics on the right-hand side. Still, there are an unlimited number of ways to numerically approximate these quantities on a cell wall. Viscous, dissipative stresses are generally approximated by using second-order-accurate central differences which numerically reflect the zone of dependence of dissipative processes due to the random, thermal velocity of molecules. However, inviscid fluxes are defined by more complex wave interactions. (In fact, the physical zones of dependence of viscous and inviscid terms are interrelated, but the relationship can be ignored for the purposes of defining approximation schemes.) Additional physical insight as to how information is propagated in a flow field can be used to better model the physics of the flow.

Such insight is provided by a class of approximation schemes referred to as "upwind differencing algorithms." These algorithms can be derived for the inviscid terms by considering the Riemann problem defined by the end states at cell centers on either side of a cell wall or by performing an eigenvalue analysis of the Jacobian of the flux vector at the cell wall with respect to the vector of conserved variables (refs. 1 and 2). The flux at a cell wall can be split to account for the contributions of waves coming from the left and the right (refs. 3 and 4). Such algorithms are most easily defined within the context of first-order-accurate approximation schemes which are inherently very dissipative. Simple second-order (or higher) corrections to the upwind approximations usually cause severe oscillations in the computed solution in the vicinity of high gradient regions in the flow. Often these corrections contribute a negative artificial dissipation which overwhelms the natural, positive dissipation of the base first-order, upwind scheme and leads to catastrophic instabilities. A physical interpretation of these difficulties notes that higher order correction difference stencils will, at times, cross discontinuities in the flow or will cross the true zone of dependence of the approximated flux. Mathematical analyses of this problem have been used to construct high-order corrections to the scheme that will compare fluxes (or flux differences) in the vicinity of cell walls and will choose a stencil that ensures a stable, nonoscillatory solution. Correction schemes which approach this task in varied ways are known as flux-corrected transport (FCT) (refs. 5 and 6), total variation diminishing (TVD) (refs. 7 through 12), and essentially nonoscillatory (ENO) (refs. 13 through 16). Some of these correction schemes may violate the actual upwind zone of dependence but still function properly (ref. 10).

The selection of a differencing scheme for the right-hand side is generally based on knowledge of the scheme's performance on similar problems. The scheme used in the present study is based primarily on the earlier work of Roe (ref. 2), Harten (ref. 7), and Yee (ref. 10). The application focus of this work is on hypersonic flows over blunt bodies, including the base and near wake, such as would occur on aeroassisted orbital transfer vehicles (AOTV) (ref. 17). Roe's scheme appeared to be well suited for capturing the strong bow shock associated with this application where a typical free-stream Mach number is greater than 30. Also, anticipated modifications to the basic perfect-gas formulation for the case of real gases and finite-rate chemistry were considered to be relatively straightforward. Coakley's application of a very similar scheme to the problem of three-dimensional flow provided additional evidence of the method's capabilities. (See ref. 18.) The evolution of the scheme to the form described herein, with emphasis on the blunt body applications, is described in references 19 through 22.

The second task, as mentioned earlier, deals with the formulation of the left-hand side for defining the evolution of the dependent variables. For problems in which a time-accurate evolution of the dependent variables is required, the options are restricted with regard to the kinds of modifications that can be made in the relaxation algorithm for driving the solution to a steady state (assuming one exists). Both spatial and temporal accuracies must be maintained. However, if only the steady-state solution is required, one is free to evaluate any element of the difference stencil at any iteration (pseudo time) level which facilitates the relaxation process. This approach was utilized by Allen and Cheng (ref. 23) in the way they suppressed the viscous stability limit in a low Reynolds number flow. Graves (ref. 24) treated variables at the cell center of interest and its nearest neighbors at the advanced time step (implicitly) in the "partial implicitization" relaxation technique which greatly enhanced the stability of his basic algorithm. In problems utilizing structured grids for multidimensional flow, the factored implicit schemes and line Gauss-Siedel relaxation methods permit compromises in the way the solution is advanced which significantly reduce the total computational effort while retaining solution accuracy (ref. 25). The point-implicit relaxation strategy described herein makes similar compromises which make the algorithm ideally suited for problems involving unstructured grids and/or executing on massively parallel computer architectures. The essence of the strategy is to treat the variables at the cell center of interest implicitly at the advanced

iteration level and to use the latest available data from neighbor cells in defining the left-hand-side numerics. The success of this approach is made possible by the robust stability characteristics of the underlying upwind difference scheme. This strategy was motivated by some encouraging results obtained by Lombard et al. (ref. 26) using a sweeping strategy for one dimension in conjunction with another upwind algorithm. The basic algorithm requires only a single level of storage, and numerical experiments show excellent stability characteristics, even when working directly with the steady-state equations (i.e., local Courant number much greater than 1). A time-accurate version of the present method is made possible by saving at least one extra level of conserved variable data and employing an iterative strategy before proceeding to the next time level.

2. Finite-Volume Fundamentals

The integral form of the conservation laws applied to a single cell in the computational domain is written

$$\iiint \frac{\partial \mathbf{q}}{\partial t} d\Omega + \iint \vec{\mathbf{f}} \cdot \vec{\mathrm{n}}\, d\sigma = \iiint \omega\, d\Omega \tag{2.1}$$

In equation (2.1), the first term describes the time rate of change of conserved quantity $\mathbf{q}$ in the control volume, the second term describes convective and dissipative flux through the cell walls, and the third term accounts for sources or sinks of conserved quantities within the control volume associated with thermochemical nonequilibrium. The third term is identically zero for the perfect-gas flows considered here. The vectors $\mathbf{q}$ and $\vec{\mathbf{f}}$ are defined as follows for viscous flow of a perfect gas:

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho \mathrm{u} \\ \rho \mathrm{v} \\ \rho \mathrm{w} \\ \rho E \end{bmatrix} \tag{2.2}$$

$$\vec{\mathbf{f}} = \begin{bmatrix} \rho \mathrm{u} \\ \rho \mathrm{uu} + p - \tau_{xx} \\ \rho \mathrm{uv} - \tau_{xy} \\ \rho \mathrm{uw} - \tau_{xz} \\ \rho \mathrm{u} H - \dot{q}_x - \vec{\mathrm{u}} \cdot \vec{\tau}_x \end{bmatrix} \vec{\mathrm{i}} + \begin{bmatrix} \rho \mathrm{v} \\ \rho \mathrm{vu} - \tau_{yx} \\ \rho \mathrm{vv} + p - \tau_{yy} \\ \rho \mathrm{vw} - \tau_{yz} \\ \rho \mathrm{v} H - \dot{q}_y - \vec{\mathrm{u}} \cdot \vec{\tau}_y \end{bmatrix} \vec{\mathrm{j}} + \begin{bmatrix} \rho \mathrm{w} \\ \rho \mathrm{wu} - \tau_{zx} \\ \rho \mathrm{wv} - \tau_{zy} \\ \rho \mathrm{ww} + p - \tau_{zz} \\ \rho \mathrm{w} H - \dot{q}_z - \vec{\mathrm{u}} \cdot \vec{\tau}_z \end{bmatrix} \vec{\mathrm{k}} \tag{2.3}$$

The finite-volume approximation to equation (2.1) for a rectangularly ordered, structured grid is written

$$\begin{aligned} &\left[\frac{\delta \mathbf{q} \Omega}{\delta t}\right]_{I,J,K} + \left[\vec{\mathbf{f}}_{i+1} \cdot \vec{\mathrm{n}}_{i+1} \sigma_{i+1} - \vec{\mathbf{f}}_i \cdot \vec{\mathrm{n}}_i \sigma_i\right]_{J,K} \\ &+ \left[\vec{\mathbf{f}}_{j+1} \cdot \vec{\mathrm{n}}_{j+1} \sigma_{j+1} - \vec{\mathbf{f}}_j \cdot \vec{\mathrm{n}}_j \sigma_j\right]_{I,K} \\ &+ \left[\vec{\mathbf{f}}_{k+1} \cdot \vec{\mathrm{n}}_{k+1} \sigma_{k+1} - \vec{\mathbf{f}}_k \cdot \vec{\mathrm{n}}_k \sigma_k\right]_{I,J} = 0 \end{aligned} \tag{2.4}$$

where $\delta \mathbf{q} = \mathbf{q}^{n+1} - \mathbf{q}^n$ and $\delta t = t^{n+1} - t^n$. The dependent variable $\mathbf{q}$ is defined at cell centers. The cell volume Ω and cell wall area σ are functions of the independent variable $\vec{\mathrm{r}} = x\vec{\mathrm{i}} + y\vec{\mathrm{j}} + z\vec{\mathrm{k}}$ which is defined at cell corners. A shorthand notation for equation (2.4) that will be used

throughout this paper is

$$\left[\frac{\delta \mathbf{q}\Omega}{\delta t}\right]_L + \sum_{l=i,j,k} \left[\vec{\mathbf{f}}_{l+1} \cdot \vec{\mathrm{n}}_{l+1}\sigma_{l+1} - \vec{\mathbf{f}}_l \cdot \vec{\mathrm{n}}_l\sigma_l\right] = 0 \tag{2.5}$$

Note in equations (2.4) and (2.5) that uppercase variables I, J, K, and L denote computational coordinates at cell centers and lowercase variables i, j, k, and l denote cell faces or cell corners. For example, $\sigma_{i,J,K}$ refers to the cell wall corresponding to indices $I - \frac{1}{2}$, J, K, and $x_{i+1,j+1,k}$ refers to the cell corner corresponding to indices $I + \frac{1}{2}$, $J + \frac{1}{2}$, $K - \frac{1}{2}$. A schematic of the indexing system is found in figure 1. In the shorthand notation of equation (2.5), the lowercase variable l is used as a generic index for i, j, or k. This notation is convenient because most of the formulations for quantities at cell faces are independent of the coordinate direction. The geometric quantities Ω, σ, and $\vec{\mathrm{n}}$ are defined in appendix A.

The formulations of the inviscid and viscous contributions to the overall conservation laws can be considered separately for convenience. Therefore, one can express

$$\vec{\mathbf{f}} \cdot \vec{\mathrm{n}} = \mathbf{g} + \mathbf{h} \tag{2.6}$$

where $\mathbf{g}$ defines the inviscid terms and $\mathbf{h}$ defines the viscous terms. The formulation of these terms follows in sections 3 and 4.

3. Formulation of Inviscid Terms

The inviscid flux vector at cell face l is defined as

$$\mathbf{g}_l = \{a'_l\mathbf{g}_{L,l} + b'_l\mathbf{g}_{L-1,l}\} - \left\{\frac{1}{2\left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l}\mathbf{R}_l|\mathbf{\Lambda}_l|\left[\mathbf{s}_l - \theta\mathbf{s}_l^{\min}\right]\right\} \tag{3.1}$$

where

$$\mathbf{g}_{L,l} = \left[\vec{\mathbf{f}}_L \cdot \vec{\mathrm{n}}_l\right]_{\mathrm{inv}}$$

The first term in braces is a second-order accurate distance-weighted interpolation formula for $\mathbf{g}_l$. The factors a'_l and b'_l are geometric weighting functions. These functions, defined as follows, account for the relative position of the cell wall with respect to the cell centers:

$$a'_l = \frac{D_{L-1}}{D_L + D_{L-1}} \tag{3.2a}$$

$$b'_l = \frac{D_L}{D_L + D_{L-1}} = 1 - a'_l \tag{3.2b}$$

$$D_L = \frac{\Omega_L}{\sigma_{l+1} + \sigma_l} \tag{3.2c}$$

Geometric weighting improves computed profiles through highly stretched grid regions. An earlier version of the algorithm (ref. 21) used volume weighting, in which $D_L = \Omega_L$. Volume weighting tends to magnify the contribution of cells near the axis singularity of spherical coordinate systems.

The second term in braces provides the upwind-biased numerical dissipation. It is a first-order dissipation when θ equals 0 and it is a second-order dissipation when θ equals 1. The variables which define both first- and second-order dissipations are discussed in sections 3.1 and 3.2.

3.1. First-Order Numerical Dissipation

The vector $\mathbf{s}_l$ is defined as

$$\mathbf{s}_l = \left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l \mathbf{R}_l^{-1}(\mathbf{q}_L - \mathbf{q}_{L-1}) \tag{3.3}$$

The matrix $\mathbf{R}_l^{-1}$ in equation (3.3) and the matrices $\mathbf{R}_l$ and $\mathbf{\Lambda}_l$ in equation (3.1) are related to the Jacobian of the inviscid flux vector $\mathbf{g}$ with respect to $\mathbf{q}$ in the following manner.

$$\mathbf{A} = \frac{\partial \mathbf{g}}{\partial \mathbf{q}} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1} \tag{3.4}$$

The matrices $\mathbf{R}$ and $\mathbf{R}^{-1}$ are composed of the right and left eigenvectors, respectively, of $\mathbf{A}$, and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}$. These matrices are defined in appendix B. The matrix $|\mathbf{\Lambda}|$ is a diagonal matrix containing the absolute values of the eigenvalues of $\mathbf{A}$ with constraints on the minimum-allowed magnitude of an eigenvalue described in section 6.1.

The term $\left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l$ is defined in appendix A. It is related to the ratio of a cell wall area to cell volume and may be thought of as the inverse of the projected distance between cell centers L and $L-1$ on a direction normal to cell face l. The variable χ is a generic computational coordinate running in the direction of increasing generic index L. Given these definitions, the vector $\mathbf{s}_l$ may be thought of as an approximation to the gradient of the vector of characteristic variables (Riemann invariants) in the direction normal to cell face l. Note that the term $\left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l$ in the definition of $\mathbf{s}_l$ is canceled by the leading factor of the dissipation term in equation (3.1). This leading factor serves to rescale contributions to the second-order dissipation and is discussed in more detail in section 3.2.

The upwind-biased nature of the dissipation term may be understood by considering the following difference relation that derives from equation (3.4):

$$\mathbf{g}_{L,l} - \mathbf{g}_{L-1,l} \approx \mathbf{A}_l(\mathbf{q}_L - \mathbf{q}_{L-1}) = \frac{1}{\left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l}\mathbf{R}_l\mathbf{\Lambda}_l\mathbf{s}_l \tag{3.5}$$

The elements of $\mathbf{A}_l$ or its associated factors $\mathbf{R}_l$, $\mathbf{\Lambda}_l$, and $\mathbf{R}_l^{-1}$ must be determined by some suitable average of the dependent variables at the adjacent cell centers L and $L-1$. Equation (3.5) is only approximately satisfied for general averaging schemes because of the nonlinear relation between $\mathbf{g}$ and $\mathbf{q}$. However, Roe (ref. 2) has introduced an averaging scheme which exactly satisfies equation (3.5) for a perfect gas. This scheme is defined in appendix B.

The first-order formulation of $\mathbf{g}_l$ on an equally spaced grid can now be simplified as follows:

$$\mathbf{g}_l = \frac{1}{2}\left[\mathbf{g}_{L,l} + \mathbf{g}_{L-1,l} - \mathbf{R}_l|\mathbf{\Lambda}_l|\mathbf{R}_l^{-1}\left(\mathbf{q}_L - \mathbf{q}_{L-1}\right)\right] \qquad \text{(First-order, equal spacing)} \tag{3.6}$$

The eigenvalues of $\mathbf{\Lambda}$, as defined in appendix B, are equal to U, $U+a$, and $U-a$ where U is the normal velocity component through cell face l and a is the sound speed. Consequently, if the flow through cell face l is supersonic and positive (moving from cell center $L-1$ to cell center L), then all the eigenvalues are positive, $|\mathbf{\Lambda}_l| = \mathbf{\Lambda}_l$, and equation (3.6) reduces to the following relation:

$$\mathbf{g}_l = \frac{1}{2}\left[\mathbf{g}_{L,l} + \mathbf{g}_{L-1,l} - \left(\mathbf{g}_{L,l} - \mathbf{g}_{L-1,l}\right)\right] = \mathbf{g}_{L-1,l} \qquad \text{(Supersonic, positive)} \tag{3.7a}$$

In like manner, if the flow through cell face l is supersonic and negative (moving from cell center L to cell center $L-1$) then all the eigenvalues are negative, $|\mathbf{\Lambda}_l| = -\mathbf{\Lambda}_l$, and equation (3.6)

reduces to the following relation:

$$\mathbf{g}_l = \frac{1}{2}\left[\mathbf{g}_{L,l} + \mathbf{g}_{L-1,l} + (\mathbf{g}_{L,l} - \mathbf{g}_{L-1,l})\right] = \mathbf{g}_{L,l} \qquad \text{(Supersonic, negative)} \tag{3.7b}$$

Intermediate situations involving subsonic flow split the contributions to the projected Riemann invariants from the right or left according to the sign of the associated eigenvalue. In the most general cases, when there are unequal cell sizes, eigenvalues close to zero (to be explained in section 6), or second-order modifications, the "upwind" approximation to the inviscid flux vector is not quite so exact. However, the present formulation retains the strong stability characteristic of upwind schemes over a range of numerical tests that challenge the computation.

3.2. Second-Order Numerical Dissipation

A total variation-diminishing (TVD) scheme is an algorithm which is guaranteed to prevent the introduction of any new maxima or minima in the evolving distribution of a function or functions determined by nonlinear scalar hyperbolic conservation laws or constant coefficient hyperbolic systems. The scheme can be second-order accurate and is oscillation free across computed discontinuities under these conditions (ref. 10). Although the time-dependent Euler equations constitute a nonlinear hyperbolic system, TVD formulations for linear hyperbolic systems generally produce oscillation-free solutions when applied to this more complex equation set. Conditions for TVD are usually satisfied through the introduction of a limiting procedure within the algorithm. For example, several classes of limiters are discussed in references 12 and 27. The limiters used in the present work were introduced by Yee (refs. 10 and 28). They involve symmetric functions of gradients in the neighborhood of the cell face, and algorithms based on these limiters are referred to as symmetric total variation-diminishing (STVD) schemes.

STVD schemes involve little extra programming work over simple first-order algorithms because most of the quantities required in their implementation are already available, as can be seen in the subsequent formulations:

$$\mathbf{s}_l^{\min} = \min\text{mod}\,(\mathbf{s}_{l+1}, \mathbf{s}_l) + \min\text{mod}\,(\mathbf{s}_{l-1}, \mathbf{s}_l) - \mathbf{s}_l \tag{3.8a}$$

$$\mathbf{s}_l^{\min} = \min\text{mod}\,(\mathbf{s}_{l+1}, \mathbf{s}_l, \mathbf{s}_{l-1}) \tag{3.8b}$$

$$\mathbf{s}_l^{\min} = \min\text{mod}\left(2\mathbf{s}_{l+1}, 2\mathbf{s}_l, 2\mathbf{s}_{l-1}, \frac{1}{2}\left(\mathbf{s}_{l-1} + \mathbf{s}_{l+1}\right)\right) \tag{3.8c}$$

where the min mod function returns the argument of smallest absolute magnitude when all the arguments are of the same sign or returns 0 if the arguments are of opposite sign. (If all arguments are positive, then a positive result is returned. If all arguments are negative, then a negative result is returned.) Also note that equations (3.8) are evaluated element by element of vector $\vec{\mathbf{s}}$.

The STVD limiter does not yield a strictly upwind bias on the formulation of the flux vector because the functional form of equations (3.8) allows for both upstream and downstream dependencies. Also, the scheme reduces to first order at cell faces where there is a sign change in the arguments of the min mod function. The present formulation differs from that of reference 28 in that the differences, $\mathbf{R}_l^{-1}(\mathbf{q}_L - \mathbf{q}_{L-1})$, have been scaled by $(\vec{\nabla}\chi \cdot \vec{\mathrm{n}})_l$. This scaling, which is eventually removed by the leading factor of the numerical dissipation term in braces, reduces the effects of grid stretching on the argument returned by the min mod function. The present treatment has been found to be well suited to the range of numerical test problems considered herein.

The dissipation term is defined by the product of the leading factor $\frac{1}{(\vec{\nabla}\chi \cdot \vec{n})_l}$ (which is a first-order difference in the distance between adjacent cell centers projected on the normal to the common cell face) and the difference in square brackets $[\mathbf{s}_l - \theta \mathbf{s}_l^{\min}]$ (which is zero order if θ or $\mathbf{s}_l^{\min}$ equals 0 and is first order otherwise). Thus, the dissipation term is proportional to a difference squared when $\theta = 1$ except for the isolated points in the flow described earlier.

3.3. Point-Implicit Relaxation of Inviscid Terms

Equation (3.1) can be approximately linearized with respect to $\delta \mathbf{q}_L$ in the following manner. Define

$$\mathbf{g}_{l,L}^* = \left\{ \frac{1}{2}\mathbf{g}_{L,l}^{n+1} + \left(a_l' - \frac{1}{2}\right)\mathbf{g}_{L,l}^n + b_l' \mathbf{g}_{L-1,l}^* \right\} - \left\{ \frac{1}{2(\vec{\nabla}\chi \cdot \vec{n})_l} \mathbf{R}_l |\mathbf{\Lambda}_l| \left[\left(\vec{\nabla}\chi \cdot \vec{n}\right)_l \mathbf{R}_l^{-1} \left(\mathbf{q}_L^{n+1} - \mathbf{q}_{L-1}^*\right) - \theta \mathbf{s}_l^{*\min} \right] \right\} \tag{3.9}$$

where superscript n refers to the current value at cell center L, superscript $n+1$ refers to the new value to be computed at cell center L, and superscript * refers to the latest available value at neighbor cell $L-1$. The notation $\mathbf{g}_{l,L}^*$ refers to the inviscid flux through cell face l evaluated with the latest available data from cell center $L-1$ and the predicted data at cell center L. Elements of the matrices $\mathbf{R}_l$, $\mathbf{\Lambda}_l$, and $\mathbf{R}_l^{-1}$ are computed by using Roe's averaging of the current data at cell centers L and $L-1$. Elements of the vector $(\mathbf{s}_l^*)^{\min}$ are also computed by using current data at cell centers L and $L-1$. Substitute $\mathbf{g}_{L,l}^n + \mathbf{A}_{L,l}\delta\mathbf{q}_L$ for $\mathbf{g}_{L,l}^{n+1}$ and $\mathbf{q}_L^n + \delta\mathbf{q}_L$ for $\mathbf{q}_L^{n+1}$ in equation (3.9) to obtain

$$\mathbf{g}_{l,L}^* = \mathbf{g}_l + \frac{1}{2}\left(\mathbf{A}_{L,l} - |\mathbf{A}_l|\right)\delta\mathbf{q}_L \tag{3.10a}$$

where $|\mathbf{A}_l| = \mathbf{R}_l|\mathbf{\Lambda}_l|\mathbf{R}_l^{-1}$. In like manner, one can show that

$$\mathbf{g}_{l+1,L}^* = \mathbf{g}_{l+1} + \frac{1}{2}(\mathbf{A}_{L,l+1} + |\mathbf{A}_{l+1}|)\delta\mathbf{q}_L \tag{3.10b}$$

The point-implicit discretization of the inviscid part of equation (2.5) can now be expressed by

$$\frac{\delta\mathbf{q}_L\Omega_L}{\delta t} = -\sum_{l=i,j,k}\left[\mathbf{g}_{l+1,L}^*\sigma_{l+1} - \mathbf{g}_{l,L}^*\sigma_l\right] \tag{3.11}$$

Use the result of equations (3.10) in equation (3.11) and combine terms to obtain

$$\left\{\mathbf{I} + \frac{\delta t}{2\Omega_L}\sum_{l=i,j,k}\left[\left(\mathbf{A}_{L,l+1} + |\mathbf{A}_{l+1}|\right)\sigma_{l+1} - \left(\mathbf{A}_{L,l} - |\mathbf{A}_l|\right)\sigma_l\right]\right\}\delta\mathbf{q}_L = -\frac{\delta t}{\Omega_L}\sum_{l=i,j,k}\left[\mathbf{g}_{l+1}\sigma_{l+1} - \mathbf{g}_l\sigma_l\right] \tag{3.12}$$

An application of Stokes' theorem to the summation of $\mathbf{A}_{L,l}$ and $\mathbf{A}_{L,l+1}$ in equation (3.12) shows that

$$\sum_{l=i,j,k} \left[\mathbf{A}_{L,l+1}\sigma_{l+1} - \mathbf{A}_{L,l}\sigma_l\right] \delta\mathbf{q}_L = \sum_{l=i,j,k} \left[\delta\mathbf{g}_{L,l+1}\sigma_{l+1} - \delta\mathbf{g}_{L,l}\sigma_l\right]$$

$$= \sum_{l=i,j,k} \delta\vec{\mathbf{f}}_{L,\text{inv}} \cdot \left[\vec{\mathbf{n}}_{l+1}\sigma_{l+1} - \vec{\mathbf{n}}_l\sigma_l\right] = 0 \qquad (3.13)$$

Therefore, equation (3.12) simplifies to the following relaxation equation:

$$\left\{\mathbf{I} + \frac{\delta t}{\Omega_L}\mathbf{M}_{L,\text{inv}}\right\} \delta\mathbf{q}_L = -\frac{\delta t}{\Omega_L} \sum_{l=i,j,k} \left[\mathbf{g}_{l+1}\sigma_{l+1} - \mathbf{g}_l\sigma_l\right] \qquad (3.14)$$

where

$$\mathbf{M}_{L,\text{inv}} = \frac{1}{2} \sum_{l=i,j,k} \left[|\mathbf{A}_{l+1}|\sigma_{l+1} + |\mathbf{A}_l|\sigma_l\right] \qquad (3.15)$$

4. Formulation of the Viscous Terms

4.1. Evaluation of Shear Stresses and Conduction

The viscous stresses on a cell wall with unit normal $\vec{\mathrm{n}}$ in the orthogonal directions $\vec{\mathrm{n}}$, $\vec{\mathrm{l}}$, and $\vec{\mathrm{m}}$ are given by

$$\tau_{\mathrm{nn}} = \frac{\lambda_l}{\mathrm{Re}}\left(\frac{\partial U}{\partial \mathrm{n}} + \frac{\partial V}{\partial \mathrm{l}} + \frac{\partial W}{\partial \mathrm{m}}\right) + \frac{2\mu_l}{\mathrm{Re}}\frac{\partial U}{\partial \mathrm{n}} \qquad (4.1a)$$

$$\tau_{\mathrm{nl}} = \frac{\mu_l}{\mathrm{Re}}\left(\frac{\partial V}{\partial \mathrm{n}} + \frac{\partial U}{\partial \mathrm{l}}\right) \qquad (4.1b)$$

$$\tau_{\mathrm{nm}} = \frac{\mu_l}{\mathrm{Re}}\left(\frac{\partial W}{\partial \mathrm{n}} + \frac{\partial U}{\partial \mathrm{m}}\right) \qquad (4.1c)$$

where U, V, and W are velocity components and n, l, and m are arc lengths in the $\vec{\mathrm{n}}$, $\vec{\mathrm{l}}$, and $\vec{\mathrm{m}}$ directions, respectively. The variables μ and λ are the viscosity coefficients.

The derivative of some quantity f with respect to arc length in the $\vec{\mathrm{n}}$ direction, for example, is expressed as

$$\frac{\partial f}{\partial \mathrm{n}} = \vec{\nabla} f \cdot \vec{\mathrm{n}} = \frac{\partial f}{\partial x}\mathrm{n}_x + \frac{\partial f}{\partial y}\mathrm{n}_y + \frac{\partial f}{\partial z}\mathrm{n}_z \qquad (4.2)$$

Expanding the partial of f with respect to x, y, and z in terms of the natural coordinates ξ, η, and ζ yields

$$\frac{\partial f}{\partial \mathrm{n}} = \left[\frac{\partial f}{\partial \xi}\vec{\nabla}\xi + \frac{\partial f}{\partial \eta}\vec{\nabla}\eta + \frac{\partial f}{\partial \zeta}\vec{\nabla}\zeta\right] \cdot \vec{\mathrm{n}} \qquad (4.3a)$$

Similar expressions, presented below, can be derived for the derivatives of f with respect to l and m:

$$\frac{\partial f}{\partial \mathrm{l}} = \left[\frac{\partial f}{\partial \xi}\vec{\nabla}\xi + \frac{\partial f}{\partial \eta}\vec{\nabla}\eta + \frac{\partial f}{\partial \zeta}\vec{\nabla}\zeta\right] \cdot \vec{\mathrm{l}} \qquad (4.3b)$$

$$\frac{\partial f}{\partial \mathrm{m}} = \left[\frac{\partial f}{\partial \xi}\vec{\nabla}\xi + \frac{\partial f}{\partial \eta}\vec{\nabla}\eta + \frac{\partial f}{\partial \zeta}\vec{\nabla}\zeta\right] \cdot \vec{\mathrm{m}} \qquad (4.3c)$$

Expanding the derivatives in equations (4.1) according to the formula in equations (4.3) and combining like terms yields the following relations for shear stresses:

$$\tau_{nn} = \frac{(2\mu+\lambda)_l}{Re}\left(\frac{\partial U}{\partial \xi}\vec{\nabla}\xi + \frac{\partial U}{\partial \eta}\vec{\nabla}\eta + \frac{\partial U}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{n}$$
$$+ \frac{\lambda_l}{Re}\left[\left(\frac{\partial V}{\partial \xi}\vec{\nabla}\xi + \frac{\partial V}{\partial \eta}\vec{\nabla}\eta + \frac{\partial V}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{l} + \left(\frac{\partial W}{\partial \xi}\vec{\nabla}\xi + \frac{\partial W}{\partial \eta}\vec{\nabla}\eta + \frac{\partial W}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{m}\right] \quad (4.4a)$$

$$\tau_{nl} = \frac{\mu_l}{Re}\left[\left(\frac{\partial V}{\partial \xi}\vec{\nabla}\xi + \frac{\partial V}{\partial \eta}\vec{\nabla}\eta + \frac{\partial V}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{n} + \left(\frac{\partial U}{\partial \xi}\vec{\nabla}\xi + \frac{\partial U}{\partial \eta}\vec{\nabla}\eta + \frac{\partial U}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{l}\right] \quad (4.4b)$$

$$\tau_{nm} = \frac{\mu_l}{Re}\left[\left(\frac{\partial W}{\partial \xi}\vec{\nabla}\xi + \frac{\partial W}{\partial \eta}\vec{\nabla}\eta + \frac{\partial W}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{n} + \left(\frac{\partial U}{\partial \xi}\vec{\nabla}\xi + \frac{\partial U}{\partial \eta}\vec{\nabla}\eta + \frac{\partial U}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{m}\right] \quad (4.4c)$$

The component of shear stress acting in the s direction (s being a dummy variable for x, y, or z) acting on a cell wall with unit norm $\vec{n}$ can be expressed

$$\tau_{ns} = \tau_{nn}n_s + \tau_{nl}l_s + \tau_{nm}m_s \quad (4.5)$$

Substitution of equations (4.4) into (4.5), collecting terms, and simplifying according to the geometric identities provided by equations (A14) and (A15) yield the following relation for shear stress in the s direction:

$$\tau_{ns} = \frac{\mu_l}{Re}\left(\left(\frac{\partial \nu}{\partial \xi}\vec{\nabla}\xi + \frac{\partial \nu}{\partial \eta}\vec{\nabla}\eta + \frac{\partial \nu}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot \vec{n} + \frac{\partial U}{\partial \xi}\frac{\partial \xi}{\partial s} + \frac{\partial U}{\partial \eta}\frac{\partial \eta}{\partial s} + \frac{\partial U}{\partial \zeta}\frac{\partial \zeta}{\partial s}\right)$$
$$+ \frac{\lambda_l}{Re}\left(\frac{\partial \vec{u}}{\partial \xi}\cdot\vec{\nabla}\xi + \frac{\partial \vec{u}}{\partial \eta}\cdot\vec{\nabla}\eta + \frac{\partial \vec{u}}{\partial \zeta}\cdot\vec{\nabla}\zeta\right)n_s \quad (4.6)$$

where ν is a dummy variable for u, v, or w corresponding to $s = x$, y, or z. Observe that there is no functional dependence on the choice of tangential directions $\vec{l}$ and $\vec{m}$ in equation (4.6).

