A Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Flowfields

P. A. Gnoffo*
NASA Langley Research Center, Hampton, Virginia

An adaptive grid, finite volume method has been applied to problems of planetary entry in computing complete flowfields. The adaptation algorithm is implicit in nature and is keyed to resolve user-specified gradients. The finite-volume algorithm is explicit, utilizing a maximum time step advancement at each grid point to accelerate convergence to the steady state. The present version of the code is for the laminar flow of a perfect gas. The role of the adaptation algorithm in resolving various features of blunt body/wake flow for planetary entry conditions is emphasized. The algorithm is demonstrated on problems involving massive blowing from the surface of the Galileo probe and moderate blowing from the surface of a sphere.

Nomenclature

c	= adaptation parameter in Eq. (5)
$\tilde{C}_p$	= heat coefficient, m^2/s$^2 \cdot$K
$d_{j,m}$	= array of dependent variables on adapting grid
e	= internal energy nondimensionalized by $\tilde{V}^2_\infty$
$f_{j,m}$	= array of dependent variables on old grid
$\tilde{h}$	= enthalpy, m^2/s^2
i,j	= mesh point indices
I	= number of iterations between rezonings
JN	= number of intervals in j, (η) direction
$\tilde{k}$	= thermal conductivity, W/m$\cdot$K
K_j	= spring constant for jth interval
$K_{\min}$	= adaptation parameter
L_2	= root mean square error norm
$\dot{m}$	= blowing rate, nondimensionalized by $\tilde{\rho}_\infty \tilde{V}_\infty$
M	= Mach number
n_j	= distance to jth mesh point on adapting grid
$\hat{n}_j$	= array equal to n_j at beginning of adaptation iteration
N	= number of passes through smoothing algorithm, Eq. (6)
N_{Nu}	= Nusselt number $\tilde{q}\tilde{R}_N\tilde{C}_{pst}/[(\tilde{h}_{st}-\tilde{h}_w)\tilde{k}_{st}]$
N_R	= pseudo-Reynolds number, $[\tilde{\rho}_{st}\tilde{R}_N(\tilde{h}_{st})^{1/2}/\tilde{\mu}_{st}] \times (\tilde{p}_{st}/\tilde{\rho}_{st}\tilde{h}_{st})^{1/2}$
p	= pressure, nondimensionalized by $\tilde{\rho}_\infty \tilde{V}^2_\infty$
Pr	= Prandtl number
$\tilde{q}$	= heat transfer, W/m^2
Re	= Reynolds number
$\tilde{R}_N$	= nose radius, m
s_j	= distance to jth mesh point on old grid
s	= surface arc length in Figs. 3, 4, and 6, nondimensionalized by $\tilde{R}_N$
Δt	= time increment, nondimensionalized by $\tilde{R}_N/\tilde{V}_\infty$
u,v	= Cartesian velocity components, nondimensionalized by $\tilde{V}_\infty$
$\tilde{V}_\infty$	= freestream velocity
x,y	= Cartesian coordinates
γ	= ratio of specific heats
ξ,η	= computational coordinates
$\tilde{\mu}$	= viscosity, N$\cdot$s/m^2
ρ	= density, nondimensionalized by $\tilde{\rho}_\infty$
Δ_i,Δ_j	= forward differences in the i and j directions, respectively
∇_i,∇_j	= backward differences in the i and j directions, respectively

Subscripts

st	= stagnation conditions
w	= wall
∞	= freestream

Superscripts

$(\ )'$	= interpolated value
$(\tilde{\ })$	= dimensional quantity

Introduction

WHEN generating computational grids for calculating complete flowfields around planetary entry bodies, there are three flow structures that require special consideration—shocks, boundary layers, and free shear layers. Typically, the bow shock is treated either as a discontinuity forming one boundary of a computational plane[1,2] or as captured using a closely aligned grid.[3,4] Internal shocks such as the wake recompression shock are usually captured with somewhat less concern for aligning the grid with the shock.[1,4,5] Laminar and turbulent boundary-layer resolution can be accomplished with grid-stretching functions[6] if given a sufficient number of grid points. However, resolution of free shear layers is considerably more difficult, because their position in physical space is unknown beforehand and because there is no natural computational boundary from which to anchor a stretching function. For example, the free shear layer separating the outer and inner blown gas layers for massive blowing problems lies at some unknown distance between the body and the shock, depending on the blowing rate and distribution. In cases with moderate to no blowing, the boundary layer separates from the rear of the forebody to form the free shear layer between a recirculating region behind the base and the outer supersonic flow. (In Ref. 4, an additional computational plane is embedded in the recirculation region whose boundary points can be used to resolve the free shear layer.) In general, neither the separation point nor the separation angle is known beforehand. One would like to capture these free shear layers in the same sense that one captures shocks. By using properly conservative algorithms[3,7,8] shocks can be captured, preserving the correct jump conditions, by using two to four mesh points. However, the mass, momentum, and energy transfer across the free shear layers is dominated by diffusion (laminar flows) or turbulent mixing (turbulent flows) and consequently requires careful resolution of the gradients through the layer. Because the position of a free shear layer is unknown beforehand and a careful resolution of the free shear layer is required, it

Presented as Paper 82-1018 at the AIAA/ASME Third Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Mo., June 7-11, 1982; submitted June 17, 1982; revision received Dec. 13, 1982. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Aero-Space Technologist, Aerothermodynamics Branch, Space Systems Division.

becomes clear that a solution-adaptive algorithm to move the grid is highly desirable for efficient computation.

