NASA Technical Paper 3068

March 1991

An Upwind-Biased Space Marching Algorithm for Supersonic Viscous Flow

Francis A. Greene

NASA

NASA Technical Paper 3068

1991

An Upwind-Biased Space Marching Algorithm for Supersonic Viscous Flow

Francis A. Greene
Langley Research Center
Hampton, Virginia

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

Symbols

$\mathbf{A}$	inviscid flux Jacobian matrix
a, b	inviscid flux volume weighting parameters
C_N	Courant number
c	nondimensional sound speed
$\mathbf{D}$	flux difference
E	nondimensional total energy per unit volume
E'	dimensional total energy per unit volume, kg/m-s^2
$\mathbf{E}$	flux vector in ξ direction
e	nondimensional internal energy per unit mass
e'	dimensional internal energy per unit mass, m^2/s^2
$\mathbf{F}$	flux vector in η direction
$\mathbf{G}$	flux vector in ζ direction
H	nondimensional total enthalpy per unit mass
$\mathbf{H}$	flux vector normal to cell face
$\vec{\mathbf{h}}$	Cartesian flux tensor
i, j, k	indices of cell centers in ξ, η, ζ directions
L	normalizing reference length, m
$\vec{l}$	unit vector tangent to cell face
l_x, l_y, l_z	components of $\vec{l}$ in x, y, z directions
M	Mach number
M_ξ	streamwise Mach number
$\mathbf{M}$	right inviscid Jacobian eigenvector matrix
$\mathbf{M}^{-1}$	left inviscid Jacobian eigenvector matrix
$\vec{m}$	unit vector tangent to cell face
m_x, m_y, m_z	components of $\vec{m}$ in x, y, z directions
$\vec{n}$	unit vector normal to cell face
n_x, n_y, n_z	components of $\vec{n}$ in x, y, z directions
p	dummy index for i, j, or k
p	nondimensional pressure
p'	dinensional pressure, N/m^2
Pr	Prandtl number
q	heating rate
q'	dimensional heating rate, w/m^2
$\dot{q}_x$	nondimensional heating rate in x direction
$\dot{q}_y$	nondimensional heating rate in y direction

$\dot{q}_z$	nondimensional heating rate in z direction
$\mathbf{q}$	dependent variable vector
Re	Reynolds number
R_N	nose radius
S'	velocity, m/s
$\vec{S}$	nondimensional Cartesian velocity vector
T	nondimensional temperature
T'	dimensional temperature, K
t	time, s
U	nondimensional velocity in direction of $\vec{n}$
u	nondimensional Cartesian velocity component in x direction
u'	dimensional Cartesian velocity component in x direction, m/s
V	nondimensional velocity in direction of $\vec{l}$
v	nondimensional Cartesian velocity component in y direction
v'	dimensional Cartesian velocity component in y direction, m/s
W	nondimensional velocity in direction of $\vec{m}$
w	nondimensional Cartesian velocity component in w direction
w'	dimensional Cartesian velocity component in z direction, m/s
x, y, z	nondimensional Cartesian coordinates
x', y', z'	dimensional Cartesian coordinates, m
β	$= \gamma - 1$
γ	ratio of specific heats
Δ	spatial change in quantity, $(\ )_R - (\ )_L$
δ	time change in quantity, $(\ )^{n+1} - (\ )^n$
$\boldsymbol{\lambda}$	inviscid Jacobian eigenvalue matrix
μ	nondimensional viscosity
μ'	viscosity, kg/m-s
ν	$= -\frac{2}{3}\mu$
ξ, η, ζ	computational coordinates
ρ	nondimensional density
ρ'	dimensional density, kg/m^3
σ	nondimensional cell face area
τ	shear stress
Ω	nondimensional cell volume
Subscripts:	
I	inviscid quantity

L	quantity at cell center to left of cell face
R	quantity at cell center to right of cell face
V	viscous quantity
∞	free-stream value
Superscripts:	
$n, *$	time level
T	transpose
$+$	quantity associated with positive eigenvalue
$-$	quantity associated with negative eigenvalue

Summary

The Langley Aerothermodynamic Upwind Relaxation Algorithm, which globally iterates the thin-layer Navier-Stokes equations, is converted to an implicit space marching algorithm. The original code is used to generate the starting solution for its newly developed counterpart. Because the two algorithms differ only in the computation of the streamwise flux, an added degree of consistency between the starting and marching codes is achieved. The space marching algorithm is first- or second-order accurate with Roe's upwind differencing or symmetric total variation diminishing differencing, respectively. The upwind formulation of the governing equations inherently limits the pressure difference across a cell and avoids the need for a limiting variable placed directly within the governing equations. Each cross-flow plane is locally iterated in pseudo time until converged, then marched in space to the next station where the convergence process is repeated. This procedure gives a complete solution in a single sweep over the geometry. The algorithm is tested on a sphere-cone geometry at 0 incidence and at angle of attack on a geometry which models the windward surface of the Space Shuttle orbiter. Center-line surface pressure comparisons for the sphere-cone are made between the global and space marching algorithms. In the overexpansion recompression region of the sphere-cone, the space marching algorithm was very sensitive to the step size. The effects of streamwise grid refinement on solution quality in the overexpansion recompression region of the sphere-cone are discussed. In addition, computational results for surface heating are compared with ground-based experimental data for both the sphere-cone and the shuttle-like geometries.

1. Introduction

Upwind methods are well suited for the numerical simulation of hypersonic flows, since their robust nature enables them to capture strong bow shocks without oscillations. Initially, upwind techniques such as the flux-difference split methods developed by Godunov (ref. 1), Osher (ref. 2), and Roe (ref. 3), and the flux-vector split methods attributed to Steger and Warming (ref. 4) and Van Leer (ref. 5) were developed for the Euler equations. These methods have since been used with the Navier-Stokes equations (refs. 6 to 9). More recently, the flux-difference split method of Roe has been used with space marching techniques to combine the efficiency of space marching with the shock capturing capability of upwind methods in a single code (refs. 10 and 11).

Prior to upwind differenced algorithms, space marching codes that solved the parabolized Navier-Stokes (PNS) equations used solution algorithms based on central difference methods (refs. 12 and 13). Unlike solution algorithms based on upwind differencing, centrally differenced algorithms can introduce spurious oscillations in values across shock waves. Because of these oscillations, centrally differenced algorithms usually have dissipation added into the solution procedure to maintain stability. Upwind methods are naturally dissipative and as such require no additional dissipation.

Even though flow conditions are more restrictive for space marching algorithms compared with globally iterated algorithms, the space marching aspect brings a higher degree of efficiency when compared with global algorithms. Efficiency in terms of computer time is enhanced with space marching techniques. This results in quicker turnaround times and enables space marching algorithms to be used on a wider range of computers.

The thin-layer Navier-Stokes (TLNS) equations are a viable and practical alternative to the full Navier-Stokes equations when streamwise viscous effects are negligible. With the same viscous terms as the PNS equations, the TLNS equations retain the unsteady terms. Even though the TLNS equations retain time-dependent terms, they can be spatially marched in the spirit of the PNS equations as long as the appropriate provisions are made at the outflow boundary. The globally iterated thin-layer code provides the initial data needed for the marching algorithm. Data at the station where the solution is to be obtained are locally iterated in pseudo time until they are converged. Once converged, the solution is marched to the next station where the local iteration process is repeated. Unlike a globally iterated solution, once a station has been converged one time it is never recomputed. This solution procedure has been demonstrated with flux-vector splitting for inviscid flow by Walters and

Dwoyer (ref. 14) and for viscous flow by Newsome, Walters, and Thomas (ref. 15). In both cases, a globally iterated algorithm was modified to space march. Walters and Dwoyer handle the subsonic region with a hybrid procedure. Global iterations are first performed identifying the subsonic regions, and the solution is then globally iterated in the subsonic regions and space marched in the supersonic regions. Newsome et al. use the approach of Vigneron, Rakich, and Tannehill (ref. 16) to avoid departure solutions.

The Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA) (ref. 6) has been converted to a space marching algorithm. This report details the modifications made to the original algorithm and presents comparisons with experimental data. In converting the algorithm to perform space marching, the original formulation was not disturbed so that in the future a single formulation could be used to calculate globally iterated solutions, space-marched solutions, or a hybird of both.

To accurately describe the physics of the flow, the flux (viscous and inviscid) at a cell face must be formulated to represent the proper zone of dependence. Second-order accurate central difference approximations to the dissipative terms in the viscous flux vector can be used to adequately describe the viscous zone of dependence but cannot be used to approximate the terms in the inviscid flux vector such that the inviscid zone of dependence is properly reflected. To reflect the inviscid zone of dependence, the motion of waves relative to a cell face must be properly accounted for. The propagation of information within the inviscid zone of dependence, in terms of speed and direction, is reflected in the eigenvalues of the unsteady inviscid flux vector Jacobian. The unsteady Jacobian is considered since the equations to be solved will be iterated in time. The space marching algorithm employs the flux-difference split method of Roe to define an upwind-biased inviscid flux at cell face in the streamwise as well as the cross-flow directions. Inherent in an upwind-biased flux is the knowledge that it contains the amount of upstream and downstream information needed to correctly describe the inviscid zone of dependence. As a consequence, the flow physics is correctly modeled in one dimension and approximately modeled in two and three dimensions. Supersonic inviscid fluxes with eigenvalues which are all positive are comprised solely of upstream information, whereas fluxes with mixed-sign eigenvalues contain both upstream and downstream information.

The streamwise pressure gradient propagates information upstream through the subsonic portion of the boundary layer. If the elliptic nature of this upstream movement of information, which is associated with the negative eigenvalue, is not properly treated, departure solutions will result (ref. 17). To avoid departure solutions, the governing equations can be globally iterated where the sweep direction is alternated. This approach is typically used when time-dependent terms are present but can also be used on equation sets without time-dependent terms such as the PNS equations (refs. 18 to 20). By performing multiple relaxation sweeps over the body, information is allowed to propagate through the subsonic portion of the boundary layer without causing the solution to diverge. A consequence of this method is the increased computational time to perform multiple sweeps compared with a single sweep method.

To prevent single sweep space marching methods from being ill-posed, the aforementioned elliptic effects must be suppressed. The suppression requires that the influence of the negative eigenvalue be limited, this is usually accomplished through a modification of the streamwise pressure gradient. To use single sweep space marching methods, the streamwise pressure gradient must be favorable (negative) or not so adverse as to cause reversed flow, and the streamwise flow should be supersonic everywhere except in the boundary layer. Through an eigenvalue analysis of the PNS equations, Vigneron et al. (ref. 16) determined that a fraction of the streamwise pressure gradient could be directly included in the PNS equations and the remaining portion would either have to be neglected or explicitly imposed. The fraction included directly in the PNS equations represents the amount of the pressure gradient associated with the positive eigenvalues, whereas the remaining portion is associated with the negative eigenvalues. Another approach incorporated within single sweep space marching PNS solvers to handle departure solutions is attributed to Schiff and Steger (ref. 21). Their method imposes the pressure gradient from the first supersonic point outside the boundary layer throughout

the subsonic boundary layer. Lubard and Helliwell (ref. 22) tried lagging the gradient and calculating it explicitly. This worked with moderate success.

In the present report, the full pressure in the momentum flux is retained. When the upwind-biased flux is computed, the pressure in the momentum flux consists of upstream and downstream information. This flux, when used in conjunction with a proper specification for a supersonic outflow boundary condition, results in the streamwise pressure difference across a cell being a fraction of the total change in pressure across the cell. This approach is similar to Vigneron's; however it does not require an explicitly defined limit on the streamwise pressure gradient in the governing equations. The limit on the pressure gradient used in Vigneron's and in this approach is different. The differences and similarities are described in this report.

The space marching algorithm is applied on a 15° sphere-cone geometry at zero incidence and at angle of attack on a geometry which models the windward surface of the Space Shuttle orbiter (ref. 23). This geometry is referred to as the HALIS shuttle geometry. The sphere-cone solutions for surface heating are compared with ground-based experimental data, whereas surface pressures are compared with pressures from the globally iterated algorithm. Streamwise grid refinement in the overexpansion recompression region of the cone is performed to determine the space marching algorithm's dependence on step size in regions of adverse pressure gradients. To provide a rigorous test for the new algorithm, space marching solutions on the geometry which models the windward surface of the Space Shuttle were computed. Computed surface heating distributions at an angle of attack of 10° are compared with ground-based experimental data.

2. Algorithm Development

The thin-layer Navier-Stokes equations for a compressible, perfect gas are employed in the present work. The thin-layer equations are derived from the full Navier-Stokes equations by neglecting derivatives in the streamwise and circumferential directions found in the viscous terms. This equation set provides adequate solutions for flows in which viscous gradients in the streamwise and circumferential directions are small compared with viscous gradients in the normal direction. Under these conditions, expending computer resources to resolve viscous effects parallel to the body is an inefficient use of computer memory and time, since their effect on the converged solution is insignificant.

The governing equations upon which the space marching algorithm was based were taken from reference 6. The program mentioned in reference 6 is the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA). The inviscid fluxes are either first-order accurate with an upwind technique attributed to Roe (ref. 3) or second-order accurate with the symmetric total variation diminishing (STVD) scheme of Yee (ref. 24). Possible entropy violations are alleviated by a procedure of imposing a minimum on the absolute value of the eigenvalues of the inviscid Jacobian matrix. This fix was suggested by Harten (ref. 25). The terms in the viscous flux vector are approximated with second-order central differences.

The integral form of the governing equations can be expressed as

$$\int\int\int \mathbf{q}_t \, d\Omega + \int\int \vec{\mathbf{h}} \cdot \vec{n} \, d\sigma = 0 \tag{2.1}$$

where $\mathbf{q}_t$ is the time rate of change of the dependent variable vector at a cell center, $\vec{\mathbf{h}}$ is the Cartesian flux tensor, and $\vec{n}$ is the outward unit normal vector to a cell face. The cell volume and cell face area are represented by Ω and σ, respectively. The quantity $\vec{\mathbf{h}} \cdot \vec{n}$ is the flux vector normal to a cell face. The tensor $\vec{\mathbf{h}}$ can be written in terms of its components in the x, y, and z directions as

$$\vec{\mathbf{h}} = (\mathbf{E}_I - \mathbf{E}_V)_x + (\mathbf{F}_I - \mathbf{F}_V)_y + (\mathbf{G}_I - \mathbf{G}_V)_z \tag{2.2}$$

where

$$\mathbf{q} = [\rho, \rho u, \rho v, \rho w, E]^T \tag{2.3a}$$

$$\mathbf{E}_I = [\rho u, \rho u^2 + p, \rho uv, \rho uw, (E+p)u]^T \tag{2.3b}$$

$$\mathbf{F}_I = [\rho v, \rho uv, \rho v^2 + p, \rho vw, (E+p)v]^T \tag{2.3c}$$

$$\mathbf{G}_I = [\rho w, \rho uw, \rho vw, \rho w^2 + p, (E+p)w]^T \tag{2.3d}$$

$$\mathbf{E}_v = [0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \dot{q}_x]^T \tag{2.3e}$$

$$\mathbf{F}_v = [0, \tau_{xy}, \tau_{yy}, \tau_{yz}, u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - \dot{q}_y]^T \tag{2.3f}$$

$$\mathbf{G}_v = [0, \tau_{xz}, \tau_{yz}, \tau_{zz}, u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - \dot{q}_z]^T \tag{2.3g}$$

All quantities have been nondimensionalized as shown in the following equations:

$$\rho = \frac{\rho'}{\rho'_\infty} \qquad u = \frac{u'}{S'_\infty} \qquad v = \frac{v'}{S'_\infty} \qquad w = \frac{w'}{S'_\infty}$$

$$p = \frac{p'}{\rho'_\infty (S'_\infty)^2} \qquad e = \frac{e'}{(S'_\infty)^2} \qquad T = \frac{T'}{T'_\infty} \qquad E = \frac{E'}{\rho'_\infty (S'_\infty)^2}$$

$$x = \frac{x'}{L'} \qquad y = \frac{y'}{L'} \qquad z = \frac{z'}{L'} \qquad \mu = \frac{\mu'}{\mu'_\infty \mathrm{Re}_\infty}$$

The value L is a user-defined reference length. The equation set is closed by using the ideal gas law.