A thin-layer approximation in the χ coordinate direction ($\chi = \xi$, η, or ζ) simplifies equation (4.6) by neglecting derivatives in the other two coordinate directions. Consequently,

$$\tau'_{ns} = \frac{\mu_l}{Re}\left(\frac{\partial \nu}{\partial \chi}\vec{\nabla}\chi\cdot\vec{n} + \frac{\partial U}{\partial \chi}\frac{\partial \chi}{\partial s}\right) + \frac{\lambda_l}{Re}\left(\frac{\partial \vec{u}}{\partial \chi}\cdot\vec{\nabla}\chi\right)n_s \quad (4.7)$$

where n refers to the direction normal to a surface χ = Constant, and the superscript prime refers to the thin-layer approximation. The viscous terms on the other two coordinate surfaces are also neglected in the thin-layer approximation because their contribution to overall momentum and energy balance is small. These approximations are valid as long as the boundary layer is relatively thin and the χ direction is approximately normal to the high gradient region.

Equation (4.7) can be further simplified through the use of the geometric relations defined by equations (A13) and (A15) and Stokes' relation (ref. 29), $\lambda = -\frac{2\mu}{3}$ to yield

$$\tau'_{ns} = \frac{\mu_l}{Re}\left(\frac{\partial \nu}{\partial \chi} + \frac{1}{3}\frac{\partial U}{\partial \chi}n_s\right)\vec{\nabla}\chi\cdot\vec{n} \quad (4.8)$$

The heat flux through a cell wall is expressed as

$$\dot{q}_n = \frac{\mu_l}{Re\ Pr}\vec{\nabla}(\gamma e)\cdot\vec{n}$$
$$= \frac{\mu_l}{Re\ Pr}\left(\frac{\partial \gamma e}{\partial \xi}\vec{\nabla}\xi + \frac{\partial \gamma e}{\partial \eta}\vec{\nabla}\eta + \frac{\partial \gamma e}{\partial \zeta}\vec{\nabla}\zeta\right)\cdot\vec{n} \quad (4.9)$$

The thin-layer approximation to $\dot{q}_n$ is expressed as

$$\dot{q}'_{\mathrm{n}} = \frac{\mu_l}{\mathrm{Re\ Pr}} \left(\frac{\partial \gamma e}{\partial \chi} \vec{\nabla} \chi \cdot \vec{\mathrm{n}} \right) \tag{4.10}$$

Let l be the index running in the χ direction. Recall that lowercase l refers to cell walls and uppercase L refers to cell centers. The solution at cell center L requires evaluation of the viscous terms on cell walls l and $l+1$. The viscous part of $\vec{\mathbf{f}}_l \cdot \vec{\mathrm{n}}_l$ is now written as

$$\mathbf{h}_l = - \begin{bmatrix} 0 \\ \tau_{\mathrm{n}x} \\ \tau_{\mathrm{n}y} \\ \tau_{\mathrm{n}z} \\ \mathrm{u}\tau_{\mathrm{n}x} + \mathrm{v}\tau_{\mathrm{n}y} + \mathrm{w}\tau_{\mathrm{n}z} + \dot{q}_{\mathrm{n}} \end{bmatrix} \tag{4.11}$$

In the case of the thin-layer approximation, $\vec{\mathbf{h}}_l$ would be a function of τ' and $\dot{q}'_{\mathrm{n}}$. The viscosity, μ_l, is calculated as the average of μ_L and μ_{L+1}. Expressions for the metrics to second-order accuracy are presented in appendix A.

4.2. Point-Implicit Relaxation of Viscous Terms

Derivatives in the χ direction are evaluated to second-order accuracy in space as follows:

$$\left(\frac{\partial \mathrm{u}}{\partial \chi} \right)^*_{l,L} = \mathrm{u}^{n+1}_L - \mathrm{u}^*_{L-1} = \mathrm{u}^n_L + \delta \mathrm{u}_L - \mathrm{u}^*_{L-1} = \left(\frac{\partial \mathrm{u}}{\partial \chi} \right)_l + \delta \mathrm{u}_L \tag{4.12a}$$

$$\left(\frac{\partial \mathrm{u}}{\partial \chi} \right)^*_{l+1,L} = \mathrm{u}^*_{L+1} - \mathrm{u}^{n+1}_L = \mathrm{u}^*_{L+1} - \mathrm{u}^n_L - \delta \mathrm{u}_L = \left(\frac{\partial \mathrm{u}}{\partial \chi} \right)_{l+1} - \delta \mathrm{u}_L \tag{4.12b}$$

Terms like $\frac{\partial \mathrm{u}}{\partial \eta}$ for χ equal to ξ are evaluated as follows assuming a rectangular ordering of mesh points:

$$\left(\frac{\partial \mathrm{u}}{\partial \eta} \right)_{i+1,J,K} = \frac{1}{4} \left(\mathrm{u}^*_{I,J+1,K} - \mathbf{u}^*_{I,J-1,K} + \mathrm{u}^*_{I+1,J+1,K} - \mathbf{u}^*_{I+1,J-1,K} \right) \tag{4.13}$$

Note that the derivatives in the directions along the face (i.e., those derivatives neglected in the thin-layer approximation) have no functional dependence on the cell center. Therefore, the point-implicit treatment of the full Navier-Stokes equations is identical to that of the thin-layer Navier-Stokes equations.

Now, define $\mathbf{h}_l$ as a function of differences evaluated with currently available data, for example $\left(\frac{\partial \mathrm{u}}{\partial \chi} \right)_l$, and define $\mathbf{h}^*_{l,L}$ as a function of differences by using predicted values at cell center L, for example $\left(\frac{\partial \mathrm{u}}{\partial \chi} \right)^*_{l,L}$. These definitions permit the linearization of the viscous terms to be expressed as follows:

$$\mathbf{h}^*_{l,L} = \mathbf{h}_l - \mathbf{B}_{l,L}\ \delta \mathbf{q}_L \tag{4.14a}$$

$$\mathbf{h}^*_{l+1,L} = \mathbf{h}_{l+1} + \mathbf{B}_{l+1,L}\ \delta \mathbf{q}_L \tag{4.14b}$$

where

$$\mathbf{B}_{l,L} = -\frac{\partial \mathbf{h}^*_{l,L}}{\partial \mathbf{q}_L} \equiv -\frac{\partial \mathbf{h}'^*_{l,L}}{\partial \mathbf{q}_L} \tag{4.15a}$$

$$\mathbf{B}_{l+1,L} = \frac{\partial \mathbf{h}^*_{l+1,L}}{\partial \mathbf{q}_L} \equiv \frac{\partial \mathbf{h}'^*_{l+1,L}}{\partial \mathbf{q}_L} \tag{4.15b}$$

Further details of the definition of matrix $\mathbf{B}_{l,L}$ and $\mathbf{B}_{l+1,L}$ are presented in appendix C.

The point-implicit implementation of the viscous terms follows the example set in equations (3.11) through (3.15).

$$\frac{\delta \mathbf{q}_L \Omega_L}{\delta t} = -\sum_{l=i,j,k} \left[\mathbf{h}^*_{l+1,L} \sigma_{l+1} - \mathbf{h}^*_{l,L} \sigma_l \right] \tag{4.16}$$

Note that in the case of the thin-layer Navier-Stokes equations, the summation would only include one of the i, j, or k directions, depending on the orientation of the computational coordinates with the body. Expand equation (4.16) and combine terms to obtain

$$\left\{ \mathbf{I} + \frac{\delta t}{\Omega_L} \mathbf{M}_{L,\text{vis}} \right\} \delta \mathbf{q}_L = -\frac{\delta t}{\Omega_L} \sum_{l=i,j,k} \left[\mathbf{h}_{l+1} \sigma_{l+1} - \mathbf{h}_l \sigma_l \right] \tag{4.17}$$

where

$$\mathbf{M}_{L,\text{vis}} = \sum_{l=i,j,k} \left[\mathbf{B}_{l+1,L} \sigma_{l+1} + \mathbf{B}_{l,L} \sigma_l \right] \tag{4.18}$$

5. Relaxation Strategies

The governing relaxation equation for both the inviscid and viscous components of the Navier-Stokes equations is obtained by combining the results of equations (3.14) and (4.17). Thus,

$$\left\{ \mathbf{I} + \frac{\delta t}{\Omega_L} \mathbf{M}_L \right\} \delta \mathbf{q}_L = \mathbf{r}_L \tag{5.1}$$

where

$$\begin{aligned} \mathbf{M}_L &= \mathbf{M}_{L,\text{inv}} + \mathbf{M}_{L,\text{vis}} \\ &= \sum_{l=i,j,k} \left[\left(\frac{1}{2} |\mathbf{A}_{l+1}| + \mathbf{B}_{l+1,L} \right) \sigma_{l+1} + \left(\frac{1}{2} |\mathbf{A}_l| + \mathbf{B}_{l,L} \right) \sigma_l \right] \end{aligned} \tag{5.2a}$$

and

$$\mathbf{r}_L = -\frac{\delta t}{\Omega_L} \sum_{l=i,j,k} \left[(\mathbf{g}_{l+1} + \mathbf{h}_{l+1}) \sigma_{l+1} - (\mathbf{g}_l + \mathbf{h}_l) \sigma_l \right] \tag{5.2b}$$

The relaxed value of $\mathbf{q}_L^{n+1}$ can now be determined by

$$\mathbf{q}_L^{n+1} = \mathbf{q}_L^n + \left[\mathbf{I} + \frac{\delta t}{\Omega_L} \mathbf{M}_L \right]^{-1} \mathbf{r}_L \tag{5.3}$$

Equation (5.3) involves the inversion of a single 5×5 matrix for three-dimensional flow of a perfect gas. In practice, Gauss elimination is used to solve for $\delta \mathbf{q}_L$ directly from equation (5.1). Note that $\mathbf{M}_{L,\text{inv}}$ is obtained as the sum of matrices that have all real, positive eigenvalues and

that $\mathbf{M}_{L,\text{vis}}$ is obtained as the sum of matrices that have all real, nonnegative eigenvalues. Such summations will not generally result in a matrix with real, positive eigenvalues. However, the Gauss elimination is routinely implemented without pivoting to facilitate code vectorization in the numerical tests, and no evidence of ill-posed behavior has been observed.

The strategy used to drive the right-hand side of equation (5.1) to zero should take advantage of the host computer architecture and the physics of the problem. On computers with serial architecture, the relaxation of equation (5.1) (with one or more local iterations) proceeds from cell to cell in an ordered fashion. Thus, an updated value of $\mathbf{q}^{n+nlocal}$ is obtained at cell center (1,1,1) followed by (2,1,1), (3,1,1), ... (1,2,1), ... until the dependent variables at every cell in the domain have been locally iterated *nlocal* times using the most recent available data from neighbor cells. Vectorizable code is far more efficient than scalar code on machines with vector architecture. In this case, all the cells in a plane of the domain are iterated *nlocal* times before proceeding to the next plane.

Numerical tests indicate that sweeps which run from a wall across the boundary layer to the opposite boundary and then back again are the most efficient for the blunt body problem. Effects of a perturbation at a wall are felt at the opposite boundary after one sweep. Effects of a perturbation at one cell in a plane parallel to the wall require n iterations to be felt by a cell whose index differs from the source cell by n.

The ordering of the sweeps and the use of local iterations may be used to speed convergence, but in numerical tests performed to date, they do not affect the final, converged steady-state solution. Thus, it should be possible to solve a large number of cells using a massively parallel processing computer in which each cell (or small group of cells) is relaxed semi-independently of its neighbor cells (cell groups) using its own processor. The expression semi-independently means that a cell (cell group) will need updated information from its neighbor cells (cell groups), but neither the order that it receives this information nor the lag time it takes for this information to arrive is critically important. (There may be some upper limit on allowable communication delays on actual parallel systems before convergence is inhibited. This issue is beyond the scope of the present paper.) As long as each processor has immediate access to some level of information from its neighbors (which could be stored locally), the execution stream could proceed uninterrupted in a parallel, asynchronous mode (ref. 30).

6. Numerical Issues

6.1. Zero Eigenvalues and Entropy Violations

Harten (ref. 7) has shown that the numerical dissipation for waves associated with eigenvalues equal to zero is also zero when the inviscid flux is defined as in equation (3.9). Thus, incorrect nonphysical weak solutions of the governing conservation laws which violate the second law of thermodynamics (i.e., expansion shocks) are permitted. In the present algorithm it has been found that if no provision has been made for zero eigenvalues, the scheme will either fail to converge or will converge to a nonunique solution (usually on a coarse grid) that is relaxation path dependent. A simple fix for this problem, suggested by Harten (ref. 7), is to restrict the minimum value of the eigenvalue magnitude. Therefore,

$$\tilde{\lambda}_{ieqn} = \begin{cases} |\lambda_{ieqn}| & (|\lambda_{ieqn}| \geq 2\epsilon) \\ \frac{(\lambda_{ieqn})^2}{4\epsilon} + \epsilon & (|\lambda_{ieqn}| < 2\epsilon) \end{cases} \tag{6.1}$$

and

$$|\mathbf{\Lambda}| = \begin{bmatrix} \tilde{\lambda}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 & 0 & 0 \\ 0 & 0 & \tilde{\lambda}_3 & 0 & 0 \\ 0 & 0 & 0 & \tilde{\lambda}_4 & 0 \\ 0 & 0 & 0 & 0 & \tilde{\lambda}_5 \end{bmatrix} \tag{6.2}$$

where λ_{ieqn} is an eigenvalue of $\mathbf{A}$ and is defined in appendix B.

The magnitude of ϵ, which may be viewed as a nondimensional velocity, is problem dependent. It is also a function of the method used to nondimensionalize the problem. Numerical tests on the blunt body problems, in which there are large zones of subsonic flow and stagnation points, indicate that ϵ should be restricted by

$$0.1 \leq \epsilon \leq 0.3 \qquad \text{(Approximate range for blunt bodies)} \tag{6.3a}$$

Numerical tests on flows that are predominantly supersonic indicate that ϵ should be restricted by

$$0.005 \leq \epsilon \leq 0.05 \qquad \text{(Approximate range for slender bodies)} \tag{6.3b}$$

The lower limits on ϵ arise from convergence or uniqueness considerations. The upper limits arise from concerns regarding excessive numerical dissipation and its effects on predicted surface heating, adiabatic wall temperatures, and shock thicknesses. Large values of ϵ have also been observed to slow convergence. In some cases for the blunt body problem, the value of ϵ was gradually reduced to 0 for cell walls parallel to the body. This treatment improved heat-transfer predictions and will be discussed further in section 7.

Recently, Yee (ref. 12) has suggested a functional dependence of ϵ on the local values of sound speed and velocity. This relation has been adapted for use in the present work as follows:

$$\epsilon = \epsilon_o(a_l + |U_l| + |V_l| + |W_l|) \tag{6.4}$$

where ϵ_o is a constant which generally varies from 0.01 to 0.4. The magnitude of this quantity follows similar guidelines as defined in equation (6.3); however, the resultant magnitude of ϵ, which is now tied to local velocities, has been found to be generally less dissipative than the use of equation (6.3).

6.2. Relaxation Factors and Frozen Matrices

Equation (3.14) has been found to be unstable when $\theta = 1$ (second-order dissipation) and the Courant number is greater than 1. (The Courant number, which determines the magnitude of δt, is defined in appendix D.) Because time accuracy is not an issue in the present formulation, one is free to adjust $\mathbf{M}_L$ in any way which improves stability, accelerates convergence, or decreases computational time per relaxation step. The instability can be removed by multiplying $\mathbf{M}_{L,\text{inv}}$ by a relaxation factor Φ_{inv} as follows.

$$\mathbf{M}'_{L,\text{inv}} = \Phi_{\text{inv}}\mathbf{M}_{L,\text{inv}} \tag{6.5}$$

where the recommended bounds on the relaxation factor as determined by numerical tests are given by

$$1.5 \leq \Phi_{\text{inv}} \leq 2.0 \tag{6.6}$$

Examination of equation (3.14) shows that the application of the relaxation factor reduces the magnitude of $\delta\mathbf{q}_L$ for a given value of the right-hand-side residual. Thus, Φ_{inv} may be viewed

as an underrelaxation factor. It also serves to better approximate the contribution of the flux limiter to $\mathbf{r}_L$.

No instabilities have been encountered that are associated with the present treatment of the viscous terms. However, an overrelaxation factor associated with the viscous contributions to $\mathbf{r}_L$ can be specified as follows:

$$\mathbf{M}'_{L,\text{vis}} = \Phi_{\text{vis}}\mathbf{M}_{L,\text{vis}} \tag{6.7}$$

where the recommended bounds on the relaxation factor as determined by numerical tests are given by

$$0.5 \leq \Phi_{\text{vis}} \tag{6.8}$$

Convergence rates as a function of Φ_{inv} and Φ_{vis} are presented in section 7.

Equation (5.2) for $\mathbf{M}_L$ is now replaced by

$$\mathbf{M}'_L = \mathbf{M}'_{L,\text{inv}} + \mathbf{M}'_{L,\text{vis}} \tag{6.9}$$

and equation (5.3) for $\mathbf{q}_L^{n+1}$ is replaced by

$$\mathbf{q}_L^{n+1} = \mathbf{q}_L^n - \mathbf{C}_L\mathbf{r}_L \tag{6.10}$$

where

$$\mathbf{C}_L = \left[\mathbf{I} + \frac{\delta t}{\Omega_L}\mathbf{M}'_L\right]^{-1} \tag{6.11}$$

Note that $\mathbf{C}_L$ does not need to be recomputed every relaxation step. In fact, an advantage of this formulation is that it requires no more memory or work to save (freeze) $\mathbf{C}_L$ at every point than is required to save $\mathbf{M}'_L$. This is in contrast to the various classes of implicit methods which require the solution of large, block tridiagonal systems of equations in which each of the component blocks (which are as large as $\mathbf{M}'_L$) may be saved, but it is numerically intractable to compute and save the inverse matrix. Average computational time per relaxation sweep is reduced by a factor of 2 to 3 when $\mathbf{C}_L$ is kept frozen for 10 to 40 sweeps. This reduction occurs because the work required to recompute $\mathbf{M}'_{L,\text{inv}}$ and $\mathbf{M}'_{L,\text{vis}}$ as well as the work required to perform the inversion specified by equation (6.11) is eliminated from approximately 90 percent of the relaxation sweeps. The algorithm demands no more work per computational cell than a purely explicit formulation except for the effort required to multiply a 5×1 vector of residuals by a 5×5 matrix for three-dimensional flow of a perfect gas. The robust stability for any Courant number is not sacrificed with this treatment of $\mathbf{C}_L$.

6.3. Positive-Definite Quantities and Initialization Procedures

Flow-field initialization is usually achieved by imposing uniform flow conditions at every cell. (The only time this initialization procedure failed was in the near wake at cells adjacent to the vehicle base in a hypersonic flow. In this case, a linear variation in velocity was introduced which varied from 0 at the wall to free-stream values in the middle of the domain.) A stability enhancing procedure which has been found useful in the early, highly nonlinear solution adjustment period is to force positive-definite quantities (such as density or energy) to remain positive definite and to quell any explosive growth caused by physically unrealistic initial conditions. The adjustment to the algorithm can be expressed as follows.

$$\rho^{n+1} = \tilde{\omega}_1\rho^n \tag{6.12a}$$

$$e^{n+1} = \tilde{\omega}_2 e^n \tag{6.12b}$$

where

$$\omega_1 = 1 + \frac{\delta q_1}{\rho^n} \tag{6.13a}$$

$$\omega_2 = 1 + \frac{\delta e}{e^n} \tag{6.13b}$$

$$\delta e = \frac{1}{\rho}\left[(u^2 + v^2 + w^2 - E)\delta q_1 - u\delta q_2 - v\delta q_3 - w\delta q_4 + \delta q_5\right] \tag{6.14}$$

$$\widetilde{\omega} = \left\{ \begin{array}{ll} 0.5 & (\omega < 0.5) \\ \omega & (0.5 \leq \omega \leq 2.0) \\ 2.0 & (\omega > 2.0) \end{array} \right\} \tag{6.15}$$

7. Results and Discussion

The algorithm has been documented in its various stages of development in references 19 through 22. It is referred to as the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA). Boundary conditions are discussed in appendix E. Modifications required for an unstructured grid are discussed in appendix F. Opportunities for exploiting asynchronous relaxation strategies on parallel computers are discussed in appendix G. The material on asynchronous iteration and some of the blunt body results first appeared in reference 21. Some material on the two-dimensional flow problems appeared first in references 31 and 32.

7.1. Convergence and Grid Refinement Studies

Two-dimensional flow problems are used to illustrate convergence properties and solution dependence on numerical parameters. The first test case is one studied recently by Rudy et al. (ref. 31) involving laminar, supersonic flow through an inlet. The upper wall is straight and parallel to the incoming flow. The lower wall is offset 2 cm from the upper wall. Two centimeters into the inlet there is a 10° compression followed by a 10° expansion at 4 cm. The total length of the domain L is 10 cm. An illustration of the geometry and flow physics in the problem is presented in figure 2. The inflow Mach number is 5.0, and the Reynolds number based on the length of the domain is 1.14×10^7. Adiabatic, no-slip boundary conditions are applied at the upper and lower walls. Extrapolation is used at the outflow boundary.

Grid refinement studies were implemented for discretizations involving 50×50, 100×100, and 200×200 cells. Equation (3.8b) was used to define the limiter, and equation (6.4) was used to define ϵ with $\epsilon_o = 0.1$. Cell Reynolds numbers for this test case were evaluated at the lower wall in front of the inlet. Values of N_c for the three cases are 120, 60, and 30 for the three grids, respectively, where

$$N_c = \frac{\rho a \, \Delta z}{\mu} \tag{7.1}$$

Here, a is the sound speed and Δz is the cell size in the direction normal to the wall. This definition of cell Reynolds number is used to evaluate the adequacy of grid resolution for the purpose of resolving thermal gradients. It depends on a thermal velocity (the local sound speed) and not on the local value of a velocity gradient, which can pass through zero at stagnation or separation points. Experience with this parameter indicates that it should not exceed a value of approximately 2.0 for adequate resolution of the thermal layer.

The distributions of pressure (fig. 3) and skin friction (fig. 4) show continually sharper profiles, particularly in the vicinity of the reflected shock on the upper wall, with increasing resolution. It is clear that the separation region in front of the reflected shock ($C_f < 0$) is not

adequately resolved with the 50×50 and 100×100 cell discretizations. The truncation errors cause significant differences in the extent and magnitude of the separation as compared with the 200×200 discretization. The values of N_c exceed the recommended limit for adequate resolution of dissipative phenomena at the surface. On this basis, the 200×200 cell discretization is still somewhat coarse. The test cases were repeated with a geometry a factor of 10 smaller than the original test. This reduces the free-stream Reynolds number by a factor of 10 and effectively reduces the cell Reynolds numbers used above by a factor of 10. (In this case, the finest grid is marginally acceptable with a value of N_c equal to 3.) These results are presented in figure 5 for pressure and figure 6 for skin friction. Here again, the finer grids give a sharper pressure rise on the upper wall and an earlier separation in front of the reflected shock. The effects of truncation error are much less severe (though still observable) for these order 1 values of N_c (12, 6, and 3, respectively, for the 50×50, 100×100, and 200×200 grids). The small differences between the two finest grid solutions can be attributed to the relative coarseness of grid transverse to the oblique, reflected shock. Shocks which are oblique to a cell face are generally found to be smeared over more cells than in the case of shocks aligned with cell faces.

A second test case was studied to further address the issue of truncation error. The test case involves laminar, hypersonic flow ($M_\infty = 14.1$, $\mathrm{Re}_\infty = 2.36 \times 10^5/\mathrm{m}$) over a 15° ramp. The free-stream temperature is 88.9 K (160°R) and the wall temperature is 297.22 K (535°R). This case and related tests involving larger ramp angles are extensively discussed in reference 32. Four different grids are used in this test: 50×50, 100×100, 100×200, and 200×100. The first index refers to the number of cells along the wall and the second index refers to the number of cells normal to the wall. Values of N_c for these four cases, evaluated on the wall in front of the ramp at $x \approx 1.35$, are approximately 1.2, 0.6, 0.3, and 0.6. Values of N_c near the peak skin friction point on the ramp ($x \approx 2.40$) are approximately 14.0, 7.0, 3.5, and 7.0. The surface pressure and skin friction coefficients as computed on all four grids are presented in figures 7 and 8. The boundary layer is well resolved according to the values of N_c ahead of the ramp and the solutions for both pressure and skin friction coefficients on all four grids are in excellent agreement. The solutions on the ramp near the peak pressure and skin friction points are in good agreement, though observable differences in magnitude on the order of less than 5 percent on the three finest grids are observable. The boundary layer thickens but does not separate ahead of the ramp. The pressure and skin friction are in excellent agreement among all four grids in this region. It is clear that boundary-layer resolution is not the only issue in the prediction of surface quantities. The 100×100 and 200×100 cell cases have identical grids in the direction normal to the wall, but differences are observed due to the improved resolution across the captured shock in the second case. (The computational speed for the 100×100 test case is 32×10^{-6} sec/iteration/grid point on the Cray Y-MP computer using a single processor with Cray FORTRAN CFT77 3.0.2.2. Approximately 2500 relaxation steps are required for convergence.)

The enhanced numerical dissipation provided by increasing the lower limit on eigenvalue magnitude ϵ is shown in the pressure distributions of figure 9 and the skin friction distributions in figure 10 in the inlet. These tests were conducted on the 50×50 grid for the free-stream Reynolds Number equal to 1.14×10^7. The pressure rise across the reflected shock steepens with decreasing ϵ_o. In this coarse grid solution, there is no separation in front of the reflected shock; however, the smaller values of ϵ_o clearly better resolve the sharp decrease in skin friction in this area. Computational stability at Courant numbers greater than 5 was compromised with values of ϵ_o less than 0.1. Convergence rates deteriorate and solution profiles become overly smeared when the eigenvalue limiter is set too large. Nonphysical solutions or a failure to converge can result from eigenvalue limiters set too small.

The effects of the STVD limiters considered herein on the computed solutions are presented in figures 11 and 12. The limiters given by equations (3.8a) and (3.8b) differ only where there is a local maximum or minimum in the value of $\vec{s}_l$. The solutions given by these two limiters are

nearly identical. The solution given by the third limiter (eq. (3.8c)) gives sharper pressure jumps than the first two limiters and lower skin friction predictions. The predictions using this limiter are, generally, in better agreement with the predictions given by three other relaxation schemes on the coarsest grid, as documented in reference 31 and shown here in figure 13. (The results of reference 31 were obtained with a constant eigenvalue limiter equal to 0.005.) Convergence characteristics of the scheme with this limiter were not as attractive as those obtained with the other two limiters. Usually, the solution would converge to an L_2 norm and then stall at that level. Global checks of mass conservation and momentum show gentle oscillations in the fourth (or greater) decimal place around a constant level in these quantities. The magnitude of these oscillations does not appear to be abated significantly by lowering the Courant number from order 10^6 to order 10^{-1}. It would appear that equation (3.8c) is the least dissipative of the three limiters based on the sharper profiles in figures 11 and 12 and the oscillatory convergence characteristics.

Convergence histories for two of the test cases are presented in figures 14 and 15 using the limiter given by equation (3.8b). The measure of convergence is given by the L_2 norm defined by

$$L_2 = \frac{1}{C_N^2} \sum_{L=1}^{N} \mathbf{r}_L \cdot \mathbf{r}_L \tag{7.2}$$

where C_N is the Courant number, N is the total number of cells, and $\mathbf{r}_L$ is the right-hand side of equation (5.1). Scaling by C_N reduces the Courant number dependency in the calculation. Note that L_2 is not scaled by the total number of cells. Figure 14 shows the rapid convergence rate for the coarse 50×50 grid solution with a Courant number of 5×10^6 and $\epsilon_o = 0.2$. Figure 15 shows the convergence history for the fine 200×200 grid solution with a Courant number of 5.0 and $\epsilon_o = 0.1$. The minimum value of C_f on the upper wall and the maximum value of p/p_∞ on the upper wall are also plotted as a function of iteration number in figure 15. The minimum value of skin friction is found in the separation region in front of the reflected shock (fig. 6) and the maximum pressure ratio is found behind the reflected shock (fig. 5). The pressure is converged after approximately 600 iterations when the L_2 norm is less than 3×10^{-3} and the skin friction is converged after approximately 1200 iterations when the L_2 norm is less than 2×10^{-4} in this case.