Solution-adaptive algorithms have been developed and demonstrated for problems in surface heat transfer[9] and shock tracking.[10] The present algorithm has evolved from concepts presented in these works as discussed in Ref. 11. The material presented herein deals with the application of that solution-adaptive finite volume algorithm to problems in planetary entry, focusing in particular on resolution of free shear layers.

The general problem of adapting a grid to high-gradient regions that do not lie along any computational boundary is complicated by the fact that the computed gradients are themselves a function of the grid being moved. As the solution is generated, transients in the solution can appear, needlessly distorting the grid and causing grid-induced errors that interact with and exacerbate the solution being generated until the calculations become unstable. Observations of this type of behavior are presented and the safeguards incorporated against this behavior in the adaptation algorithm are discussed.

Finite Volume Formulation

In the present finite volume formulation[11] (FVF) the integral form of the governing conservation laws is approximated on cells whose corners are defined by the position of grid points in physical space. Each cell is indexed the same as the lower left-hand corner point in computational space (Fig. 1). Mass, momentum, and energy flux through the $(i,j; i,j+1)$ wall are calculated using information from the (i,j) point on the predictor step and the $(i,j+1)$ point on the corrector step. Similarly, flux through the $(i,j; i+1,j)$ wall are calculated using information from the (i,j) point on the predictor step and the $(i+1,j)$ point on the corrector step. The integrated changes of properties in the $[i,j]$ cell are then applied to the (i,j) point on the predictor step and the $(i+1, j+1)$ point on the corrector step. Consequently, for a two-dimensional system of governing equations in the form

$$\iiint_{\Omega} \frac{\partial q}{\partial t}\,\mathrm{d(volume)} + \iint_{\delta\Omega} (fi+gj)\cdot n\,\mathrm{d(surface)} = 0$$

where vector $\boldsymbol{n}$ is a unit outward normal from the surface $\delta\Omega$ of a finite-control volume Ω and $\boldsymbol{i}$ and $\boldsymbol{j}$ are unit vectors in the x and y directions, respectively, the FVF is written as

$$(\overline{Q_{ij}^{n+1}} - Q_{ij}^{n})A_{ij}/\Delta t_{ij}$$

$$= -\{\Delta_i[F_{ij}\Delta_j y_{ij} - G_{ij}\Delta_j x_{ij}] + \Delta_j[G_{ij}\Delta_i x_{ij} - F_{ij}\Delta_i y_{ij}]\}^n$$

$$(\overline{\overline{Q_{i+1,j+1}^{n+1}}} - Q_{i+1,j+1}^{n})A_{ij}/\Delta t_{i+1,j+1}$$

$$= -\{\nabla_i[F_{i+1,j+1}\nabla_j y_{i+1,j+1} - G_{i+1,j+1}\nabla_j x_{i+1,j+1}]$$

$$+ \nabla_j[G_{i+1,j+1}\nabla_i x_{i+1,j+1} - F_{i+1,j+1}\nabla_i y_{i+1,j+1}]\}^{\overline{n+1}}$$

$$Q_{ij}^{n+1} = (\overline{Q_{ij}^{n+1}} + \overline{\overline{Q_{ij}^{n+1}}})/2$$

This relation assumes that the $x_{i,j}$ and $y_{i,j}$ coordinates of the grid points defining the control volumes are not functions of time. This is consistent with the rezoning technique described in the section on grid generation. The relation can be expanded to include the effects of grid motion for problems that use time-dependent grid adaption or require temporal accuracy. The vectors q, f, and g for the Navier-Stokes equations are

$$q = (\rho, \rho u, \rho v, \rho E)$$

$$f = [\rho u, p + \rho u^2 - \sigma_{xx}, \rho uv - \tau_{xy}, u(\rho E + p - \sigma_{xx}) - v\tau_{xy} - q_x]$$

$$g = [\rho v, \rho uv - \tau_{xy}, p + \rho v^2 - \sigma_{yy}, v(\rho E + p - \sigma_{yy}) - u\tau_{xy} - q_y]$$

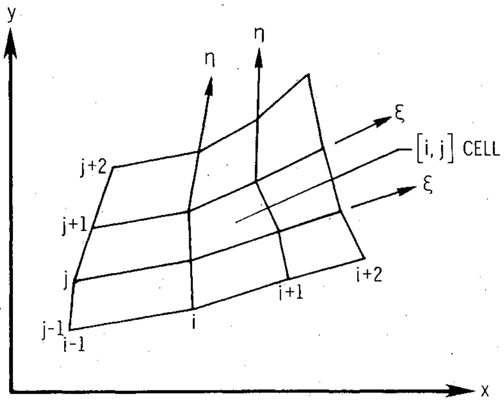

Fig. 1 Schematic showing relation of indexing system, physical (x,y) coordinates, and computational (ξ,η) coordinates.