Expressing equation (2.1) in finite volume form for a single six-sided cell in the domain gives

$$\delta \mathbf{q}_{i,j,k} = \left(\frac{\delta t}{\Omega}\right)_{i,j,k} \left(\mathbf{H}^*_{i-1/2,j,k}\, \sigma_{i-1/2,j,k} - \mathbf{H}^*_{i+1/2,j,k}\, \sigma_{i+1/2,j,k} \right.$$

$$\left. + \mathbf{H}^*_{i,j-1/2,k}\, \sigma_{i,j-1/2,k} - \mathbf{H}^*_{i,j+1/2,k}\, \sigma_{i,j+1/2,k} + \mathbf{H}^*_{i,j,k-1/2}\, \sigma_{i,j,k-1/2} - \mathbf{H}^*_{i,j,k+1/2}\, \sigma_{i,j,k+1/2} \right) \tag{2.4a}$$

A shorthand index notation that is used enables equation (2.4a) to be expressed as

$$\delta \mathbf{q}_{i,j,k} = \left(\frac{\delta t}{\Omega}\right)_{i,j,k} \sum_{p=i,j,k} \left[(\mathbf{H}^* \sigma)_{p-1/2} - (\mathbf{H}^* \sigma)_{p+1/2} \right] \tag{2.4b}$$

where $\delta \mathbf{q} = (\mathbf{q}^{n+1} - \mathbf{q}^n)$ is the change in $\mathbf{q}$ per time step, $\mathbf{H}$ is the flux vector normal to a cell face, δt is the time step, and n is the time level. The lower case subscripts in equations (2.4) represent cell-centered values, unless offset by one half, then they represent values at the center of a cell face. A diagram of the indexing is shown in figure 1. The equations used to compute the cell volume, the cell face area, the time step, and the metrics are found in reference 6. The asterisk * represents the time level and may be at either the n or $n+1$ level, depending on the cell face being evaluated (ref. 6). Since $\mathbf{H}$ is comprised of inviscid and viscous contributions, it can be expressed as

$$\mathbf{H}^* = \mathbf{H}^*_I(\mathbf{q}_L, \mathbf{q}_R) + \mathbf{H}^*_V(\mathbf{q}_L, \mathbf{q}_R)$$

The first-order inviscid flux is computed with Roe's flux-difference splitting to derive its upwind nature. The flux difference at a cell face comes from the solution of a Riemann initial value problem posed by Roe. The initial data for the problem are found at the two cell centers to which the face is common. The flux difference is split into two parts, one associated with the positive eigenvalues of the unsteady inviscid Jacobian matrix and the other with the negative eigenvalues. The speed and direction of information propagating toward a cell face are directly proportional to the absolute magnitude of the eigenvalues and sign of the eigenvalues, respectively. This is to say negative eigenvalues send information from the right toward a cell face, whereas positive ones send information from the left, and how quickly this information travels is related to the eigenvalue's absolute magnitude. Roe's method correctly interprets

this wave motion relative to a cell face and computes a flux which correctly models the physics of the flow. The flux for a supersonic flow with all positive eigenvalues is solely comprised of information from the left, and a supersonic flow with all negative eigenvalues is comprised solely of information from the right. If the flow has mixed eigenvalues, the flux is comprised of information from the right and left.

In general, an inviscid flux at a cell face can be written as

$$\mathbf{H}_I^*(\mathbf{q}_L, \mathbf{q}_R) = \frac{1}{2}\left[a\mathbf{H}_I^*(\mathbf{q}_L) + b\mathbf{H}_I^*(\mathbf{q}_R)\right] - \frac{1}{2}\mathbf{D}^*(\mathbf{q}_L, \mathbf{q}_R) \tag{2.5}$$

where L and R denote the indices of the cell centers to the left and right, respectively, of the face being evaluated. The functions, a and b, are flux weighting parameters defined in terms of cell volumes as

$$a = \frac{2\Omega_R}{\Omega_R + \Omega_L} \qquad b = \frac{2\Omega_L}{\Omega_R + \Omega_L}$$

The parameters, a and b, lessen the effects of grid stretching, and their formulation is empirical. The variable $\mathbf{D}^*$, shown in equation (2.5), is the previously mentioned flux difference. It can be written as

$$\mathbf{D}^*(\mathbf{q}_L, \mathbf{q}_R) = \mathbf{M}\,|\boldsymbol{\lambda}|\,\mathbf{M}^{-1}\,\Delta\mathbf{q}^* = |\mathbf{A}|\,\Delta\mathbf{q}^* \tag{2.6}$$

where $\mathbf{A}$ is the flux vector Jacobian. In equation (2.6) the Jacobian is also shown in terms of its right ($\mathbf{M}$), and left ($\mathbf{M}^{-1}$) eigenvectors, and a diagonal matrix of its eigenvalues ($\boldsymbol{\lambda}$). The matrices $\mathbf{A}$, $\boldsymbol{\lambda}$, $\mathbf{M}$, and $\mathbf{M}^{-1}$ are given in appendix A.

With equation (2.6), the first-order inviscid flux at a cell face is

$$\mathbf{H}_I^*(\mathbf{q}_L, \mathbf{q}_R) = \frac{1}{2}\left[a\mathbf{H}_I^*(\mathbf{q}_L) + b\mathbf{H}_I^*(\mathbf{q}_R) - |\mathbf{A}|\,(\mathbf{q}_R, \mathbf{q}_L)\,\Delta\mathbf{q}^*\right] \tag{2.7}$$

The inviscid flux shown in equation (2.7) can be thought of as being composed of a second-order approximation to the flux at a cell face (first two terms) minus a dissipation term (remaining term). If this dissipation is not included, the algorithm is equivalent to a centrally differenced algorithm.

For second-order accurate solutions with Yee's STVD approach, $\mathbf{D}^*$ in equation (2.5) is as follows:

$$\mathbf{D}^*(\mathbf{q}_L, \mathbf{q}_R) = \mathbf{M}_2|\boldsymbol{\lambda}|_2\left[\mathbf{M}_2^{-1}\,\Delta\mathbf{q}_2^* - \min\mathrm{mod}(\mathbf{M}_1^{-1}\,\Delta\mathbf{q}_1, \mathbf{M}_2^{-1}\Delta\mathbf{q}_2, \mathbf{M}_3^{-1}\,\Delta\mathbf{q}_3)\right] \tag{2.8}$$

The subscript 2 references the face at which the flux is being computed; 1, the face behind; and 3, the face ahead. The min mod function compares differences in characteristic variables at these locations and chooses the smallest in absolute magnitude if the signs of the values are the same or zero if the signs are different. The first term in equation (2.8) is the first-order term; the second is a correction which makes $\mathbf{D}^*$ second-order accurate if the signs of the differences in characteristic variables are the same.

From reference 6, the viscous stress in a direction normal to a cell wall can be expressed in terms of its x, y, and z components as

$$\begin{aligned}\tau_{\mathrm{nd}} =& \frac{\nu}{\mathrm{Re}}\left[(\vec{u}_\xi \cdot \vec{\nabla}\xi) + (\vec{u}_\eta \cdot \vec{\nabla}\eta) + (\vec{u}_\zeta \cdot \vec{\nabla}\zeta)\right] n_d \\ &+ \frac{\mu}{\mathrm{Re}}\left[U_\xi \xi_d + U_\eta \eta_d + U_\zeta \zeta_d + \dot{d}_\xi\left(\vec{\nabla}\xi \cdot \vec{n}\right) + \dot{d}_\eta\left(\vec{\nabla}\eta \cdot \vec{n}\right) + \dot{d}_\zeta(\vec{\nabla}\zeta \cdot \vec{n})\right]\end{aligned} \tag{2.9}$$

where $d = (x, y, z)$, and $\dot{d}$ is the Cartesian velocity component ($\dot{d} = u$, v, or w) corresponding to d. If the thin-layer assumption and Stokes hypothesis are invoked, the thin-layer viscous stresses on a cell wall with a unit normal $\vec{n}$ become

$$\tau_{\mathrm{nd}} = \frac{\mu}{\mathrm{Re}}\left(\dot{d}_\chi + \frac{1}{3}U_\chi n_d\right)(\vec{\nabla}\chi \cdot \vec{n}) \tag{2.10}$$

where χ represents the computational coordinates (ξ, η, or ζ). For the thin-layer approximation, χ is evaluated in the body normal (ζ) direction only. Including the remaining directions (ξ and η) is equivalent to neglecting only the cross derivative terms from the stress tensor.

From reference 6, the heat flux through a cell wall can be expressed as

$$\dot{q} = \frac{\mu\gamma}{\mathrm{Pr}}\left[e_\xi(\vec{\nabla}\xi \cdot \vec{n}) + e_\eta(\vec{\nabla}\eta \cdot \vec{n}) + e_\zeta(\vec{\nabla}\zeta \cdot \vec{n})\right] \tag{2.11}$$

Again, if the thin-layer assumption is made for the heat flux, equation (2.11) becomes

$$\dot{q} = \frac{\mu\gamma}{\mathrm{Pr}}\left[e_\chi(\vec{\nabla}\chi \cdot \vec{n})\right] \tag{2.12}$$

where the treatment of χ is identical to that of the stress terms. The viscous portion of $\mathbf{H}$ is shown as follows:

$$\mathbf{H}_V^*(\mathbf{q}_L, \mathbf{q}_R) = -\left[0, \tau_{\mathrm{nx}}, \tau_{\mathrm{ny}}, \tau_{\mathrm{nz}}, \vec{\tau}_n \cdot \vec{S} + \dot{q}\right]^T \tag{2.13}$$

If variables with the superscript $*$ contain information referenced to the i, j, k cell center, they are linearized by the following equation:

$$\mathbf{K}^* = \mathbf{K}^n + \left(\frac{\partial \mathbf{K}^n}{\partial \mathbf{q}}\right)_{i,j,k} \delta\mathbf{q}_{i,j,k}$$

The variable $\mathbf{K}$ is a dummy variable which represents the value being linearized.

Upon substituting equations (2.5) and (2.13) into equation (2.4b) and performing the requisite linearizations as described in reference 6, the governing equations can be written as

$$\left\{I + \left(\frac{\delta t}{2\Omega}\right)_{i,j,k} \sum_{p=i,j,k}\left[\left(|\mathbf{A}| + 2\frac{\partial \mathbf{H}_V^n}{\partial \mathbf{q}}\right)_{p+1/2}\sigma_{p+1/2} + \left(|\mathbf{A}| - 2\frac{\partial \mathbf{H}_V^n}{\partial \mathbf{q}}\right)_{p-1/2}\sigma_{p-1/2}\right]\right\}\delta\mathbf{q}_{i,j,k}$$

$$= \left(\frac{\delta t}{2\Omega}\right)_{i,j,k} \sum_{p=i,j,k}\Big(\{[a\mathbf{H}_I^n(\mathbf{q}_L) + b\mathbf{H}_I^n(\mathbf{q}_R) - \mathbf{D}^n(\mathbf{q}_L, \mathbf{q}_R)] + 2\mathbf{H}_V^n(\mathbf{q}_L, \mathbf{q}_R)\}_{p-1/2}\sigma_{p-1/2}$$

$$- \{[a\mathbf{H}_I^n(\mathbf{q}_L) + b\mathbf{H}_I^n(\mathbf{q}_R) - \mathbf{D}^n(\mathbf{q}_L, \mathbf{q}_R)] + 2\mathbf{H}_V^n(\mathbf{q}_L, \mathbf{q}_R)\}_{p+1/2}\sigma_{p+1/2}\Big) \tag{2.14}$$

where the inviscid and viscous Jacobians are given in appendix A.

2.1. Boundary and Initial Conditions

Pseudo (imaginary) cells are used to specify all boundary conditions. The location of these cell centers is 1/2 cell beyond the boundary. The volume of each pseudo cell is set equal to the volume directly across the boundary being considered.

At the body pseudo cell centers, Cartesian velocity components are specified with no slip conditions, the wall temperature is given, and a normal pressure gradient equal to zero is employed. Because the boundary layer is resolved to have a cell Reynolds number (eq. (4.1)) of order 1, the distance of 1/2 cell is very small, and as a result, the specification of the wall

boundary conditions at these locations appears to have a negligible effect on the solution (ref. 6). Values for the free-stream pseudo cells are taken to be the free-stream quantities. Reflective boundary conditions are used in the upper and lower symmetry plane pseudo cells and in the axis-singularity pseudo cells. Pseudo cell values at the supersonic outflow boundary are set equal to the computed values across the boundary. For this specification of pseudo cell values to be accurate, the streamwise flow at the outflow boundary must be supersonic. Although flow in the boundary layer is not totally supersonic, the use of a supersonic outflow boundary condition does not appear to adversely affect the solution in this region. Additional information on the boundary conditions for the globally relaxed algorithm is in reference 6.

The flow field is initialized with the free-stream values occupying all the cell centers. When the computation starts, it is as if the body instantaneously materialized in the free-stream flow.

2.2. Space Marching Algorithm

The original code globally sweeps in planes moving from the body toward the shock and vice versa, updating the dependent variables at the cell centers of the plane before moving to the next plane. This scheme can be thought of as being point Jacobi within the plane and Gauss Siedel outside the plane, since values used from within the plane are at the previous time step, whereas values used from outside the plane are the most recently computed. To enable marching in the streamwise direction, the global sweep direction was changed to the streamwise direction, and the capability of performing multiple iterations (local iterations) within a plane before preceding to the next was added. In this state, the code can be used as a space marching code provided

1. A starting solution is used to begin the computation
2. Each marching plane is converged by using local iterations before moving to the next
3. Only one global pass is used

Solutions from this initially modified code were obtained for a 15° sphere-cone and can be found in section 4. This modification is referred to as "MOD1."

From MOD1, efforts were made to reduce the computer resource requirements and make the code resemble a PNS solver. The number of axial stations in the streamwise direction was reduced from the number required to define the entire geometry to five, the number needed for space marching. Grid generation, which facilitated the space marching solution procedure, was included and is addressed in section 3. The dissipation portion of the inviscid flux was modified to enable the computation of second-order accurate fluxes in the streamwise direction. These issues, as well as boundary and initial conditions and the handling of the streamwise pressure gradient, are discussed in the following sections. The code incorporating all modifications has been tagged "LAURA-SM"; the additional SM denotes Space Marching.

2.2.1. Boundary conditions. The space marching algorithm also employs pseudo cells to specify boundary conditions. The methods used to determine pseudo cell values for the upper and lower symmetry planes, free stream, and the body are identical to those used in the globally relaxed algorithm. The axis singularity boundary condition is replaced with an inflow boundary condition. The values at the inflow boundary occupy actual rather than pseudo cells. At the start of the space marching procedure, the inflow boundary is part of the starting solution. As the solution proceeds down the body, inflow boundary information is saved and carried along until no longer required. When updating dependent variables at the p station with first-order accurate streamwise fluxes, only inflow boundary information at the $p-1$ station is required, but for second-order accurate fluxes, information at both the $p-1$ and $p-2$ stations is required. The stations are shown in figure 2. The values at the $p+1$ station occupy a position equivalent to that of the pseudo cells for the outflow boundary in the globally relaxed algorithm. However, in the space marching algorithm, actual cells and not pseudo cells are used for values at the $p+1$ station. Because the location $p+1$ is treated like a supersonic outflow boundary, the values at the $p+1$ station are specified with the outflow boundary condition used

in the globally relaxed algorithm. As before with the globally relaxed algorithm, this boundary condition is valid only for streamwise supersonic flow. In section 2.3.2, it is shown that, when in the subsonic portion of the boundary layer, this specification of the outflow boundary condition causes the pressure difference across a cell to be a fraction of the total difference and enables stable space marching.

Each data plane is separated by a body normal grid system. Initially, grid coordinates at the $p - 5/2$, $p - 3/2$, $p - 1/2$, and $p + 1/2$ stations are provided with the starting solution, and the $p + 3/2$ face is computed within the space marching code prior to beginning the solution procedure. Just as with the information needed for the inflow boundary, the required grid lines are saved and carried along as the solution proceeds down the body until they are no longer needed. The starting solution is computed by the global relaxation algorithm (LAURA).

*2.2.2. **Second-order fluxes.*** To compute second-order accurate fluxes using Yee's STVD differencing, differences in characteristic variables ($\mathbf{M}^{-1}\ \Delta\mathbf{q}$) across a cell face must be defined not only at the face at which the flux is desired but also at the face ahead of it and the face behind it. All information necessary to compute second-order accurate fluxes in the circumferential and normal directions is available. However, in the streamwise direction, a problem arises because of the use of the supersonic outflow boundary condition. Referring to figure 2, the use of this boundary condition results in the dependent variables at the $p + 1$ and p locations being identical, and hence the difference in characteristic variables across the $p + 1/2$ face is zero, since $\Delta\mathbf{q}$ across this face is zero. If the difference in characteristic variables is zero at the face where the flux is desired, the face ahead, or the face behind, the min mod function returns a value of zero. Because the difference in characteristic variables across the $p + 1/2$ face is zero, and this value is input to the min mod function when computing a second-order flux at both the $p + 1/2$ and $p - 1/2$ faces, the value returned by the min mod function is zero. This results in the streamwise flux being first-order accurate. This is true for the MOD1 solutions.

To avoid this condition, differences in the characteristic variables across the $p + 1/2$ face must not be used in the min mod function. To compute a second-order flux at the $p - 1/2$ face, the min mod function compares differences from the $p - 3/2$ and $p - 1/2$ faces only. The value returned by the min mod function for the flux at the $p - 1/2$ face was also assumed to be the value returned by the min mod function for the flux at the $p + 1/2$ face. Although this method ensures that the streamwise flux at a cell face will not automatically be first-order accurate, it no longer assures that the streamwise flux is TVD.

The formulation of the equations for the space marching algorithm is identical to the global formulation. Dependent variables in streamwise planes are updated by using equation (2.18) with the above modifications in the min mod function until the solution for the plane is converged. This single formulation for the two algorithms enables easy movement from one to the other.

*2.2.3. **Streamwise pressure gradient.*** The handling of departure solutions is similar in nature to that employed by Vigneron et al. Vigneron multiplies the streamwise pressure gradient by a coefficient ω (ref. 16):

$$\omega = \frac{\gamma M_\xi^2}{1 + (\gamma - 1)M_\xi^2}$$

The value of ω, which enables stable space marching through the subsonic boundary layer, was determined through an eigenvalue analysis of the steady inviscid and viscous flux Jacobian. This value limits the streamwise pressure gradient so that the inviscid eigenvalues remain positive and the viscous eigenvalues remain real and positive. The physical interpretation of ω is that it represents the fraction of the pressure gradient associated with upstream information. The remaining portion $(1 - \omega)$ is associated with downstream information. In single sweep space marching PNS codes, the $(1 - \omega)$ portion is usually neglected. Including this portion creates an ill-posed problem when space marching. In LAURA-SM, the pressure at a cell face in the flux vector is monitored rather than the pressure gradient. The pressure at a cell face can be divided into contributions associated with downstream and upstream information. To determine the

respective contributions, the flux differences must be known and summed according to the signs of the Jacobian eigenvalues.