An optimum value of $\Phi_{\text{inv}} = 1.5$ was found from numerical tests on the 100×100 grid ($N_c = 6$) as shown in figure 16. All these tests were computed with $\epsilon_o = 0.1$ and equation (3.8b) for the flux limiter. The first curve shows the L_2 norm after 700 iterations from a uniform initial flow with $\Phi_{\text{vis}} = \Phi_{\text{inv}}$ and $C_N = 5.0$. The solution diverged for values of $\Phi_{\text{inv}} \leq 1.4$. The second curve shows the L_2 norm as a function of Φ_{inv} with $\Phi_{\text{vis}} = 6.0$ and $C_N = 5.0$. Tests on other problems substantiate the choice of $\Phi_{\text{inv}} = 1.5$ as optimum for reducing the L_2 norm. (In some tests with particularly severe initialization errors, larger values of Φ_{inv} would enhance stability and convergence during the initial adjustment phase.) The effects of Φ_{vis} and C_N on the L_2 norm after 700 iterations are shown in figure 17. The relaxation factor Φ_{vis} has a relatively weak influence on convergence over the range tested, with an optimum value occurring at approximately 7.0. The large value for Φ_{vis} is required as a consequence of stability problems, particularly at large Courant numbers, associated with the abrupt transition from free-stream conditions at the inflow to no-slip conditions at the wall. The choice of Courant number has a stronger impact on convergence, with an optimum value of 5.0 in this test. Optimum values for these parameters are expected to be problem dependent, but experience to date has not shown significant variation from the values obtained in this test for Courant number or Φ_{inv}. Optimum values for Φ_{vis} in the blunt body problem discussed in section 7.2 (which has no boundary condition singularity) vary between 0.5 and 1.0.

7.2. Blunt Body Applications on Aeroassist Flight Experiment (AFE) Configuration

The present algorithm was developed specifically for application to hypersonic flows over blunt bodies. The excellent capabilities of total variation-diminishing schemes with regard to shock capturing, resolution of severe expansions, and robust stability characteristics motivated the development of the present code for these applications.

The Aeroassist Flight Experiment (AFE) configuration is a 60° elliptic cone raked at 73° on the base to form a circular backplane. The nose is defined by an ellipsoid of eccentricity equal to 2.0. The shoulder region is defined by circular arcs on planes passing through the axis of the elliptic nose. More details of the forebody definition are given in reference 33. An illustration of the vehicle is presented in figure 18. The body size is defined relative to the base plane diameter. The base plane diameter is equal to 3.9116 m (154 in.) for the flight vehicle. The base plane diameter for the wind-tunnel test model is 9.3218 cm (3.67 in.). The inviscid relaxation factor Φ_{inv} was set equal to 1.5. The viscous relaxation factor Φ_{vis} was decreased from 1.0 to 0.5, the Courant number was decreased from 5.0 to 0.5, and the eigenvalue limiter ϵ_o was decreased from 0.4 to 0.1 over the period of convergence. (There was no comprehensive retesting of numerical parameters to achieve optimum efficiency.) Free-stream conditions are defined by $V_\infty = 1429$ m/s, $p_\infty = 60.136$ N/m^2, $T_\infty = 52.22$ K, and $M_\infty = 9.86$. The unit Reynolds number is 17 040/cm (43 282/in.). Wall temperature is 300 K.

Predictions from the LAURA code and the HALIS code (ref. 34) are in excellent agreement with experimental data for the pitching-moment coefficient in figure 19. The contributions of viscous forces to the pitching moment are negligible at this test condition.

Calculations and experimental data (ref. 35) for pressure coefficient and heat-transfer coefficient are presented in figure 20. The data were run at angles of attack of 0°, 5°, and −5°. The heat-transfer coefficient for the cone flank compares well with the experimental data; however, as angle of attack is decreased from 5° to −5°, the comparison in the stagnation region gets progressively worse. At an angle of attack of −5°, where the stagnation point crosses the axis singularity, there is an abrupt overshoot/undershoot in the data crossing the axis singularity. A grid refinement involving double the number of points across the shock layer eliminated most of the irregularity in this region and improved the agreement with experimental data in figure 20(c). A grid restructuring, in which the axis singularity was removed by redistributing cell faces across the plane of symmetry, also smoothed the prediction curves for heating in this region. The strong influence of grid structure on computed heating indicates that the variation of truncation error around the axis, as well as the magnitude of the truncation error, is responsible for the irregularities in the stagnation region heating. Computed pressure distributions are insensitive to these effects. The grid refinement shows little change in the heating on the cone flank, which indicates that truncation error is not a significant factor elsewhere in the flow field.

Heat-transfer distributions on the sting in the near wake (ref. 36) compare well with calculated data in figure 21. A grid refinement in which a solution was generated with double the number of cells in all four domains in the direction normal to the forebody and sting shows only a slight change in the computed solution for heat transfer in figure 21(b). The calculation assumes laminar flow. There is no direct evidence in the experimental data to indicate if the shear layer flow or the sting boundary-layer flow is laminar, transitional, or turbulent. The experimental data show no evidence of transition to a turbulent boundary layer on the sting (over the length that data were taken) as judged by the heat-transfer distribution. The value of the eigenvalue limiter ϵ_o was set to 0.010. The near wake flow field does not require the larger values of eigenvalue limiters needed for stable computation of the stagnation region of blunt body flows.

The four-domain grid used for this case is presented in figure 22 on the symmetry plane and on the surface. Grid 1 is the forebody grid which consists of $39 \times 24 \times 64$ cells. Grid 2

extends out from the outflow boundary of grid 1 and consists of $31 \times 24 \times 32$ cells which approximately span the region extending from above the free shear layer across the captured bow shock. Grid 3 fills in the region between the cylindrical upper surface of the afterbody with $31 \times 24 \times 19$ cells and spans much of the free shear layer. Grid 4 resolves the boundary-layer flow over the sting and the region behind the afterbody with $26 \times 24 \times 21$ cells. The low densities in the near wake region simplify the task of constructing grids which satisfy the cell Reynolds number requirement at the body surface discussed after equation (7.1). The cell walls on adjoining domain boundaries are aligned to simplify flux calculations. The outflow boundary condition of the forebody grid system (grid 1) employed second-order accurate extrapolation. The inflow boundary conditon for grid 2 picked up from the converged solution in grid 1. This boundary condition precludes feedback of information from the wake to the forebody and was implemented out of convenience rather than necessity. It is considered reasonable because the information from grid 2 can only influence the solution in grid 1 through the subsonic portion of the boundary layer which is very thin in these test cases. Grid 2 was constructed so that exactly two cell walls of the common boundary in grid 1 formed a single cell wall in grid 2.

Contours of pressure and Mach number in the symmetry plane at angles of attack equal to $0°$ and $-5°$ are shown in figures 23 and 24. Contour lines pass smoothly across grid domain boundaries. Gaps in the contour lines are an artifact of the data base supplied to the plotting routine. Information is stored at cell centers. The gaps in the contour lines equal the distance between cell centers of adjacent domains. The captured shock in the near wake over the sting is clearly visible in figure 23. The free shear layer extending from the aft corner of the shoulder is evident in the Mach contours in figure 24.

7.3. Some Observations of the Relaxation Process

The variation of Courant number, eigenvalue limiter, and viscous relaxation factor noted in section 7.2 was implemented to improve convergence. The solution for the AFE forebody is initialized assuming that the body materializes in the undisturbed flow at time $t = 0$. A coarse grid, first-order accurate solution is developed on a grid which adapts to the position of the captured shock (ref. 21). When the coarse grid solution is judged sufficiently developed, the grid is refined and the second-order flux limiters are implemented. A Courant number equal to 5.0, a viscous relaxation factor $\Phi_{\text{vis}} \geq 1.0$, and an eigenvalue limiter $\epsilon_o \geq 0.2$ are typically applied to speed convergence and maintain sufficient dissipation. Grid adaption is turned off once the L_2 norm is observed to be level. Further relaxation can be used to drop the L_2 norm about another order of magnitude, but then convergence stalls. At this point, the computed surface pressure distribution is generally quite satisfactory, but the computed surface heating is likely to differ from the fully converged values by 25 percent or more, depending on the amount of grid stretching across the boundary layer and the magnitude of ϵ_o. The L_2 norm can be reduced several more orders of magnitude by reducing the Courant number to 0.5 and the viscous relaxation factor to 0.5. The eigenvalue limiter is reduced as much as possible, generally to a value of 0.1, in order to minimize the numerical dissipation on the computed heating. The eigenvalue limiter can be turned off in the direction crossing the boundary layer in order to remove any influence of this parameter on the computed heating. The computation remains stable with this adjustment when the solution is close to convergence and it is recommended when there is severe grid stretching (for example, when $\frac{\Omega_{I,J,K}}{\Omega_{I,J,K-1}} > 1.5$ where K is the index crossing the boundary layer). This adjustment was not used in the present set of calculations.

The convergence rate in the three wake region domains is better than the rate obtained in the forebody region. Typical convergence histories are presented in figures 25 through 30 for the forebody and wake regions to illustrate this point. Figure 25 shows the initialization of the case for $\alpha = 5°$ on the regular grid starting from a converged case for $\alpha = 0°$ considering only the forebody. The Courant number used in this case was equal to 5.0, and ϵ_o was equal

to 0.25. The grid was allowed to adapt to the moving shock during the first 500 relaxation steps and then was held fixed for the remainder of the test. Convergence has obviously stalled at an L_2 norm of approximately 0.02. The curve in figure 26 picks up from the stalled results of figure 25 with a larger value of ϵ_o which is 0.4. After 1000 relaxation steps, the L_2 norm drops 2 orders of magnitude with the larger value of ϵ_o and a Courant number of 5.0. The convergence then stalls at L_2 of approximately 0.0003. A decrease in the Courant number to 0.5 with the same value of ϵ_o allows the L_2 norm to drop another 5 orders of magnitude during a subsequent 1500 relaxation steps. This behavior is indicative of a low level limit cycle that can occur at the larger Courant numbers.

Figure 27 records the drop in the L_2 norm for the three base region domains for the case for $\alpha = 0°$. The wake region is initialized by inserting the computed values of the solution vector from the outflow boundary of the forebody domain into the computational cells in the three base region domains. The L_2 norm decreases by more than 4 orders of magnitude in 2600 relaxation steps. Several Courant numbers were used, as indicated in figure 27. There were no limit cycles apparent in these tests, except at a Courant number of 8×10^6. The eigenvalue limiter ϵ_o during this test was held constant at 0.25. Recall that this value was too small to achieve acceptable convergence behavior in the forebody tests.

The convergence history for the fine grid solution over the forebody at $\alpha = -5°$ is presented in figure 28. The initial condition is interpolated from the initial grid solution. The Courant number in this case was greater than or equal to 5.0. The eigenvalue limiter ϵ_o during this test was held constant at 0.2. Convergence in this case stalled at a value of $L_2 \approx 10^{-5}$. This level is still nearly 4 orders of magnitude lower than the level that could be obtained in the regular grid tests with $\epsilon_o = 0.25$ and illustrates the role that truncation error can play in the limit cycle behavior. The limit cycle in this case can be even further reduced by reducing the Courant number to 0.5, as shown in figure 29, which picks up from the solution state of the previous case.

The convergence history for the fine grid solution in the three base domains at $\alpha = -5°$ is presented in figure 30. The initial condition is interpolated from the regular grid solution. The Courant number in this case was 0.5. The value of the eigenvalue limiter ϵ_o is 0.01 which is an order of magnitude smaller than the value used in the forebody region. Apparently, the greater influence of viscous dissipation in the wake and the larger region of supersonic flow surrounding the wake core contribute to the improved stability and convergence of the computation in this region.

The two-dimensional inlet calculations did not require any of the parameter adjustments used on the blunt body calculations. In particular, there was no "stalling" of convergence with large Courant numbers and the required eigenvalue limiter was 2 orders of magnitude smaller than used for the blunt body computations. Here again, the absence of any extensive stagnation region and the predominance of supersonic flow are believed to improve the stability and convergence (fig. 15) of the computation in the inlet relative to the blunt body flow computations.

Concluding Remarks

An upwind-biased, point-implicit relaxation algorithm for obtaining the numerical solution to the governing equations for three-dimensional, viscous, compressible, perfect-gas flows has been described. The algorithm is derived by using a finite-volume formulation in which the inviscid components of flux across cell walls are described with Roe's averaging and Harten's entropy fix with second-order corrections based on Yee's symmetric total variation-diminishing scheme. Viscous terms are discretized using central differences. The scheme is second-order accurate in computational space except at isolated points where the flux limiter restricts a second-order antidissipative correction which could introduce catastrophic instabilities. The

scheme is second-order accurate in physical space so long as metric coefficients can be assumed to vary continuously from cell to cell. Metric coefficients are defined by the ratio of cell areas to cell volumes within the context of a finite-volume scheme.

The relaxation scheme requires only a single level of storage for the unknowns if time accuracy is not required. Time-accurate solutions can be constructed by employing additional storage levels to save the results of previous iterations. Because of the point-implicit relaxation strategy, the algorithm remains stable at large Courant numbers without the necessity of solving large block tridiagonal systems. On machines with large memories, the inverse of the Jacobian for the right-hand-side residual vector with respect to the solution vector can be saved over large blocks of iterations, causing the algorithm to require very little overhead as compared with a standard explicit scheme. These features of the relaxation strategy make the algorithm well suited for computers employing either vector or parallel architectures. It is also well suited to the numerical solution of the governing equations on unstructured grids, although this option has not been developed in the present work.

Two-dimensional flow problems are used to illustrate convergence properties and sensitivity of the solution on numerical parameters. The first test case involves laminar, supersonic flow through an inlet. The inflow Mach number is 5.0 and two test Reynolds numbers, based on the length of the domain, of 1.14×10^7 and 1.14×10^6 were used. Convergence of the computed solution is evaluated through a grid refinement study. A second test case, involving Mach 14 flow over a 15° ramp, is also used to evaluate convergence through a grid refinement study. The enhanced numerical dissipation provided by increasing the lower limit on eigenvalue magnitude ϵ is demonstrated. Three different symmetric total variation-diminishing (STVD) flux limiters were evaluated in the first test problem. The solution given by the third limiter gives sharper pressure jumps and lower skin friction predictions than the first two limiters. The predictions using this limiter are, generally, in better agreement with the predictions given by three other relaxation schemes.

The application of separate relaxation factors to the inviscid and viscous contributions to the Jacobian within the context of point-implicit relaxation has been found to accelerate convergence. The inviscid relaxation factor must be greater than or equal to 1.5 in order for the second-order accurate flux-limited schemes to converge. As a converged state is approached, a value of 1.5 for the inviscid relaxation factor was found to be the optimum in several numerical tests. The optimum value for the viscous relaxation factor was problem dependent in the present set of tests, though this behavior may be due to a leading-edge singularity in the boundary condition of the inlet problem. The optimum value of the viscous relaxation factor in the blunt body problems varied between 1.0 and 0.5, with 0.5 giving the best convergence rate as the converged state is approached.

Several numerical tests were conducted on the Aeroassist Flight Experiment (AFE) configuration, including the base flow region of a wind-tunnel model. Comparisons of computed aerodynamic coefficients and surface pressure with another numerical method and with experimental data show good agreement. Comparisons of computed heat-transfer coefficients with experimental data at three different angles of attack show good agreement on the cone flank, but reveal a strong influence of the coordinate singularity at the elliptic nose in the stagnation region on the computed heating distributions. Grid refinement in the direction normal to the boundary layer and grid restructuring in the vicinity of the axis can be used to improve the computed heating distribution in the stagnation region. A four-domain grid was used to discretize the space surrounding the forebody and sting of the complete AFE wind-tunnel model. Computed contour lines pass smoothly across domain boundaries. Heating distributions on the sting are in good agreement with experimental data under the assumption of steady, laminar flow. A grid refinement for this case shows only slight truncation error effects on the computed sting heating.

Appendix A

Cell Geometry

Expressions for cell volume and cell wall area in a cell-centered scheme in which the dependent variables are at cell centers and the independent variables (x, y, z) are at cell corners are presented in this appendix. A similar derivation may be found in reference 37.

Let i, j, k be indices of a cell corner at which point the values of the independent variables x, y, z are known. Let I, J, K be indices of a cell center at which point the values of the dependent variable q are to be determined. Note, for example, that the computational planes represented by $I - \frac{1}{2}$ and i are equivalent in this indexing system.

Let

$$d\vec{r}_1 = \vec{r}_{i,j+1,k} - \vec{r}_{i,j,k+1} = \left[\left(\tilde{x}_\eta - \tilde{x}_\zeta\right)\vec{i} + \left(\tilde{y}_\eta - \tilde{y}_\zeta\right)\vec{j} + \left(\tilde{z}_\eta - \tilde{z}_\zeta\right)\vec{k}\,\right]_{i,J,K} \tag{A1a}$$

$$d\vec{r}_2 = \vec{r}_{i,j+1,k+1} - \vec{r}_{i,j,k} = \left[\left(\tilde{x}_\eta + \tilde{x}_\zeta\right)\vec{i} + \left(\tilde{y}_\eta + \tilde{y}_\zeta\right)\vec{j} + \left(\tilde{z}_\eta + \tilde{z}_\zeta\right)\vec{k}\,\right]_{i,J,K} \tag{A1b}$$

$$d\vec{r}_3 = \vec{r}_{i,j,k+1} - \vec{r}_{i+1,j,k} = \left[\left(\tilde{x}_\zeta - \tilde{x}_\xi\right)\vec{i} + \left(\tilde{y}_\zeta - \tilde{y}_\xi\right)\vec{j} + \left(\tilde{z}_\zeta - \tilde{z}_\xi\right)\vec{k}\,\right]_{I,j,K} \tag{A1c}$$

$$d\vec{r}_4 = \vec{r}_{i+1,j,k+1} - \vec{r}_{i,j,k} = \left[\left(\tilde{x}_\zeta + \tilde{x}_\xi\right)\vec{i} + \left(\tilde{y}_\zeta + \tilde{y}_\xi\right)\vec{j} + \left(\tilde{z}_\zeta + \tilde{z}_\xi\right)\vec{k}\,\right]_{I,j,K} \tag{A1d}$$

$$d\vec{r}_5 = \vec{r}_{i+1,j,k} - \vec{r}_{i,j+1,k} = \left[\left(\tilde{x}_\xi - \tilde{x}_\eta\right)\vec{i} + \left(\tilde{y}_\xi - \tilde{y}_\eta\right)\vec{j} + \left(\tilde{z}_\xi - \tilde{z}_\eta\right)\vec{k}\,\right]_{I,J,k} \tag{A1e}$$

$$d\vec{r}_6 = \vec{r}_{i+1,j+1,k} - \vec{r}_{i,j,k} = \left[\left(\tilde{x}_\xi + \tilde{x}_\eta\right)\vec{i} + \left(\tilde{y}_\xi + \tilde{y}_\eta\right)\vec{j} + \left(\tilde{z}_\xi + z_\eta\right)\vec{k}\,\right]_{I,J,k} \tag{A1f}$$

$$d\vec{r}_7 = \vec{r}_{i+1,j+1,k+1} - \vec{r}_{i,j,k} = \left[\left(\tilde{x}_\xi + \tilde{x}_\eta + \tilde{x}_\zeta\right)\vec{i} + \left(\tilde{y}_\xi + \tilde{y}_\eta + \tilde{y}_\zeta\right)\vec{j} + \left(\tilde{z}_\xi + \tilde{z}_\eta + \tilde{z}_\zeta\right)\vec{k}\,\right]_{I,J,K} \tag{A1g}$$

where

$$\vec{r}_{i,j,k} = \left[x\vec{i} + y\vec{j} + z\vec{k}\right]_{i,j,k}$$

The generic differences, $\tilde{s}_\xi$, $\tilde{s}_\eta$, $\tilde{s}_\zeta$, used in equations (A1) are defined as follows:

$$\left[\tilde{s}_\eta\right]_{i,J,K} = \left(s_{i,j+1,k} - s_{i,j,k} + s_{i,j+1,k+1} - s_{i,j,k+1}\right)/2 \tag{A2a}$$

$$\left[\tilde{s}_\zeta\right]_{i,J,K} = \left(s_{i,j,k+1} - s_{i,j,k} + s_{i,j+1,k+1} - s_{i,j+1,k}\right)/2 \tag{A2b}$$

$$\left[\tilde{s}_\xi\right]_{I,j,K} = \left(s_{i+1,j,k} - s_{i,j,k} + s_{i+1,j,k+1} - s_{i,j,k+1}\right)/2 \tag{A2c}$$

$$\left[\tilde{s}_\zeta\right]_{I,j,K} = \left(s_{i,j,k+1} - s_{i,j,k} + s_{i+1,j,k+1} - s_{i+1,j,k}\right)/2 \tag{A2d}$$

$$\left[\tilde{s}_\xi\right]_{I,J,k} = \left(s_{i+1,j,k} - s_{i,j,k} + s_{i+1,j+1,k} - s_{i,j+1,k}\right)/2 \tag{A2e}$$

$$\left[\tilde{s}_\eta\right]_{I,J,k} = \left(s_{i,j+1,k} - s_{i,j,k} + s_{i+1,j+1,k} - s_{i+1,j,k}\right)/2 \tag{A2f}$$

$$\begin{aligned}\left[\tilde{s}_\xi\right]_{I,J,K} = \big[&2\left(s_{i+1,j,k} - s_{i,j,k} + s_{i+1,j+1,k+1} - s_{i,j+1,k+1}\right)\\ &+ \left(s_{i+1,j+1,k} - s_{i,j+1,k} + s_{i+1,j,k+1} - s_{i,j,k+1}\right)\big]/6\end{aligned} \tag{A2g}$$

$$\begin{aligned}\left[\tilde{s}_\eta\right]_{I,J,K} = \big[&2\left(s_{i,j+1,k} - s_{i,j,k} + s_{i+1,j+1,k+1} - s_{i+1,j,k+1}\right)\\ &+ \left(s_{i,j+1,k+1} - s_{i,j,k+1} + s_{i+1,j+1,k} - s_{i+1,j,k}\right)\big]/6\end{aligned} \tag{A2h}$$

$$\begin{aligned}\left[\tilde{s}_\zeta\right]_{I,J,K} = \big[&2\left(s_{i,j,k+1} - s_{i,j,k} + s_{i+1,j+1,k+1} - s_{i+1,j+1,k}\right)\\ &+ \left(s_{i,j+1,k+1} - s_{i,j+1,k} + s_{i+1,j,k+1} - s_{i+1,j,k}\right)\big]/6\end{aligned} \tag{A2i}$$

where s is a dummy variable representing the independent variables, x, y, z, and ξ, η, ζ are computational coordinates in the i, j, k directions, respectively. The cell volume Ω and cell wall directed area $\vec{\sigma}$ can now be defined by simple vector relations as follows:

$$\vec{\sigma}_{i,J,K} = \frac{d\vec{r}_1 \times d\vec{r}_2}{2} \tag{A3a}$$

$$\vec{\sigma}_{I,j,K} = \frac{d\vec{r}_3 \times d\vec{r}_4}{2} \tag{A3b}$$

$$\vec{\sigma}_{I,J,k} = \frac{d\vec{r}_5 \times d\vec{r}_6}{2} \tag{A3c}$$

$$\Omega_{I,J,K} = d\vec{r}_7 \cdot \frac{\left(\vec{\sigma}_{i,J,K} + \vec{\sigma}_{I,j,K} + \vec{\sigma}_{I,J,k}\right)}{3} \tag{A4}$$

where the magnitude of $\vec{\sigma}$ is the cell wall area, the direction of $\vec{\sigma}$ is normal to the cell wall toward increasing I, J, K, respectively, and Ω is the cell volume.

The transformation metrics, such as ξ_x, ξ_y, ξ_z, can be expressed as ratios of cell wall areas to cell volumes. We would like these expressions to be second-order accurate with respect to a face-centered index (i.e., i, J, K) or a cell-centered index (I, J, K). Equation (A4) for cell volume does not meet this criterion, and use of this expression causes jags in contours running across an axis singularity. A better formulation utilizes symmetric averages of differences about the cell center (I, J, K) as follows:

$$\Omega_{I,J,K} = \frac{d\vec{r}_8 \cdot \left(\vec{\sigma}_{i,J,K} + \vec{\sigma}_{i+1,J,K} + \vec{\sigma}_{I,j,K} + \vec{\sigma}_{I,j+1,K} + \vec{\sigma}_{I,J,k} + \vec{\sigma}_{I,J,k+1}\right)}{6} \tag{A5a}$$

where

$$d\vec{r}_8 = \left[\left(\hat{x}_\xi + \hat{x}_\eta + \hat{x}_\zeta\right)\vec{\mathrm{i}} + \left(\hat{y}_\xi + \hat{y}_\eta + \hat{y}_\zeta\right)\vec{\mathrm{j}} + \left(\hat{z}_\xi + \hat{z}_\eta + \hat{z}_\zeta\right)\vec{\mathrm{k}}\right]_{I,J,K} \tag{A5b}$$

and

$$[\hat{s}_\xi]_{I,J,K} = \frac{[\tilde{s}_\xi]_{I,j,K} + [\tilde{s}_\xi]_{I,j+1,K}}{2} = \frac{[\tilde{s}_\xi]_{I,J,k} + [\tilde{s}_\xi]_{I,J,k+1}}{2} \tag{A5c}$$

$$[\hat{s}_\eta]_{I,J,K} = \frac{[\tilde{s}_\eta]_{i,J,K} + [\tilde{s}_\eta]_{i+1,J,K}}{2} = \frac{[\tilde{s}_\eta]_{I,J,k} + [\tilde{s}_\eta]_{I,J,k+1}}{2} \tag{A5d}$$

$$[\hat{s}_\zeta]_{I,J,K} = \frac{[\tilde{s}_\zeta]_{I,j,K} + [\tilde{s}_\zeta]_{I,j+1,K}}{2} = \frac{[\tilde{s}_\zeta]_{i,J,K} + [\tilde{s}_\zeta]_{i+1,J,K}}{2} \tag{A5e}$$

Note that the $\tilde{s}$ differences are second-order accurate with respect to face centers and that the $\hat{s}$ differences are second-order accurate with respect to cell centers.

The transformation metrics are defined by

$$\begin{bmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{bmatrix}^{-1} \tag{A6}$$

In the limit as cell volume goes to zero, using equations (A1) through (A6) gives

$$\vec{\nabla}\xi_{i,J,K} = \frac{\vec{\sigma}_{i,J,K}}{\Omega_{I,J,K}} \tag{A7a}$$

$$\vec{\nabla}\eta_{I,j,K} = \frac{\vec{\sigma}_{I,j,K}}{\Omega_{I,J,K}} \tag{A7b}$$

$$\vec{\nabla}\zeta_{I,J,k} = \frac{\vec{\sigma}_{I,J,k}}{\Omega_{I,J,K}} \tag{A7c}$$

Equations (A7) are only first-order accurate with respect to any point in the I, J, K cell. This is because the $\vec{\sigma}$ terms are functions of $\tilde{s}$ differences which are second-order accurate with respect to face centers but first-order accurate with respect to cell centers, and $\Omega_{I,J,K}$ is a function of $\hat{s}$ differences which are second-order accurate with respect to cell centers but first-order accurate with respect to face centers. Symmetric averages of cell volumes or cell faces can be used to achieve second-order accurate metrics at either face centers or cell centers. Several options were studied in reference 1. The recommended formulations for face-centered metrics, which are required in the evaluation of the viscous dissipation terms across cell faces, are as follows:

$$\vec{\nabla}\xi_{i,J,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{i-1,J,K} + \vec{\sigma}_{i,J,K}) + \Omega_{I-1,J,K}(\vec{\sigma}_{i,J,K} + \vec{\sigma}_{i+1,J,K})}{4\Omega_{I,J,K}\Omega_{I-1,J,K}} \tag{A8a}$$

$$\vec{\nabla}\xi_{I,j,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{i,J-1,K} + \vec{\sigma}_{i+1,J-1,K}) + \Omega_{I,J-1,K}(\vec{\sigma}_{i,J,K} + \vec{\sigma}_{i+1,J,K})}{4\Omega_{I,J,K}\Omega_{I,J-1,K}} \tag{A8b}$$

$$\vec{\nabla}\xi_{I,J,k} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{i,J,K-1} + \vec{\sigma}_{i+1,J,K-1}) + \Omega_{I,J,K-1}(\vec{\sigma}_{i,J,K} + \vec{\sigma}_{i+1,J,K})}{4\Omega_{I,J,K}\Omega_{I,J,K-1}} \tag{A8c}$$

$$\vec{\nabla}\eta_{i,J,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I-1,j,K} + \vec{\sigma}_{I-1,j+1,K}) + \Omega_{I-1,J,K}(\vec{\sigma}_{I,j,K} + \vec{\sigma}_{I,j+1,K})}{4\Omega_{I,J,K}\Omega_{I-1,J,K}} \tag{A8d}$$

$$\vec{\nabla}\eta_{I,j,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I,j-1,K} + \vec{\sigma}_{I,j,K}) + \Omega_{I,J-1,K}(\vec{\sigma}_{I,j,K} + \vec{\sigma}_{I,j+1,K})}{4\Omega_{I,J,K}\Omega_{I,J-1,K}} \tag{A8e}$$

$$\vec{\nabla}\eta_{I,J,k} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I,j,K-1} + \vec{\sigma}_{I,j+1,K-1}) + \Omega_{I,J,K-1}(\vec{\sigma}_{I,j,K} + \vec{\sigma}_{I,j+1,K})}{4\Omega_{I,J,K}\Omega_{I,J,K-1}} \tag{A8f}$$

$$\vec{\nabla}\zeta_{i,J,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I-1,J,k} + \vec{\sigma}_{I-1,J,k+1}) + \Omega_{I-1,J,K}(\vec{\sigma}_{I,J,k} + \vec{\sigma}_{I,J,k+1})}{4\Omega_{I,J,K}\Omega_{I-1,J,K}} \tag{A8g}$$

$$\vec{\nabla}\zeta_{I,j,K} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I,J-1,k} + \vec{\sigma}_{I,J-1,k+1}) + \Omega_{I,J-1,K}(\vec{\sigma}_{I,J,k} + \vec{\sigma}_{I,J,k+1})}{4\Omega_{I,J,K}\Omega_{I,J-1,K}} \tag{A8h}$$

$$\vec{\nabla}\zeta_{I,J,k} = \frac{\Omega_{I,J,K}(\vec{\sigma}_{I,J,k-1} + \vec{\sigma}_{I,J,k}) + \Omega_{I,J,K-1}(\vec{\sigma}_{I,J,k} + \vec{\sigma}_{I,J,k+1})}{4\Omega_{I,J,K}\Omega_{I,J,K-1}} \tag{A8i}$$

The unit normal and tangent vectors for the i faces are defined as follows:

$$\vec{n}_{i,J,K} = \frac{\vec{\sigma}_{i,J,K}}{|\vec{\sigma}_{i,J,K}|} \tag{A9a}$$

$$\vec{l}_{i,J,K} = \frac{d\vec{r}_1}{|d\vec{r}_1|} \tag{A9b}$$

$$\vec{m}_{i,J,K} = \vec{n}_{i,J,K} \times \vec{l}_{i,J,K} \tag{A9c}$$

The unit normal vectors for the j and k cell faces are defined similarly. If the cell wall area is equal to zero, extrapolation of $\vec{n}$ from interior cell walls may be used.