The vectors Q_{ij}, F_{ij}, and G_{ij} are the net function approximations to q, f, and g at the (ij) grid point. The shear stresses σ_{xx}, σ_{yy}, τ_{xy}, and heat transfer q_x, q_y are approximated using second-order accurate central differences. For axisymmetric flow, the cell volumes and surface areas are determined by multiplying the area of the two-dimensional cell and the lengths of the two-dimensional cell walls by the respective distances of their centroids from the axis. A source term is also introduced on the right-hand side of the FVF equal to $H_{i,j}^{n} A_{i,j}$ (predictor step) and $\overline{H_{i,j}^{n+1}}$ $A_{i-1,\ j-1}$ (corrector step), where $H_{i,j} = [0, 0, (p - \sigma_{\theta\theta})_{i,j}, 0]$.

The scheme reduces exactly to MacCormack's method[12] on a rectangular grid. The algorithm vectorizes very easily with a minimum of temporary long-vector storage locations required. It is also strictly conservative in the sense of summation without residue[13] because of the manner in which flux passing through cell walls is treated.

On some problems it was found that instabilities would result if the predictor-corrector pattern described above were not alternated between opposite corners of the cell on even and odd time steps. In the Galileo probe calculations to be described, it was found that some stability problems, which would manifest themselves near the expansion corner of the probe or behind the lip just above the free shear layer, could be overcome by cycling the predictor-corrector pattern across all four corners of the cell.

The formulation yields a consistent approximation to the governing differential equations that is second-order accurate in space and second-order accurate in time if a constant time step is used. However, the solution is usually advanced at each grid point by a local maximum time step (multiplied by a safety factor) in order to accelerate convergence to a steady state. This treatment introduces a first-order (Δt) error in the wave speed proportional to the gradient of $\Delta t(x,y)/\Delta t_{i,j}$. Consequently, when implementing the adaptation routine, checks must be implemented to prevent severe grid stretching or skewness that could cause large gradients in the locally evaluated time steps. Excessive stretching or skewness can also introduce large factors of second-order spatial errors. Some of the methods used to control these errors are discussed in the section on grid adaptation. Further details of the finite-volume formulation used herein, including specifics on the difference equations, are given in Ref. 11.

Initialization

Lines of constant j index (ξ coordinates) define the shock and the body/wake centerline. Lines of constant i index (η coordinates) run between the shock and the body/wake centerline. A typical grid is shown in Fig. 2. The grid can be mapped onto a rectangular computational plane, which simplifies programming on a vector-processing computer.

In the first set of solutions to be presented on the Galileo probe, the body is divided into a finite number of segments

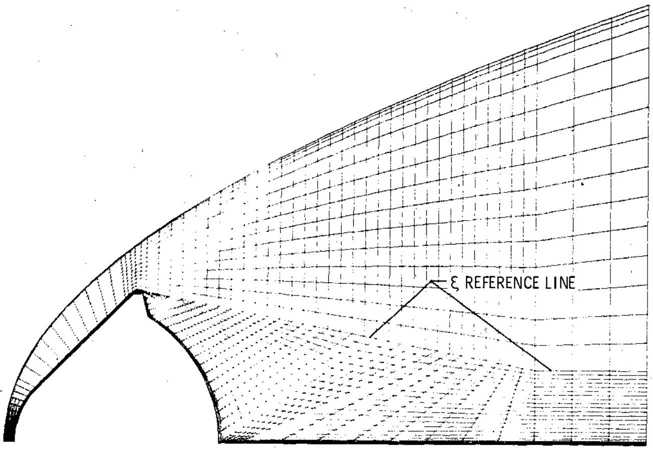

Fig. 2 Initial grid used for calculations over Galileo probe.

consisting of straight lines and circular arcs. The user specifies the number of points on each segment, which are then distributed evenly along that segment. A reference ξ coordinate line is set a constant distance away from the forebody and is composed of straight line segments in the wake as shown in Fig. 2. Straight η coordinate lines running between the body and the reference line are normal to the forebody. A modified hyperbola[5] at a user-specified standoff distance initializes the shock and the straight lines emanating from the reference line are used to complete the grid.

In the latest set of calculations a polynomial-blending function[14] is used to provide a smoother variation of arc length along the ξ coordinate lines from one body segment to another. The reference line is omitted and the straight lines emanating from the body to the shock at specified angles are used to complete the grid. The generation of severely skewed grids is far less of a problem when adapting along straight lines as opposed to lines with large curvature or sharp bends, as occurred in the wake with the previous grids.

Pressure on a ξ reference coordinate line is calculated using Newtonian theory. When $p_{i,j\mathrm{ref}} < p_\infty$, the pressure is set equal to p_∞. Internal energy e on the reference line equals $\cos^2(\theta) e_{st}$ where $\cos^2(\theta) = (\partial y/\partial\xi)^2 / [(\partial y/\partial\xi)^2 + (\partial x/\partial\xi)^2]_{i,j\mathrm{ref}}$. When $e_{i,j\mathrm{ref}} < 0.7 e_{st}$, the internal energy is set equal to $0.7 e_{st}$. With p,e known, density is obtained from the equation of state. With p,ρ,e known, velocity squared V^2 is obtained from a specification of constant total enthalpy equal to H_∞. The velocity components are then defined as

$$v = \cos(\theta) V, \quad u = \sin(\theta) V$$

Conditions at the shock are obtained from the Rankine-Hugoniot equations. Conditions at the body are set as

$$\rho_{i,1} = \rho_{i,j\mathrm{ref}}, \quad e_{i,1} = e_{i,j\mathrm{ref}}, \quad u_{i,1} = v_{i,1} = 0$$

Values for ρ,u,v,e in the interior of the domain are obtained using linear interpolation in η between the shock and the reference line or between the reference line and the body.