Within the formulation of an upwind-biased flux computed with flux-difference splitting is the knowledge that the flux contains portions of upstream and downstream information. The value κ, which represents the contributions to the pressure at a cell face from upstream information, and the remaining portion $(1-\kappa)$, which represents information from downstream, occur naturally within the formulation of the upwind-biased momentum flux at a cell face unlike Vigneron's ω, which must be computed.

As an illustration, consider the first momentum-flux term in the streamwise flux vector. The characteristic variables $(\mathbf{M}^{-1}\ \Delta\mathbf{q})$ are

$$r_1 = c^2(\rho_R - \rho_L) + (p_R - p_L) \tag{2.15a}$$

$$r_2 = \rho_R^{1/2}\rho_L^{1/2}(V_R - V_L) \tag{2.15b}$$

$$r_3 = \rho_R^{1/2}\rho_L^{1/2}(W_R - W_L) \tag{2.15c}$$

$$r_4 = c\rho_R^{1/2}\rho_L^{1/2}(U_R - U_L) + (p_R - p_L) \tag{2.15d}$$

$$r_5 = c\rho_R^{1/2}\rho_L^{1/2}(U_L - U_R) + (p_R - p_L) \tag{2.15e}$$

where

$$U = un_x + vn_y + wn_z$$

$$V = ul_x + vl_y + wl_z$$

$$W = um_x + vm_y + wm_z$$

and c is Roe's averaged speed of sound as defined in appendix A. The variables with subscripts L and R represent variables at the cell center to the left and right of the cell face, respectively. Variables with the subscript L can be thought of as upstream information, since they are upstream of the cell face, whereas those with R can be thought of as downstream information. Multiplying through by the absolute value of the eigenvalue matrix and the second row of the right eigenvector matrix $\mathbf{M}$ and isolating the pressure contributions gives the pressure difference associated with the second element of the vector $\mathbf{D}^*(\mathbf{q}_R, \mathbf{q}_L)$ in equation (2.6) as

$$\Delta p_\xi = [M_x(1 - M) + Mn_x](p_R - p_L) \tag{2.16}$$

where

$$M = \frac{U}{c}$$

$$M_x = \frac{u}{c}$$

(U and u are computed with Roe's variables.) The derivation of equation (2.16) is given in appendix B. Equation (2.16) was formulated by assuming subsonic flow in the streamwise direction and is only valid for Mach numbers from 0 to 1. For Mach numbers outside this range, equation (2.16) must be recomputed to account for the different signs of the eigenvalues.

The pressure at a cell face is

$$p(\mathbf{q}_L, \mathbf{q}_R) = \theta_1 p_R + \theta_2 p_L \tag{2.17}$$

where

$$\theta_1 = \frac{1}{2}\left[(1-M)n_x + (M-1)M_x\right] \tag{2.18a}$$

$$\theta_2 = \frac{1}{2}\left[(M+1)n_x + (1-M)M_x\right] \tag{2.18b}$$

Replacing n_x with n_y or n_z will yield similar expressions for the remaining two momentum fluxes.

For one-dimensional flows, equations (2.18) become

$$\theta_1 = \frac{1}{2}(M-1)^2 = 1-\kappa$$

$$\theta_2 = 1 - \frac{1}{2}(M-1)^2 = \kappa$$

The pressure at a cell face can then be written in terms of κ as

$$p(\mathbf{q}_L, \mathbf{q}_R) = (1-\kappa)p_R + \kappa p_L \tag{2.19}$$

where κ represents the fraction of the upstream influence and $1-\kappa$ represents the fraction of downstream influence.

Examining κ as a function of Mach number shows that it is fully consistent with the upwind formulation, and as such, the correct physics is reflected in the pressure. For a Mach number equal to 0, the pressure at a cell face is composed of equal fractions of upstream and downstream pressure, whereas for a Mach number of 1, the pressure contains no downstream influence.

The streamwise pressure difference across a cell is

$$\frac{dp}{d\xi} = -\left[(1-\kappa)p_i + \kappa p_{i-1}\right]_{i-1/2} + \left[(1-\kappa)p_{i+1} + \kappa p_i\right]_{i+1/2} \tag{2.20a}$$

Because of the specification used for the supersonic outflow boundary condition, p_{i+1} is equal to p_i, and enables the gradient to be expressed as

$$\frac{dp}{d\xi} = \kappa_{i-1/2}(p_i - p_{i-1}) \tag{2.20b}$$

Although κ and Vigneron's ω serve the same purpose, differences do exist. The value κ modifies the pressure, whereas ω does the same for the pressure gradient. Because of this difference, equation (2.20b) is not interpreted as the allowable portion of the pressure gradient, but as an allowable prescription for averaging the pressure at a cell face. Computations with this formulation have demonstrated excellent agreement with the global algorithm and experimental data as is shown in section 4.

3. Grid Generation

To generate the grid at each $p + 3/2$ station (fig. 2), the axial step size Δz must be determined. Within the starting solution, an initial axial step size is given. For the sphere-cone, subsequent step sizes are computed by multiplying the previous value of Δz by a constant factor.

For the HALIS shuttle shown in figure 3, rather than computing a step size based on an arbitrary parameter, it was based on flow variables. In one dimension, the Courant number can be expressed as

$$C_N = \frac{c\ \delta t}{\Delta z} \tag{3.1}$$

where c is the speed of sound, δt is the time step, and Δz is the step size. Solving equation (3.1) for Δz gives

$$\Delta z = \frac{c\ \delta t}{C_N} \tag{3.2}$$

The time step formulation used in LAURA and LAURA-SM is given in appendix D of reference 6 as

$$\delta t = \frac{C_N\ \Omega}{K} \tag{3.3}$$

where K is the sum of the absolute values of the inverse transit time for a convective wave to cross a cell in the ξ, η, and ζ directions plus the inverse transit time for an acoustic wave to cross a characteristic length of a cell. Substituting equation (3.3) into equation (3.2) gives

$$\Delta z = \frac{c\ \Omega}{K} \tag{3.4}$$

All values used to compute Δz were taken from the leeward symmetry plane at the p station.

The circumferential locations of body points depend upon the angle σ (fig. 4). The angle σ has a value of $-\pi/2$ on the windward symmetry plane and $\pi/2$ on the leeward symmetry plane. The angle σ,

$$\sigma = \frac{\pi}{2} - \pi(1 - \phi) \tag{3.5}$$

through the specification of ϕ in equation (3.6),

$$\phi(\theta) = c_1\theta^5 + c_2\theta^4 + c_3\theta^3 + c_4\theta^2 + c_5\theta + f \tag{3.6}$$

controls the circumferential spacing and clustering of body points. The function ϕ itself specifies a percentage of points to be distributed up to the corresponding value of θ, and $\phi'(d\phi/d\theta)$ is related to the degree of clustering at that location. Values of ϕ that give the desired values of the angle σ are determined from the constraints used to solve for the constants $(c_1, c_2, c_3, c_4, c_5)$ in equation (3.6). The constraints were determined a priori through a trial-and-error process. The specific constraints for the sphere-cone are

$$\phi(0) = 0.0 \qquad \phi(0.5) = 0.5 \qquad \phi(1) = 1.0$$

$$\phi'(0) = 1.0 \qquad \phi'(0.5) = 1.0 \qquad \phi'(1) = 1.0$$

As θ varies linearly from 0 on the leeward surface to 1 on the windward surface, these conditions give uniform spacing around the body. The corresponding constraints used for the shuttle-like geometry are

$$\phi(0) = 0.0 \qquad \phi(\theta) = 0.5 \qquad \phi(1) = 1.0$$

$$\phi'(0) = 4.0 \qquad \phi'(\theta) = 0.4 \qquad \phi'(1) = 1.5$$

The value of θ in these constraints corresponds to the location of the orbiter's wingtip, which for each station must be determined through an iterative process. These constraints cluster points near the windward symmetry plane and wingtips. A variation of the true shuttle geometry was used in this work. This variation, which models the windward side of the shuttle, is referred to as the HALIS shuttle and is based on the QUICK geometry package (ref. 23).

The distance of the outer grid boundary (Δ_3) above the body surface for the sphere-cone was determined by modifying the exponential stretching function found in reference 17:

$$\Delta_3 = \Delta_{\text{nose}}(C_2 C_1^{\text{ex}_1} + 1.0) \tag{3.7}$$

The equations for C_1 and ex are

$$C_1 = \left[\frac{(r + 1.0) - \left(\frac{r + 1.0}{r - 1.0}\right)^{\mathrm{ex}}}{\left(\frac{r + 1.0}{r - 1.0}\right)^{\mathrm{ex}} + 1.0} \right]$$

$$\mathrm{ex} = 1.0 - \frac{z_{\mathrm{body}}}{z_{\mathrm{max}}}$$

where z_{max} is the axial length of the sphere-cone and z_{body} is the axial body coordinate value at the $p + 3/2$ station. The value of C_2 is used as a growth factor and is the ratio of the shock standoff distance at the tail of the cone (Δ_{tail}) to that at the nose (Δ_{nose}). The parameters r and ex_1 are used to control how rapidly Δ_3 grows as the algorithm marches downstream and have values of 25.0 and 0.3, respectively.

To determine the outer boundary for the shuttle grid, a subroutine from the global algorithm (ALGNSHK) was modified for use in the space marching algorithm. In the global method, ALGNSHK adjusts the grid to the captured bow shock and clusters points near the body. In the space marching algorithm, ALGNSHK was modified to estimate the shock standoff distance at the $p-1/2$ and $p+1/2$ stations. A distance for the outer grid boundary at the $p+3/2$ station was then extrapolated from this information.

A body normal grid system is used. A unit vector normal to the surface is computed, and each component of the normal is multiplied by a constant. The constant increases as a function of normal distance away from the body such that the points at the $p+3/2$ station are distributed proportionally to the points at the $p + 1/2$ station.

4. Results and Discussion

4.1. MOD1—Cone

The effects of the changes made in MOD1 were examined by computing the flow over a 15° sphere-cone at an angle of attack of 0°. The cone had a nose radius equal to 0.009 meter and a length of 70 nose radii. Experimentally, this geometry was studied by Cleary (ref. 26). The nominal free-stream conditions are given in the following table:

Mach number	10.6
Reynolds number per meter	120 000
Pressure, $\mathrm{N/m^2}$	132.2
Wall temperature, K	294.0
Temperature, K	473.0
Velocity, m/s	1461.0
Density, $\mathrm{kg/m^3}$	0.00972

The predicted values of surface pressure and heating from the MOD1 solutions were compared with their corresponding values from the globally iterated solution. In addition, the MOD1 surface heating values were compared with the experimental data of Cleary (ref. 26). In all cases, the heating was normalized by a theoretical value of the stagnation point heating, and the pressure plotted was the nondimensional pressure. Both pressure and heating were plotted against z'/R_N, where z' was the axial coordinate value in meters and R_N was the nose radius, also, in meters.

Two MOD1 cases are presented, one encompassing the full length of the cone (70 nose radii) and one covering just beyond the overexpansion recompression region (24 nose radii). Both these cases use a body normal grid consisting of 21 equally spaced circumferential points, 65 points between the body and the free stream, and 27 points along the body. The resolution of the boundary layer necessary for accurate heating predictions was achieved by clustering points near the body to maintain a cell Reynolds number of order 1. The cell Reynolds number is defined as

$$(\mathrm{Re})_{\mathrm{wall}} = \left(\frac{\rho c l}{\mu}\right)_{\mathrm{wall}} \tag{4.1}$$

where l is the cell height.

The grid generation routine has not been modified, and as a consequence, the grid used for the first MOD1 computation has spacing in all computational directions equal to that used to obtain the global solution. Attempting to lessen the effect of grid-induced differences between the global and MOD1 solutions, the grid over the nose region from the global solution was used as the grid for the starting solution. In doing this, the combination of the starting solution grid and the MOD1 grid is equivalent to the grid used to compute the global solution.

Solutions were converged with a large Courant number ($\approx$100) at the start of the iteration process to damp the low frequency errors. After a set number of iterations, the Courant number was lowered to damp the high frequency errors. Convergence was also enhanced by the use of a spatially varying time step tied to a constant Courant number. Solutions are considered converged when local iteration has reduced the error of the right-hand-side residual of the governing equations to 1.0E-05. The error is measured by using a L_2 error norm. To reach this criterion for the MOD1 solutions, 600 local iterations evenly split between Courant numbers of 100.0 and 3.0 were run.

The first MOD1 solution, covering the entire body, compares well with the global solution. The streamwise surface pressure is in good agreement even with the large marching steps taken. Figure 5 indicates that pressure agreement is within a fraction of a percent except in the recompression area. In this area, MOD1 underpredicts the minimum pressure by 4 percent compared with the global solution. As the flow recompresses, a 13-percent difference in surface pressure is noted in figure 5. Once the streamwise pressure gradient becomes small, the two methods are in excellent agreement.

The heating comparison is shown in figure 6. Figure 6 shows good agreement, except in the recompression region where MOD1 underpredicts the global heating by 10 percent. Agreement with the experimental data of Cleary (ref. 26) is slightly better in figure 7, with an underprediction of 7 percent in this area. Over the remaining portion of the cone, MOD1 and the globally relaxed solution compare equally well with the data of Cleary.

The differences in the recompression region may be due to the marching step being too large. As a result, the extrapolated value of κ at the $p + 1/2$ station may have been a poor estimate of the amount of pressure to be retained. Also, the difference in the order of accuracy of the streamwise fluxes between the space marching (first-order) and the global (second-order) algorithms may have made a small contribution to the differences seen in the recompression region. Decreasing the step size should help lessen this discrepancy by providing better resolution in the overexpansion recompression region. This is examined in the second MOD1 case.

The second MOD1 solution was computed by using a grid with the same circumferential and normal spacing and one third the streamwise spacing as the first. The streamwise spacing of the first MOD1 grid varied from 0.004 to 0.058 meter. The second MOD1 grid varied from 0.001 to 0.019 meter. For both grids, the spacing increased exponentially. The grid dimensions and convergence method were unchanged.

Solutions for surface heating and pressure are much improved over the first MOD1 predictions. The difference in the minimum pressure is only 2 percent, and figure 8 also indicates that the increased resolution of the overexpansion recompression region has removed the shift in pressure experienced in the first solution. Figure 9 reveals improved heating predictions when compared with the global values. Unlike the first MOD1 solution, good agreement is obtained in the overexpansion recompression region.

In comparing the two MOD1 solutions, the overexpansion recompression region is apparently more sensitive to step size than other parts of the flow. Poor agreement in this area is found in the first MOD1 solution, and good agreement is obtained in the second solution, with the only difference being the step size. Outside of this area both solutions compare equally as well independent of step size.

4.2. LAURA-SM—Cone

The flow over the cone described in section 4.1 was recomputed with LAURA-SM to examine the effects of the additional modifications. Again, surface pressure and heating are compared.

Four cases were run. Each had a different factor by which the axial step size grew linearly. Factors were varied to see the effect of the step size on stability and solution quality, and to determine in terms of computational time whether a large or small marching step is more efficient. The initial step size for all runs was 0.0012 meter and was chosen based on the second MOD1 case. The four axial step size stretching factors (FKT) examined were 1.006, 1.10, 1.14, and 1.17, corresponding to 233, 45, 35, and 31 solution planes. Factors beyond 1.20 were not explored, since this would lead to excessive grid stretching in the streamwise direction.

The effect of step size on surface pressure is shown in figure 10. In general, excellent agreement with the global solution is found, except at the end of the recompression region. The expanded scale on the figures indicates that, as the stretching factor is increased, a deviation between the global and space marching solutions at the end of the recompression becomes prominent. The larger the stretching factor, the longer the delay in recognizing the end of the recompression. A maximum difference of 2 percent is associated with the largest factor (FKT = 1.17) and a difference of less than a percent with the smallest factor (FKT = 1.006). In looking at the location of the minimum surface pressure, decreasing the stretching factor produced a lower minimum pressure and moved its location downstream. Comparing the solution computed with the largest factor with that of the smallest shows that the minimum pressure is lower by 1/2 percent and has moved downstream approximately 1 nose radius.

The effect of step size on surface heating and comparisons with the global solution are shown in figure 11. Overall the space marching solutions compare well with the data. The predictions for surface heating from the space marching code more closely match the predictions from the global relaxation algorithm than the experimental data of Cleary (ref. 26). As the stretching factor is decreased, differences between the global and space marching heating values are resolved to less than 8 percent. The space marching solution also shows a small inflection in the heating just downstream of the minimum in pressure. The inflection becomes less noticeable as the step size is reduced. As did the location of the minimum in pressure, the inflection in heating also moves downstream as the stretching factor is decreased. Both the changes in location and magnitude are small. In comparing results from the largest and smallest factors, a 1-percent difference in the minimum value exists, and the location moves downstream 1 nose radius.

The deviation noted between the space marching solutions appears to be step size related. Increased resolution of the flow in the streamwise direction resolves these discrepancies but adds to the computational time to obtain a solution. Because the global and space marching solutions are in good agreement outside the recompression region, regardless of the step size, including the overexpansion-recompression region in the starting solution for flows where it occurs early and is well defined may enhance computational efficiency, since larger step sizes can be used outside of this region. While a larger step requires more iterations to converge than a smaller one, the pass over the vehicle is performed in less computer time with the larger step, as is shown in the next paragraph.