In the case of the thin-layer Navier-Stokes equations, only the vectors defined by equations (A8a), (A8e), and (A8i) are required, depending on the orientation of the coordinate system. The dot product of these vectors with the corresponding unit normal to the cell face, as used in equations (4.4) through (4.10), can be approximated as follows:

$$\vec{\nabla}\xi_{i,J,K} \cdot \vec{n}_{i,J,K} \approx \frac{\Omega_{I,J,K}(\sigma_{i-1,J,K} + \sigma_{i,J,K}) + \Omega_{I-1,J,K}(\sigma_{i,J,K} + \sigma_{i+1,J,K})}{4\Omega_{I,J,K}\Omega_{I-1,J,K}} \tag{A10a}$$

$$\vec{\nabla}\eta_{I,j,K} \cdot \vec{n}_{I,j,K} \approx \frac{\Omega_{I,J,K}(\sigma_{I,j-1,K} + \sigma_{I,j,K}) + \Omega_{I,J-1,K}(\sigma_{I,j,K} + \sigma_{I,j+1,K})}{4\Omega_{I,J,K}\Omega_{I,J-1,K}} \tag{A10b}$$

$$\vec{\nabla}\zeta_{I,J,k} \cdot \vec{n}_{I,J,k} \approx \frac{\Omega_{I,J,K}(\sigma_{I,J,k-1} + \sigma_{I,J,k}) + \Omega_{I,J,K-1}(\sigma_{I,J,k} + \sigma_{I,J,k+1})}{4\Omega_{I,J,K}\Omega_{I,J,K-1}} \tag{A10c}$$

where, for example, $\sigma_{i,J,K} = |\vec{\sigma}_{i,J,K}|$ is the magnitude (area) of $\vec{\sigma}_{i,J,K}$.

Another identity that is useful in the programming of the thin-layer Navier-Stokes equations involves a geometric relation between the unit normal to a cell face and the gradient of the computational coordinate which defines the cell face. This relation is most easily established by examining the limiting behavior of equations (A7) and (A9a) as cell volume goes to zero. Consider, for example, the gradient of the computational coordinate ξ across a computational cell face defined by ξ = Constant (i = Constant) given by

$$\vec{\nabla}\xi_{i,J,K} = \frac{\vec{\sigma}_{i,J,K}}{\Omega_{I,J,K}} = \frac{\sigma_{i,J,K}}{\Omega_{I,J,K}}\vec{n}_{i,J,K} \tag{A11}$$

Take the dot product of both sides of equation (A11) with $\vec{n}$ to obtain

$$(\vec{\nabla}\xi \cdot \vec{n})_{i,J,K} = \frac{\sigma_{i,J,K}}{\Omega_{I,J,K}} \tag{A12}$$

Substitute equation (A12) back into equation (A11) and relate the individual vector components to show that

$$\left(\frac{\partial \xi}{\partial s}\right)_{i,J,K} = (\vec{\nabla}\xi \cdot \vec{n}n_s)_{i,J,K} \tag{A13a}$$

where s is a dummy variable for x, y, or z. Similar relations for the other two coordinate directions can be derived in like manner to obtain

$$\left(\frac{\partial \eta}{\partial s}\right)_{I,j,K} = (\vec{\nabla}_\eta \cdot \vec{n}n_s)_{I,j,K} \tag{A13b}$$

$$\left(\frac{\partial \zeta}{\partial s}\right)_{I,J,k} = (\vec{\nabla}\zeta \cdot \vec{n}n_s)_{I,J,k} \tag{A13c}$$

Finally, some unit vector relations which do much to simplify the evaluation of viscous stresses are presented as follows:

$$\mathbf{T}\mathbf{T}^{-1} = \mathbf{T}^{-1}\mathbf{T} = \mathbf{I} \tag{A14a}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathrm{n}_x & \mathrm{n}_y & \mathrm{n}_z \\ \mathrm{l}_x & \mathrm{l}_y & \mathrm{l}_z \\ \mathrm{m}_x & \mathrm{m}_y & \mathrm{m}_z \end{bmatrix} \tag{A14b}$$

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathrm{n}_x & \mathrm{l}_x & \mathrm{m}_x \\ \mathrm{n}_y & \mathrm{l}_y & \mathrm{m}_y \\ \mathrm{n}_z & \mathrm{l}_z & \mathrm{m}_z \end{bmatrix} \tag{A14c}$$

$$\mathbf{T} \begin{bmatrix} \mathrm{u} \\ \mathrm{v} \\ \mathrm{w} \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{A15a}$$

$$\mathbf{T}^{-1} \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \mathrm{u} \\ \mathrm{v} \\ \mathrm{w} \end{bmatrix} \tag{A15b}$$

Appendix B

Definition of Matrices A, R, Λ, and $\mathbf{R}^{-1}$

The Jacobian of **g** with respect to **q** is expressed as

$$\mathbf{A} = \begin{bmatrix} 0 & \mathrm{n}_x & \mathrm{n}_y & \mathrm{n}_z & 0 \\ \alpha\beta\mathrm{n}_x - U\mathrm{u} & -\beta\mathrm{u}\mathrm{n}_x + \mathrm{u}\mathrm{n}_x + U & -\beta\mathrm{v}\mathrm{n}_x + \mathrm{u}\mathrm{n}_y & -\beta\mathrm{w}\mathrm{n}_x + \mathrm{u}\mathrm{n}_z & \beta\mathrm{n}_x \\ \alpha\beta\mathrm{n}_y - U\mathrm{v} & -\beta\mathrm{u}\mathrm{n}_y + \mathrm{v}\mathrm{n}_x & -\beta\mathrm{v}\mathrm{n}_y + \mathrm{v}\mathrm{n}_y + U & -\beta\mathrm{w}\mathrm{n}_y + \mathrm{v}\mathrm{n}_z & \beta\mathrm{n}_y \\ \alpha\beta\mathrm{n}_z - U\mathrm{w} & -\beta\mathrm{u}\mathrm{n}_z + \mathrm{w}\mathrm{n}_x & -\beta\mathrm{v}\mathrm{n}_z + \mathrm{w}\mathrm{n}_y & -\beta\mathrm{w}\mathrm{n}_z + \mathrm{w}\mathrm{n}_z + U & \beta\mathrm{n}_z \\ \alpha\beta U - UH & -\beta\mathrm{u}U + H\mathrm{n}_x & -\beta vU + H\mathrm{n}_y & -\beta\mathrm{w}U + H\mathrm{n}_z & \beta U + U \end{bmatrix} \tag{B1}$$

The similarity transformation matrices $\mathbf{R}^{-1}$ and **R**, composed of the left and right eigenvectors of **A**, are defined as

$$\mathbf{R}^{-1} = \begin{bmatrix} a^2 - \alpha\beta & \beta u & \beta v & \beta w & -\beta \\ -V & \mathrm{l}_x & \mathrm{l}_y & \mathrm{l}_z & 0 \\ -W & \mathrm{m}_x & \mathrm{m}_y & \mathrm{m}_z & 0 \\ \alpha\beta - Ua & a\mathrm{n}_x - \beta\mathrm{u} & a\mathrm{n}_y - \beta\mathrm{v} & a\mathrm{n}_z - \beta\mathrm{w} & \beta \\ \alpha\beta + Ua & -a\mathrm{n}_x - \beta\mathrm{u} & -a\mathrm{n}_y - \beta\mathrm{v} & -a\mathrm{n}_z - \beta\mathrm{w} & \beta \end{bmatrix} \tag{B2}$$

$$\mathbf{R} = \begin{bmatrix} \frac{1}{a^2} & 0 & 0 & \frac{1}{2a^2} & \frac{1}{2a^2} \\ \frac{\mathrm{u}}{a^2} & \mathrm{l}_x & \mathrm{m}_x & \frac{\mathrm{u}+a\mathrm{n}_x}{2a^2} & \frac{\mathrm{u}-a\mathrm{n}_x}{2a^2} \\ \frac{\mathrm{v}}{a^2} & \mathrm{l}_y & \mathrm{m}_y & \frac{\mathrm{v}+a\mathrm{n}_y}{2a^2} & \frac{\mathrm{v}-a\mathrm{n}_y}{2a^2} \\ \frac{\mathrm{w}}{a^2} & \mathrm{l}_z & \mathrm{m}_z & \frac{\mathrm{w}+a\mathrm{n}_z}{2a^2} & \frac{\mathrm{w}-a\mathrm{n}_z}{2a^2} \\ \frac{\alpha}{a^2} & V & W & \frac{H+aU}{2a^2} & \frac{H-aU}{2a^2} \end{bmatrix} \tag{B3}$$

Note that $\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}^{-1}\mathbf{R} = \mathbf{I}$. The diagonal matrix of eigenvalues of **A** is defined by

$$\mathbf{\Lambda} = \begin{bmatrix} U & 0 & 0 & 0 & 0 \\ 0 & U & 0 & 0 & 0 \\ 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & U+a & 0 \\ 0 & 0 & 0 & 0 & U-a \end{bmatrix} \tag{B4}$$

The matrix definitions presented in equations (B1) through (B4) need to be evaluated at cell interfaces. Consequently, the unit normal and tangential vector components are defined with respect to cell walls by using equation (A9). The velocity components in the $\vec{\mathrm{n}}$, $\vec{\mathrm{l}}$, and $\vec{\mathrm{m}}$ directions are defined as

$$U = \vec{\mathrm{u}} \cdot \vec{\mathrm{n}} = \mathrm{u}\mathrm{n}_x + \mathrm{v}\mathrm{n}_y + \mathrm{w}\mathrm{n}_z \tag{B5a}$$

$$V = \vec{\mathrm{u}} \cdot \vec{\mathrm{l}} = \mathrm{u}\mathrm{l}_x + \mathrm{v}\mathrm{l}_y + \mathrm{w}\mathrm{l}_z \tag{B5b}$$

$$W = \vec{\mathrm{u}} \cdot \vec{\mathrm{m}} = \mathrm{u}\mathrm{m}_x + \mathrm{v}\mathrm{m}_y + \mathrm{w}\mathrm{m}_z \tag{B5c}$$

Other parameters required to complete the definition of equations (B1) through (B4) include the sound speed a given by

$$a^2 = \frac{\gamma p}{\rho} = \beta(H - \alpha) \tag{B6}$$

and the kinetic energy per unit mass α given by

$$\alpha = \frac{1}{2}(\vec{u} \cdot \vec{u}) \tag{B7}$$

The parameter β is related to the perfect-gas specific heat ratio γ by

$$\beta = \gamma - 1 \tag{B8}$$

Roe's averaging is used to define the elements of $\mathbf{R}$, $\mathbf{R}^{-1}$, and $\mathbf{\Lambda}$ so that Roe's Property U is maintained. (See ref. 2.) Property U is defined by the relation

$$\mathbf{R}_l \mathbf{\Lambda}_l \mathbf{R}_l^{-1}(\mathbf{q}_L - \mathbf{q}_{L-1}) = \mathbf{g}_L - \mathbf{g}_{L-1} \tag{B9}$$

where the grid indexing system described in appendix A and averaging in the l direction have been used. Equation (B9) is not generally satisfied exactly by standard schemes for obtaining average values at the cell interface l based on values at adjacent cell centers L and $L-1$, because of the nonlinear relation between $\mathbf{g}$ and $\mathbf{q}$. Early numerical tests showed that the sharpest normal shock captures at hypersonic conditions were obtained using Roe's averaging. Roe's averaged values at cell wall l are defined as follows:

$$\vec{u}_l = c_2(c_1 \vec{u}_L + \vec{u}_{L-1}) \tag{B10}$$

$$H_l = c_2(c_1 H_L + H_{L-1}) \tag{B11}$$

$$a_l^2 = \beta\left(H_l - \frac{\vec{u}_l \cdot \vec{u}_l}{2}\right) \tag{B12}$$

where

$$c_1 = \left(\frac{\rho_L}{\rho_{L-1}}\right)^{1/2} \tag{B13a}$$

$$c_2 = \frac{1}{1 + c_1} \tag{B13b}$$

Appendix C

Definition of $\mathbf{B}_{l,L}$ for Point-Implicit Treatment of Viscous Terms

The point-implicit treatment of the governing equations considers only the functional dependence of the conservation laws on the cell center which is undergoing a relaxation step. Consequently, only derivatives across a cell face contribute to the point-implicit treatment because these are the only terms which depend on the cell center. Furthermore, given a rectangular ordering of mesh points in which the face normal is directed in the direction of increasing index, there is a sign change in the definition of $\mathbf{B}_{l+1,L}$ as compared with $\mathbf{B}_{l,L}$ because of the different functional dependence of $\left(\frac{\partial \mathrm{u}}{\partial \chi}\right)_l^*$ and $\left(\frac{\partial \mathrm{u}}{\partial \chi}\right)_{l+1}^*$ on the cell center as can be seen in equations (4.12a) and (4.12b). Following the definitions given in equations (4.8) and (4.10) through (4.13), one can define $\mathbf{B}_{l,L}$ as

$$\mathbf{B}_{l,L} = -\frac{\partial \mathbf{h}_{l,L}^*}{\partial \mathbf{q}_L} \equiv -\frac{\partial \mathbf{h}_{l,L}'^*}{\partial \mathbf{q}_L} \tag{C1a}$$

$$\mathbf{B}_{l,L} = \frac{\mu_l \left(\vec{\nabla}\chi \cdot \vec{\mathrm{n}}\right)_l}{\mathrm{Re}\ \rho_L} \left[b_{irow,jcol}\right] \tag{C1b}$$

where

$$b_{1,1-5} = 0$$

$$b_{1-4,5} = 0$$

$$b_{2,1} = -\mathrm{u}_L - \frac{U_L(\mathrm{n}_x)_l}{3}$$

$$b_{3,1} = -\mathrm{v}_L - \frac{U_L(\mathrm{n}_y)_l}{3}$$

$$b_{4,1} = -\mathrm{w}_L - \frac{U_L(\mathrm{n}_z)_l}{3}$$

$$b_{5,1} = \mathrm{u}_l b_{2,1} + \mathrm{v}_l b_{3,1} + \mathrm{w}_l b_{4,1} - \frac{\gamma}{\mathrm{Pr}}(E - 2\alpha)_L$$

$$b_{2,2} = 1 + \frac{(\mathrm{n}_x \mathrm{n}_x)_l}{3}$$

$$b_{3,2} = b_{2,3} = \frac{(\mathrm{n}_x \mathrm{n}_y)_l}{3}$$

$$b_{4,2} = b_{2,4} = \frac{(\mathrm{n}_x \mathrm{n}_z)_l}{3}$$

$$b_{4,3} = b_{3,4} = \frac{(\mathrm{n}_y \mathrm{n}_z)_l}{3}$$

$$b_{3,3} = 1 + \frac{(\mathrm{n}_y \mathrm{n}_y)_l}{3}$$

$$b_{4,4} = 1 + \frac{(\mathrm{n}_z \mathrm{n}_z)_l}{3}$$

$$b_{5,2} = \mathrm{u}_l b_{2,2} + \mathrm{v}_l b_{3,2} + \mathrm{w}_l b_{4,2} - \frac{\gamma \mathrm{u}_L}{\mathrm{Pr}}$$

$$b_{5,3} = \mathrm{u}_l b_{2,3} + \mathrm{v}_l b_{3,3} = \mathrm{w}_l b_{4,3} - \frac{\gamma \mathrm{v}_L}{\mathrm{Pr}}$$

$$b_{5,4} = \mathrm{u}_l b_{2,4} + \mathrm{v}_l b_{3,4} + \mathrm{w}_l b_{4,4} - \frac{\gamma \mathrm{w}_L}{\mathrm{Pr}}$$

$$b_{5,5} = \frac{\gamma}{\mathrm{Pr}}$$

and u_l, v_l, w_l, and u_l have been treated as constants for the purpose of evaluating the Jacobian. The eigenvalues of $\mathbf{B}_{l,L}$ are all nonnegative and equal to $\frac{\mu_l(\vec{\nabla}\chi \cdot \vec{\mathrm{n}})_l}{\mathrm{Re}\ \rho_L}\left[0, 1, 1, \frac{4}{3}, \frac{\gamma}{\mathrm{Pr}}\right]^T$. The eigenvalue equal to 0 derives from the fact that there is no physical dissipation in the continuity equation. This does not present a problem because sufficient numerical dissipation for stability is introduced in the continuity equation from the treatment of the inviscid terms.

In like manner, the Jacobian $\mathbf{B}_{l+1,L}$ is defined by

$$\mathbf{B}_{l+1,L} = \frac{\partial \mathbf{h}^*_{l+1,L}}{\partial \mathbf{q}_L} \equiv \frac{\partial \mathbf{h}'^*_{l+1,L}}{\partial \mathbf{q}_L} \tag{C2a}$$

$$\mathbf{B}_{l+1,L} = \frac{\mu_{l+1}(\vec{\nabla}\chi \cdot \vec{\mathrm{n}})_{l+1}}{\mathrm{Re}\ \rho_L}\left[b_{irow,jcol}\right] \tag{C2b}$$

In this case, the expressions for $b_{irow,jrow}$ are exactly the same as above except that subscript $l+1$ is substituted for subscript l.

Appendix D

Definition of Time Step δt

The Courant number is defined as the ratio of a computational time step δt to the real, physical time required for an arbitrary perturbation in a flow field to traverse a computational cell. For a given value of δt, the Courant number varies from cell to cell and the largest Courant numbers generally occur at the smallest cells. Conversely, if the Courant number is held fixed across the domain, then the solution at every cell is advanced at a different computational time step. In this case, the temporal evolution of the problem is in error, but the steady-state solution is usually independent of the path taken to convergence.

For most explicit schemes, a Courant number equal to 1.0 represents the largest time step which may be used to advance the solution without developing instabilities. Stability analyses for explicit schemes are used to define this physically limiting time step. The present formulation for time step is modeled after the formulation used in reference 38. Metric coefficients in the earlier formulation are replaced by the appropriate ratios of cell volume to cell wall area as discussed in appendix A. No stability analysis was performed on the explicit counterpart of the present algorithm (as defined by eq. (6.10) with $\mathbf{C}_L = \mathbf{I}$). This formulation serves only as a point of reference for defining a Courant number.

The computational time step at cell L $(\delta t)_L$ is given as a function of the Courant number C_N by

$$(\delta t)_L = C_N \Omega_L \frac{1}{|R_1| + |R_2| + |R_3| + R_4} \tag{D1}$$

where

$$R_1 = \vec{u}_L \cdot \vec{\nabla}\xi_L \tag{D2a}$$

$$R_2 = \vec{u}_L \cdot \vec{\nabla}\eta_L \tag{D2b}$$

$$R_3 = \vec{u}_L \cdot \vec{\nabla}\zeta_L \tag{D2c}$$

$$R_4^2 = a_L^2 \left(\vec{\nabla}\xi_L \cdot \vec{\nabla}\xi_L + \vec{\nabla}\eta_L \cdot \vec{\nabla}\eta_L + \vec{\nabla}\zeta_L \cdot \vec{\nabla}\zeta_L + 2|\vec{\nabla}\xi_L \cdot \vec{\nabla}\eta_L| + 2|\vec{\nabla}\xi_L \cdot \vec{\nabla}\zeta_L| + 2|\vec{\nabla}\eta_L \cdot \vec{\nabla}\zeta_L| \right) \tag{D2d}$$

The cell-centered metric coefficients are defined as

$$\vec{\nabla}\xi_L = \frac{1}{2}\left(\vec{\nabla}\xi_{i,J,K} + \vec{\nabla}\xi_{i+1,J,K}\right) \tag{D3a}$$

$$\vec{\nabla}\eta_L = \frac{1}{2}\left(\vec{\nabla}\eta_{I,j,K} + \vec{\nabla}\eta_{I,j+1,K}\right) \tag{D3b}$$

$$\vec{\nabla}\zeta_L = \frac{1}{2}\left(\vec{\nabla}\zeta_{I,J,k} + \vec{\nabla}\zeta_{I,J,k+1}\right) \tag{D3c}$$

and the face-centered metrics are defined in equations (A8a), (A8e), and (A8i). The terms R_1, R_2, and R_3 represent the inverse of the transit time for a convective wave to cross a cell in the ξ, η, and ζ directions, respectively. The term R_4 represents the inverse of the transit time for an acoustic wave to cross a characteristic length of the cell. Viscous contributions to the formulation of the time step have been ignored.

Appendix E

Boundary Conditions

Boundary conditions considered in the present work include uniform supersonic inflow, supersonic outflow with subsonic boundary layer, no-slip wall, plane of symmetry, and axis singularity. Boundary conditions are imposed within the context of a cell-centered scheme which utilizes pseudo cells on the opposite side of the boundary. Pseudo cells are not explicitly defined, except for the cell face which lies on the real boundary and is shared with the real cell inside the computational domain. Pseudo cell volumes are extrapolated from interior cells as follows:

$$\Omega_0 = \frac{\Omega_1^2}{\Omega_2} \tag{E1}$$

Values of dependent variables at pseudo cell centers and values of s_l behind the boundary ($l = 0$) are required to fully define the problem. When the dependent variables in a pseudo cell are a function of the interior cells, the most recent available data from the interior are used.

Uniform Supersonic Inflow

Values of dependent variables at the pseudo cell center are fixed at free-stream conditions. Values of $\mathbf{s}_0$ are set equal to 0.

Supersonic Outflow With Subsonic Boundary Layer

First-order extrapolation is generally used to define dependent variables at pseudo cell centers and is a more stable boundary condition in the highly nonlinear initialization phase of the solution process. Second-order extrapolation has been used in some tests, but the results were not significantly different from those obtained with the first-order methods. Positive definite quantities are extrapolated to second-order accuracy as in equation (E1). First-order extrapolation is used to define $\mathbf{s}_0$ at the pseudo cell wall behind the boundary so that $\mathbf{s}_0 = \mathbf{s}_1$. Because of the upwind nature of the approximation scheme and the hyperbolic/parabolic nature of the outflow boundary, the influence of the pseudo cell specification on the interior points is very weak.

No-Slip Wall

A zero normal pressure gradient is enforced by imposing the pressure at the boundary cell onto the reflected pseudo cell. Adiabatic wall boundary conditions are enforced by imposing the total enthalpy at the boundary cell onto the reflected pseudo cell. Cold wall boundary conditions and no-slip velocity boundary conditions are enforced in one of two ways. The simplest and most stable method is to impose the exact wall boundary conditions at the pseudo cell center, which is in fact half a cell away from the real boundary. In practice, if the grid is fine enough to adequately resolve boundary-layer gradients, then the offset of the pseudo cell center from the boundary is small and the influence of this offset is minimal. (Grid resolution is judged adequate when the cell height Δh at the wall boundary is approximately equal to the local value of $\mu/\rho a$.) A numerically more accurate boundary condition is to specify the pseudo cell velocity and energy such that the interpolated wall boundary value is correct. Thus,

$$\vec{u}_0 = -\vec{u}_1 \tag{E2}$$

$$e_0 = 2e_{\text{wall}} - e_1 \tag{E3}$$

Equations (E2) and (E3) are more likely to lead to instabilities in the early initialization phase of the developing solution. First-order extrapolation is used to define $\mathbf{s}_0$ at the pseudo cell wall

behind the boundary so that $\mathbf{s}_0 = \mathbf{s}_1$. In effect, this specification of $\mathbf{s}_0$ restricts the flux limiter of equation (3.8) to consider only gradients at the surface and in the interior of the domain. So long as the min mod function does not return a zero argument, the accuracy of the scheme is still formally of second order.

Symmetry Plane

Pseudo cells behind the plane of symmetry correspond to actual interior cells. Except for the reflection of the velocity component normal to the symmetry plane, the pseudo cells are defined by imposing the dependent variables from the corresponding reflected cells in the interior. Values of the gradients $\mathbf{s}_0$ are calculated directly from reflected cell center data.

Axis Singularity

An axis singularity occurs when the computational domain is constructed by rotating a two-dimensional grid around some natural axis of the body or the flow. The flow itself does not have to be axisymmetric for this construction to exist. Pie-shaped cells with zero cell-wall area on the axis surround the axis singularity. Pseudo cells behind the axis correspond to actual interior cells which are reflected across the symmetry plane. Specification of dependent variables at pseudo cell centers and gradients at cell walls behind the axis proceed similarly to the symmetry plane boundary. However, if J is the index of the boundary cell and JN is the number of cells in the semicircle surrounding the axis, then the corresponding pseudo cell is a reflection of the interior cell with index $JN + 1 - J$. Pseudo cell specification has no effect on the first-order accurate algorithm ($\theta = 0$) because the common cell wall has zero area. The gradient across the axis will influence the outcome of the flux limiter in equation (3.8) for the far wall ($l = 2$) of the boundary cell in the second-order algorithm.

Appendix F

Unstructured Grids

The formulations contained in this paper are based on a structured grid system (i.e., the computational cells are rectangularly ordered with indices i, j, and k, and nearest neighbor cells differ in index by one). Unstructured grids are generally composed of triangular cells in two dimensions and tetrahedral cells in three dimensions. In the most general case, they cannot be ordered as described for structured grids; however, they have an advantage of being completely adaptable to very complex geometries and flow structures. In light of recent developments in unstructured grid formulations (ref. 39), some comments are offered with regard to modifications required in the present analysis for such applications.

The finite-volume approximation to equation (2.1) for a general, unstructured grid is written as

$$\left[\frac{\delta \mathbf{q}\Omega}{\delta t}\right]_L + \sum \vec{\mathbf{f}}_m \cdot \vec{\mathbf{n}}_m \sigma_m = 0 \tag{F1}$$

where $\delta\mathbf{q} = \mathbf{q}^{n+1} - \mathbf{q}^n$ and $\delta t = t^{n+1} - t^n$. The summation is over all the faces of cell L enclosing volume Ω. Subscript m indicates a quantity taken on cell face m with surface area σ_m. The quantity $\vec{\mathbf{n}}_m$ is a unit vector normal to cell face m in a direction facing away from the cell center.

The geometric derivations of metrics in appendix A are based on a rectangularly ordered, structured grid system. Some modification of these formulations would be required for general, unstructured grids. One point to consider is that the unit normal to a cell face in the unstructured grid formulation $\vec{\mathbf{n}}_m$ faces away from the cell center, whereas the unit normal to a cell face in the structured grid formulation $\vec{\mathbf{n}}_l$ faces in the direction of increasing index l. This convention accounts for the minus signs appearing in equations (2.4) and (2.5).

A first-order accurate formulation of the Euler equations on a structured grid is identical to the formulation on an unstructured grid. Modifications are required to achieve second-order accuracy. Clearly, the present formulation requires an ordered grid system to define $\mathbf{s}_l^{\min}$. The construction of an equivalent limiter from neighboring cells in an unstructured grid should be possible but has not been investigated. In like manner, the formulation of the viscous terms, which depends on ordered computational directions in the present case, will need to be reformulated in the unstructured grid case. However, the formulations for obtaining second-order accuracy with the complete Navier-Stokes equations will only involve modifications to the right-hand-side residual vector. The point-implicit relaxation procedure that is defined by the formulation of the left-hand-side matrix is independent of grid structure. Consequently, the development of point-implicit relaxation will carry over with minimal changes in unstructured grid formulations.

Appendix G

Asynchronous Iteration

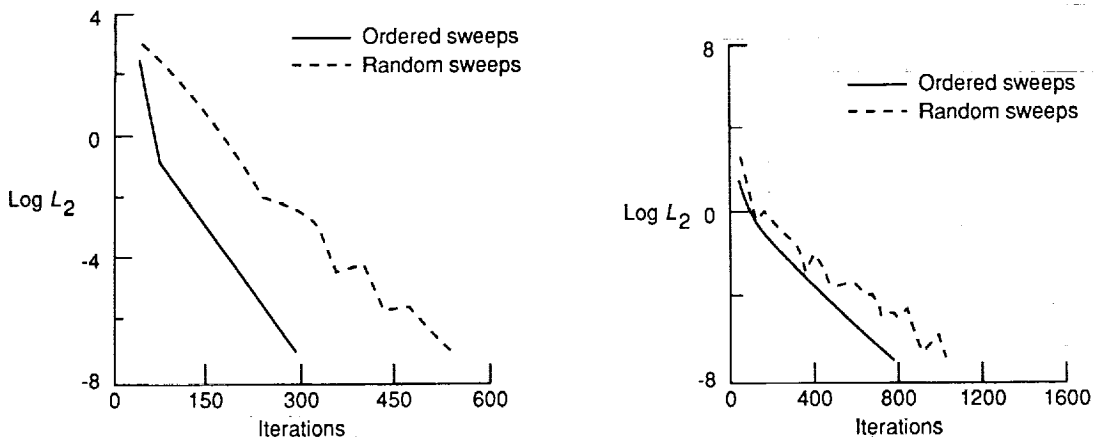
A two-dimensional scalar version of the present algorithm was modified to test the feasibility of executing on a computer with a massively parallel, asynchronous processing architecture. The test is only a simulation in that it was conducted on a serial machine. The environment to be simulated is one in which each computational cell has associated with it a single processor which is to be kept busy at all times executing its own copy of the master algorithm. When the algorithm in a particular cell (processor) requires data from a neighbor cell, it gathers the latest available data from local shared memory, regardless of the iteration level of the neighbor or where the neighbor is within its own processing of its copy of the algorithm. As the global solution develops, individual processors may get out of step. This loss of synchronization may arise from variations in instructions at boundary cells, different branches of instructions resulting from conditional branches within the algorithm, and potential hardware differences in neighboring processors.

This environment is crudely simulated on a single processor, serial machine by using a random number generator to direct the order that cells are relaxed in the computational domain. A version of the present algorithm was coded in which a single index L ordered the relaxation sweeps through a two-dimensional domain. All aspects of a relaxation step were completed at cell L before proceeding to cell $L + 1$. Convergence histories were recorded for the ordered sweeps. The loop over index L was then replaced by a loop over dummy index LL, and index L was redefined by a random number generator which ranged from 1 to LMAX, where LMAX is the total number of cells. Convergence histories were then recorded for the random sweeps, in which case a sweep is defined as one complete pass through the loop. Within a given sweep, some cells will likely be called more than one time, and other cells will not be called at all. As sweeps continue, the iteration levels of neighbor cells are not generally equal. This random offset of iteration levels of neighbor cells executed on a single processor is used to simulate the asynchronous iteration of cells on a massively parallel processor. This simulation does not model all the possible ways that processors can get out of synch on a real, parallel machine. However, it does serve to demonstrate the potential applicability of upwind-biased, point-implicit relaxation schemes on asynchronous, massively parallel processors.

