NASA TM-86674

NASA Technical Memorandum 86674

NASA-TM-86674 19850011614

Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers

Joseph L. Steger and Pieter G. Buning

FOR REFERENCE

NOT TO BE TAKEN FROM THIS ROOM

February 1985

LIBRARY COPY

MAR 25 1985

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

NASA
National Aeronautics and
Space Administration

NF01047

NASA Technical Memorandum 86674

Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers

Joseph L. Steger,
Pieter G. Buning, Ames Research Center, Moffett Field, California

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

N85-19924 #

DEVELOPMENTS IN THE SIMULATION OF COMPRESSIBLE INVISCID AND VISCOUS FLOW ON SUPERCOMPUTERS

JOSEPH L. STEGER AND PIETER G. BUNING

INTRODUCTION

The near term availability of scientific supercomputers will soon permit routine simulation of three dimensional compressible flow about relatively complex configurations.

General purpose flow simulation codes have been under development at NASA Ames for application to existing large scale computers (such as the CRAY-XMP and the Cyber 205) and future supercomputers. These codes have primarily used the finite difference or finite volume approach, but because of storage and speed limitations, the codes have previously been restricted to simulation of fairly simple geometric configurations. However, work towards generalizing the current codes to treat complex geometries and larger sized meshes has been underway for the past several years in anticipation of supercomputers.

While there is no agreement within NASA Ames as to what is the best general purpose simulation procedure, considerable experience has been obtained with a class of implicit finite difference algorithms written is terms of generalized coordinates. These implicit procedures are widely used in various CFD simulation codes, and they are rapidly being extended to treat quite complex geometries using current and future large scale computers. This paper will review these procedures and discuss how they are being deployed on supercomputers for simulation of complex three dimensional flow fields by using various mesh interface schemes. The importance of flow visualization and diagnostic methods to three dimensional flow simulation will be discussed at the end of this paper.

FINITE DIFFERENCE ALGORITHMS

A general purpose simulation code should be able to solve either steady or unsteady, viscous or inviscid flow with only minor input changes from the user. While it is unlikely to be an optimum code for all tasks, it should be good at its mainstay tasks, and have acceptable efficiency for other tasks. Ideally it will be a modular code whose basic engine can be stripped out and readily applied to other problems. Our mainstay general purpose flow simulation code has been a centrally differenced implicit approximate factorization code [1-6] which solves a conservative form of either the Euler or thin layer Navier Stokes equations cast in generalized coordinates.

1. Fluid Conservation Equations and Transformations

The conservation equations of mass momentum and energy referenced to a Cartesian coordinate system can be represented in the flux vector form (c.f. [7]):

$$\partial_t Q + \partial_x(F + F_v) + \partial_y(G + G_v) + \partial_z(H + H_v) = 0 \tag{1}$$

The viscous flux terms F_v, G_v and H_v contain derivatives and throughout a nondimensional form of the equations will be used. The conservative form of the equations is maintained chiefly to capture the Rankine Hugoniot shock jump relations as accurately as possible.

New independent variables $\tau, \xi, \eta, \varsigma$ are generally chosen to map a curvilinear body conforming discretization into a uniform computational space as shown in Fig. 1. Body conforming curvilinear meshes are generally used in finite difference and finite volume computations for a variety of reasons: to simplify the application of boundary conditions, to allow clustering of grid points to flow field action regions, to help maintain the well-orderliness that is useful for vector processing and for various implicit methods that employ approximate factorization techniques such as ADI, etc. Indeed, the calculation of high Reynolds number viscous flow is simply impractical without the use of a body conforming curvilinear mesh that employs clustering in the direction normal to the body surface.