Two advantages of using a space marching algorithm are that less computer time is needed to obtain a solution when compared with a globally iterated algorithm and the space marching algorithm can be used on a wider range of computers. The following table shows the computer time required to compute the global and space marching solutions:

Code	Time, sec	Factor	Steps
LAURA-Global	12 800	1.00	39
LAURA-SM (FKT = 1.006)	7 429	1.72	233
LAURA-SM (FKT = 1.100)	7 880	1.62	45
LAURA-SM (FKT = 1.140)	7 256	1.76	35
LAURA-SM (FKT = 1.170)	6 949	1.84	31

The time shown is the CPU time required for the VPS-32 at the Langley Research Center. The first column lists the code used, the second the CPU time, the third lists the ratio of the global to space marching CPU time, and the fourth shows the number of streamwise steps taken. The CPU times for the space marching solutions include the 3102 sec required to obtain the starting solution. The starting solution accounts for approximately 45 percent of the CPU time in the space marching cases. This is due to the subsonic region at the nose. It is possible to reduce the computational time of the space marching and global algorithms approximately 40 percent by freezing the left-hand side of the governing equations, and updating it periodically (ref. 6).

4.3. LAURA-SM—Shuttle Orbiter

To evaluate how the space marching algorithm performs when exposed to significant three-dimensional effects, the flow over the HALIS shuttle geometry was computed. This geometry models the windward surface of the Space Shuttle orbiter. The HALIS model fills in the leeward surface with elliptic cross sections. Figure 12 shows both geometries.

Predicted surface heating values from the space marching algorithm are compared with the unpublished experimental heating data of Miller. At higher angles of attack, Miller's data have been published (ref. 27), but are not applicable for use in this instance since the windward surface is subsonic. The flow was computed to correspond to the wind tunnel conditions used by Miller. The nominal flow conditions are as follows:

Mach number	10.0
Reynolds number per meter	690 000
Pressure, N/m^2	110.7
Wall temperature, K	300.0
Temperature, K	492.0
Velocity, m/s	1 410.0
Density, kg/m^3	0.00784
Angle of attack, deg	10

It should be noted that in Miller's experiment, the actual shuttle geometry was used and not the HALIS shuttle geometry shown in figure 12(b). Even though the leeward surface used to obtain the predicted results is different for that used in the experiment, comparisons of windward surface values can be made. The circumferential flow around the chines and wingtips is supersonic, and minimal influence is transmitted circumferentially through the boundary layer; therefore, the leeward surface shape has negligible effect on windward values.

The space marching solution begins at $z'/z_{\text{ref}} = 0.062$ and continues toward a value of 0.942. The axial distance of the experimental model from the nose to the base z_{ref} is 0.196 meter. The space marching algorithm computed values at 97 streamwise stations. A Courant number of 50.0 was used to converge each station. A station was considered converged when local iteration had reduced the L_2 norm below a value of 0.2E-06. A change in the grid system was also made at $z'/z_{\text{ref}} = 0.350$. At this location, the grid system was changed from body normal to axis normal. The change was made to circumvent the occurrence of normal grid lines crossing in the area where the wings begin to flare.

The heating predictions agree within 5 percent when compared with the experimental data of Miller (ref. 27). Figure 13 shows the predicted values following experiment, until predicted values begin to diverge at a location z/z_{ref} of 0.55. At this location, a cross-flow shock on the leeward surface intensifies. Beyond this location, the heating agreement worsens. The space marching algorithm fails at 0.70, where the algorithm encounters reversed streamwise flow on the leeward surface near the symmetry plane.

Two potential causes of this divergence have been identified. First, the developing cross-flow shock is oblique to the grid. Such orientations cause enhanced dissipation across the captured shock and may induce physically incorrect pressure fields in the immediate vicinity that can induce separation. Limitations in the present grid generation capabilities preclude resolution of this problem. Second, the implementation of the second-order corrections in Yee's

STVD scheme must be modified in the streamwise marching direction. The min mod function used in the STVD approach selects the gradient of characteristic variables which is guaranteed not to introduce any new maximums or minimums into the problem or revert to first-order accuracy. By doing this, shocks and other discontinuities are captured without oscillations. The modification made in the streamwise direction, which excludes extrapolated gradients at downstream cells from consideration in the min mod function, may degrade the solution. This change may neutralize the TVD characteristics of the original scheme, particularly in the subsonic portion of the boundary layer. However, this restriction cannot be removed without sacrificing second-order accuracy in the marching direction.

Pressure contours for the global and space marching algorithms at three cross-sectional locations are shown in figure 14. The presence of the cross-flow shock is confirmed in these contours. The patterns of the contours are similar for the global and space marching solutions. Both indicate a high gradient region near the wingtip and the formation of a cross-flow shock near the leeward symmetry plane. A thickening of the shock is also noted in the space marching contours as the solution proceeds downstream.

5. Concluding Remarks

The Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA) code has been converted to perform space marching of the thin-layer Navier-Stokes equations in the spirit of a parabolized Navier-Stokes (PNS) solver. This study has demonstrated that the intrinsic upwind treatment of the streamwise pressure combined with an appropriate outflow boundary condition avoids departure solutions and stable marching is possible without the explicit inclusion of a limiting variable within the governing equations. Excellent agreement with the globally iterated algorithm for surface pressure was obtained with the new code on a sphere-cone geometry, and computed heat-transfer distributions on the sphere-cone and the HALIS shuttle orbiter were in good agreement when compared with experimental values.

Care must be exercised when computing through an overexpansion-recompression region. Taking too large a step in this area may not cause the solution to diverge but to yield an inaccurate solution. Unfortunately, a prescription for the best combination of initial step size and stretching factor which would give the best solution and minimize computer time is not known. The best combination must be determined through experience.

The LAURA-SM algorithm was faster than the globally relaxed algorithm, but only by a factor of approximately 1.6. Room for improvement may lie in including regions which require a small step size, such as the overexpansion-recompression region associated with the cone, in the starting solution and initiating space marching beyond the overexpansion-recompression region. Another approach may be to freeze the inverted implicit part (left-hand side) of the governing equations and update it periodically. Doing this can reduce computer time by approximately 40 percent and would have no effect on the converged solution. This same freezing procedure can be used with the global algorithm, saving a similar percentage of computer resources.

The space marching HALIS shuttle orbiter solution fails at the cross-flow shock on the leeward surface. Since the second-order streamwise flux is no longer total variation diminishing, it may introduce spurious oscillations in the solution which cause divergence. If the second-order streamwise flux is at fault and the space marching formulation of the governing equations is to remain unchanged, a way of specifying the gradients of characteristic variables consistent with the global approach must be developed if the min mod is to function as intended. However, if second-order accurate streamwise fluxes are to be computed, this is not possible, since downstream gradients of characteristic variables are required in subsonic regions and are unavailable. The only other way around this condition is to use first-order streamwise fluxes and sacrifice overall second-order accuracy. Another factor which may have caused the space marching algorithm to fail is grid alignment. Not aligning the grid with a shock can result in the pressure field being different from when the grid is aligned. The resulting pressure field from the space marching solution could have caused the reversed flow found on the leeward surface. This question was not resolved because the required adaptive-grid capabilities are not incorporated in this code.

Appendix A

Jacobian Matrices

The unsteady inviscid flux Jacobian matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 0 & n_x & n_y & n_z & 0 \\ -uU + \beta\alpha n_x & U + un_x - un_x\beta & un_y - vn_x\beta & un_x - wn_x\beta & \beta n_x \\ -vU + \beta\alpha n_y & vn_x - un_y\beta & U + vn_y - vn_y\beta & vn_z - wn_y\beta & \beta n_y \\ -wU + \beta\alpha n_z & wn_x - un_z\beta & wn_y - vn_z\beta & U + wn_z - wn_z\beta & \beta n_z \\ U(\alpha\beta - H) & Hn_x - uU\beta & Hn_y - vU\beta & Hn_z - wU\beta & \gamma U \end{bmatrix} \tag{A1}$$

The matrix $\mathbf{A}$ can be decomposed into it right ($\mathbf{M}$) and left ($\mathbf{M}^{-1}$) eigenvectors and a diagonal matrix of its eigenvalues ($\boldsymbol{\lambda}$). These matrices are

$$\mathbf{M} = \begin{bmatrix} \frac{-1}{c^2} & 0 & 0 & \frac{1}{2c^2} & \frac{1}{2c^2} \\ \frac{-u}{c^2} & l_x & m_x & \frac{u + cn_x}{2c^2} & \frac{u - cn_x}{2c^2} \\ \frac{-v}{c^2} & l_y & m_y & \frac{v + cn_y}{2c^2} & \frac{v - cn_y}{2c^2} \\ \frac{-w}{c^2} & l_z & m_z & \frac{w + cn_z}{2c^2} & \frac{w - cn_z}{2c^2} \\ -\frac{\alpha}{c^2} & V & W & \frac{\alpha + Uc}{2c^2} + \frac{1}{2\beta} & \frac{\alpha - Uc}{2c^2} + \frac{1}{2\beta} \end{bmatrix} \tag{A2}$$

$$\mathbf{M}^{-1} = \begin{bmatrix} \alpha\beta - c^2 & -\beta u & -\beta v & -\beta w & \beta \\ -V & l_x & l_y & l_z & 0 \\ -W & m_x & m_y & m_z & 0 \\ \alpha\beta - Uc & cn_x - \beta u & cn_y - \beta v & cn_z - \beta w & \beta \\ \alpha\beta + Uc & -cn_x - \beta u & -cn_y - \beta v & -cn_z - \beta w & \beta \end{bmatrix} \tag{A3}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} U & 0 & 0 & 0 & 0 \\ 0 & U & 0 & 0 & 0 \\ 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & U + c & 0 \\ 0 & 0 & 0 & 0 & U - c \end{bmatrix} \tag{A4}$$

where

$$U = un_x + vn_y + wn_z$$

$$V = ul_x + vl_y + wl_z$$

$$W = um_x + vm_y + wm_z$$

$$\alpha = \frac{1}{2}(u^2 + v^2 + w^2)$$

$$c^2 = (H - \alpha)\beta$$

In order for equation (2.6) to be an exact equality, specially averaged variables known as Roe's averaged variables are used in the computation of the matrices in equations (A1) through (A4). These variables are defined as

$$\left.\begin{aligned} u &= a_1(a_2 u_R + u_L) \\ v &= a_1(a_2 v_R + v_L) \\ w &= a_1(a_2 w_R + w_L) \\ H &= a_1(a_2 H_R + H_L) \end{aligned}\right\} \tag{A5}$$

where

$$a_1 = \left(\frac{\rho_R}{\rho_L}\right)^{1/2} \qquad a_2 = \frac{1}{1 + a_1}$$

The varriable β is defined in terms of the ratio of specific heat as

$$\beta = \gamma - 1$$

The viscous Jacobian, shown in equation (A6), was used in equation (2.14) and was taken from reference 6:

$$\frac{\partial \mathbf{H}_V}{\partial \mathbf{q}} = \left[\frac{\mu(\vec{\nabla}_\chi \cdot \vec{n})}{\mathrm{Re}\ \rho}\right][b_{i,j}] \tag{A6}$$

where

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

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

$$b_{2,1} = -\left(\mathrm{u} + \frac{Un_x}{3}\right)$$

$$b_{3,1} = -\left(\mathrm{v} + \frac{Un_y}{3}\right)$$

$$b_{4,1} = -\left(\mathrm{w} + \frac{Un_z}{3}\right)$$

$$b_{5,1} = ub_{2,1} + vb_{3,1} + wb_{4,1} - \frac{\gamma}{\mathrm{Pr}}[E - (\mathrm{u}^2 + \mathrm{v}^2 + \mathrm{w}^2)]$$

$$b_{2,2} = 1 + \frac{n_x n_x}{3}$$

$$b_{3,2} = b_{2,3} = \frac{n_x n_y}{3}$$

$$b_{4,2} = b_{2,4} = \frac{n_x n_z}{3}$$

$$b_{4,3} = b_{3,4} = \frac{n_y n_z}{3}$$

$$b_{3,3} = 1 + \frac{n_y n_y}{3}$$

$$b_{4,4} = 1 + \frac{n_z n_z}{3}$$

$$b_{5,2} = ub_{2,2} + vb_{3,2} + wb_{4,2} - \frac{\gamma \mathrm{u}}{\mathrm{Pr}}$$

$$b_{5,3} = ub_{2,3} + vb_{3,3} + wb_{4,3} - \frac{\gamma \mathrm{v}}{\mathrm{Pr}}$$

$$b_{5,4} = ub_{2,4} + vb_{3,4} + wb_{4,4} - \frac{\gamma \mathrm{w}}{\mathrm{Pr}}$$

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

All variables are values computed at cell faces with the exception of u, v, and w, which are cell-centered values.

Appendix B

Streamwise Pressure Gradient

Equation (2.16) is computed with equation (2.6), which is

$$\mathbf{D}(\mathbf{q}_L, \mathbf{q}_R) = (\mathbf{M}\boldsymbol{\lambda}^+\mathbf{M}^{-1} - \mathbf{M}\boldsymbol{\lambda}^-\mathbf{M}^{-1})(\mathbf{q}_R - \mathbf{q}_L) \tag{B1}$$

where $\boldsymbol{\lambda}^+$ and $\boldsymbol{\lambda}^-$ are given by

$$\boldsymbol{\lambda}^+ = \frac{\boldsymbol{\lambda} + |\boldsymbol{\lambda}|}{2} \qquad \boldsymbol{\lambda}^- = \frac{\boldsymbol{\lambda} - |\boldsymbol{\lambda}|}{2} \tag{B2}$$

The first term in equation (B1) represents information associated with the positive eigenvalues and the second term is associated with the negative eigenvalues. For stable space marching, reverse flow is not allowed; therefore, the streamwise Mach number must be greater than or equal to 0 throughout the flow field. In the boundary layer, the streamwise Mach number is further restricted to be less than 1, since the flow is subsonic. Accordingly in the subsonic boundary layer, four eigenvalues will be positive $(\tilde{U}, \tilde{U}, \tilde{U}, \tilde{U} + \tilde{c})$ and one will be negative $(\tilde{U} - \tilde{c})$. Knowing the eigenvalue signs, equation (B1) can be expanded accordingly as shown below.

$$\begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{bmatrix} = \mathbf{M} \times \frac{1}{2} \begin{bmatrix} (\tilde{U} + |\tilde{U}|)(\tilde{c}^2\ \Delta\rho + \Delta p) \\ (\tilde{U} + |\tilde{U}|)\rho_L^{1/2}\rho_R^{1/2}\ \Delta V \\ (\tilde{U} + |\tilde{U}|)\rho_L^{1/2}\rho_R^{1/2}\ \Delta W \\ \left[(\tilde{U} + \tilde{c}) + (|\tilde{U} + \tilde{c}|)\right]\left(\tilde{c}\rho_L^{1/2}\rho_R^{1/2}\ \Delta U + \Delta p\right) \\ 0 \end{bmatrix}$$

$$- \mathbf{M} \times \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \left[(\tilde{U} - \tilde{c}) - (|\tilde{U} - \tilde{c}|)\right]\left(-\tilde{c}\rho_L^{1/2}\rho_R^{1/2}\ \Delta U + \Delta p\right) \end{bmatrix} \tag{B3}$$

Multiplying the vectors on the right-hand side of equation (B3) by the second row of $\mathbf{M}$ gives

$$D_2 = (\tilde{U}\tilde{c}^2\ \Delta\rho + \tilde{U}\ \Delta p)\left(-\frac{\tilde{u}}{\tilde{c}^2}\right) + \left(\tilde{U}\rho_L^{1/2}\rho_R^{1/2}\ \Delta V\right) l_x + \left(\tilde{U}\rho_L^{1/2}\rho_R^{1/2}\ \Delta W\right) m_x$$
$$+ \left[(\tilde{U} + \tilde{c})\left(\tilde{c}\rho_L^{1/2}\rho_R^{1/2}\ \Delta U + \Delta p\right)\right]\left(\frac{\tilde{u} - \tilde{c}n_x}{2\tilde{c}^2}\right) - \left[(\tilde{U} - \tilde{c})\left(\tilde{c}\rho_L^{1/2}\rho_R^{1/2}\ \Delta U + \Delta p\right)\right]\left(\frac{\tilde{u} - \tilde{c}n_x}{2\tilde{c}^2}\right) \tag{B4}$$

Isolating the pressure terms in equation (B4) gives the same pressure contribution to the momentum flux associated with the flux difference. The result is

$$\Delta p_\xi = \Delta p\left(\frac{-\tilde{U}\tilde{u}}{\tilde{c}^2} + \frac{\tilde{U}n_x}{\tilde{c}} + \frac{\tilde{u}}{\tilde{c}}\right) \tag{B5}$$