The tests were conducted on two supersonic flows over cylinders. Both tests used initial conditions for a body materializing in a uniform, supersonic flow at time $t = 0$. The first test involved Mach 3 inviscid flow and used the first-order scheme with variable time steps based on a Courant number of 2.0. The second test involved Mach 1.9 viscous flow (using the thin-layer Navier-Stokes approximation) at a Reynolds number of 105 based on cylinder radius and used the second-order scheme with variable time steps based on a Courant number of 0.75. The convergence histories of the two tests are presented in figure G1. The random sweeps take twice as long as ordered sweeps in the inviscid case and 1.25 times as long in the viscous case. In both cases, the converged solutions were identical to at least four significant figures, indicating that the order of relaxation contributes to the rate of convergence but not to the final converged solution.

The converged viscous solution and experimental data (ref. 40) for the pressure coefficient are presented in figure G2. The comparison in this case is good in the stagnation region but breaks down near the outflow boundary. A calculation using the full Navier-Stokes equations with slip-flow boundary conditions would be more appropriate for this case, but this capability was not included in the serial version of the code.

The success of these tests bodes well for the possibility of using an upwind-biased, point-implicit relaxation scheme on asynchronous, massively parallel computers. One recognizes that there are potential obstacles in the design of a computer that could exploit the synergism



(a) Inviscid calculation; $M_\infty = 3.0$.

(b) Viscous calculation; $M_\infty = 1.9$.

Figure G1. Convergence history for calculations on serial computer using ordered and random sweeps across domain for two-dimensional supersonic flow over cylinder.

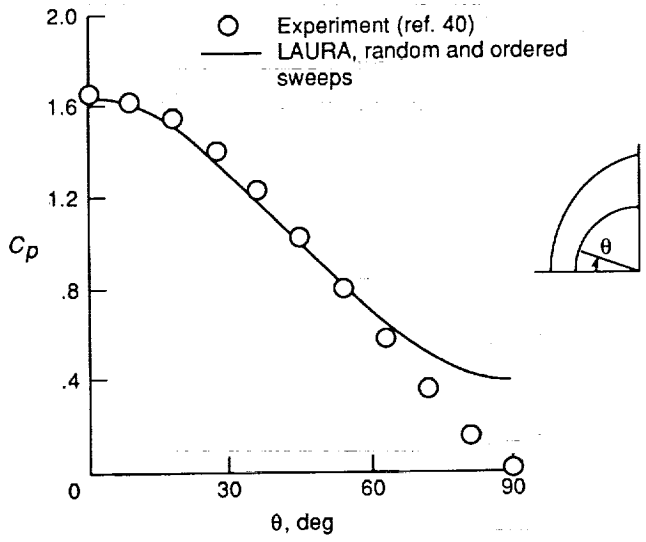


Figure G2. Experimental data for pressure coefficient and prediction for $M_\infty = 1.9$ case with ordered and random sweeps.

of algorithm and architecture described here. Problems involving sharing memory among so many processors may overwhelm any potential benefits. However, the real-world challenges facing computational fluid dynamicists today, not to mention the computational challenges faced by other disciplines, call for algorithms which must deal with millions of computational cells that are patched together in ordered subsets or that are completely unstructured. Hardware restrictions limit significant speedup of machines with only vector architecture for these problems. The optimum blend of vector and parallel features is likely to be algorithm dependent, and benchmarking of tomorrow's supercomputers should not be based strictly on today's algorithms that were optimized for today's architectures. In anticipation of these developments, an upwind-biased, point-implicit relaxation scheme should offer the flexibility required for efficient implementation on either vector or massively parallel machines (or some combination thereof) utilizing structured or unstructured grids.

References

1. Osher, Stanley: Riemann Solvers, the Entropy Condition, and Difference Approximations. *SIAM J. Numer. Anal.*, vol. 21, no. 2, Apr. 1984, pp. 217–235.
2. Roe, P. L.: Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comput. Phys.*, vol. 43, no. 2, Oct. 1981, pp. 357–372.
3. Steger, Joseph L.; and Warming, R. F.: Flux Vector Splitting of the Inviscid Gasdynamic Equations With Application to Finite-Difference Methods. *J. Comput. Phys.*, vol. 40, no. 2, Apr. 1981, pp. 263–293.
4. Van Leer, Bram: Flux-Vector Splitting for the Euler Equations. *Eighth International Conference on Numerical Methods in Fluid Dynamics*, E. Krause, ed., Volume 170 of *Lecture Notes in Physics*, Springer-Verlag, 1982, pp. 507–512.
5. Zalesak, Steven T.: Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids. *J. Comput. Phys.*, vol. 31, no. 3, June 1979, pp. 335–362.
6. Boris, Jay P.; and Book, David L.: Flux-Corrected Transport. I. SHASTA, a Fluid Transport Algorithm That Works. *J. Comput. Phys.*, vol. 11, no. 1, Jan. 1973, pp. 38–69.
7. Harten, Ami: High Resolution Schemes for Hyperbolic Conservation Laws. *J. Comput. Phys.*, vol. 49, no. 2, Feb. 1983, pp. 357–393.
8. Osher, Stanley; and Chakravarthy, Sukumar: *Very High Order Accurate TVD Schemes.* ICASE Rep. No. 84-44 (Contract NAS1-17070), 1984.
9. Sweby, P. K.: High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.*, vol. 21, no. 5, Oct. 1984, pp. 995–1011.
10. Yee, H. C.: *On Symmetric and Upwind TVD Schemes.* NASA TM-86842, 1985.
11. Yee, H. C.: *Numerical Experiments With a Symmetric High-Resolution Shock-Capturing Scheme.* NASA TM-88325, 1986.
12. Yee, H. C.; Klopfer, G. H.; and Montagné, J.-L.: *High-Resolution Shock-Capturing Schemes for Inviscid and Viscous Hypersonic Flows.* NASA TM-100097, 1988.
13. Harten, Ami; Engquist, Bjorn; Osher, Stanley; and Chakravarthy, Sukumar R.: *Uniformly High Order Accurate Essentially Non-Oscillatory Schemes III.* NASA CR-178101, ICASE Rep. No. 86-22, 1986.
14. Harten, Ami: *ENO Schemes With Subcell Resolution.* NASA CR-178362, ICASE Rep. No. 87-56, 1987.
15. Shu, Chi-Wang; and Osher, Stanley: *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes.* NASA CR-178304, ICASE Rep. No. 87-33, 1987.
16. Chang, Shih-Hung; and Liou, Meng-Sing: A Comparison of ENO and TVD Schemes. *Technical Papers, Part 1—AIAA/ASME/SIAM/APS 1st National Fluid Dynamics Congress*, July 1988, pp. 427–434. (Available as AIAA-88-3707.)
17. Walberg, Gerald D.: A Survey of Aeroassisted Orbit Transfer. *J. Spacecr. & Rockets*, vol. 22, no. 1, Jan.–Feb. 1985, pp. 3–18.
18. Coakley, T. J.: Implicit Upwind Methods for the Compressible Navier-Stokes Equations. *A Collection of Technical Papers—6th AIAA Computational Fluid Dynamics Conference*, July 1983, pp. 505–514. (Available as AIAA-83-1958.)
19. Gnoffo, Peter A.: Application of Program LAURA to Three-Dimensional AOTV Flowfields. AIAA-86-0565, Jan. 1986.
20. Gnoffo, Peter A.; and McCandless, Ronald S.: Three-Dimensional AOTV Flowfields in Chemical Nonequilibrium. AIAA-86-0230, Jan. 1986.
21. Gnoffo, Peter A.; McCandless, Ronald S.; and Yee, H. C.: Enhancements To Program LAURA for Computation of Three-Dimensional Hypersonic Flow. AIAA-87-0280, Jan. 1987.
22. Gnoffo, Peter A.; and Greene, Francis A.: A Computational Study of the Flowfield Surrounding the Aeroassist Flight Experiment Vehicle. AIAA-87-1575, June 1987.
23. Allen, J. S.; and Cheng, S. I.: Numerical Solutions of the Compressible Navier-Stokes Equations for the Laminar Near Wake. *Phys. Fluids*, vol. 13, no. 1, Jan. 1970, pp. 37–52.
24. Graves, Randolph Anderson, Jr.: Solutions to the Navier-Stokes Equations for Supersonic Flow Over Blunt Bodies With Massive Wall Blowing. Ph.D. Diss., George Washington Univ., 1977.
25. Thomas, James L.; and Walters, Robert W.: Upwind Relaxation Algorithms for the Navier Stokes Equations. *A Collection of Technical Papers—AIAA 7th Computational Fluid Dynamics Conference*, July 1985, pp. 117–128. (Available as AIAA-85-1501.)
26. Lombard, C. K.; Bardina, J.; and Venkatapathy, E.: AOTV Bluff Body Flow—Relaxation Algorithm. *Thermal Design of Aeroassisted Orbital Transfer Vehicles*, H. F. Nelson, ed., American Institute of Aeronautics and Astronautics, Inc., c.1985, pp. 85–112. (Available as AIAA-84-1699.)
27. Montagné, J.-L.; Yee, H. C.; and Vinokur, M.: *Comparative Study of High-Resolution Shock-Capturing Schemes for a Real Gas.* NASA TM-100004, 1987.

28. Yee, H. C.: Construction of Explicit and Implicit Symmetric TVD Schemes and Their Applications. *J. Comput. Phys.*, vol. 68, no. 1, Jan. 1987, pp. 151–179.
29. Vincenti, Walter G.; and Kruger, Charles H., Jr.: *Introduction to Physical Gas Dynamics.* John Wiley & Sons, Inc., c.1965.
30. Voigt, Robert G.: *Where Are the Parallel Algorithms?* NASA CR-172516, ICASE Rep. No. 85-2, 1985.
31. Rudy, David H.; Kumar, Ajay; Thomas, James L.; Gnoffo, Peter A.; and Chakravarthy, Sukumar R.: A Comparative Study and Validation of Upwind and Central-Difference Navier-Stokes Codes for High-Speed Flows. Paper presented at the AGARD Symposium on Validation of Computational Fluid Dynamics (Lisbon, Portugal), May 2–5, 1988.
32. Rudy, David H.; Thomas, James L.; Kumar, Ajay; Gnoffo, Peter A.; and Chakravarthy, Sukumar R.: A Validation Study of Four Navier-Stokes Codes for High-Speed Flows. AIAA-89-1838, June 1989.
33. Cheatwood, F. McNeil; DeJarnette, Fred R.; and Hamilton, H. Harris, II: *Geometrical Description for a Proposed Aeroassist Flight Experiment Vehicle.* NASA TM-87714, 1986.
34. Weilmuenster, K. James; and Hamilton, H. Harris, II: A Comparison of Computed and Measured Aerodynamic Characteristics of a Proposed Aeroassist Flight Experiment Configuration. AIAA-86-1366, June 1986.
35. Micol, John R.: Experimental and Predicted Pressure and Heating Distributions for an Aeroassist Flight Experiment Vehicle in Air at Mach 10. AIAA-89-1731, June 1989.
36. Wells, William L.: Measured and Predicted Aerodynamic Heating on a Cylinder in Wake of AFE Configuration at Incidence. AIAA-89-2162, Aug. 1989.
37. Hung, C. M.; and Kordulla, W.: A Time-Split Finite-Volume Algorithm for Three-Dimensional Flow-Field Simulation. *A Collection of Technical Papers—6th AIAA Computational Fluid Dynamics Conference*, July 1983, pp. 494–504. (Available as AIAA-83-1957.)
38. Weilmuenster, K. James; and Hamilton, H. Harris, II (appendix C by M. J. Hamilton): *Calculations of Inviscid Flow Over Shuttle-Like Vehicles at High Angles of Attack and Comparisons With Experimental Data.* NASA TP-2103, 1983, pp. 37–49.
39. Thareja, Rajiv R.; Stewart, James R.; Hassan, Obey; Morgan, Ken; and Peraire, Jaime: A Point Implicit Unstructured Grid Solver for the Euler and Navier-Stokes Equations. AIAA-88-0036, Jan. 1988.
40. Tewfik, O. K.; and Giedt, W. H.: *Heat Transfer, Recovery Factor and Pressure Distributions Around a Cylinder Normal to a Supersonic Rarefied Air Stream—Part I. Experimental Data.* Rep. HE-150-162, Inst. of Engineering Research, Univ. of California, Jan. 30, 1959.

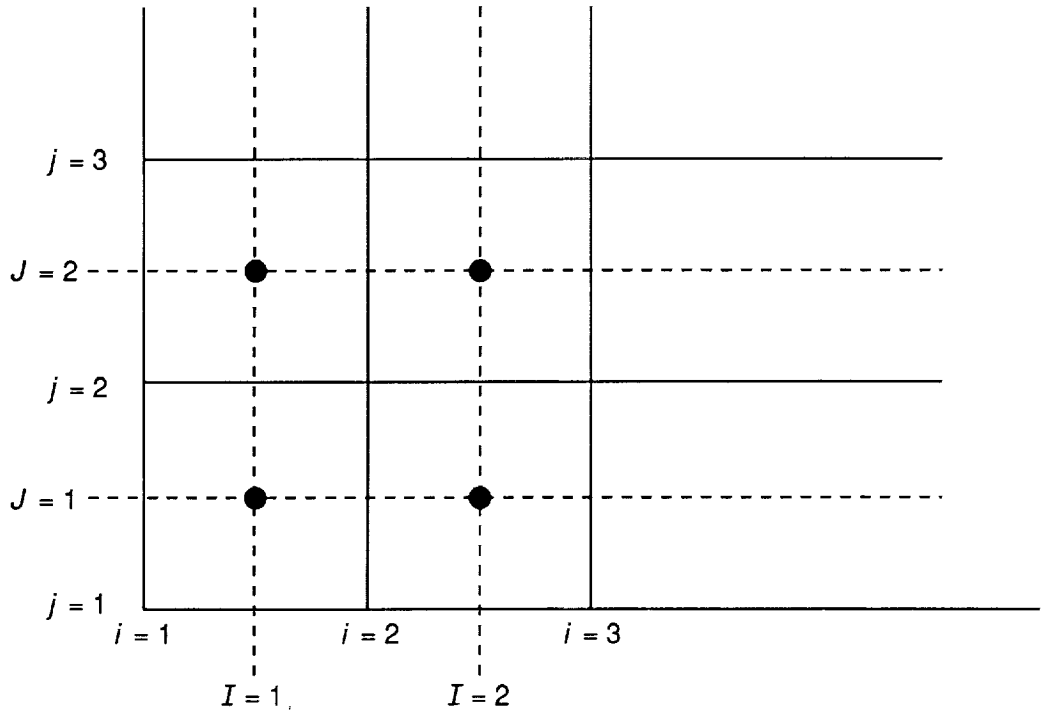


Figure 1. Cell indexing system with cell corners defined by lowercase symbols and cell centers defined by uppercase symbols.

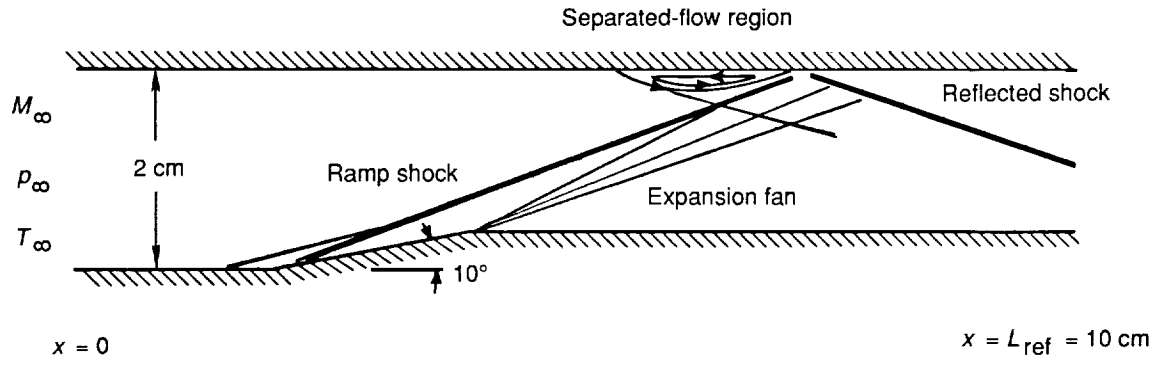
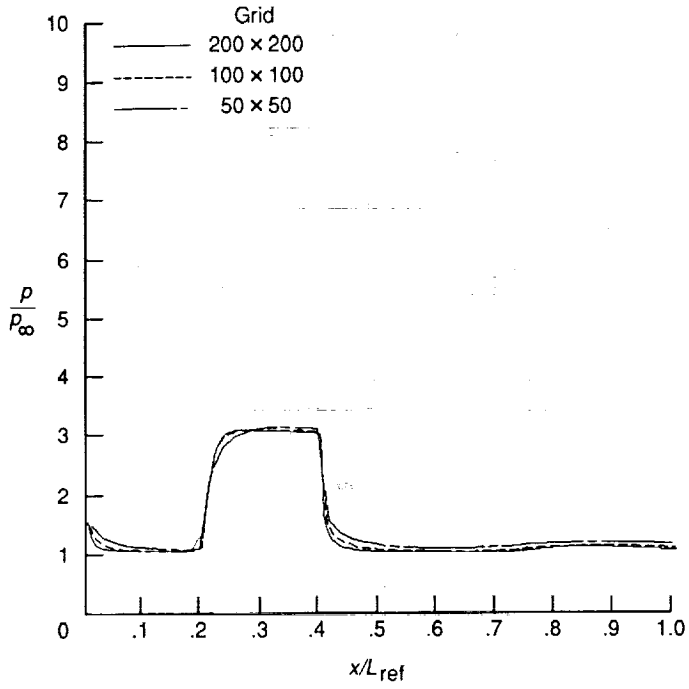
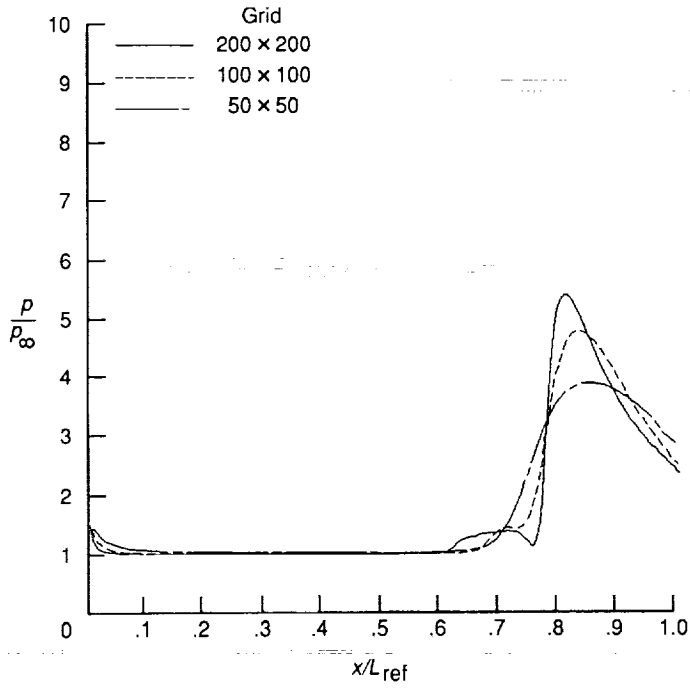


Figure 2. High-speed inlet model problem. $M_\infty = 5.0$; $\mathrm{Re}_{\infty,L} = 11.4 \times 10^6$.

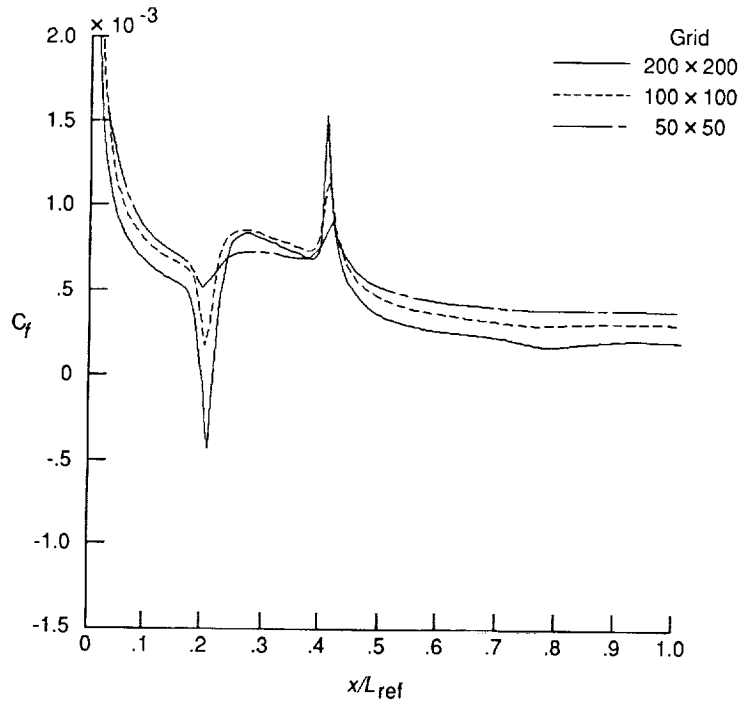


(a) Lower wall.

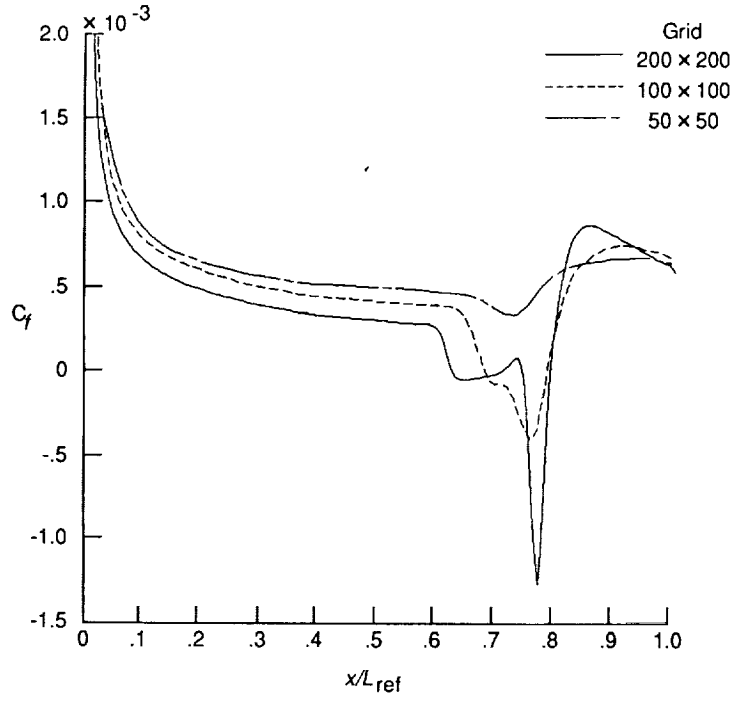


(b) Upper wall.

Figure 3. Effect of grid refinement on surface pressure for inlet with Re = 1.14×10^7 and $\epsilon_o = 0.1$.

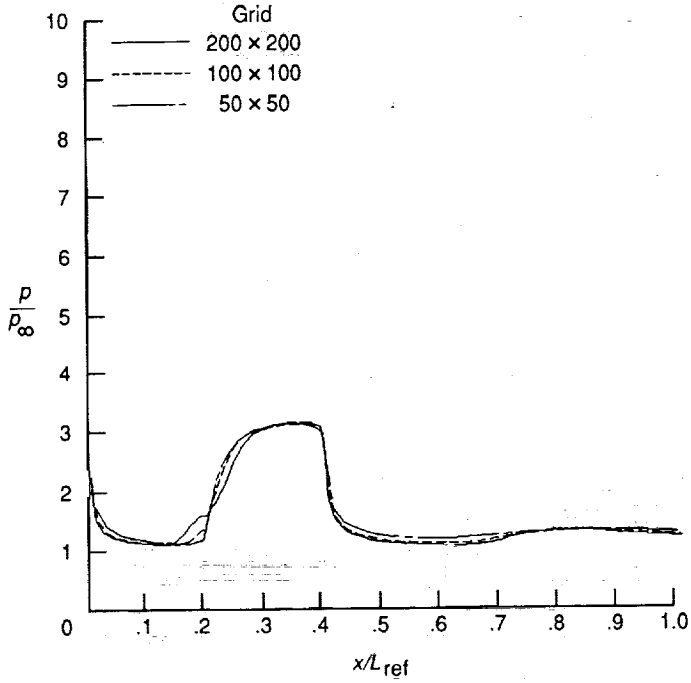


(a) Lower wall.

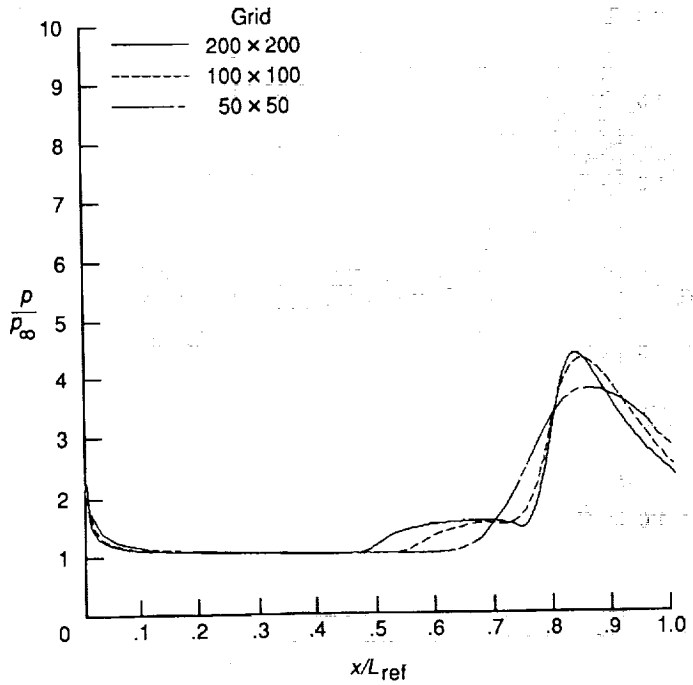


(b) Upper wall.

Figure 4. Effect of grid refinement on skin friction for the inlet with Re = 1.14×10^7 and $\epsilon_o = 0.1$.

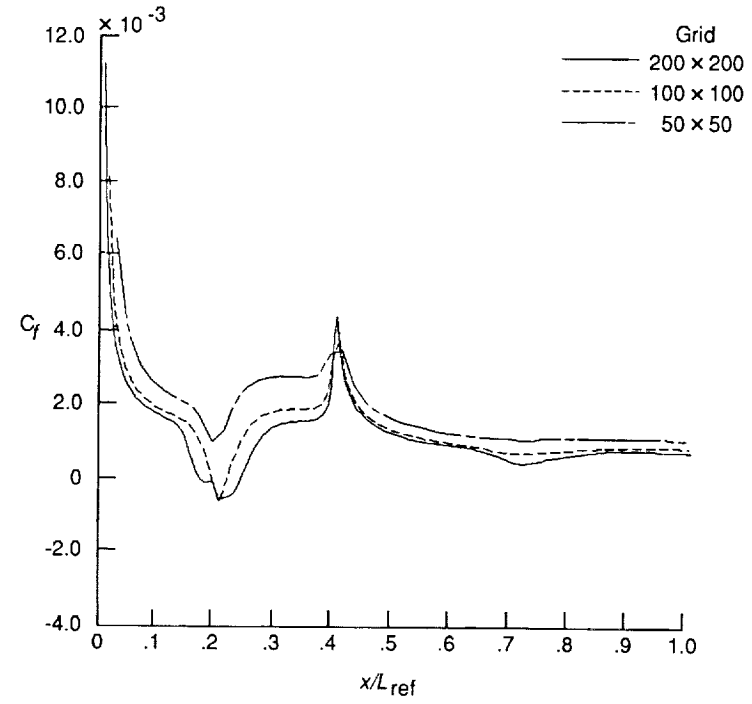


(a) Lower wall.

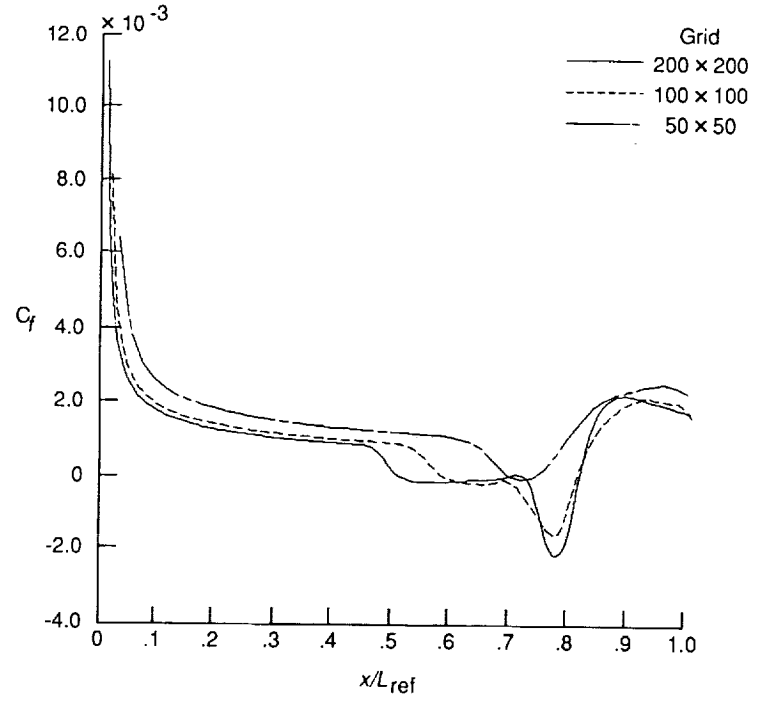


(b) Upper wall.

Figure 5. Effect of grid refinement on surface pressure for the inlet with Re = 1.14×10^6 and $\epsilon_o = 0.1$.



(a) Lower wall.



(b) Upper wall.

Figure 6. Effect of grid refinement on skin friction for the inlet with Re = 1.14×10^6 and $\epsilon_o = 0.1$.