In the new independent variables the transformed equations can be represented as (c.f. [7] for the detailed terms):

$$\partial_\tau \widehat{Q} + \partial_\xi(\widehat{F} + \widehat{F}_v) + \partial_\eta(\widehat{G} + \widehat{G}_v) + \partial_\varsigma(\widehat{H} + \widehat{H}_v) = 0 \tag{2}$$

where the original dependent variables are maintained. (The flux vectors of the transformed equations can be made to resemble their Cartesian counterparts by combining terms into contravarient velocity components [3-4].) If a body conforming coordinate is used, then for high Reynolds number flow it is generally permissable to make a thin layer assumption and to discard viscous terms except for those in the normal-like direction. If ς is the coordinate away from the surface, the thin layer equations can be represented as [3,4]

$$\partial_\tau \widehat{Q} + \partial_\xi \widehat{F} + \partial_\eta \widehat{G} + \partial_\varsigma \widehat{H} = Re^{-1}\partial_\varsigma \widehat{S} \tag{3}$$

where the viscous terms in ς have been collected into the vector $\widehat{S}$ and the nondimensional reciprocal Reynolds number is extracted to indicate a viscous flux term.

In differencing these equations it is often advantageous to difference about a base solution denoted by subscript $_0$ as

$$\begin{aligned}\delta_\tau(\widehat{Q} - \widehat{Q}_0) + \delta_\xi(\widehat{E} - \widehat{E}_0) + \delta_\eta(\widehat{F} - \widehat{F}_0) + \delta_\varsigma(\widehat{G} - \widehat{G}_0) - Re^{-1}\delta_\varsigma(\widehat{S} - \widehat{S}_0) \\ = -\partial_\tau \widehat{Q}_0 - \partial_\xi \widehat{E}_0 - \partial_\eta \widehat{F}_0 - \partial_\varsigma \widehat{G}_0 + Re^{-1}\partial_\varsigma \widehat{S}_0\end{aligned} \tag{4}$$

where δ indicates a general difference operator, and ∂ is the differential operator. If the base state is properly chosen, the differenced quantities can have smaller and smoother variation and therefore less differencing error. If a the base solution exactly satisfies the partial differential equation, then the right hand side of Eq.(4)

is identically zero. A uniform free stream satisfies the governing equations and is frequently taken as the base solution so as to minimize far field differencing errors. Such an error can occur if the coordinate transformation terms that have been embedded into the transformed fluxes are not consistently differenced [3,4,8]. There are also anologies between the finite volume method and the transformed difference equations, and the actual discretized equations that result from the two schemes can be made similar.

2. Implicit Central Difference Algorithm

Equations (3) or (4) have been solved using a Beam-Warming noniterative approximate factorization implicit scheme of the form [3-6]

$$\begin{aligned}&\left[I+h\delta_\xi\widehat{A}^n-D_i|_\xi\right]\left[I+h\delta_\eta\widehat{B}^n-D_i|_\eta\right]\left[I+h\delta_\varsigma\widehat{C}^n-hRe^{-1}\bar{\delta}_\varsigma J^{-1}\widehat{M}^n J-D_i|_\varsigma\right]\Delta\widehat{Q}^n\\&=-\Delta t\left[\delta_\xi(\widehat{E}^n-\widehat{E}_\infty)+\delta_\eta(\widehat{F}^n-\widehat{F}_\infty)+\delta_\varsigma(\widehat{G}^n-\widehat{G}_\infty)-Re^{-1}\bar{\delta}_\varsigma(\widehat{S}^n-\widehat{S}_\infty)\right]\\&-D_e(\widehat{Q}^n-\widehat{Q}_\infty)\end{aligned}\tag{5}$$

where $h = \Delta t$ or $(\Delta t)/2$ and the free stream base solution is used. Here δ is typically a three point second order accurate central difference operator, while $\bar{\delta}$ is a midpoint operator used with the viscous terms. The matrices $\widehat{A}, \widehat{B}, \widehat{C}$, and $\widehat{M}$ result from local linearization about the previous time level and J is the Jacobian of the coordinate transformation. The factored left hand side operators form block tridiagonal matrices.