Equation (B5) can be simplified to

$$\Delta p_\xi = p_R\left[M_x(1 - M) + Mn_x\right] - p_L\left[M_x(1 - M) + Mn_x\right] \tag{B6}$$

where

$$M = \frac{\widetilde{U}}{\tilde{c}} \qquad M_x = \frac{\tilde{u}}{\tilde{c}}$$

Variables with a tilde ($\tilde{}$) over them are Roe's averaged variables and should be computed as shown in appendix A (eqs. (A5)). The delta (Δ) indicates a difference across a cell face computed with cell-centered values, for example

$$\Delta p = p_R - p_L$$

References

1. Godunov, S. K.: *A Difference Method for the Numerical Calculation of Discontinuous Solutions of Hydrodynamic Equations.* JPRS 7225, U.S. Dep. of Commerce, Nov. 29, 1960.
2. Osher, Stanley: Riemann Solvers, the Entropy Condition, and Difference Approximations. *SIAM J. Numer. Anal.*, vol. 21, no. 2, Apr. 1984, pp. 217–235.
3. Roe, P. L.: Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *J. Comput. Phys.*, vol. 43, no. 2, Oct. 1981, pp. 357-372.
4. 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.
5. 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.
6. Gnoffo, Peter A.: *An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows.* NASA TP-2953, 1990.
7. Coakley, T. J.: Implicit Upwind Methods for the Compressible Naiver-Stokes Equations. *A Collection of Technical Papers—6th AIAA Computational Fluid Dynamics Conference*, July 1983, pp. 505–514. (Available as AIAA 83-1958.)
8. Lombard, C. K.; Bardina, J.; and Venkatapathy, E.: AOTV Bluff Body Flow—Relaxation Algorithm. AIAA-84-1699, June 1984.
9. 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.)
10. Lawrence, Scott L.; Tannehill, J. C.; and Chaussee, Denny S.: An Upwind Algorithm for the Parabolized Navier-Stokes Equations. AIAA-86-1117, May 1986.
11. Korte, John J.; and McRae, D. Scott: Explicit Upwind Algorithm for the Parabolized Navier-Stokes Equations. AIAA-88-0716, Jan. 1988.
12. Beam, Richard M.; and Warming, R. F.: An Implicit Factored Scheme for the Compressible Navier-Stokes Equations. AIAA J., vol. 16, no. 4, Apr. 1978, pp. 393–402.
13. MacCormack, R. W.: A Numerical Method for Solving the Equations of Compressible Viscous Flow. AIAA-81-0110, Jan. 1981.
14. Walters, Robert W.; and Dwoyer, Douglas L.: *Efficient Solutions to the Euler Equations for Supersonic Flow With Embedded Subsonic Regions.* NASA TP-2523, 1987.
15. Newsome, Richard W.; Walters, Robert W.; and Thomas, James L.: An Efficient Iteration Strategy for Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations. *A Collection of Technical Papers– AIAA 8th Computational Fluid Dynamics Conference*, June 1987, pp. 126–142. (Available as AIAA-87-1113.)
16. Vigneron, Yvon C.; Rakich, John V.; and Tannehill, John C.: Calculation of Supersonic Viscous Flow Over Delta Wings With Sharp Subsonic Leading Edges. AIAA-78-1137, July 1978.
17. Anderson, Dale A.; Tannehill, John C.; and Pletcher, Richard H.: *Computational Fluid Mechanics and Heat Transfer.* Hemisphere Publ. Corp., c.1984.
18. Rakich, John V.: Iterative PNS Method for Attached Flows With Upstream Influence. *A Collection of Technical Papers—AIAA 6th Computational Fluid Dynamics Conference*, July 1983, pp. 473–482. (Available as AIAA-83-1955.)
19. Brown, James L.: *Parabolized Navier-Stokes Solutions of Separation and Trailing-Edge Flows.* NASA TM-84378, 1983.
20. Barnett, M.; and Davis, R. T.: A Procedure for the Calculation of Supersonic Flows With Strong Viscous-Inviscid Interaction. AIAA-85-0166, Jan. 1985.
21. Schiff, Lewis B.; and Steger, Joseph L.: Numerical Simulation of Steady Supersonic Viscous Flow. AIAA-79-0130, Jan. 1979.
22. Lubard, Stephen C.; and Helliwell, William S.: Calculation of the Flow on a Cone at High Angle of Attack. *AIAA J.*, vol. 12, no. 7, July 1974, pp. 965–974.
23. Vachris, Alfred F., Jr.; and Yaeger, Larry S.: QUICK-GEOMETRY—A Rapid Response Method for Mathematically Modeling Configuration Geometry. *Applications of Computer Graphics in Engineering*, NASA SP-390, 1975, pp. 49–73.
24. Yee, H. C.: *On Symmetric and Upwind TVD Schemes.* NASA TM-86842, 1985.
25. Harten, Ami: *High-Resolution Schemes for Hyperbolic Conservation Laws.* DOE/ER/03077-175, DE82 015473 (Interchange No. NCA2-OR525-101 and Contract DE-AC02-76FR03077), Courant Mathematics and Computing Lab., New York Univ., Mar. 1982.
26. Cleary, Joseph W.: *Effects of Angle of Attack and Bluntness on Laminar Heating-Rate Distributions of a 15° Cone at a Mach Number of 10.6.* NASA TN D-5450, 1969.
27. Miller, C. G.: Experimental Investigation of Gamma Effects on Heat Transfer to a 0.006 Scale Shuttle Orbiter at Mach 6. AIAA-82-0826, June 1982.

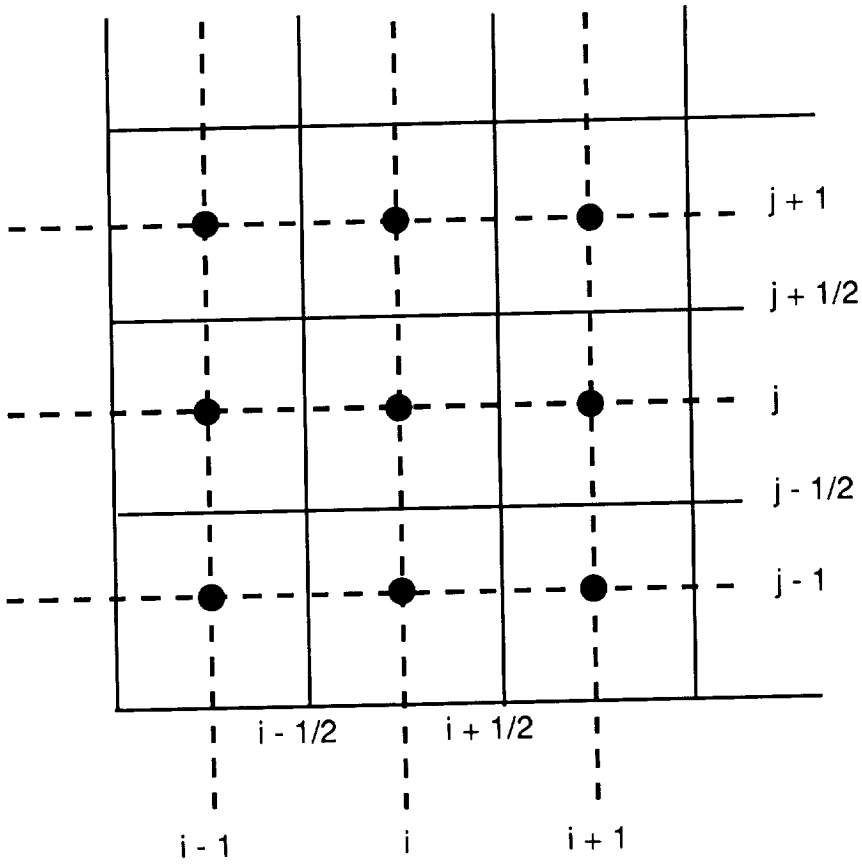


Figure 1. Indexing system.

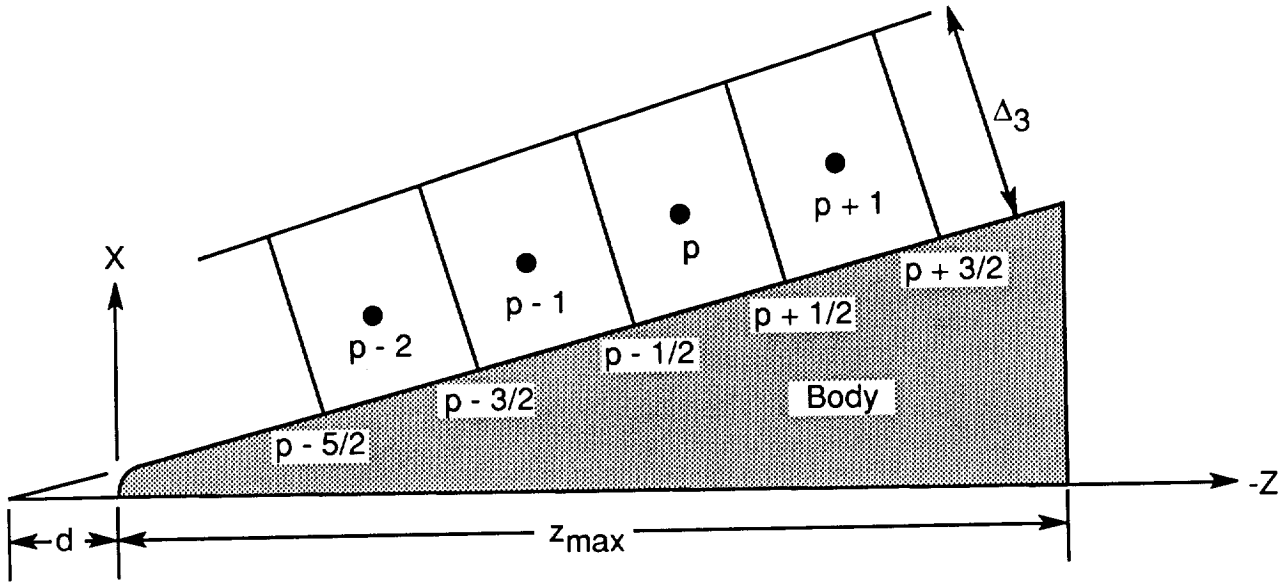


Figure 2. Cone symmetry plane.

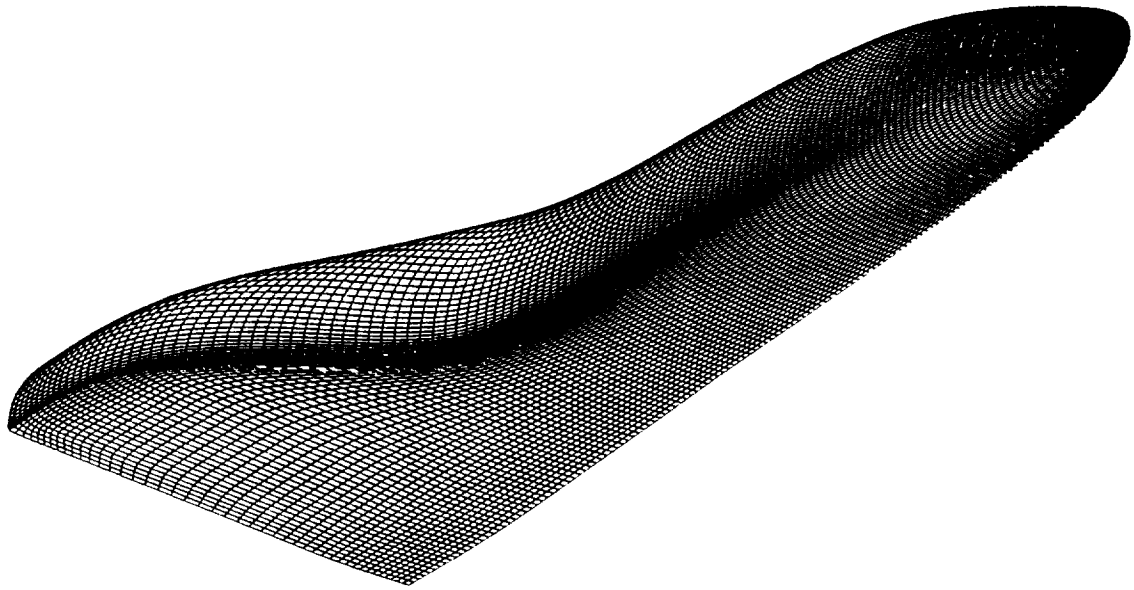


Figure 3. HALIS space shuttle surface geometry.

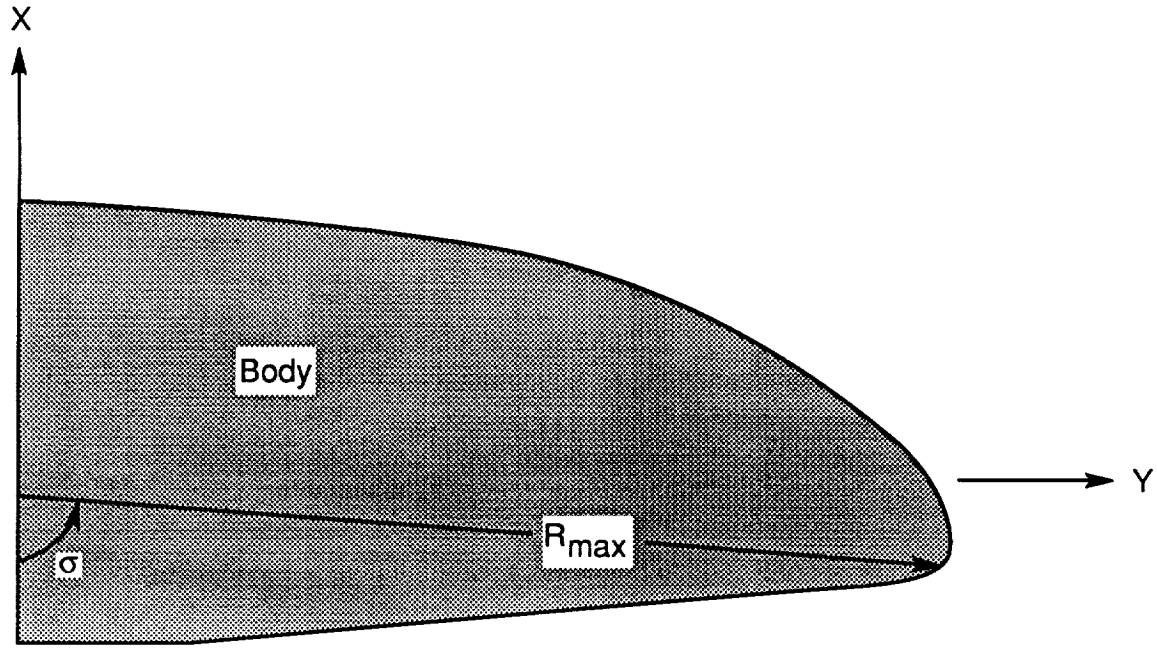


Figure 4. HALIS orbiter cross section.

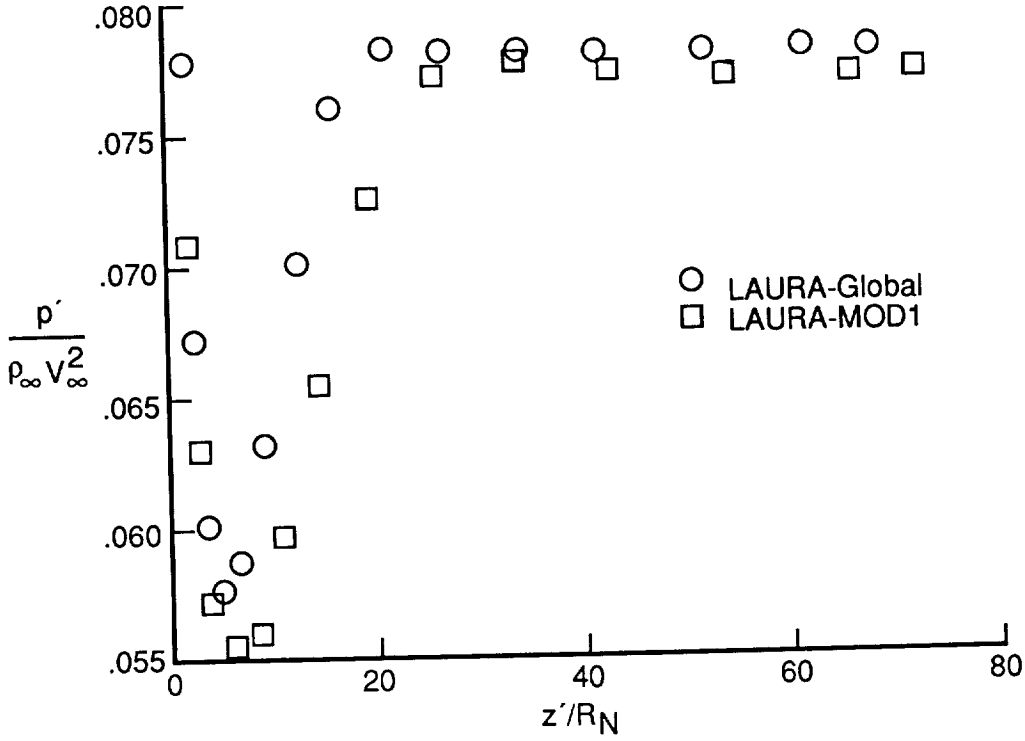


Figure 5. Global versus MOD1 surface pressures for $z'/R_N = 70$, $\rho'_\infty = 0.00972$ kg/m^3, $S'_\infty = 1461$ m/s, and $R_N = 0.89$ cm.