Once a solution has been obtained, new calculations can be performed at different freestream conditions using the old solution as the initial condition.

Grid Generation

Overview

The adaptation algorithm described in the next section is designed to act as a self-contained subroutine that can be added to existing codes with minimal modification of the host code. It is based on an equivalent spring analogy where springs connect adjacent mesh points and spring constants are a function of the gradient of some property f between the grid points. While this adaptation technique can be applied in two-dimensions,[11] the algorithm is greatly simplified if its adaptation is restricted to one coordinate direction, in which form it is related to the method of Ref. 9. The resolution of some property $f(x,y)$ can be accomplished by adapting along only one coordinate direction, provided the coordinate direction is not significantly oblique to the gradient vector ∇f. Consequently, the adaptation algorithm requires a rectangular ordering of the grid points and a flow situation in which the high-gradient regions can be resolved with grid points free to move in only one computational coordinate direction. The routine has several adaptation parameters that are specified by the user to control the grid distribution. The necessity of some free parameters to control adaptation will be explained and the guidelines for their definition will be provided subsequently.

Adaption Algorithm

Let s_j be the length of an η coordinate line from $j=1$ to the jth mesh point as defined below,

$$s_1 = 0 \quad s_j = \sum_{k=2}^{j} [(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2]^{1/2} \tag{1}$$

Let $f_{j,m}$ be an array of dependent variables at the point s_j (i.e., $f_{j,1} = x_j$, $f_{j,2} = y_j$, $f_{j,3} = \rho_j$, $f_{j,4} = u_j$, $f_{j,5} = v_j$, and $f_{j,6} = e_j$). The grid points in physical (x,y) space are mapped onto a line in s space, with the jth point connected to $j+1$ point by a spring with spring constant K_j. Let Δn_j be the distance between the j and $j+1$ points in s space. Then assuming equilibrium with endpoints fixed and spring tension equal to $K_j \Delta n_j$ we can write

$$\Delta n_j = K_1 \Delta n_1 / K_j \tag{2}$$

The known total length of s can be expressed as

$$s_{JN+1} = \sum_{j=1}^{JN} \Delta n_j = \Delta n_1 K_1 \sum_{j=1}^{JN} \frac{1}{K_j} = \Delta n_1 K_1 \mathrm{SUM} \tag{3}$$

Consequently, from Eqs. (2) and (3) one obtains

$$\Delta n_j = s_{JN+1} / (K_j \mathrm{SUM}) \tag{4}$$

In the work presented herein, K_j is defined by

$$K_j = 1 + c\,|d(n_{j+1}) - d(n_j)| / \Delta n_j \tag{5}$$

where d is some function being monitored on the adapting grid. The adaptation algorithm is then implemented as follows:

1) Start at first column of data. Compute and store all values of s_j and $f_{j,m}$.

2) Initialize n_j and $\bar{n}_j$ equal to s_j. Set ITER = 0.

3) ITER = ITER + 1.

4) Using the known stored values of s_j and $f_{j,m}$, compute the value of dependent variables $d_{j,m}$ at all n_j using a univariate interpolation routine where $d_{j,1} = x_j'$, $d_{j,2} = y_j'$, etc.

5) Compute $K_j = K_j\ (d_{j,m})$ from Eq. (5).

6) (Optional) If $K_j < K_{j\min}$, then set $K_j = K_{j\min}$ where $K_{j\min}$ is defined in section on adaptation parameters.

7) (Optional) Smooth the resulting spring constants using a filtering algorithm

$$K_j^{n+1} = (K_{j-1}^n + 2K_j^n + K_{j+1}^n)/4 \tag{6}$$

8) Compute new values of n_j from Eq. (4). Let $n_{JN+1} = s_{JN+1}$ to keep the endpoint value constant.

9) (Optional) If the convergence is slow (greater than 10 passes through adaptation algorithm), reset $n_j = (n_j + n_j)/2$.

10) Compute the error norm to determine convergence,

$$L_2=\left[\sum_{j=2}^{JN}(n_j-n_j)^2\right]^{1/2} \tag{7}$$

If $L_2<0.001$ or if ITER >200, proceed to step 9.

11) Set $n_j=n_j$ and go back to step 3.

12) Replace x_j, y_j, etc., with $d_{j,1}$, $d_{j,2}$, etc., and go to next column of data.

Problems and Remedies

Before proceeding to the next section on adaptation parameters, some comments on the somewhat "ad hoc" nature of parameter definition and smoothing routines (i.e., steps 7 and 9) are in order. In an ideal situation, one would like to minimize both the maximum norm and the Euclidean norm of some user-specified approximation to the truncation error in the solution by adjusting the grid points. If a good approximation to the total truncation error were employed, the user would not need to worry, for example, about significant grid skewness or too many points brought into a high-gradient region (which slows convergence because of the excessively small spatial resolution). In the approach taken here, truncation error, per se, is not evaluated and only one degree of adaptive freedom is permitted. Therefore, excessive grid stretching or skewness are allowed by the adaptation routine. The adaptation parameters and smoothing options give the user more control of these problems over a wide variety of conditions. While numerical experimentation may be required to define the parameters, it can be accomplished with little difficulty or expense at an interactive terminal with graphics capability. The following observations of potential problem areas and remedies employed are offered as a guide in this process.