Because central space difference operators are used, numerical dissipation terms denoted as D_i and D_e have been inserted into Eq.(5). In their simplest form these have been given as combinations of fourth differences

$$D_e = \epsilon_e \Delta t J^{-1}[(\nabla\Delta)^2_\xi + (\nabla\Delta)^2_\eta + (\nabla\Delta)^2_\varsigma]J \tag{6a}$$

and second differences

$$D_i|_\xi = \epsilon_i \Delta t J^{-1}(\nabla\Delta)_\xi J, \quad D_i|_\eta = \epsilon_i \Delta t J^{-1}(\nabla\Delta)_\eta J, \quad D_i|_\varsigma = \epsilon_i \Delta t J^{-1}(\nabla\Delta)_\varsigma J \tag{6b}$$

where ∇ and Δ are two-point backward and forward difference operators and where $\epsilon_i > 2\epsilon_e$, and $\epsilon_e = O(1)$. The implicit second difference numerical dissipation operators were chosen to keep the left hand side factors block tridiagonal, and as D_i works on ΔQ, accuracy is not impaired. More robust dissipation terms are often used in which the ϵ coefficients are scaled with some approximate modulus of the A, B, and C Jacobian matrices and gradients of an appropriate variable. For example, explicit and implicit ξ-dissipation terms such as

$$D_e = (\Delta t)J^{-1}(|\xi_x| + |\xi_y| + |\xi_z|)[\epsilon_2\bar{\delta}\frac{|\bar{\delta}^2 p|}{|(1+\bar{\delta}^2)p|}\bar{\delta} + \epsilon_4\bar{\delta}^4]J$$

$$D_i = (\Delta t)J^{-1}(|\xi_x| + |\xi_y| + |\xi_z|)[\epsilon_2\bar{\delta}\frac{|\bar{\delta}^2 p|}{|(1+\bar{\delta}^2)p|}\bar{\delta} + 3\epsilon_4\bar{\delta}^2]J$$

have been used where p is the nondimensionalized fluid pressure and ϵ_2 is $O(1)$ while ϵ_4 is $O(0.1)$. In transonic flow the term $(|\xi_x| + |\xi_y| + |\xi_z|)$ is an estimate to the

spectral radius of A for variables that have been nondimensionalized with respect to the sound speed. Since $(|\xi_x| + |\xi_y| + |\xi_z|)$ does not contain a fluid variable, it can be left as an outside coefficient to the dissipation operator without effecting the weak conservation form of the difference equations.

Body surface boundary conditions have usually been supplied by using a combination of normal-momentum, tangency or no slip, and extrapolation [3,4]. Various far field conditions have been used including characteristic-like conditions [9-11]. Because of their simplicity, in most of the application codes the boundary conditions have been imposed explicitly, or a combination of simplified implicit-explicit conditions [12] have been used. However, fully implicit boundary conditions have also been used (c.f. [1,11,13]) and an elegant implicit characteristic-like formulation has been given and tested by Chakravarthy[11].

3. Vectorization and Multi-Tasking

The structure of the above algorithm lends itself to vectorized computer coding. Vectorization has been implemented by inverting "pencils" of data [14], but FORTRAN based codes generally follow an approach coded by Benek (unpublished, circa 1980). In this approach strings of block tridiagonal matrices are inverted simultaneously so as to avoid the recursive nature of matrix elimination procedures. For example, if indices j, k, l correspond to ξ, η, ζ, the first factor of Eq.(5) forms a block tridiagonal between, say, points $j = 1$ to $j = jmax$. There is one such ξ-block tridiagonal for each k, l index so by simultaneously inverting ξ-block tridiagonals over, say, $k = 1, kmax$ a vector length of $kmax$ is achieved, see Fig. 2. However, in inverting a ξ-block tridiagonal temporary storage is needed for the backward elimination and this storage requirement is increased by $kmax$. (Means of reducing the block size are discussed in [15-17] as a way to improve efficiency, and these same techniques *reduce* temporary storage as well.) Overall, vectorization tends to complicate the coding, but not unduely so. On the CRAY-XMP the ARC2D vectorized code [5] runs about 5 times faster than the unvectorized code for a two-dimensional grid of order 200×40.

The central differenced implicit algorithm, Eq.(5), has also been experimentally coded for multi-tasking on two minicomputers by assigning each processor a portion of the code [18]. For example, processor 1 can solve the ξ-block tridiagonal in the range $l = 1, l_1$, processor 2 can solve the ξ-block tridiagonal from $l = l_1 + 1, l_2$ and so on. Because of a lack of readily available hardware and software, multi-tasking has not yet worked its way into the codes used for routine applications.