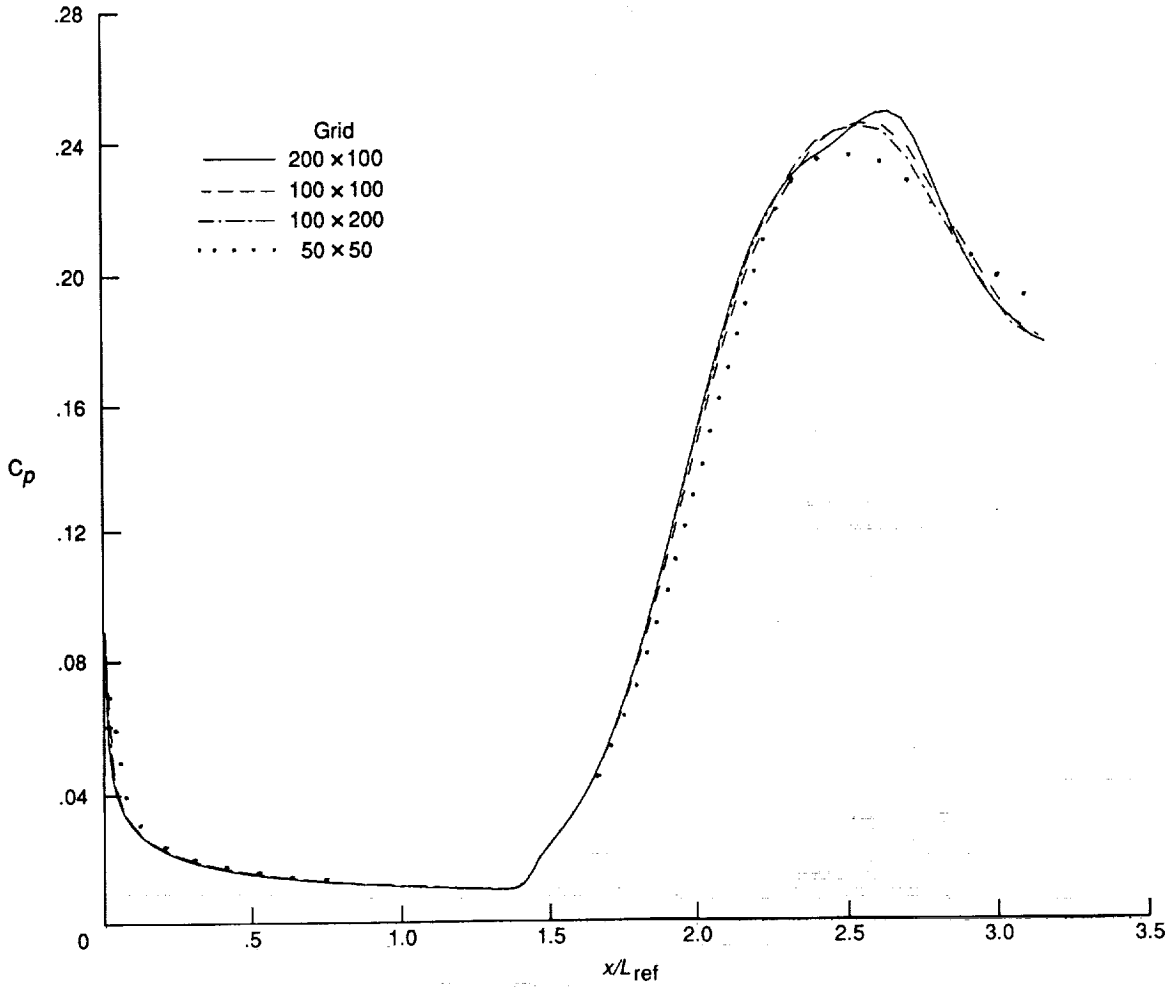


Figure 7. Effect of grid refinement on surface pressure coefficient for 15° ramp.

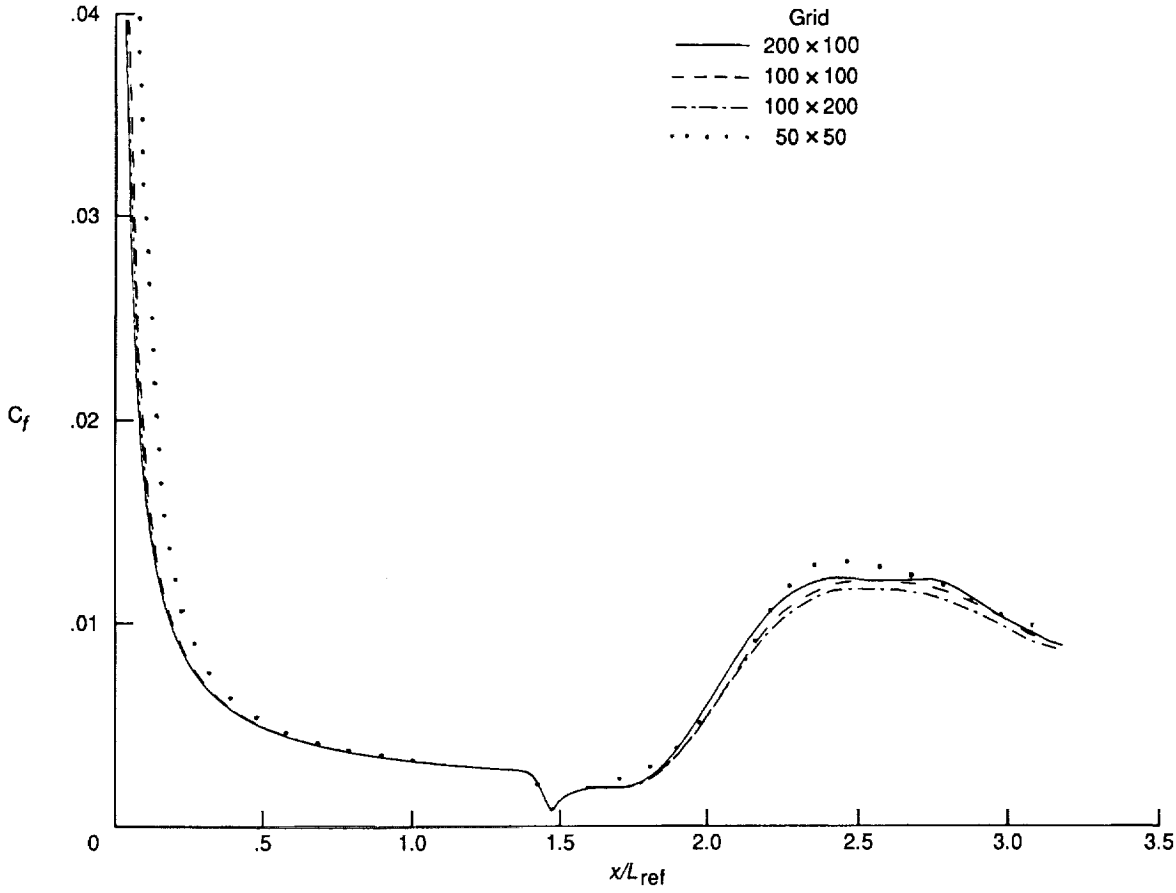
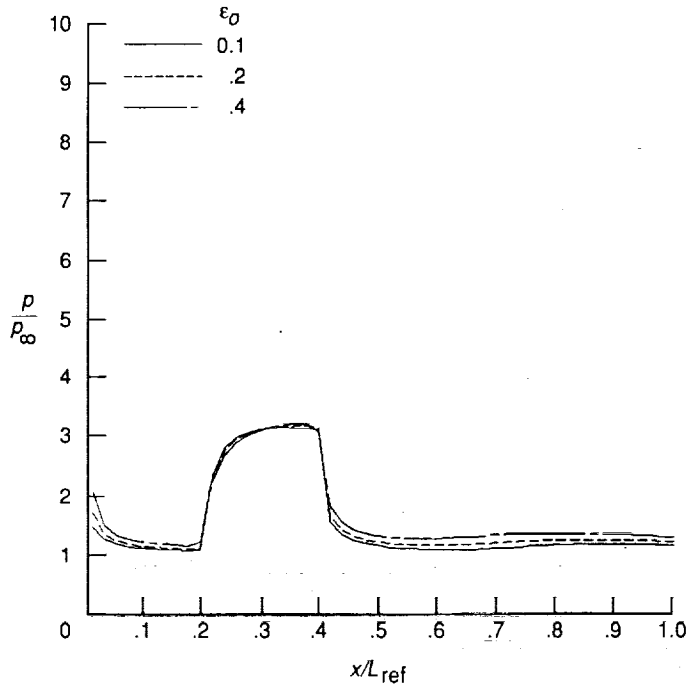
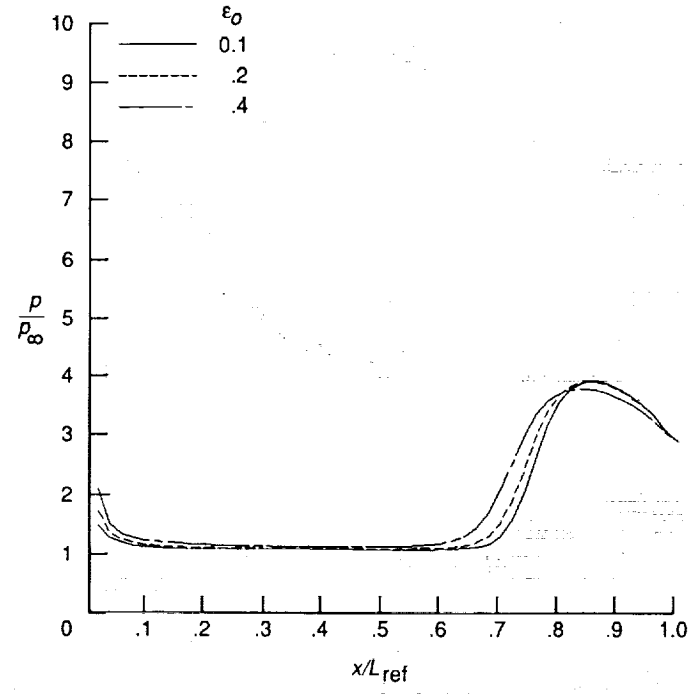


Figure 8. Effect of grid refinement on skin friction coefficient for 15° ramp.

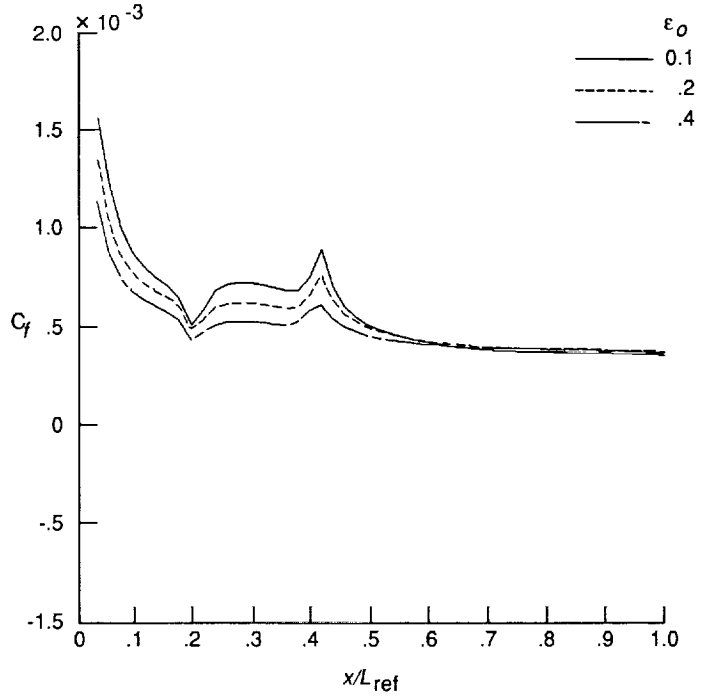


(a) Lower wall.

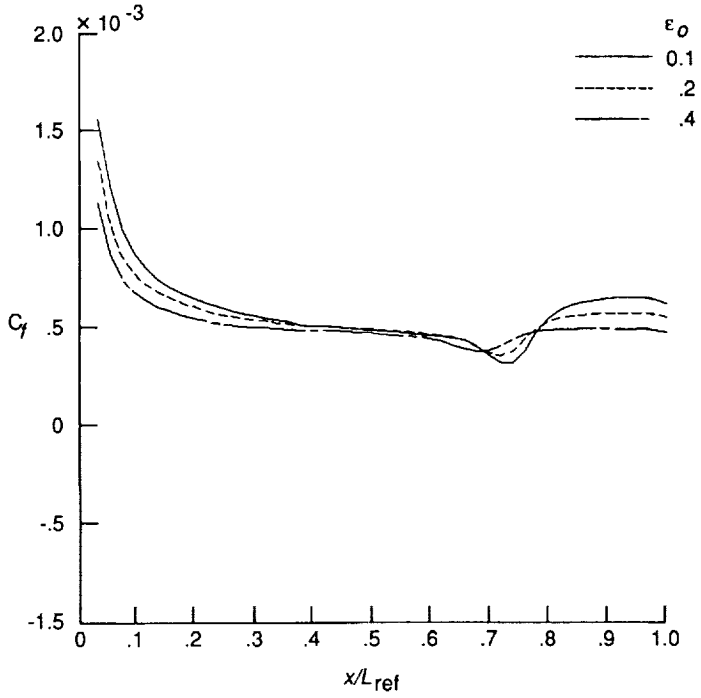


(b) Upper wall.

Figure 9. Effect of eigenvalue limiter on surface pressure for inlet with Re = 1.14×10^7 on 50×50 grid.

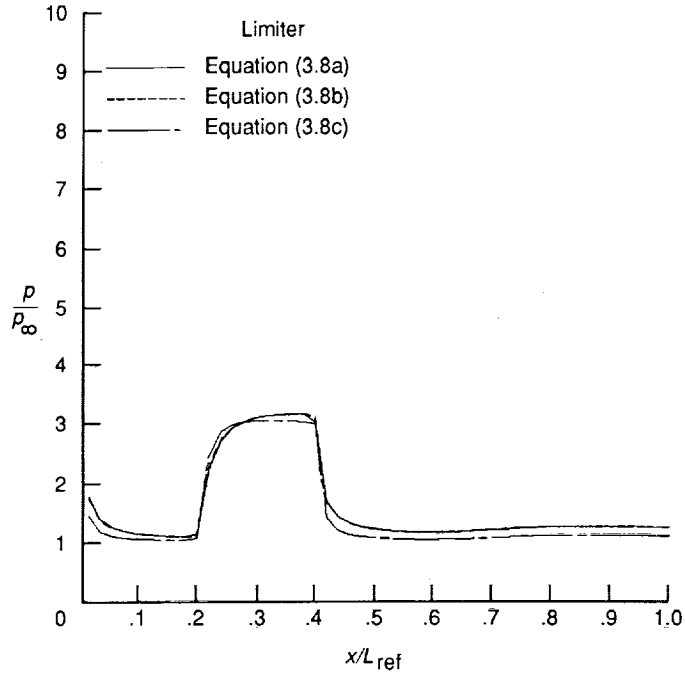


(a) Lower wall.

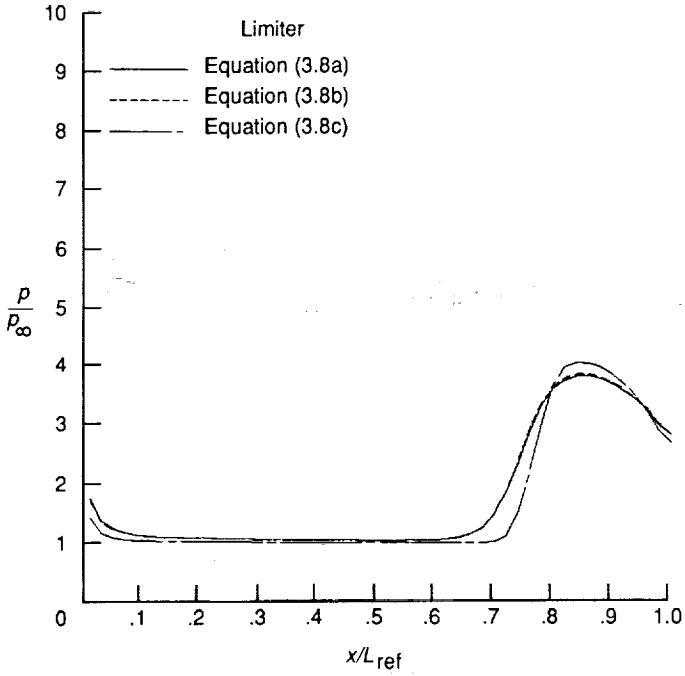


(b) Upper wall.

Figure 10. Effect of eigenvalue limiter on skin friction for inlet with $\mathrm{Re} = 1.14 \times 10^7$ on 50×50 grid.

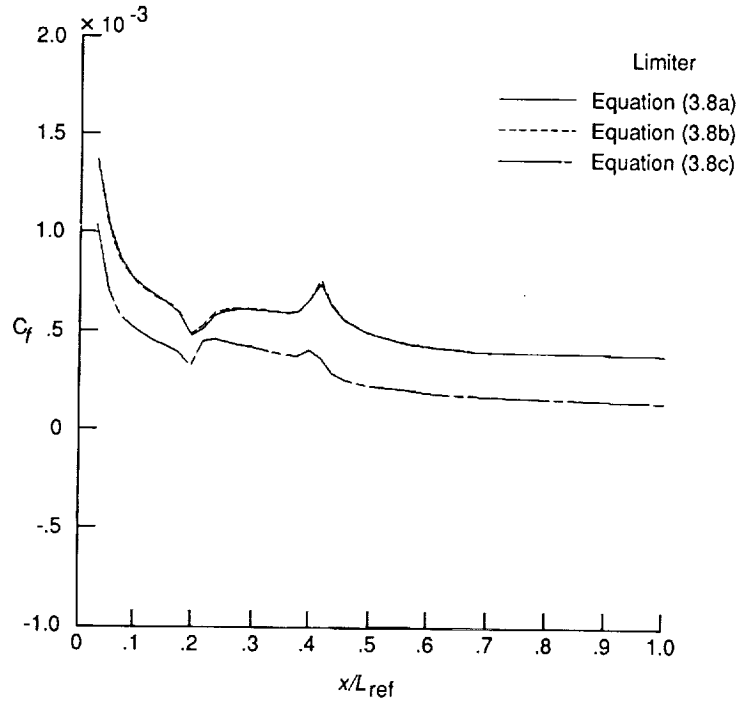


(a) Lower wall.

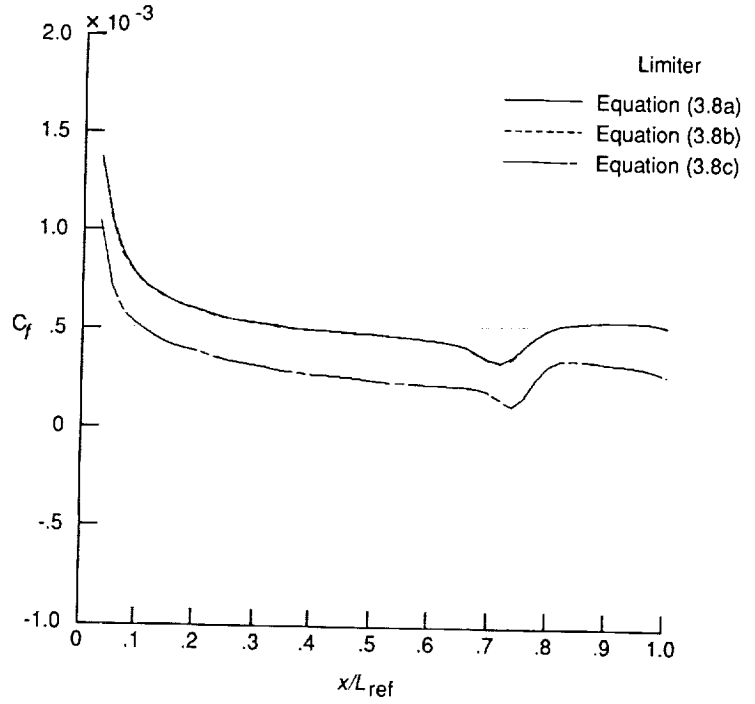


(b) Upper wall.

Figure 11. Effect of STVD limiter on surface pressure for inlet with $\mathrm{Re} = 1.14 \times 10^7$ and $\epsilon = 0.2$ on 50×50 grid.

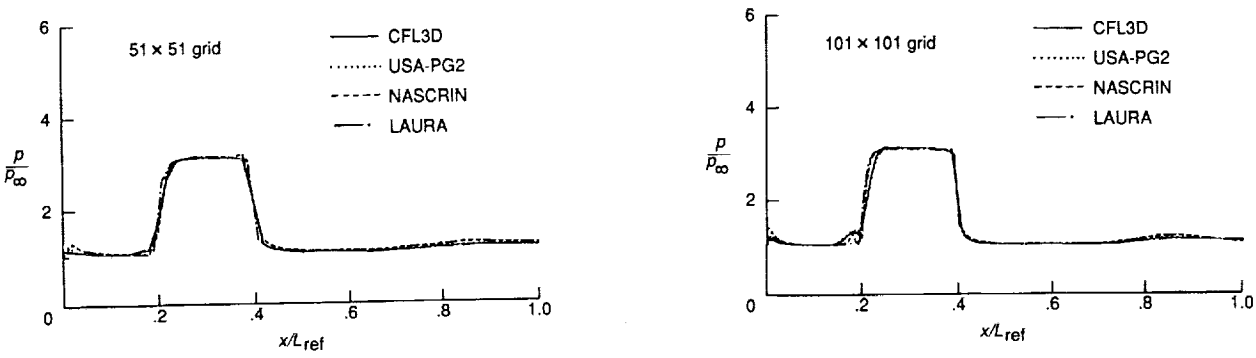


(a) Lower wall.

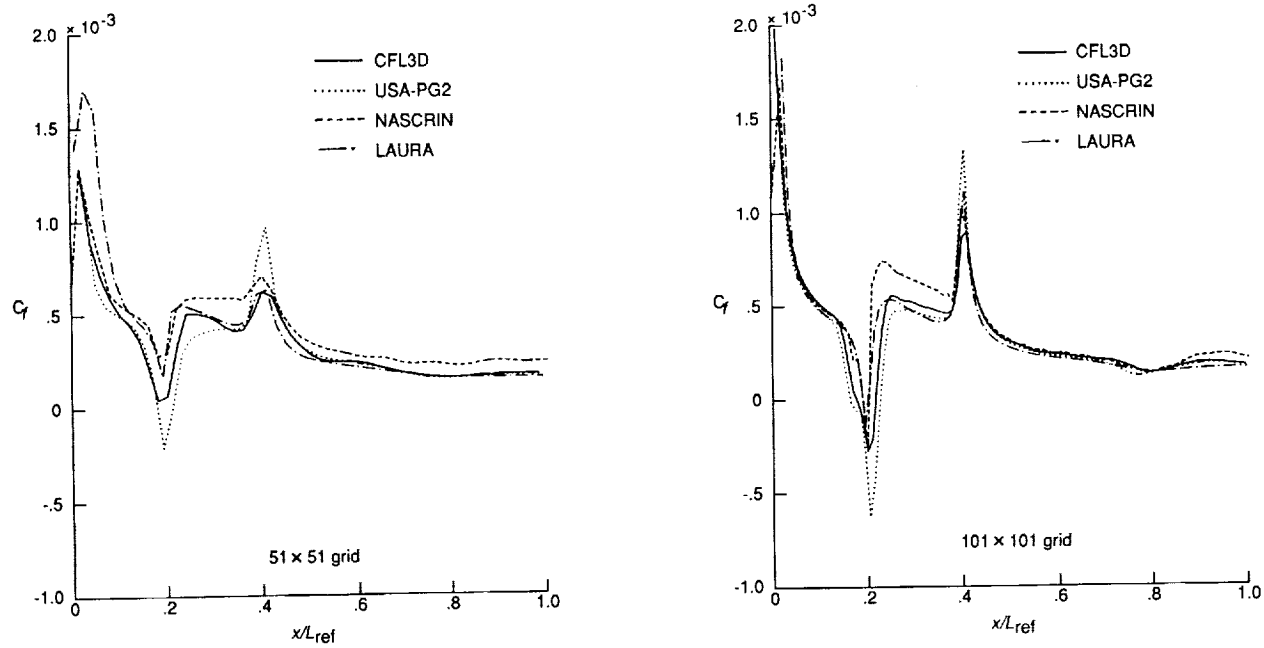


(b) Upper wall.

Figure 12. Effect of STVD limiter on skin friction for inlet with Re = 1.14×10^7 and $\epsilon = 0.2$ on 50×50 grid.

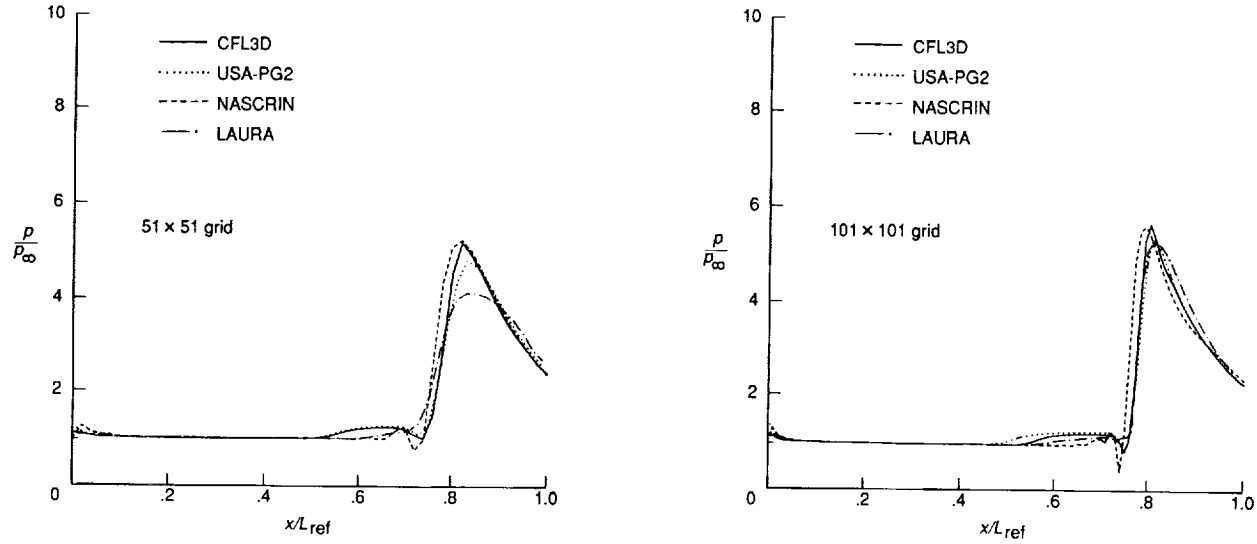


(a) Pressure; lower wall.

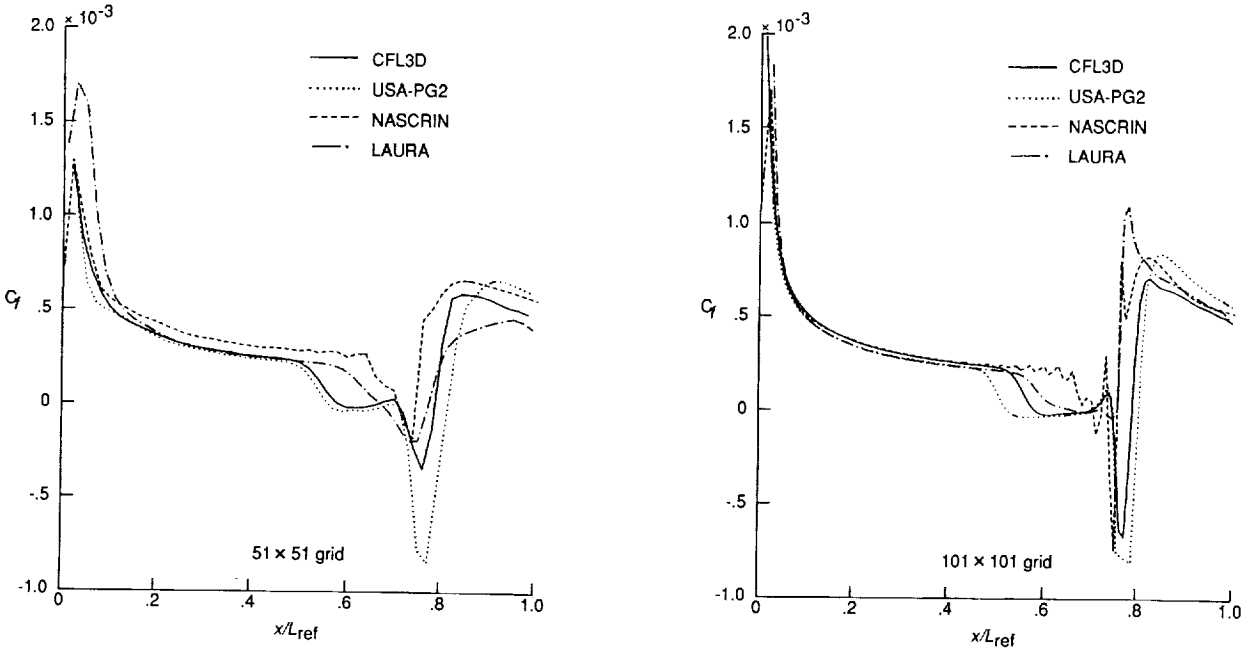


(b) Skin friction; lower wall.

Figure 13. Comparison of four algorithms for inlet problem with Re = 1.14×10^7 taken from reference 32.



(c) Pressure; upper wall.



(d) Skin friction; upper wall.

Figure 13. Concluded.