In many cases the algorithm will converge within 10 iterations. If it has not converged within 10 iterations, the damping operation in step 9 has been implemented with great success. The smoothing option in step 7 is used to avoid excessive grid stretching and to enhance convergence. It can be cycled through N times before proceeding to step 8 in order to provide a smooth variation of stiffness from spring to spring, providing a smoother transition of grid size from high- to low-gradient regions. It also reduces the problem of mathematical stiffness because a wide disparity in the magnitudes of K_j can cause significant error due to roundoff in the evaluation of SUM in Eq. (3). (All results presented herein used single precision.)

Finally, it is noted that, even with these damping and smoothing algorithms, a situation will occasionally arise where some error norm will be attained; however, the solution will never converge below the L_2 norm requirement specified in the code. In these situations, the grid distribution after 200 iterations is accepted as the final result under the assumption that the main job of redistributing grid points to resolve high-gradient regions has been accomplished and the implementation of more damping with more iterations through the algorithm serve no useful purpose. When that assumption is wrong, experience to date shows that grid-induced errors will introduce instabilities, causing divergence of the solution and thus telling the user to reconsider the various parameters chosen in the adaptation algorithm.

Adaptation Parameters

The user-specified adaptation parameters are the functions to be resolved (i.e., temperature, velocity, etc.): the coefficient c in Eq. (5), the number of passes N through the smoothing algorithm in Eq. (6), the number of iterations I, of the finite volume formulation between rezonings, and a spring stiffness parameter $K_{\min}$. The parameter $K_{\min}$ is used to focus grid points near the wall, even if the gradient near the wall is not particularly strong relative to gradients elsewhere along the η coordinate line. It is applied in step 6 of the adaptation algorithm where $K_{j\min}=aK_{\min}$ where $a=1$ if $j<5$ and $a=1-(j-4)/6$ if $j>4$. The algorithm will work with low values of $K_{\min}$ ($1<K_{\min}<5$), but in the cases considered here it has been found that surface heat transfer may be underpredicted unless a value of $K_{\min}$ on the order of the maximum value of K_j encountered globally is specified. Typical values of c vary between 0.1 and 5, with large values of c tending to pull more points into high-gradient regions with greater grid stretching. The value of N is usually between ⅛ and ¼ of the number of grid points, $JN+1$, in the η direction. The value of I varies between 20 and several thousand with a typical value of 200. A very convenient feature of this adaptation algorithm is that it does not have to be run interactively with the flowfield code. Consequently, the user can perform parametric studies on the adaptation algorithm to assess the effects of monitoring various gradients and changing the constants c, N, and $K_{\min}$. The user can also turn off the adaptation routine if the grid is satisfactory, thus saving computational costs.

It should be clear that the effect of any one parameter also depends on the values of the other parameters. For example, small values of I and N and large values of c can cause the grid to focus on initially innocuous looking gradients caused by transient phenomena or grid nonuniformities. When this happens the interaction of grid-induced errors with the adaptation algorithm further exacerbates the problem. Values of I on the order of 100 allow the transients caused by a change in the grid to smooth out before the next rezoning is attempted. The smoothing routine [Eq. (6)] also reduces this problem, although sometimes at the expense of smoothing out the grid to such a extent that gradients are not properly resolved.

Results and Discussion

All of the calculations to be presented are for the laminar flow of a perfect gas, $\gamma=1.4$, $Pr=0.72$. Sutherlands's law is used to calculate viscosities. The first set of calculations for flow over a sphere is not particularly interesting from the point of view of grid adaptation, but does serve to demonstrate the accuracy of the method through comparisons with experiment and another calculation., The second set of calculations on the Galileo probe demonstrates the capabilities of the adaptation algorithm.

Sphere with Zero to Moderate Blowing

The flowfield over a sphere with $M_\infty=6$ and $Re_\infty=264{,}000$ was obtained with a grid of 45 points around the body, encompassing a 120 deg arc from the stagnation point, and 41 points between the body and the shock. Average run time for these cases was 3.75 min for 6000 iterations on the CYBER 203 computer. Very good comparisons of pressure distributions were obtained with experimental data of Ref. 15 and calculations of Ref. 2. Fair comparisons of heat transfer were obtained, as presented in Figs. 3 and 4 for blowing rates of $\dot{m}_{st}=0$ and 0.02406 flowing from a porous plug of radius 0.166 R_N. Wall temperature was a constant with

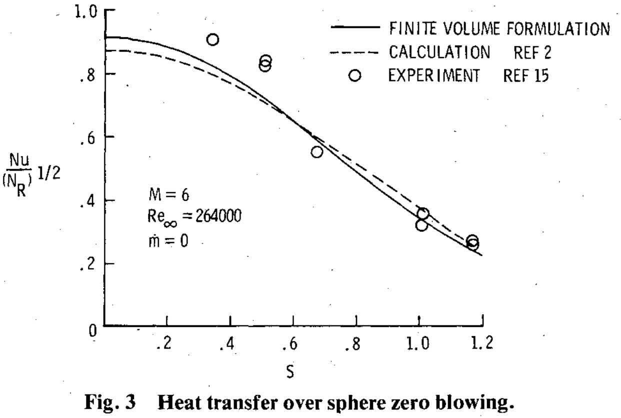

Fig. 3 Heat transfer over sphere zero blowing.