4. Improvements in Efficiency and Accuracy

The introduction of a faster computer (that is still easy to use) drives the development of much more efficient numerical algorithms. This is because as the machines become more powerful, it is easier to experiment will new ideas in numerical algorithms. New algorithms can be quickly verified, problem areas can be isolated, and optimum convergence parameters can be found. As a result, improvements in numerical algorithms have kept pace with improvements in computer hardware. Such has been the experience with the central differenced implicit algorithm. Originally coded for the Control Data 7600, it was transferred to the CRAY-1S and vectorized [5] to run an order of magnitude faster. Increased computer speed has in turn has-

tened the numerical optimization process indicated below which has lead to another order of magnitude improvement in the code's steady state performance.

Improvements to the basic algorithm in both efficiency and accuracy have been made by a variety of contributors. To improve its overall efficiency, simple changes have been optionally implemented into Eq.(5). These changes have been selected so that they do not unduely complicate the basic algorithm. They include the use of space varying Δt, use of a sequence of coarsened grids to provide a good initial guess, cutting inversion costs by using either diagonalization [15,16] or block reduction [17] methods, implementation of better numerical dissipation terms, and more implicit treatment of the numerical dissipation terms. As described in [5,6], these combined changes can improve steady state efficiency by an order of magnitude. Details and appropriate references are described in [5,6].

Although the implicit algorithm has been presented with three point central differencing, versions of the algorithm that have fourth order accuracy in space have been available [1-4] and are preferred unless strong shock waves are present. Reddy [19] has also demonstrated a version in which a pseudo-spectral operator is used in place of the right hand side convection operators. Improved dissipation models [5,6] and total variation diminishing (TVD) implementations [20] have also been carried out to better capture shocks, and perturbation about approximate base solutions has been used to reduce the number of needed grid points [21].

5. Applications

The factored implicit scheme described above has been ultilized in a variety of flow applications. Two-dimensional simulations include steady and unsteady flow about airfoils [c.f. 3,5,6,22-25], cascades [9], projectiles [26-27], and inlets[16]. Three dimensional calculations have been carried out about simple configurations using relatively coarse meshes. Simulations include the supersonic blunt body problem[28], simple bodies [4-6], wings [29], afterbodies [30], and the space shuttle [31]. Figures 3 and 4 are reproduced from the publications of Diewert [30] and Chaussee et al [31] and show representative three dimensional viscous flow solutions that have been obtained on a CRAY-1S. To save computer time and storage, both of these calculations where computed in stages. In the boattail case a complete solution was first obtained using a grid that inadequately resolved the base flow region. A refined grid was then introduced about the afterbody section and the computation for only this region was carried out using inflow conditions taken from the coarsened grid result. In the shuttle calculation a parabolized Navier Stokes code was used in those flow regions in which the main outer flow remains supersonic and the inner viscous layer remains attached with respect to the mainstream direction. These calculations and a majority of the viscous calculations carried out using the factored implicit algorithm at high Reynolds numbers have ultilized a simple algebraic turbulence model [32].

6. Upwind Schemes

As an alternative to using central space differencing for the convection terms, upwind (i.e. backward and forward) space differencing can be used if the fluxes are properly split according to their characteristic properties. As discussed in [33] upwind schemes can have several advantages over central difference schemes, in-

cluding natural numerical dissipation, better explicit stability, and more readily inverted implicit schemes. Conversely, upwind schemes for systems of equations have generally been more complicated and computationally expensive than central difference schemes and are not very suitable for treating viscous terms.

Although upwind schemes are not as extensively ultilized in aerodynamics simulations as central difference schemes, upwind schemes have been used on curvilinear grids in both finite difference and finite volume formulations and good results have been obtained. Figures 5a and 5b, for example, show an inviscid flow result for a NACA 0012 airfoil at $M_\infty = 0.8$ and an angle of attack of 1.25° using a flux split class of upwind scheme [34] (see [35] for similar results). Various upwind schemes have also proved to be quite effective for capturing strong shock waves, c.f. [36] and [37]. However, because of their complexity and cost, upwind based schemes have not yet been widely distributed as general purpose simulation codes for either inviscid or viscous flow.