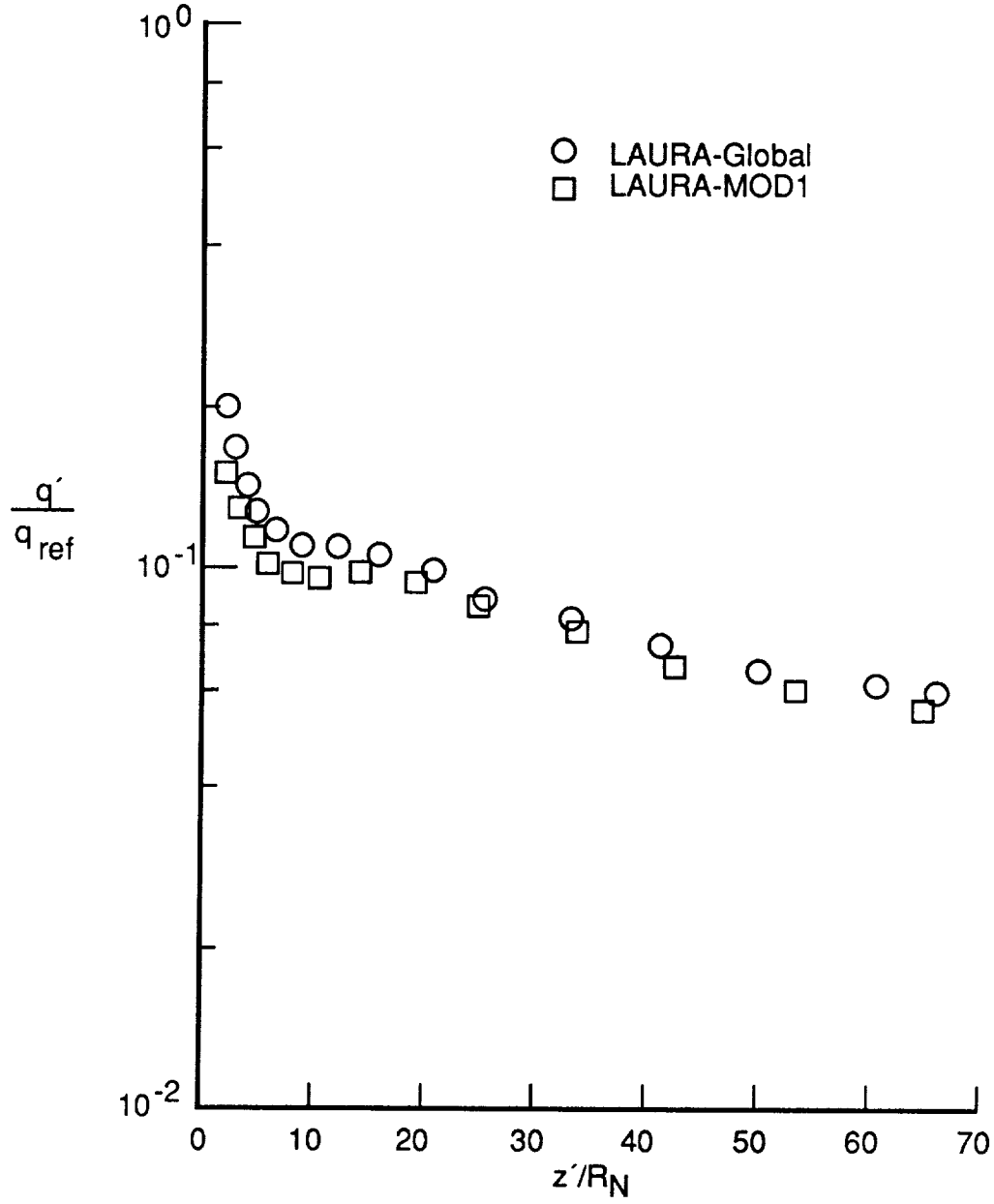


Figure 6. Global versus MOD1 surface heating for $z'/R_N = 70$, $\rho'_\infty = 0.00972$ kg/m^3, $S'_\infty = 1461$ m/s, $R_N = 0.89$ cm, and $q_{\text{ref}} = 0.2156$ MW/m^2.

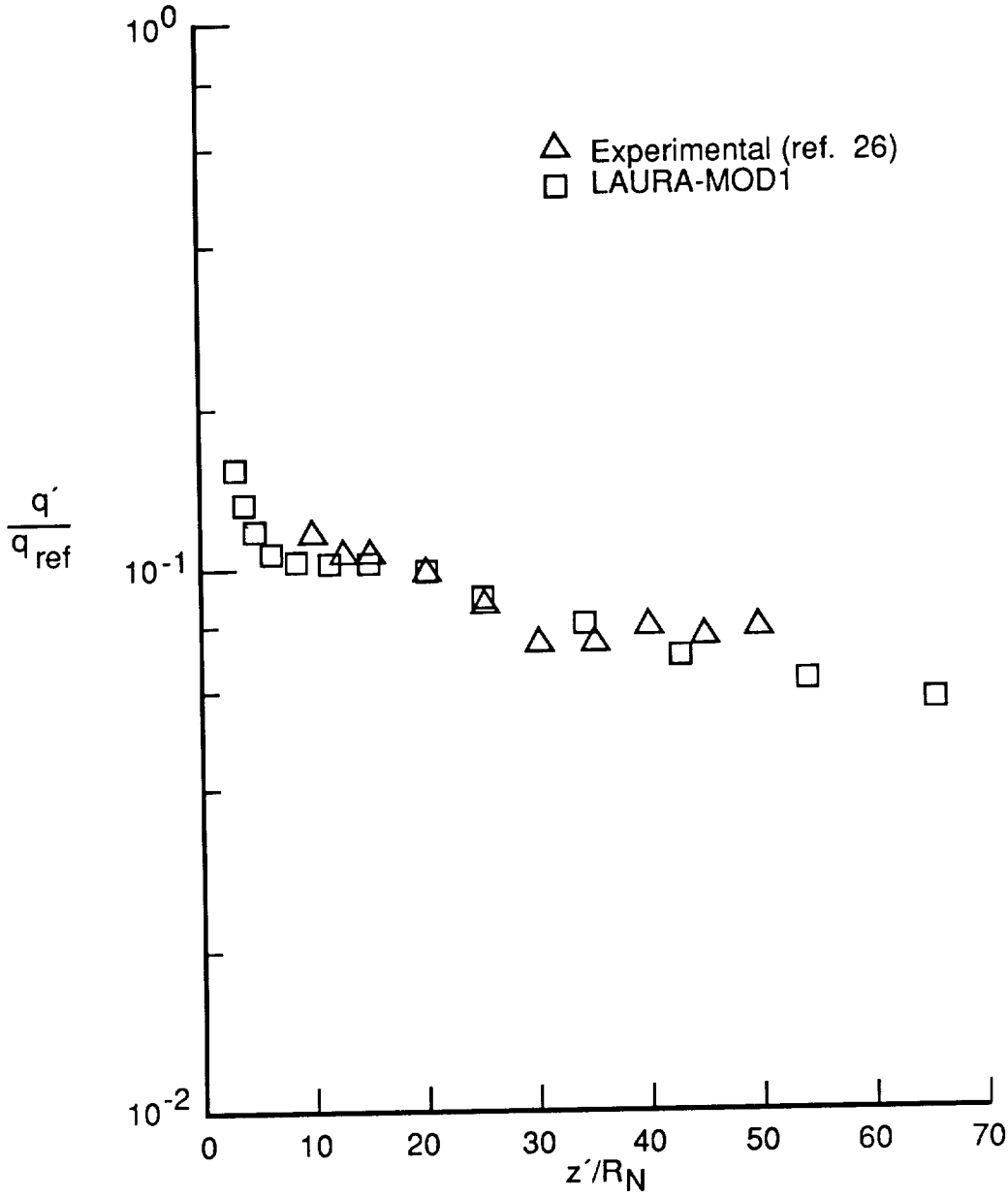


Figure 7. Experimental versus MOD1 surface heating for $z'/R_N = 70$, $\rho'_\infty = 0.00972$ kg/m^3, $S'_\infty = 1461$ m/s, $R_N = 0.84$ cm, and $q_{ref} = 0.2156$ MW/m^2.

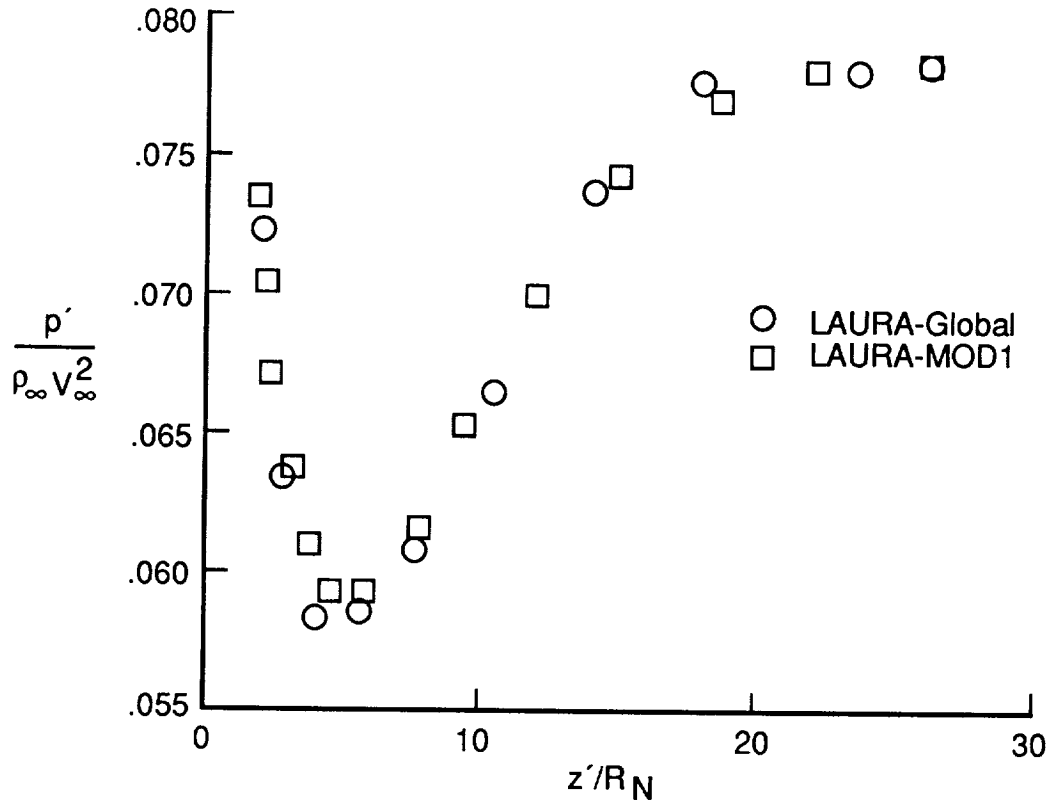


Figure 8. Global versus MOD1 surface pressure for $z'/R_N = 30$, $\rho'_\infty = 0.00972$ kg/m^3, $S'_\infty = 1461$ m/s, and $R_N = 0.89$ cm.

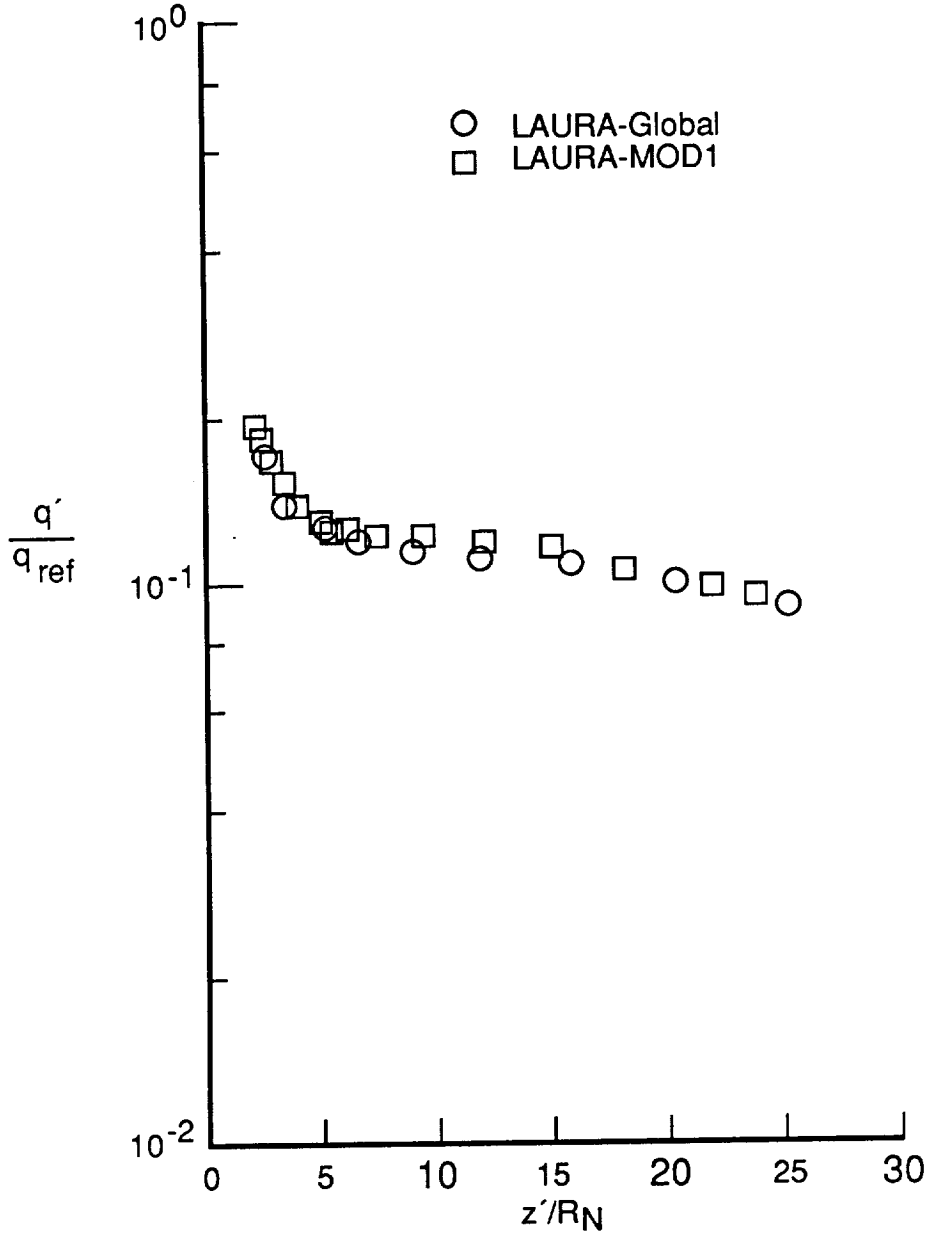
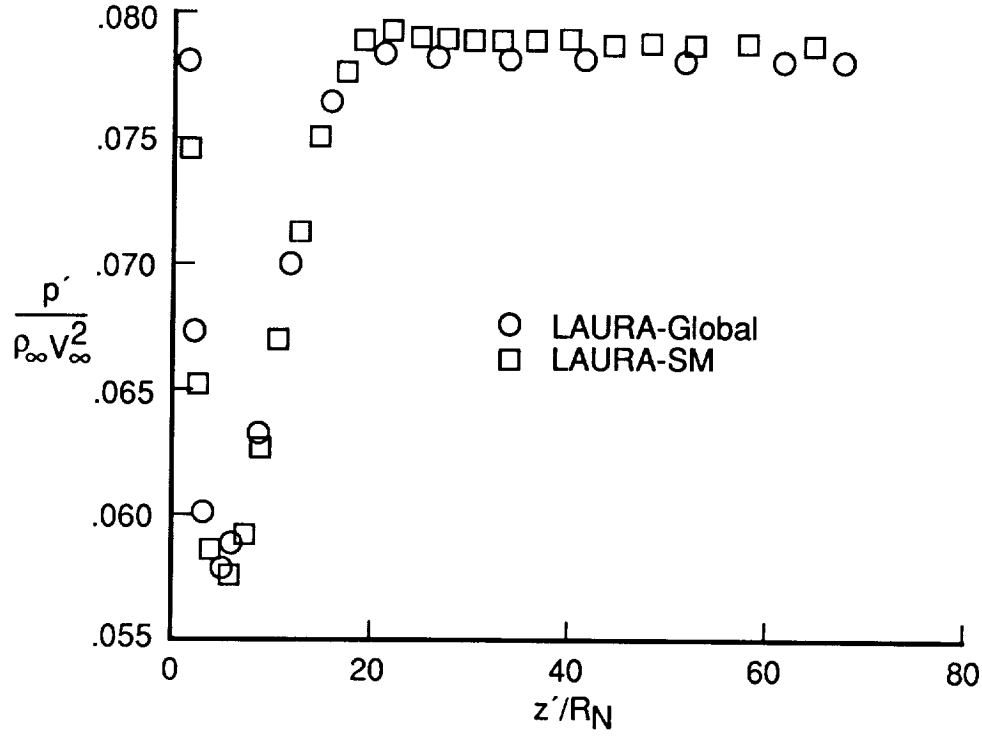
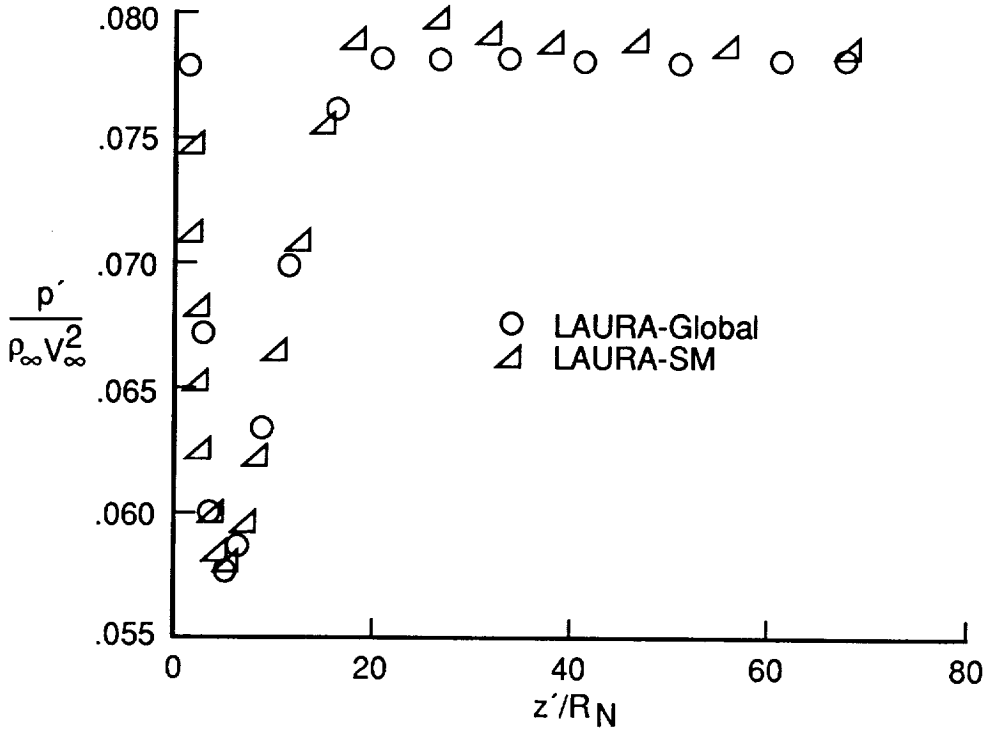


Figure 9. Global versus MOD1 surface heating for $z'/R_N = 30$, $\rho'_\infty = 0.00972$ kg/m^3, $S'_\infty = 1461$ m/s, and $q_{\text{ref}} = 0.2156$ MW/m^2.