$T_w/T_\infty = 2.945$. A large value of $K_{\min}$ was used to bring in more points near the wall to better resolve the temperature gradient with the given grid resources. When resolution of boundary layers is a problem, increasing $K_{\min}$ as opposed to c is a quicker and more reliable fix for adjusting the grid.

Galileo Probe

Flowfields over the Galileo probe, a 45 deg spherically capped cone with a hemisphere afterbody, for $M_\infty = 50$ and $Re_\infty = 500{,}000$ and 100,000 are presented for adiabatic and constant wall temperature boundary conditions and for blowing rates of $\dot{m}_{st} = 0$ and 0.5. The first case studied was for an adiabatic wall, $Re_\infty = 500{,}000$, on a grid of 61 points around the body/wake centerline and 41 points between the body and shock. A converged grid for this case using $K_{\min} = 2$, $c = 0.2$, and Mach number as the function being monitored, is presented in Fig. 5. Some concentration of coordinate lines in the free shear layer just behind the probe lip is evident. The ξ coordinate lines leave the boundary layer and fan out to fill in the recirculation region and free shear layer. Pressure distribution around the body is presented in Fig. 6 along with comparisons to the results of Ref. 4 obtained at a lower Reynolds number. Attempts to run at higher Reynolds numbers or at higher values of c to pull more points in the shear layer caused numerical instabilities just behind the probe lip. Note, for example, the excessive skewness of the grid in Fig. 7 computed with $c \approx 1$. While the free shear layer is resolved in much finer detail, the excessive stretching around the η coordinate bend associated with the reference coordinate line caused large grid skewness. This skewness introduced grid-induced errors that ultimately caused the calculation to diverge. These results indicate that 41 points are, at best, marginally sufficient for resolving all of the high-gradient regions between the body and shock, even with the adaptation capability. The ability to obtain a converged solution for this

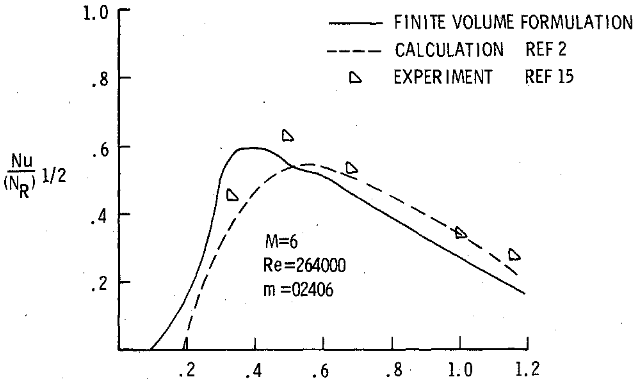

Fig. 4 Heat transfer over sphere with blowing.

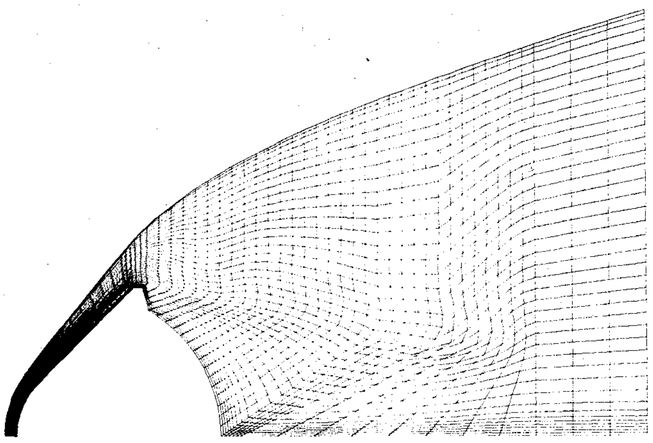

Fig. 5 Converged grid used in Galileo probe flowfield calculation.

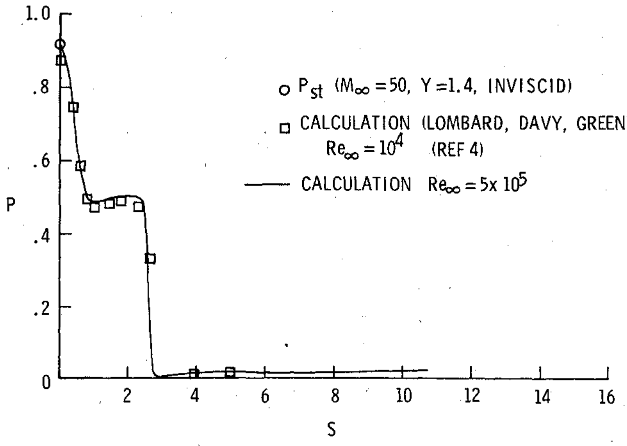

Fig. 6 Pressure distribution around Galileo probe and downwake centerline.