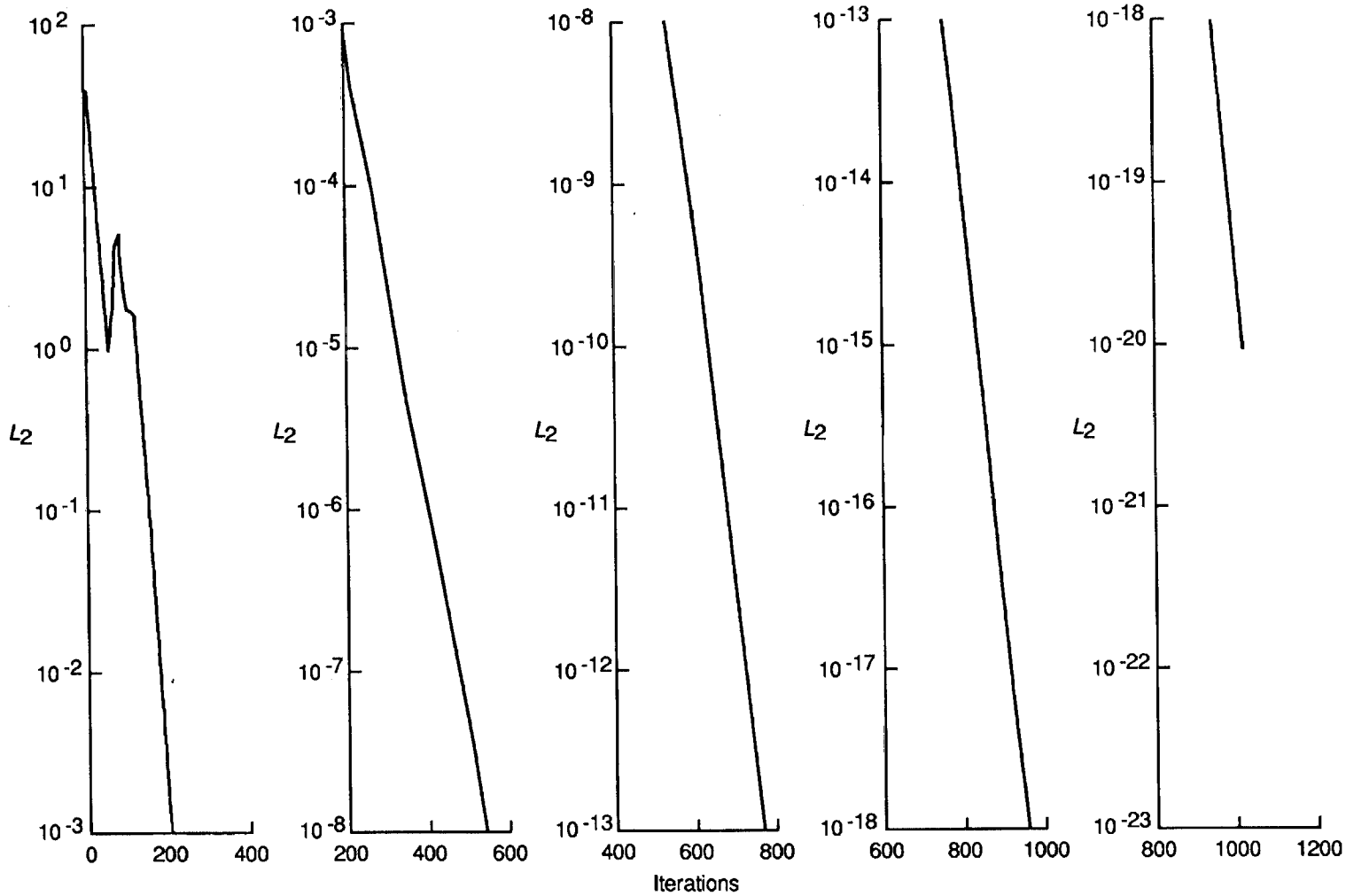


Figure 14. Convergence history for coarse (50×50) grid with $\epsilon_o = 0.2$ and $C_N = 5 \times 10^6$.

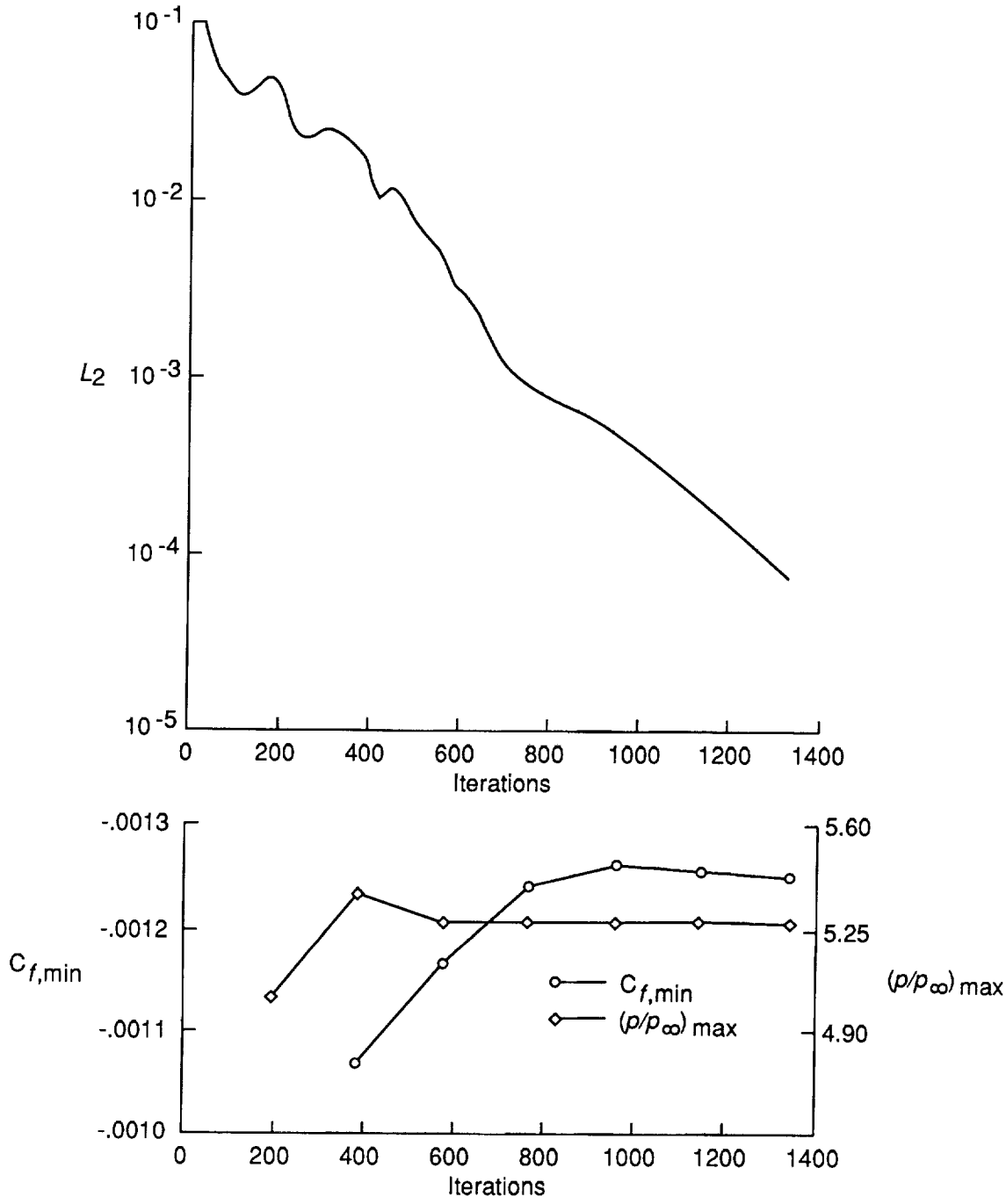


Figure 15. Convergence history for inlet problem using fine (200 × 200) grid with $\epsilon_o = 0.1$. Bottom of figure shows maximum pressure on upper wall behind reflected shock and minimum skin friction (maximum negative quantity) in separated region in front of reflected shock.

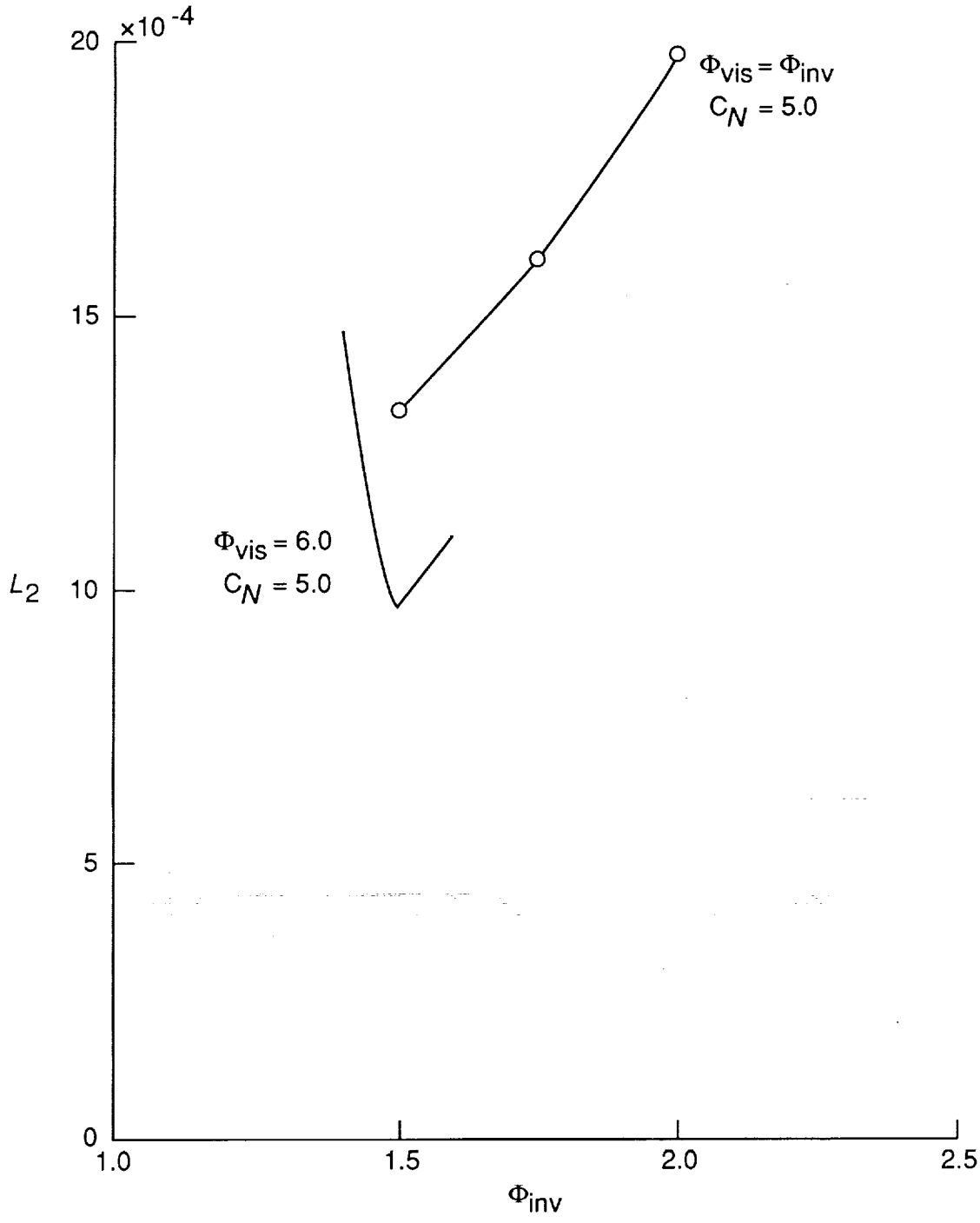


Figure 16. Effects of variations of inviscid relaxation factor on convergence levels after 700 iterations on 100×100 grid.

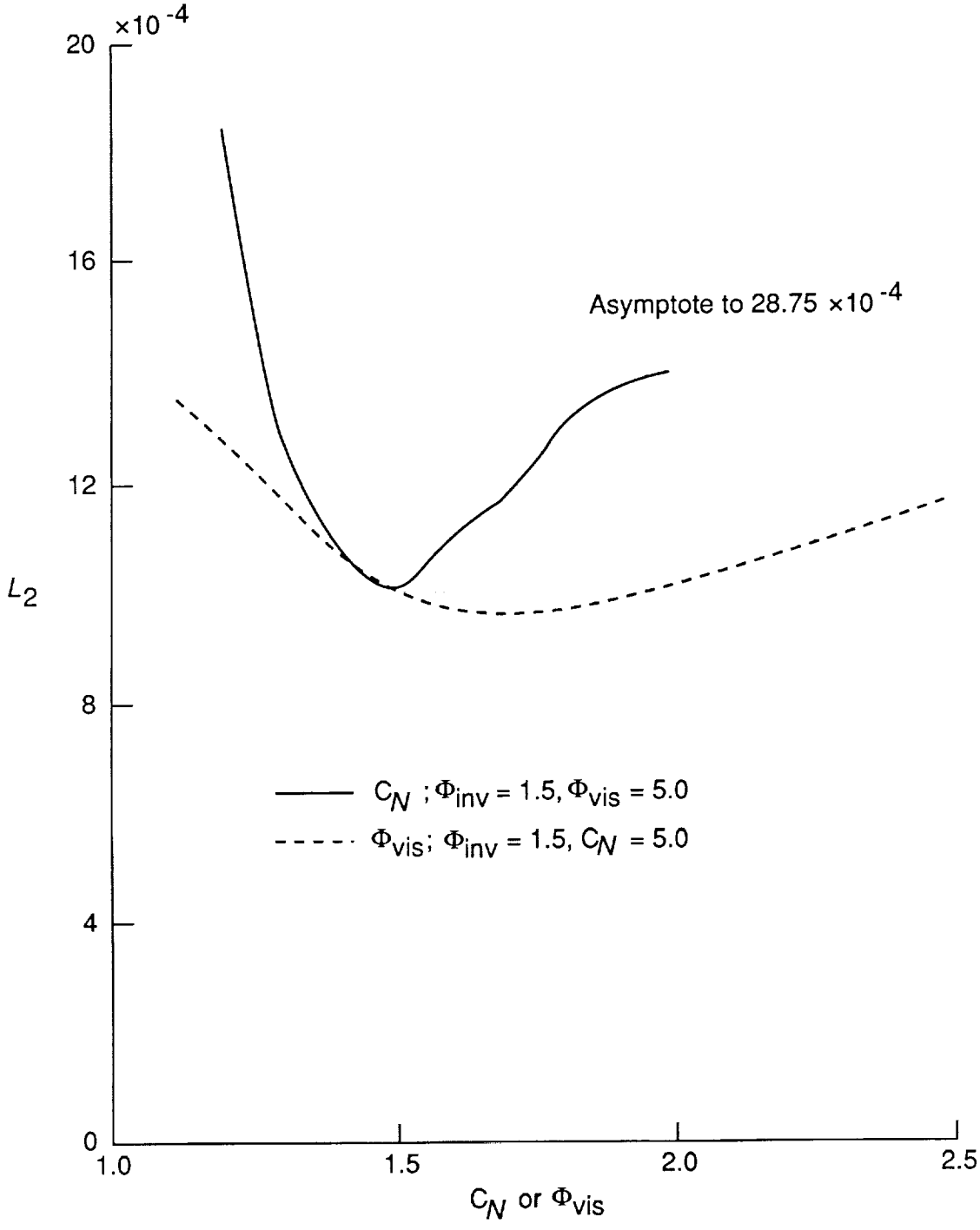


Figure 17. Effects of variations in Courant number and viscous relaxation factor on convergence levels after 700 iterations on 100 × 100 grid.

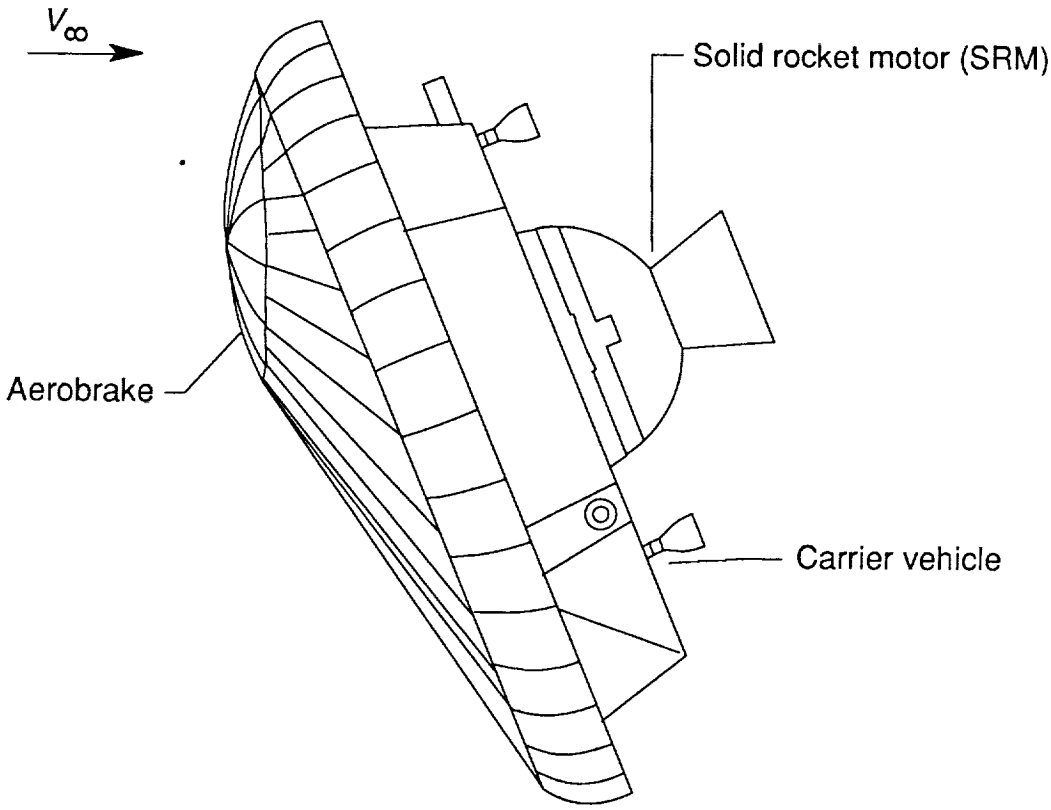


Figure 18. Aeroassist Flight Experiment (AFE).

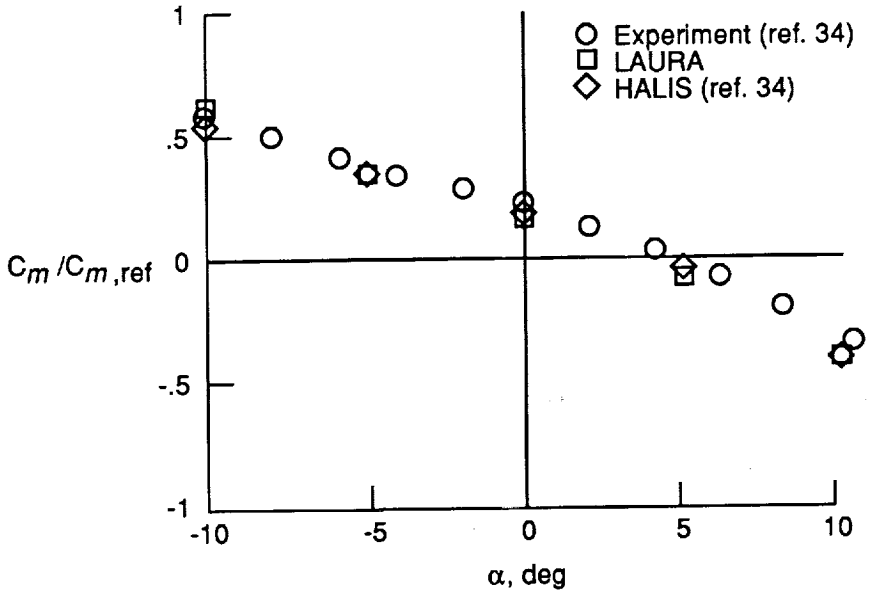
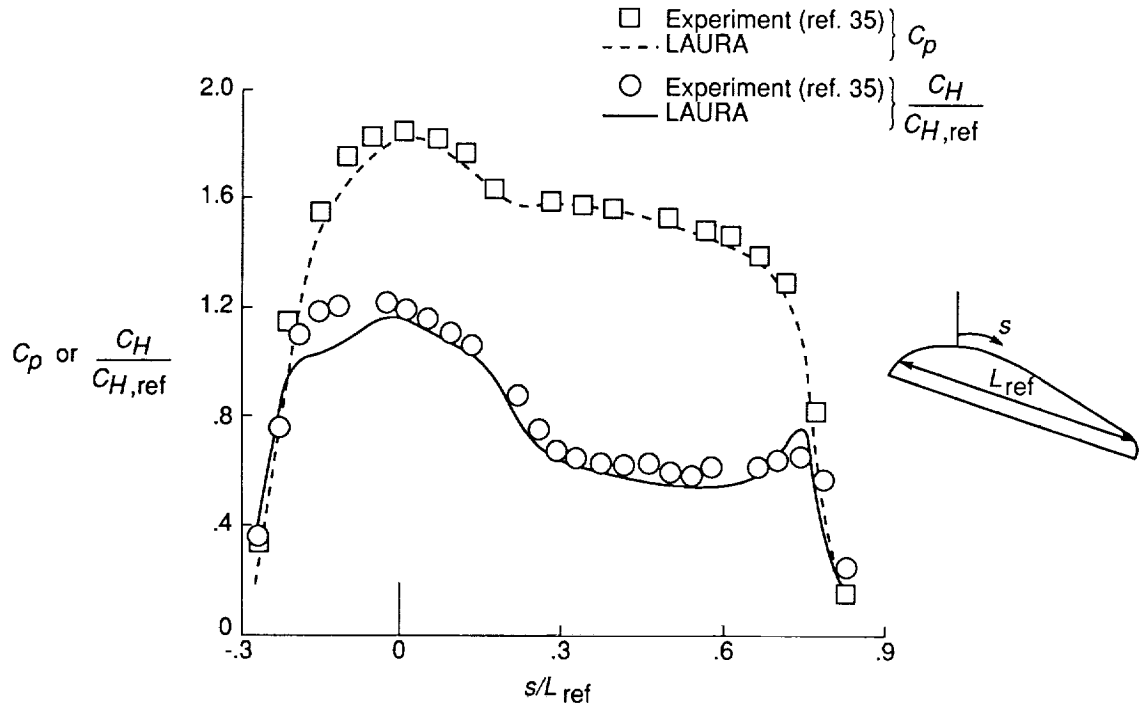
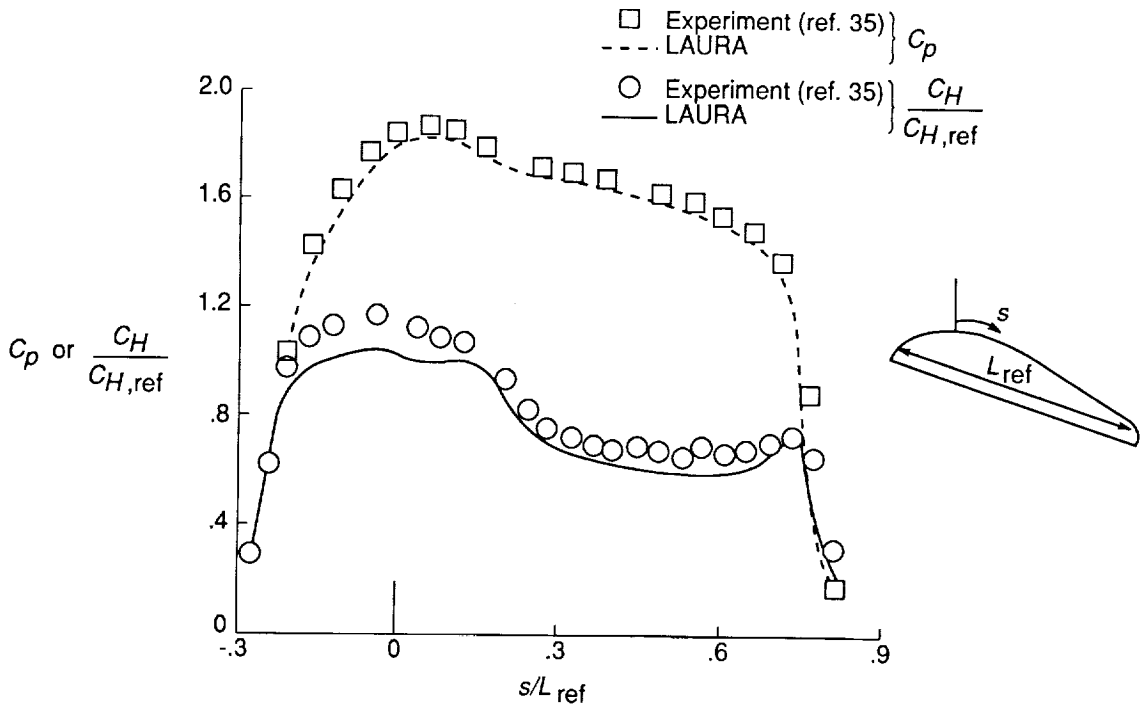


Figure 19. Experimental data and prediction for pitching moment of aerobrake at Mach 10.

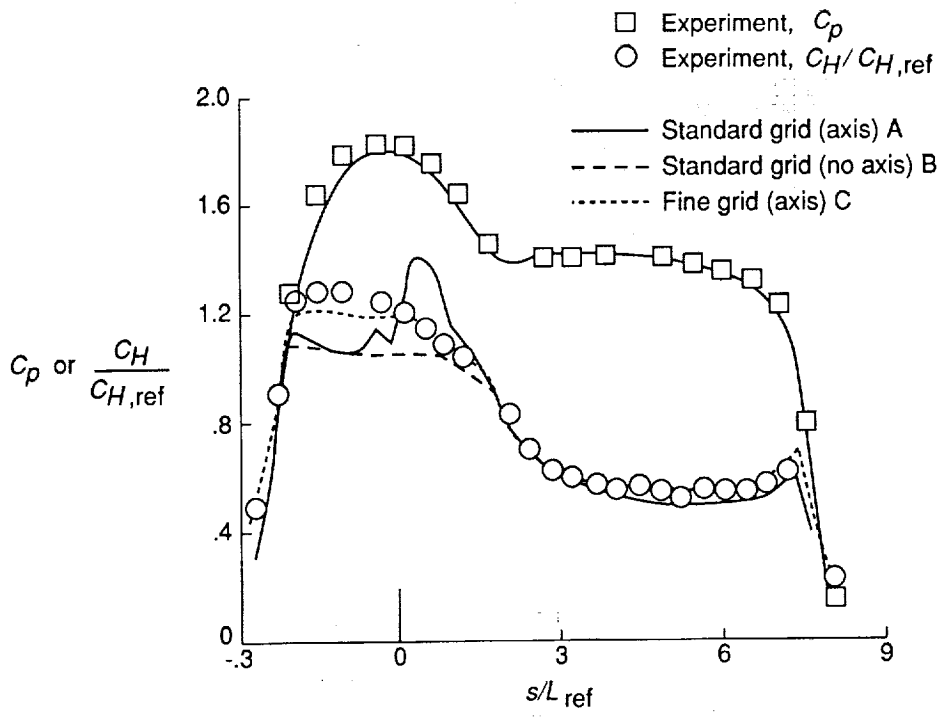


(a) $\alpha = 0°$.



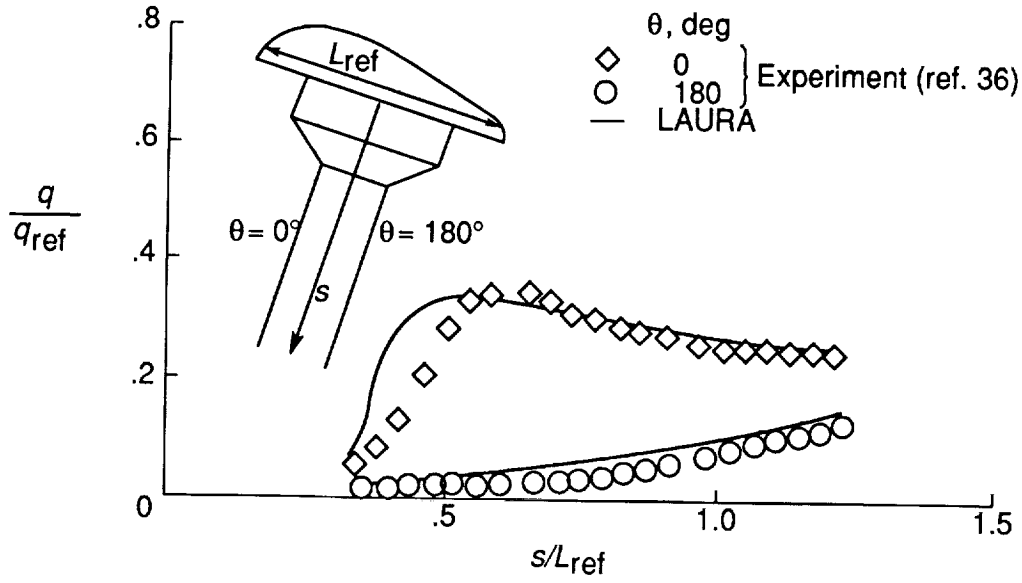
(b) $\alpha = 5°$.

Figure 20. Prediction and experiment for pressure and heating over AFE at Mach 10.

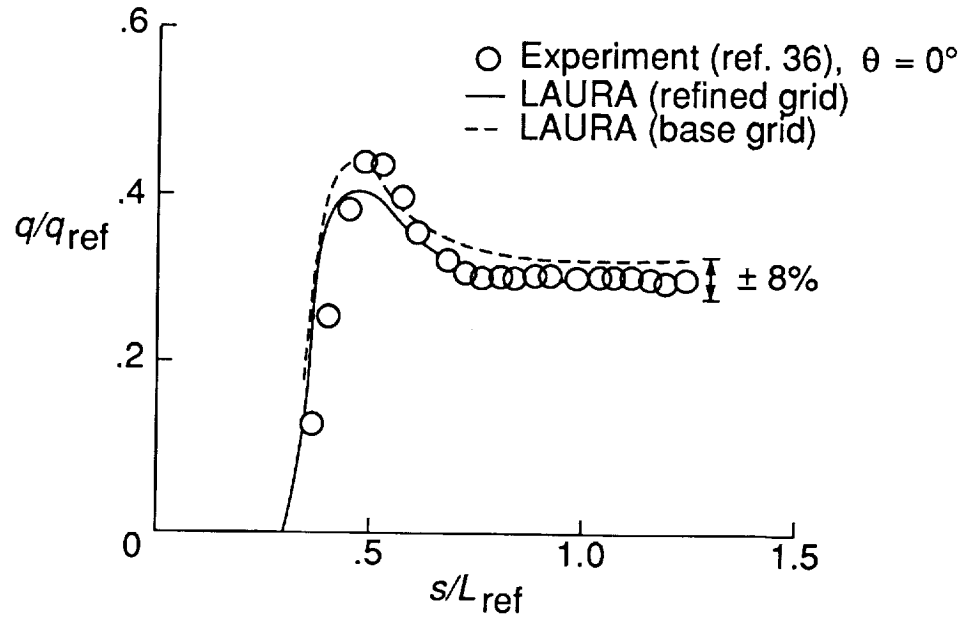


(c) $\alpha = -5°$; Cases A–C.

Figure 20. Concluded.



(a) $\alpha = 0°$.



(b) $\alpha = -5°$.

Figure 21. Prediction (laminar) and experimental data for heating on sting supporting AFE model in Mach 10 test case.