Upwind and partially upwind based schemes will likely first emerge as important flow simulation schemes in two areas: 1) supersonic and hypersonic flows in which there are very strong embedded shocks, and, 2) transonic and supersonic flows in which one coordinate direction can be mostly aligned with the flow streamlines. In this latter case, a mixed upwind-central difference scheme can be very effective if the upwind differencing is used in the streamline direction. For example, an implicit algorithm for the thin layer Navier Stokes equations could have the form

$$\begin{aligned}
&\left[I + h\delta_\xi^b(\widehat{A}^+)^n + h\delta_\varsigma\widehat{C}^n - hRe^{-1}\delta_\varsigma J^{-1}\widehat{M}^n J - D_i|_\varsigma\right] \\
&\qquad \times \left[I + h\delta_\xi^f(\widehat{A}^-)^n + h\delta_\eta\widehat{B}^n - D_i|_\eta\right]\Delta\widehat{Q}^n = \\
&\qquad - \Delta t\{\delta_\xi^b[(\widehat{E}^+)^n - \widehat{E}_\infty^+] + \delta_\xi^f[(\widehat{E}^-)^n - \widehat{E}_\infty^-] + \delta_\eta(\widehat{F}^n - \widehat{F}_\infty) \\
&\qquad + \delta_\varsigma(\widehat{G}^n - \widehat{G}_\infty) - Re^{-1}\delta_\varsigma(\widehat{S}^n - \widehat{S}_\infty)\} - D_e(\widehat{Q}^n - \widehat{Q}_\infty) \qquad (7)
\end{aligned}$$

where δ_ξ^b and δ_ξ^f are backward and forward three-point difference operators and D_e contains only η and ς numerical dissipation operators. This two factor implicit scheme is readily vectorized or multi-tasked in planes of ξ = constant. A semi-implicit scheme is obtained by neglecting the calculation of $h\delta_\xi^f\widehat{A}^-$ in the implicit backsweep operating on $\Delta\widehat{Q}^n$.

COMPOSITE GRIDS

Use of a single well-ordered body conforming curvilinear mesh simplifies the application of boundary conditions and can lead to use of efficient solution procedures. However, the generation of body conforming well-clustered curvilinear grids that are not overly skewed and have smooth variation can often be quite difficult. In particular, it is generally impractical to build a single grid of this type for complex three dimensional configurations. Of course, by judiciously introducing cuts in the grid, some fairly complex bodies can be meshed with a single grid, Fig. 6 illustrates this possiblility. But the trend has been to introduce more than one grid and to patch or overset the grid systems together. The sketches shown in Fig. 7 illustrate simple patch and overset grid configurations in two dimensions. (There has also been some activity in reverting back to the use of pure Cartesian grids that intersect the body

in a random way and so require special logic at the body interface [38]. Such a procedure appears to be impractical, however, for high Reynolds number viscous flow problems unless the Cartesian scheme is patched-to or overset-with a body conforming grid that is used only near the body. In this case the Cartesian mesh approach would be a special case of either the patched or overset grid method.)

The use of a set of patched or overset grids to form a larger composite grid carries the discretization process one step further. In a sense such a finite difference process assumes some of the characteristics of a finite element scheme that uses large powerful elements in which each element is itself discretized. In this discretization process each individual grid in the system is well ordered and is thus suitable for efficient finite difference solution using any available single grid scheme. The problem with such a composite grid scheme is the difficulty of interfacing each mesh without degrading numerical accuracy or convergence. Moreover, as few such meshes should be generated as necessary to achieve grid efficiency.

Limited experience with both patched [39-41] and overset grids [42-46] has not shown which method is preferable - an optimum method will perhaps combine both patched and overset grids. Both schemes necessitate extensive bookkeeping procedures. The patched grid method has as its chief drawback a grid generation problem that is still relatively difficult because various interfaces have to be defined and grids have to be generated with both inner and outer defined boundary surfaces. Drawbacks to using overset grids include having to interpolate data points along an irregular boundary and the bookkeeping can be especially complex if more than two levels of overset grids intersect each other.