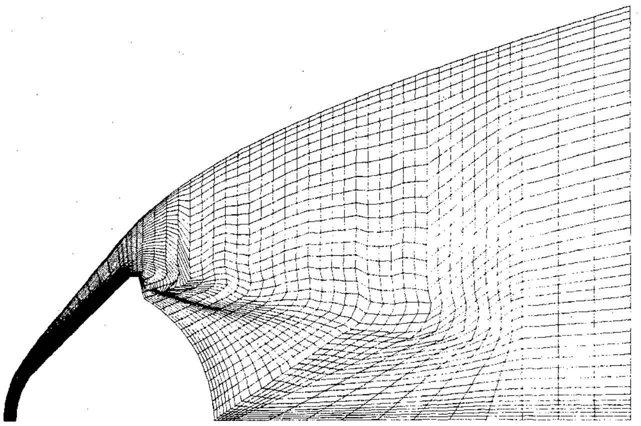

Fig. 7 Grid distribution over Galileo probe with excessive stretching through free shear layer.

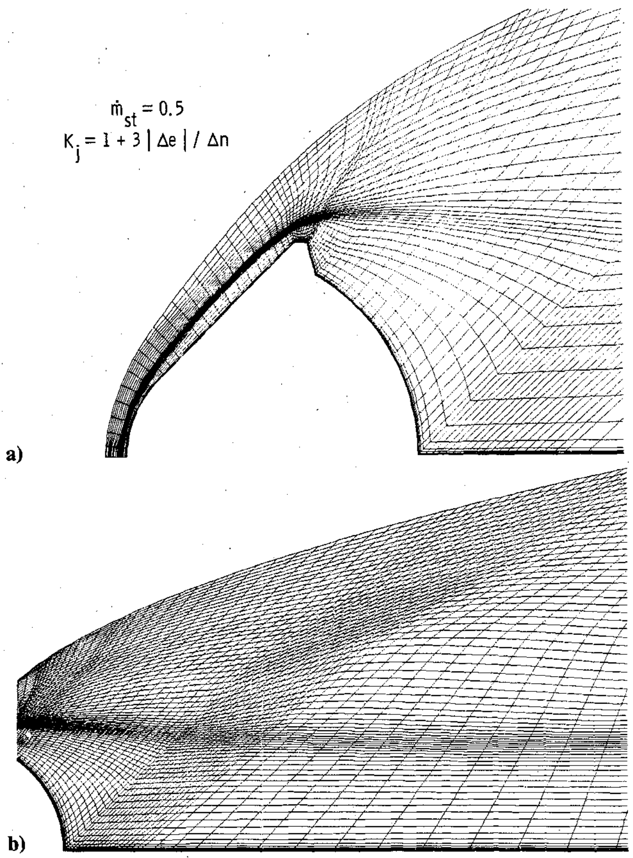

Fig. 8 Grid distribution over Galileo probe with massive blowing: a) forebody (odd number mesh lines), b) wake.

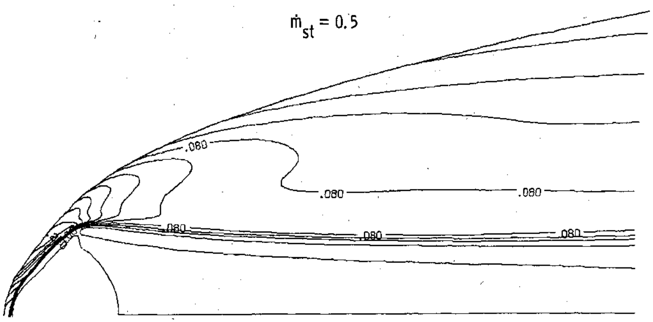

Fig. 9 Global internal energy contours over Galileo probe with massive blowing: $M_\infty = 50$, $Re_\infty = 10^5$, $\gamma = 1.4$.

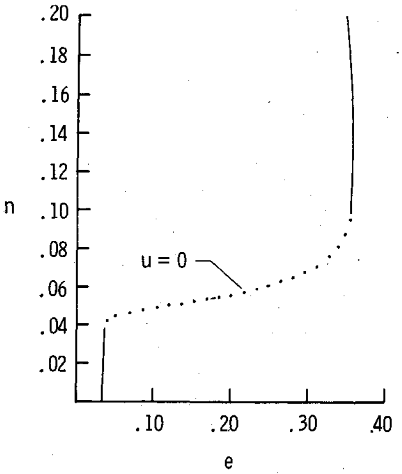

Fig. 10 Internal energy distribution along stagnation streamline across shock layer showing odd-number grid point location through high-gradient region.

problem using the given adaptation parameters and grid resources demonstrates the usefulness of this program.

The next set of calculations is for a cold-wall condition $T_w/T_\infty = 50$, $Re_\infty = 100{,}000$, and $\dot{m} = 0.5$ on a grid of 100 points in the ξ direction and 81 points in the η direction. The blowing distribution along the body surface is one taken from Ref. 16 (at 51.5 s) along the forebody with a constant blowing rate of $\dot{m} = 0.04$ along the base. The converged grid for this case is presented in Fig. 8. The adaptation parameters are $c = 3$, $K_{\min} = 100$, and $N = 20$ using internal energy e as the function being monitored. The high-density, low-velocity, low-temperature blown gas layer displaces the outer high-temperature shock-layer flow. The concentration of grid points in this high-temperature gradient region extending into the wake is obvious in Fig. 8. The inner layer flow accelerates around the corner of the body into the low-pressure region behind the base. Over the course of 9000 iterations the recirculation region (zero blowing solution) was washed out from behind the base, being filled in by a relatively uniform, high-density ($\rho \approx 1.4$), low-velocity flow with no recirculation. (Typical runs were made in blocks of 3000 iterations requiring 7.56 min of computer time on the CYBER 203 computer.)