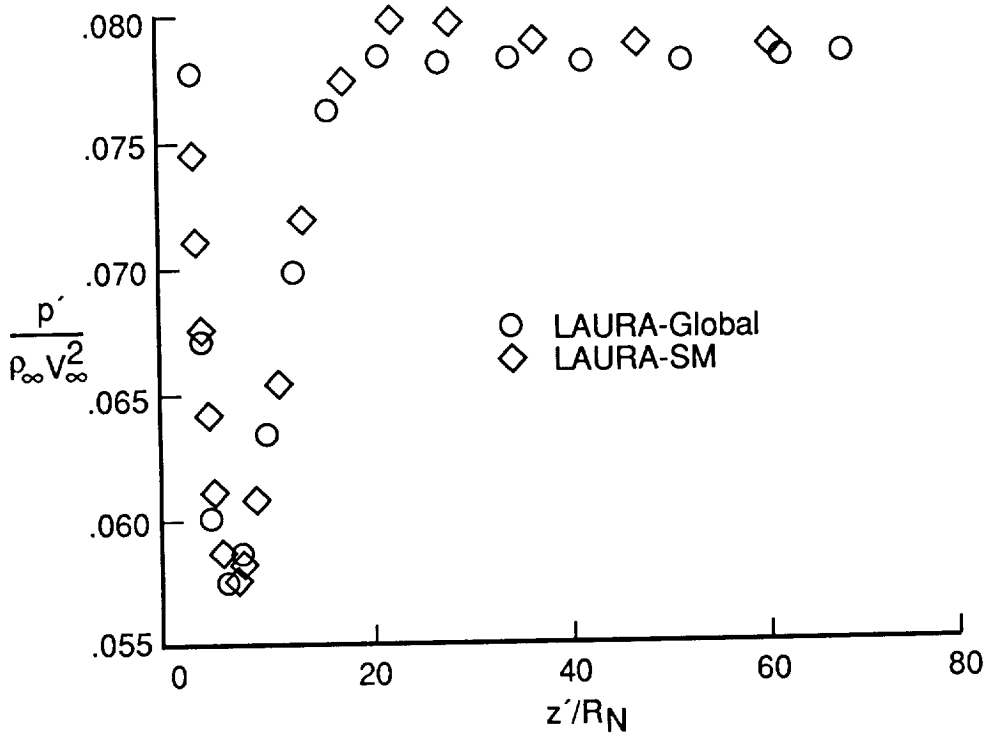


(a) FKT = 1.006.

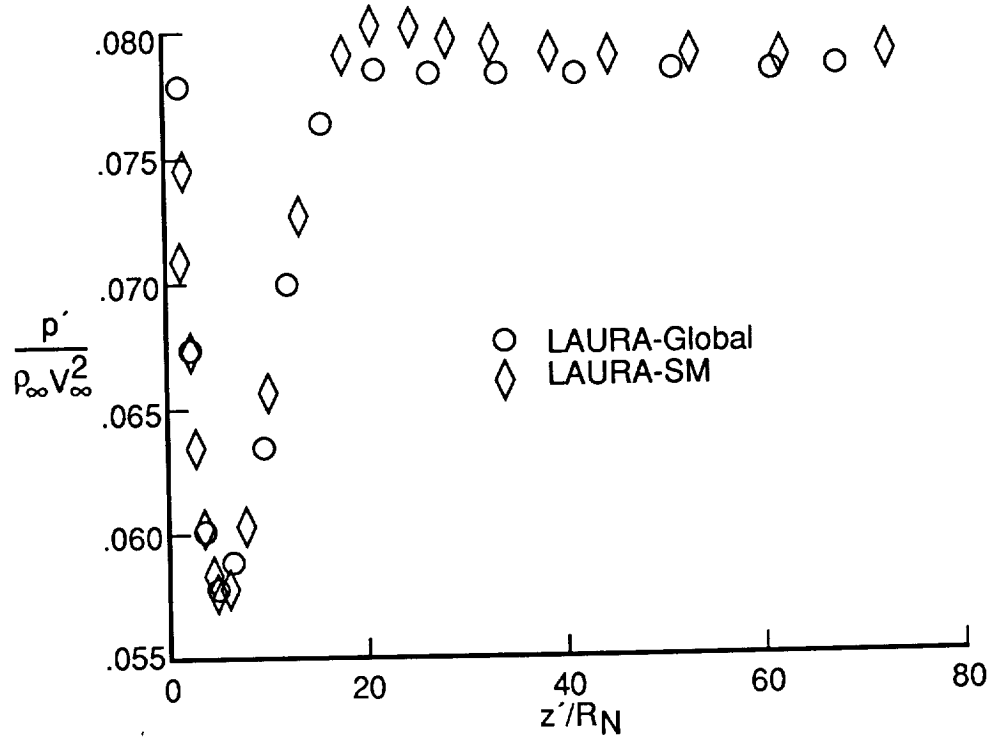


(b) FKT = 1.10.

Figure 10. Effect of step size on surface pressure.

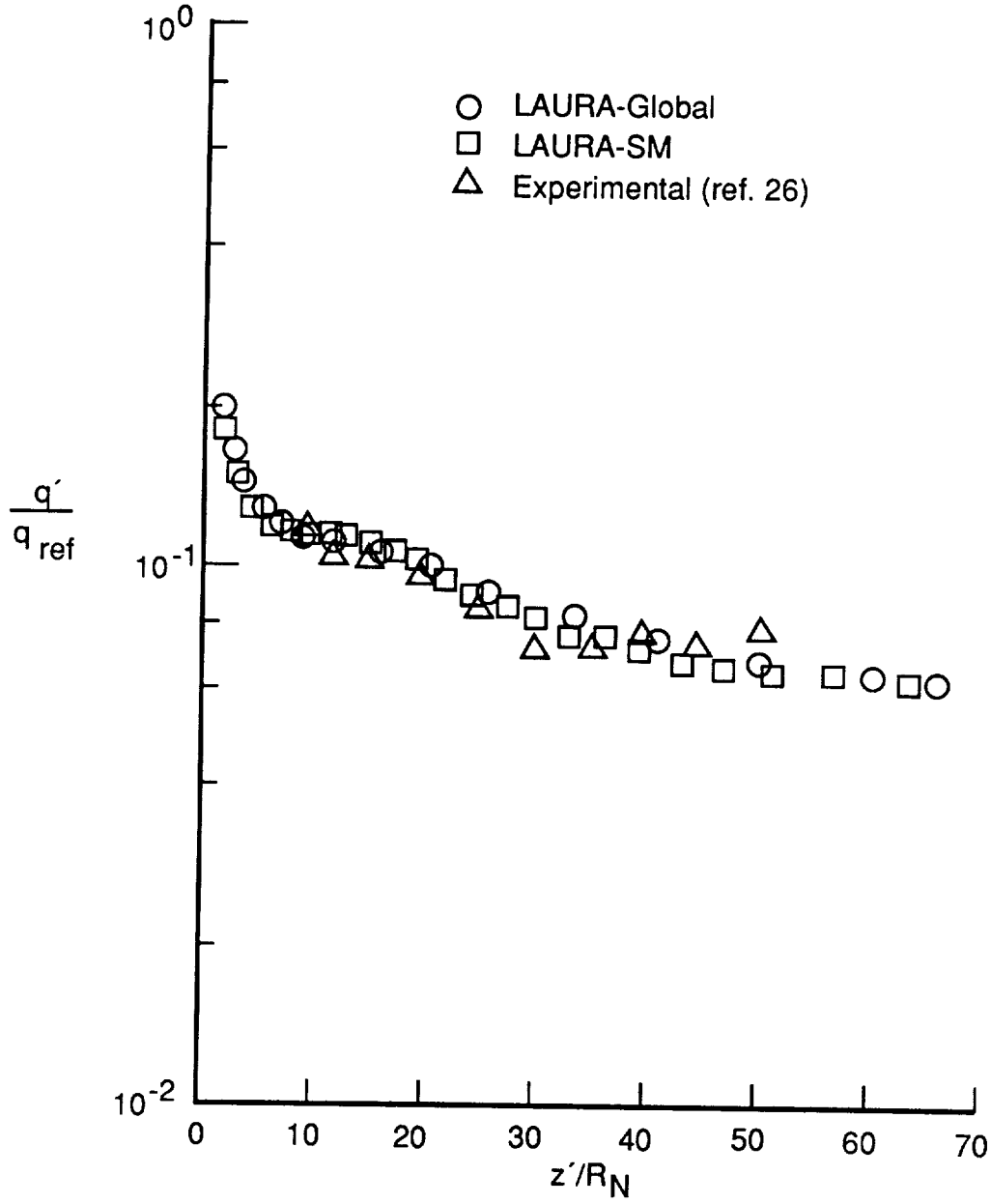


(c) FKT = 1.14.



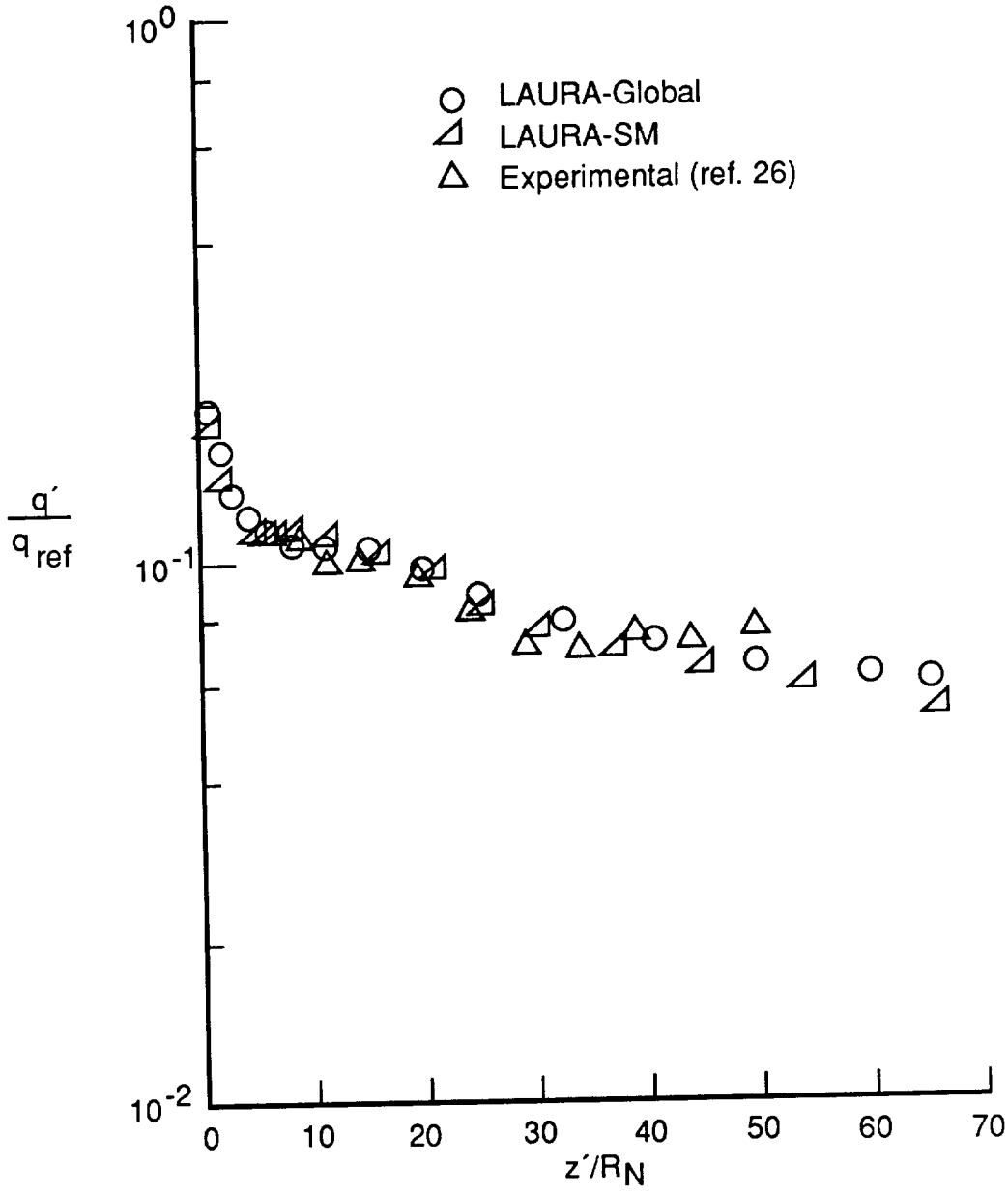
(d) FKT = 1.17.

Figure 10. Concluded.



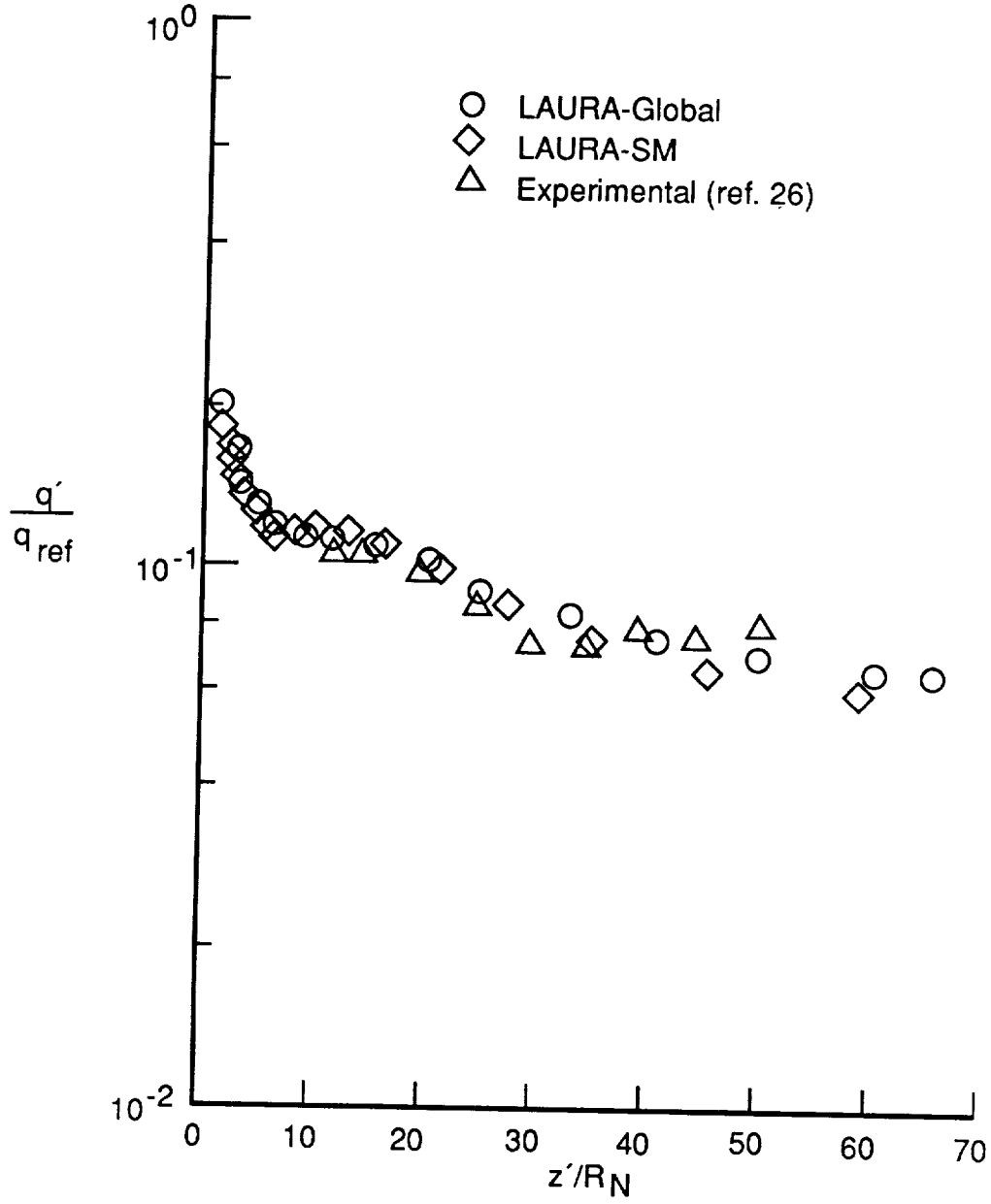
(a) FKT = 1.006.