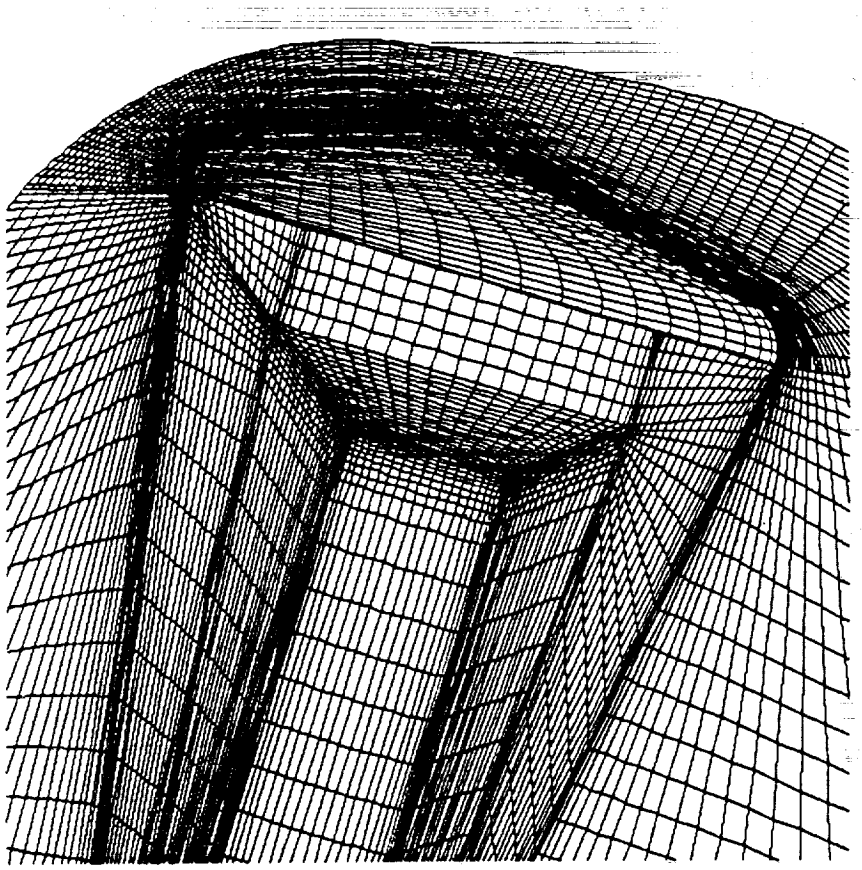
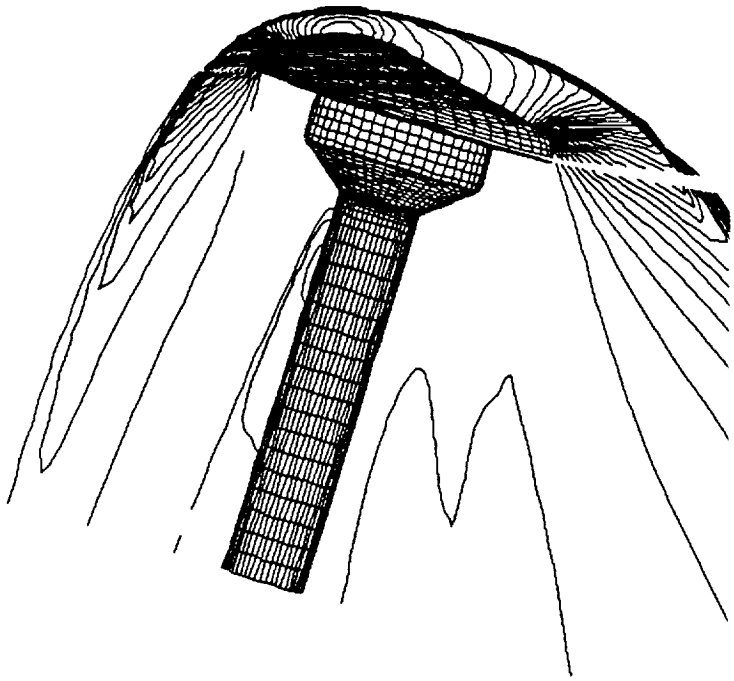
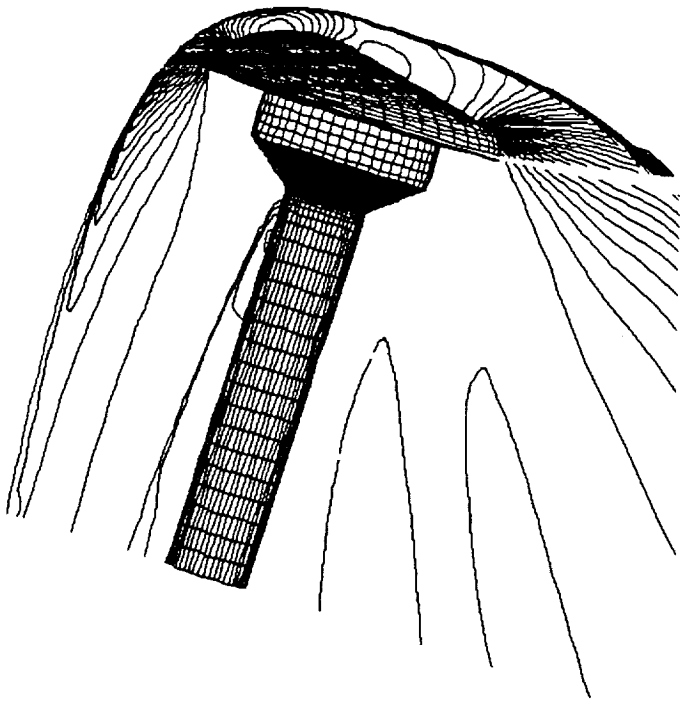


Figure 22. Four-domain grid defining surface and plane of symmetry for AFE test case. Domains are approximately divided into forebody, outer wake, shear layer behind shoulder, and inner wake core surrounding sting.

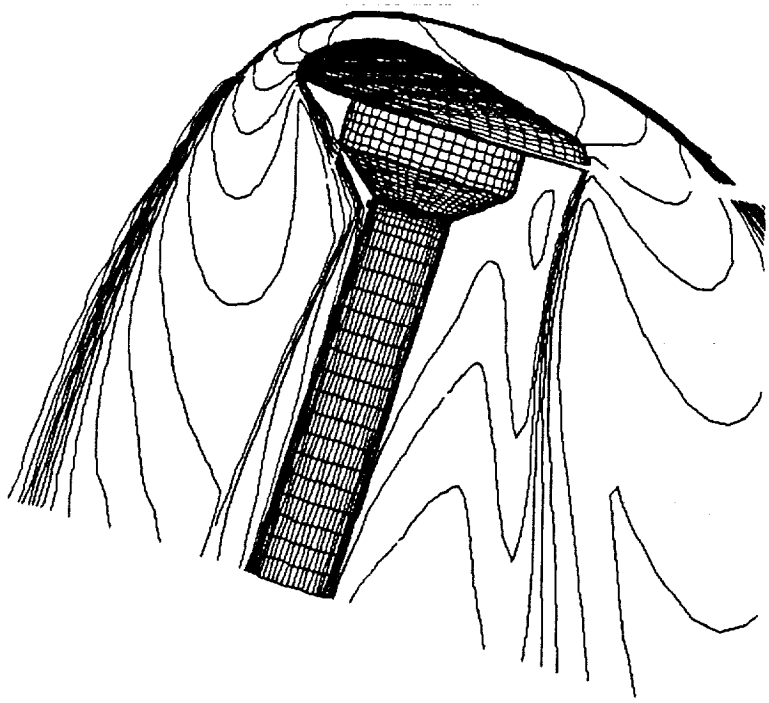


(a) $\alpha = 0°$.

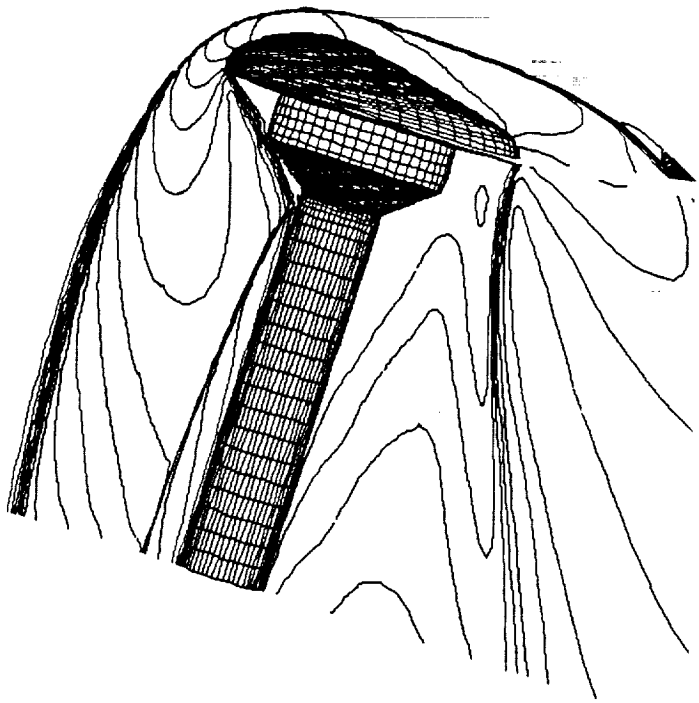


(b) $\alpha = -5°$.

Figure 23. Pressure contours in plane of symmetry about AFE at Mach 10.



(a) $\alpha = 0°$.



(b) $\alpha = -5°$.

Figure 24. Mach number contours in plane of symmetry about AFE at Mach 10.

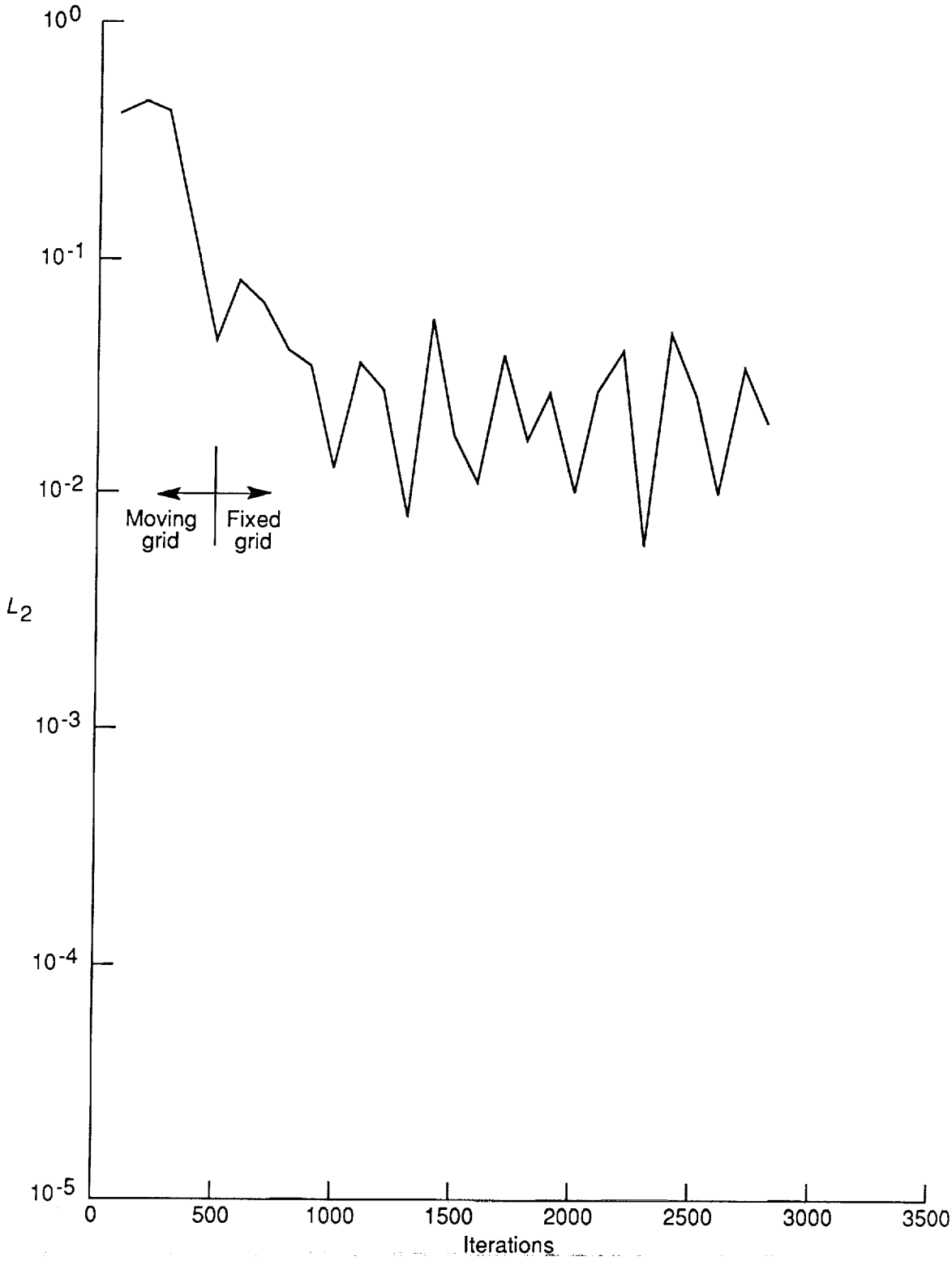


Figure 25. Convergence history for $\alpha = 5°$ and $\epsilon_o = 0.25$ on standard grid in domain 1 with $C_N = 0.5$.

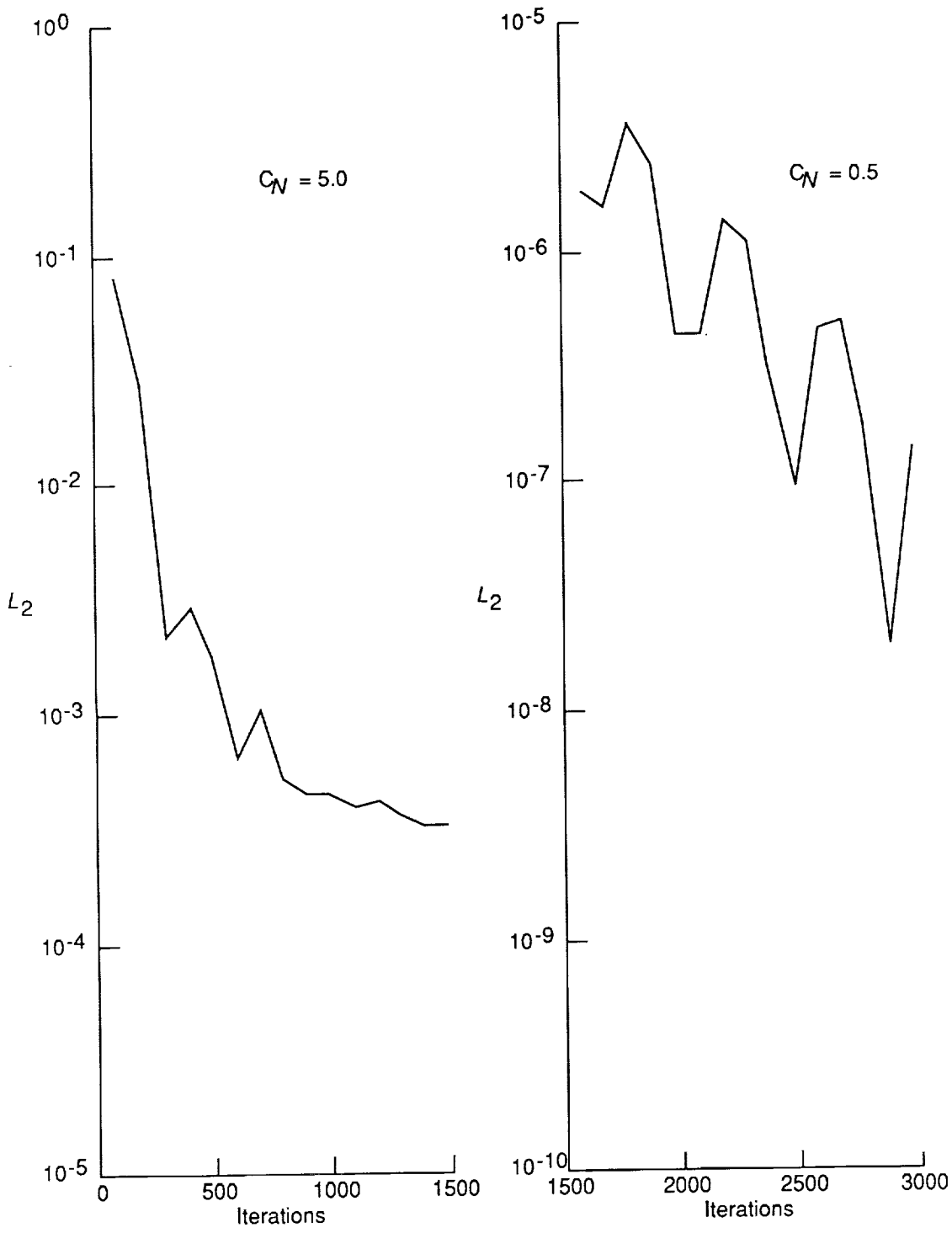


Figure 26. Convergence history for $\alpha = 5°$ and $\epsilon_o = 0.4$ on standard grid in domain 1 with $C_N = 5.0$ and 0.5.

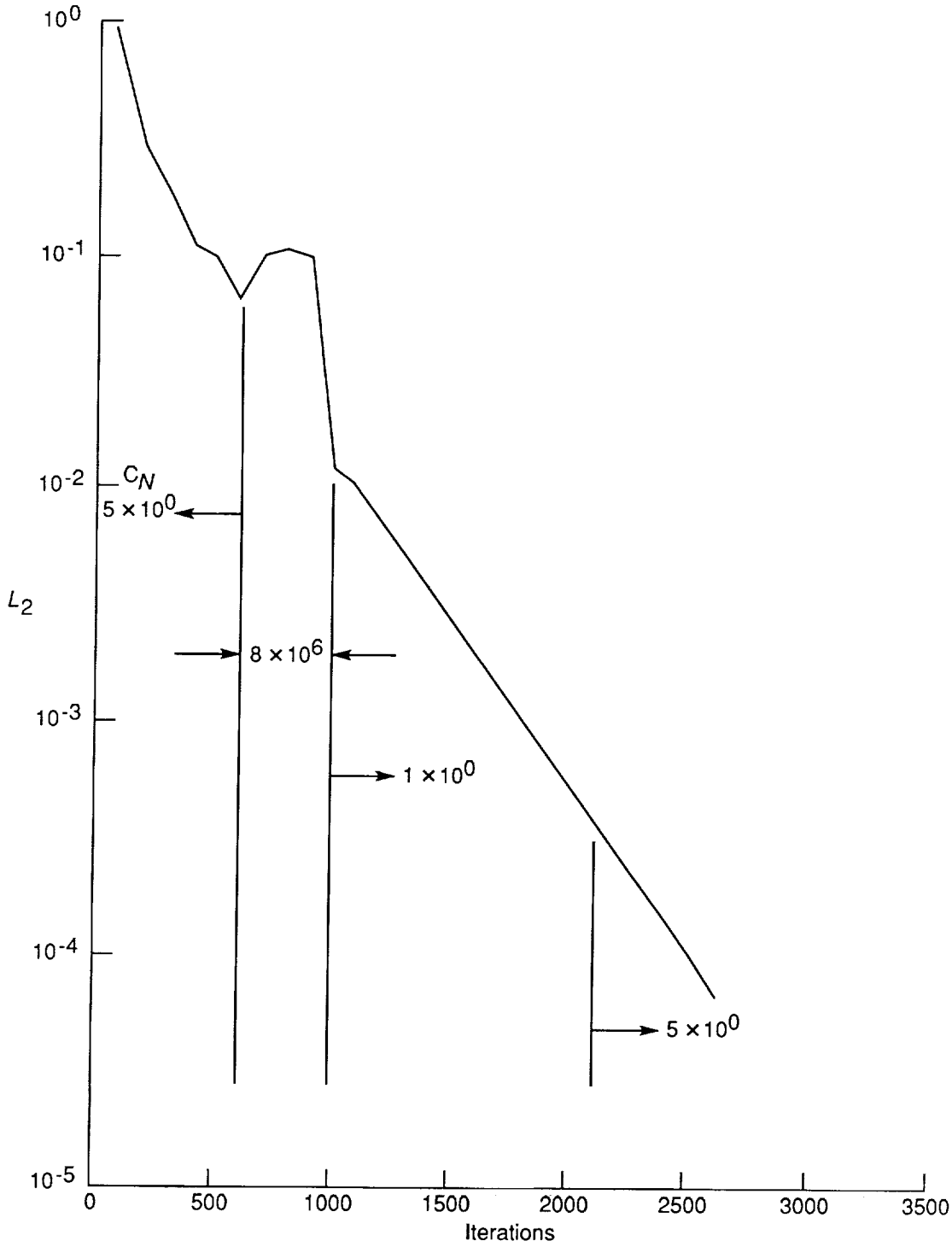


Figure 27. Convergence history for $\alpha = 0°$ and $\epsilon_o = 0.25$ on standard grid in domains 2, 3, and 4 with several Courant numbers.

Figure 28. Convergence history for $\alpha = -5°$ and $\epsilon_o = 0.2$ on fine grid in domain 1 with $C_N \geq 5.0$.

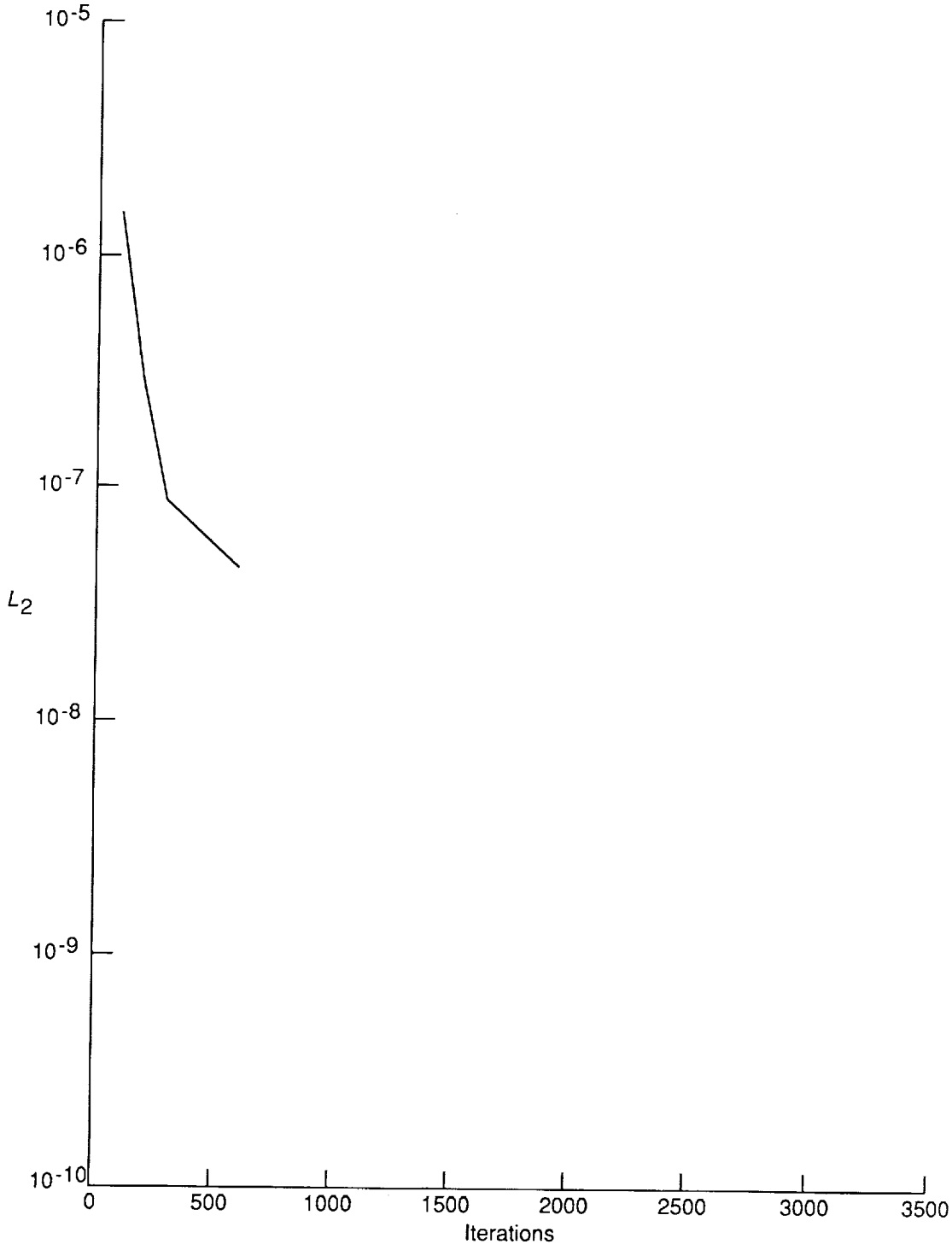


Figure 29. Convergence history for $\alpha = -5°$ and $\epsilon_o = 0.2$ on fine grid in domain 1 with $C_N = 0.5$.

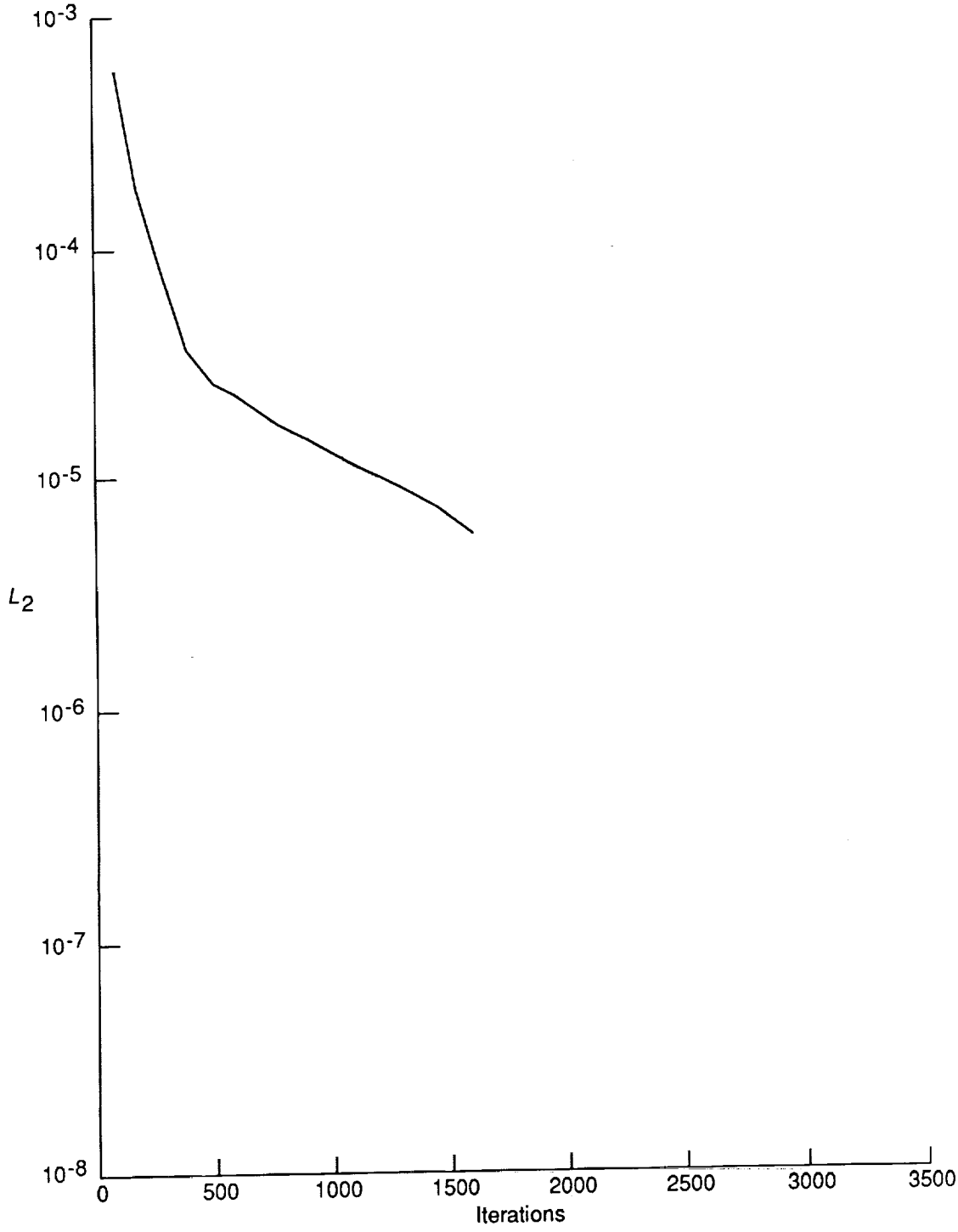


Figure 30. Convergence history for $\alpha = -5°$ and $\epsilon_o = 0.01$ on fine grid in domains 2, 3, and 4 with $C_N = 0.5$.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2953		

4. Title and Subtitle	5. Report Date
An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows	February 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Peter A. Gnoffo	L-16588
9. Performing Organization Name and Address	10. Work Unit No.
NASA Langley Research Center Hampton, VA 23665-5225	506-40-91-02
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, DC 20546-0001	Technical Paper
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An upwind-biased, point-implicit relaxation algorithm for obtaining the numerical solution to the governing equations for three-dimensional, viscous, compressible perfect-gas flows is described. The algorithm is derived by using a finite-volume formulation in which the inviscid components of flux across cell walls are described with Roe's averaging and Harten's entropy fix with second-order corrections based on Yee's symmetric total variation-diminishing scheme. Viscous terms are discretized by using central differences. The relaxation strategy is well suited for computers employing either vector or parallel architectures. It is also well suited to the numerical solution of the governing equations on unstructured grids. Because of the point-implicit relaxation strategy, the algorithm remains stable at large Courant numbers without the necessity of solving large, block tridiagonal systems. Convergence rates and grid refinement studies are conducted for Mach 5 flow through an inlet with a 10° compression ramp and Mach 14 flow over a 15° ramp. Predictions for pressure distributions, surface heating, and aerodynamic coefficients compare well with experimental data for Mach 10 flow over a blunt body.

17. Key Words (Suggested by Authors(s))
Hypersonic flows
Aeroassisted space transfer vehicle
Total variation diminishing

18. Distribution Statement
Unclassified—Unlimited

Subject Category 34

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	75	A04