Global internal energy contours in Fig. 9 again show the high-gradient region, while a plot of internal energy across the shock layer on the stagnation streamline (Fig. 10) shows that approximately 40 points were brought in by the adaptation algorithm to resolve the high-gradient region.

Conclusions

The finite-volume formulation used herein gives a consistent and conservative approximation to the Navier-Stokes equations. The present formulation is second-order accurate in space and has a first-order error in wave speed, proportional to the local value of Δt, when the local maximum time step is used to advance the solution at every point. Good comparisons with experimental data and other calculations have been obtained using this formulation.

The adaptation algorithm presented herein is based on a simple spring analogy. The one-dimensional adaptation variation has been shown to be capable of resolving the free shear layers in the flowfield around a typical planetary probe. The algorithm is very easy to apply and is versatile enough to permit dynamic adaptation (i.e., once every 20 iterations), large-scale rezonings after several thousand iterations, or parametric studies run noninteractively to study the effects on the grid of changing an adaptation parameter in a particular flowfield. Suggested ranges of adaptation parameters have been provided that avoid the problems associated with grid-induced errors interacting with the adaptation process and eventually destroying a calculation. In the massive blowing problem, one-half of the available grid points were pulled in by the adaptation algorithm to resolve the narrow temperature gradient region between the outer and inner blown gas layers in the stagnation region.

Acknowledgment

The present work is part of a dissertation recently completed under the direction of Prof. S. I. Cheng at Princeton University.

References

[1] Weilmuenster, K. J. and Graves, R. A. Jr., "Viscous Flowfield Solutions about a Finite Parabolic Body," *AIAA Journal*, Vol. 19, Sept. 1981, pp. 1089-1090.

[2] Graves, R. A. Jr., "Solutions to the Navier-Stokes Equations for Supersonic Flow over Blunt Bodies with Massive Wall Blowing," Ph.D. Dissertation, George Washington University, Washington, D.C., Nov. 1977.

[3] Rizzi, A. W. and Inouye, M., "A Time Split Finite-Volume Technique for Three-Dimensional Blunt Body Flow," AIAA Paper 73-133, Jan. 1973.

[4] Lombard, C. K., Davy, W. C., and Green, M. J., "Forebody and Base Region Real-Gas Flow in Severe Planetary Entry by a Factored Implicit Numerical Method—Part 1: Computational Fluid Dynamics," AIAA Paper 80-0065, Jan. 1980.

[5] Gnoffo, P. A., "Complete Supersonic Flowfields over Blunt Bodies in a Generalized Orthogonal Coordinate System," NASA TM 81784, March 1980.

[6] Blottner, F. G., "Variable Grid Scheme Applied to Turbulent Boundary Layers," *Computer Methods in Applied Mechanics and Engineering*, Vol. 4, Sept. 1974, pp. 179-194.

[7] Engquist, B. and Osher, S., "One-Sided Difference Approximations for Nonlinear Conservation Laws," *Mathematics of Computation*, Vol. 36, April 1981, pp. 321-351.

[8] Shubin, G. and Cheng, S. I., "Gas Dynamic Modeling and Computational Accuracy," *Journal of Computational Physics*, Vol. 32, July 1979, pp. 39-55.

[9] Dwyer, H. A., Kee, R. J., and Sanders, B. R., "An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," *AIAA Journal*, Vol. 18, Oct. 1980, pp. 1205-1212.

[10] Rai, M. M. and Anderson, D. A., "The Use of Adaptive Grids in Conjuction wit Shock Capturing Methods," AIAA Paper 81-1012, June 1981.

[11] Gnoffo, P. A., "A Vectorised Finite-Volume, Adaptive-Grid Algorithm for Navier-Stokes Calculations," *Numerical Grid Generation*, edited by J. F. Thompson, Elsevier North Holland, Inc., New York, 1982, pp. 819-835.

[12] MacCormack, R. W, "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, April-May 1969.

[13] Cheng, S. I., "A Critical Review of the Numerical Solution of Navier-Stokes Equations," *Lecture Notes in Physics*, No. 41, Springer Verlag, New York, 1974.

[14] Chaussee, D. S. and Steger, J. L., "Three Dimensional Viscous Flow Field Program, Pt. 2: A Curvilinear Grid and Body Generation Program for Generalized Configurations (Interim Report)," AFWAL-TM-81-64-FIMG, 1981.

[15] Libby, P. A. and Cresci, R. J., "Experimental Investigation of Downstream Influence of Stagnation-Point Mass Transfer," *Journal of the Aerospace Sciences*, Vol. 28, No. 1, Jan. 1961, pp. 51-64.

[16] Moss, J. N. and Kumar, A., "Significance of Turbulence and Transition Location on Radiative Heating with Ablation Injection," AIAA Paper 81-0281, Jan. 1981.