Figure 11. Effect of step size on surface heating.



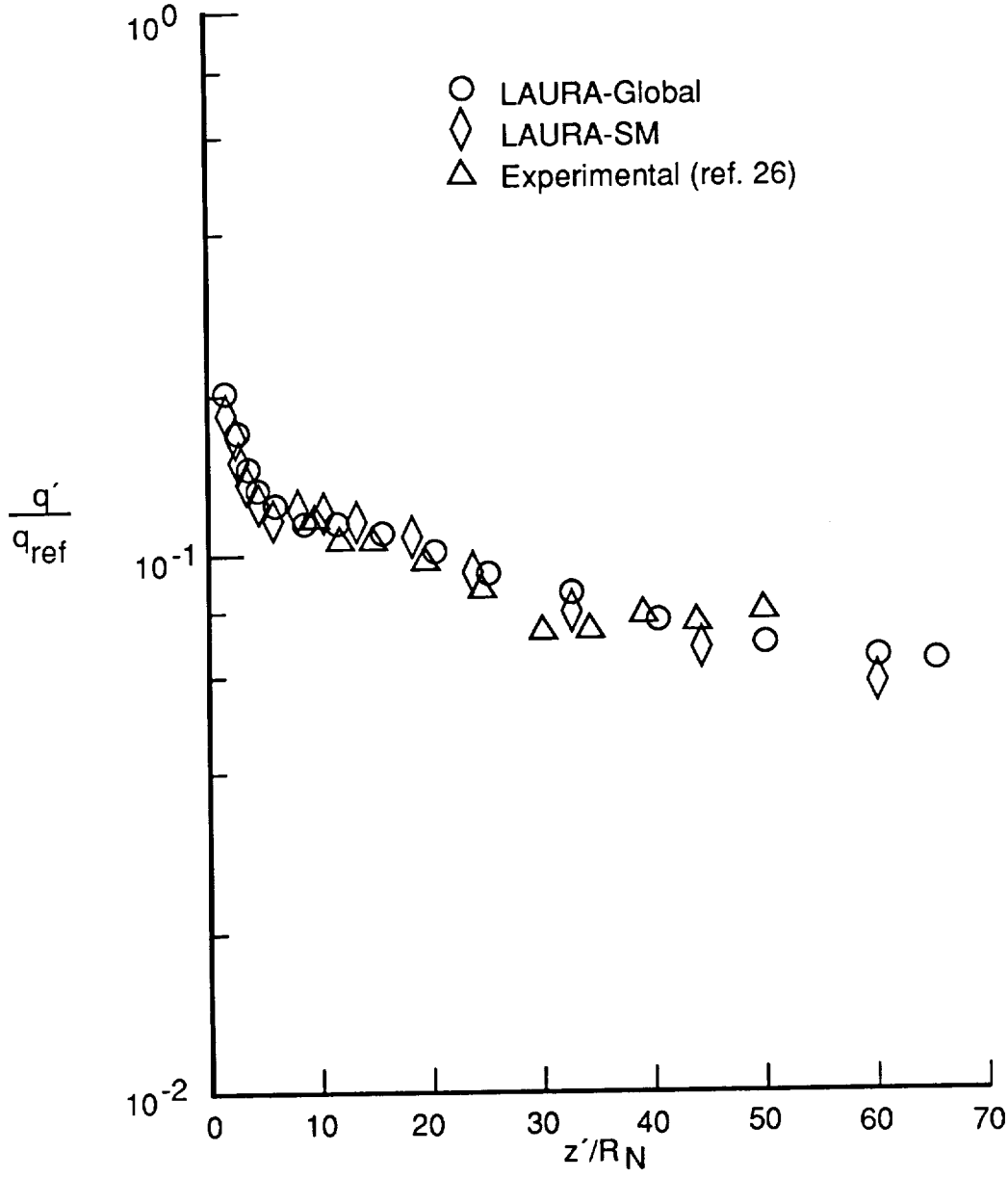
(b) FKT = 1.10.

Figure 11. Continued.



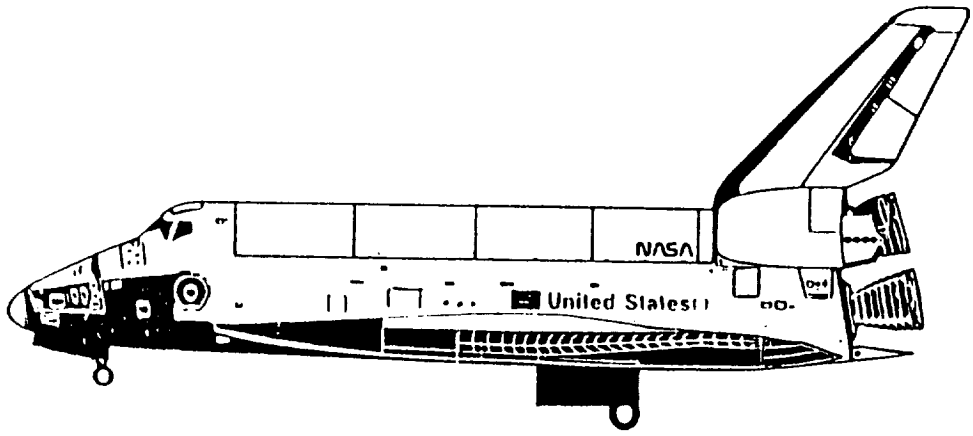
(c) FKT = 1.14.

Figure 11. Continued.

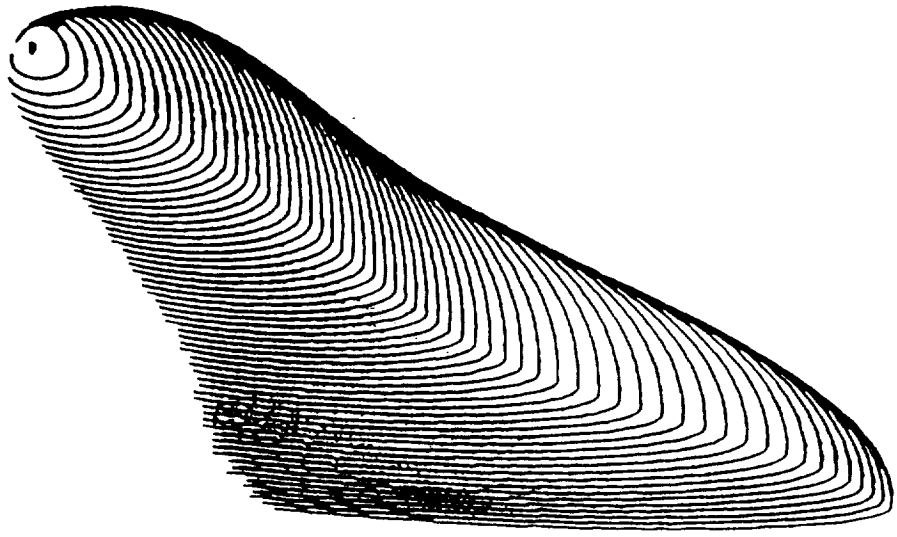


(d) FKT = 1.17.

Figure 11. Concluded.



(a) Physical model.



(b) Computational model.

Figure 12. Shuttle orbiter.

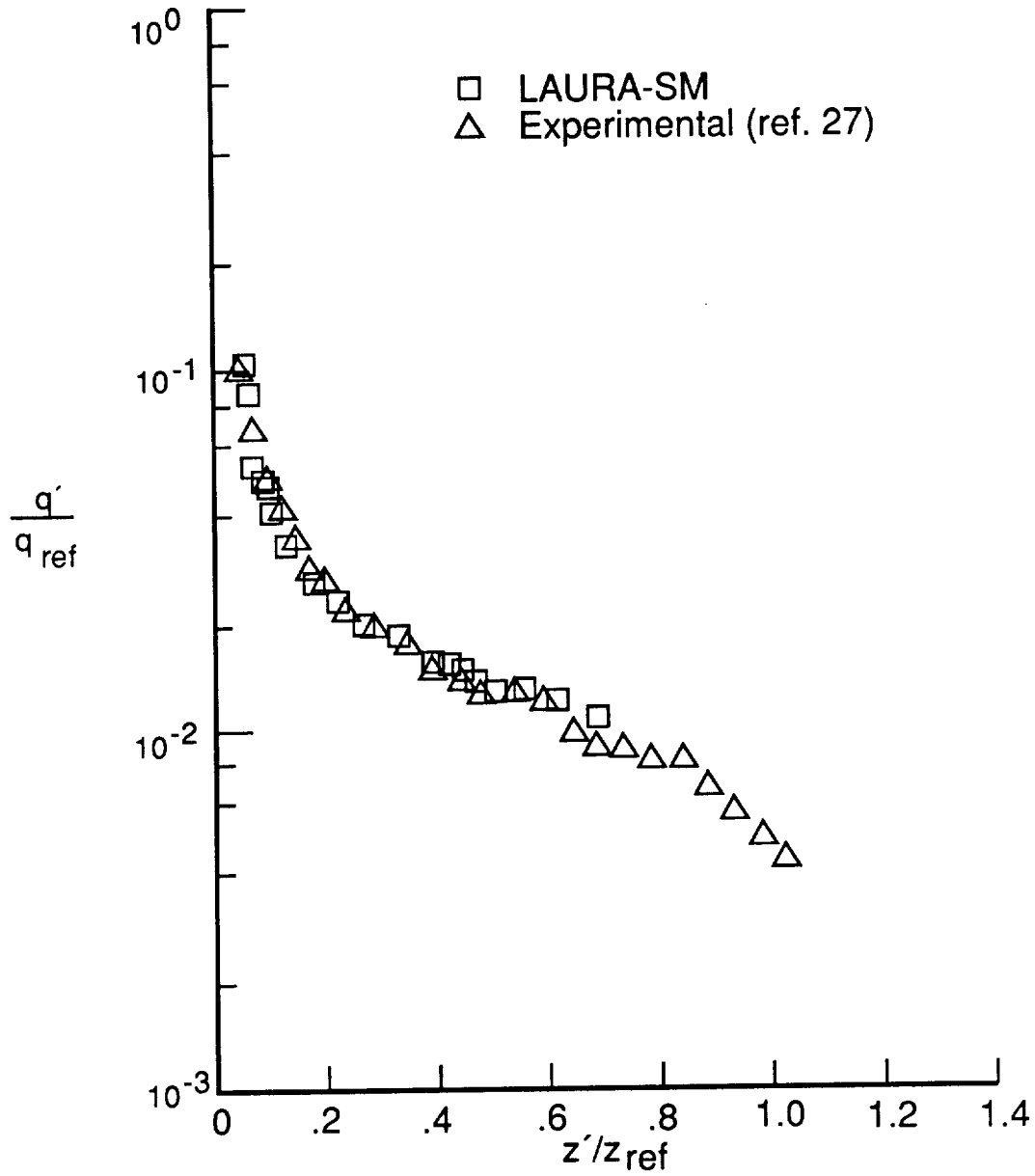
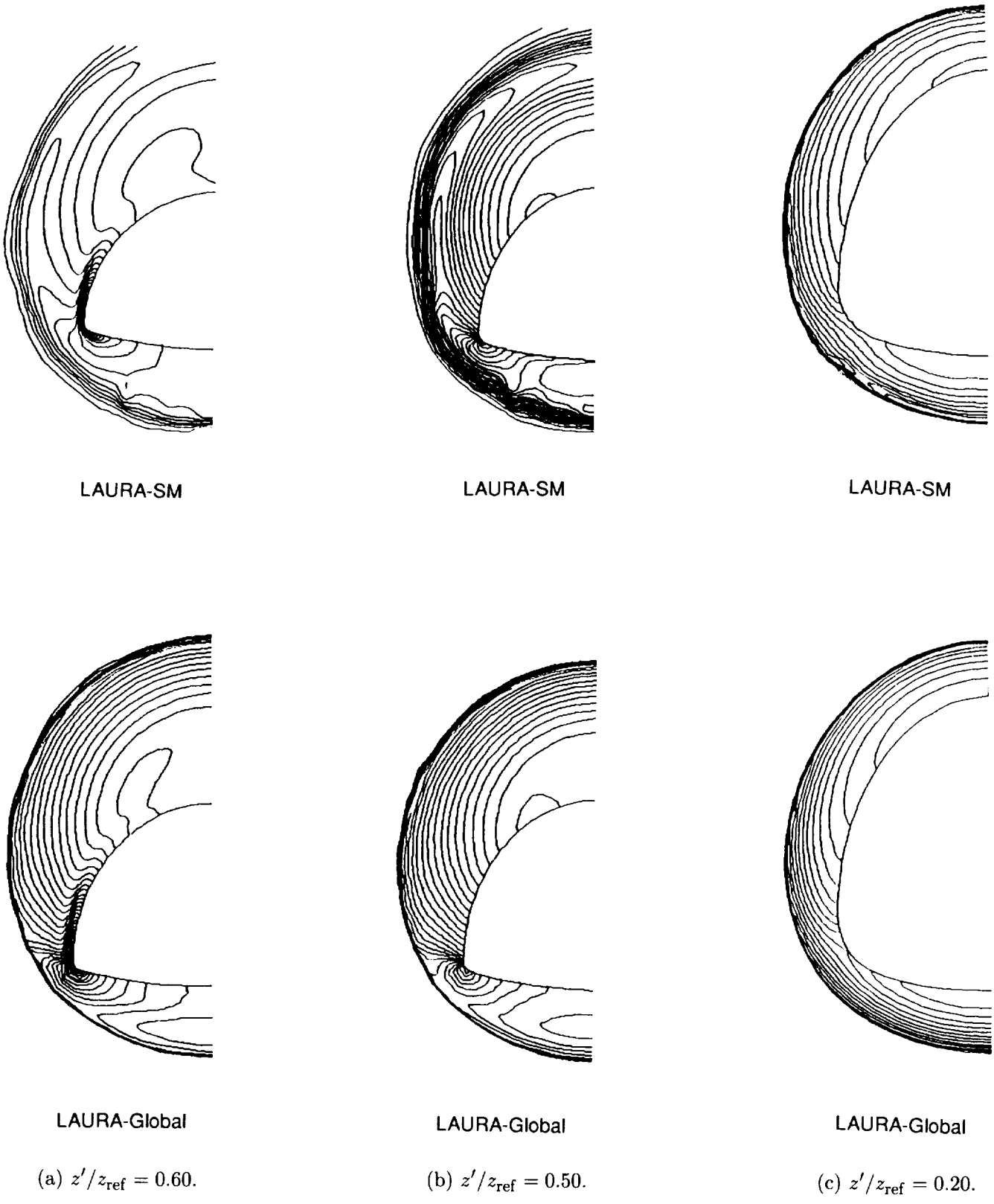


Figure 13. Shuttle in windward symmetry plane surface heating for $\rho'_\infty = 0.00782$ kg/m^3, $S'_\infty = 1410$ m/s, $z_{ref} = 32.66$ m, $\alpha = 10°$, and $q_{ref} = 687$ kW/m^2.



(a) $z'/z_{\text{ref}} = 0.60$.

(b) $z'/z_{\text{ref}} = 0.50$.

(c) $z'/z_{\text{ref}} = 0.20$.

Figure 14. Shuttle pressure contours.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-3068		

4. Title and Subtitle	5. Report Date
An Upwind-Biased Space Marching Algorithm for Supersonic Viscous Flow	March 1991
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Francis A. Greene	L-16788
	10. Work Unit No.
9. Performing Organization Name and Address	506-40-91-01
NASA Langley Research Center Hampton, VA 23665-5225	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Paper
National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The purpose of this work is to document the modifications made to the Langley Aerothermodynamic Upwind Relaxation Algorithm which allow it to compute solutions in a space marching manner. The space marching flux is formulated to be either first- or second-order accurate with Roe's upwind differencing or symmetric total variation diminishing differencing, respectively. The algorithm solves the thin-layer Navier-Stokes equations and is subject to the same flow restrictions as a parabolized Navier-Stokes solver. Each cross-flow plane is locally iterated in pseudo time until converged, then marched in space to the next station. The algorithm is tested on a sphere-cone geometry and a geometry which models the windward surface of the Space shuttle orbiter. Computational results for surface heating are compared with ground-based experimental data. In addition, space marching predictions for surface pressure are compared with values from the original algorithm.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Space marching Hypersonic Thin-layer Navier-Stokes equations	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	43	A03

NASA FORM 1626 OCT 86

NASA-Langley, 1991

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

National Aeronautics and
Space Administration
Code NTT-4

Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, $300

BULK RATE
POSTAGE & FEES PAID
NASA
Permit No. G-27

NASA

POSTMASTER: **If Undeliverable (Section 158 Postal Manual) Do Not Return**