Figures 8 - 9 show recently published two dimensional inviscid flow results obtained using patched and overset meshes. Both grids and flow contour levels are shown. The supersonic biplane results of Hessenius and Rai [41] were computed using a patched grid method in which a flux balance interface scheme is used at grid boundaries to ensure fluid conservation. In this method [40], meshes must share a common boundary, but grid lines need not have a common slope or even join together. These results were obtained using a first order accurate Osher scheme. The overset grid results of Dougherty [46] for an airfoil and flap were computed using the factored implicit algorithm. Only simple interpolation is used to interface the mesh boundaries using data from nearby points from the underlying grid. Figure 9c shows the nearby interpolation points used to update the outer boundary of the flap grid (open symbols) and grid points which are excluded from the calculation (filled symbols). Simulation of wing-body configurations using the implicit algorithm with composite grids are currently underway by Holst and coworkers and Benek and coworkers and will appear shortly.

POSTPROCESSING

1. Flow Visualization

One of the major problems that is being encountered in three dimensional flow simulation is the difficulty of displaying a limited and proper kind of data that will lead to better understanding. In the future, much of the computational aerodynamicist's time will be devoted to extracting and displaying various features of the solution. In three dimensions, flow phenomenon such as flow-reversals, shocks,

shear-layers, vortices, etc. can often be difficult to identify and visualize, especially if the flow is also unsteady. Graphic displays of contour surfaces and particle paths and the like can also be expensive to generate. As a result, more computer resource could be expended on analyzing a solution and displaying it in a meaningful way to a human being than what was needed to generate the solution in the first place. To become convinced of how costly this can become one need merely contemplate generating numerous detailed displays of a three dimensional flow field in which reliable hidden line removal is needed to keep the display from being too confusing.

Two approaches to extracting graphic information and flow visualization are evolving. In one approach the engineer works at a graphics workstation and displays various portions of the flow field, observing the solution from different vantage points, and otherwise interacts with the computed solution, perhaps by seeding particles and observing their behaviour. Because the three dimensional data base can be quite large, this approach will require very powerful graphic work stations and high speed data links between the graphics station and the supercomputer. The advantage of this approach is that one can stumble across information that might not otherwise be anticipated.

The other approach is to program the supercomputer itself to diagnose the data base. Such a computer program is needed in order to throughly search a flow field to find details that a human at a workstation would find too tedious to locate. Algorithms must be developed to search out and display special features such as shock waves, vortices, and separation lines. Because the data bases are so large, only the supercomputer itself will be able to perform many of these calculations.

Reliable software to locate interesting flow regions is not available and will be difficult to generate. For the CFD algorithm developer, developing flow visualization procedures offers interesting opportunities. Consider, for example, the difficulty of building an algorithm that automatically identifies embedded shock waves in a complex three dimensional flow, especially weak oblique shocks. Or consider the apparently straightforward task of building an efficient and accurate particle trace scheme. To trace particles given the velocity field, one need only numerically integrate simple ordinary differential equations. But if too many particles are displayed, the picture will be confusing. Interpolation of the velocity components for each particle location can also be a costly process, so efficient algorithms are needed to minimize this expense. Finally, unless the particles are seeded in the right locations, the most interesting features of the flow will not be observed. For example, particles seeded outside of a small vortex core will not be entrained inside the core.

2. Diagnostics

The accuracy of a finite difference solution is generally appraised by successively refining the grid and comparing solutions from one grid to another. If the solution is unchanged, the flow result is likely resolved. (Here we assume that a Reynolds averaged Navier Stokes solution can become invariant with grid refinement.) Because the next generation of supercomputers will not be 'super enough' to carry this process too far, the problem of determing solution accuracy may stimulate additional research in approximate methods such as potential and boundary layer schemes. We might compute, for example, a boundary layer solution to verify the correctness of the skin friction and heat transfer found from a Navier Stokes procedure. By

taking the Navier Stokes computed pressure gradient, the boundary layer equations can be solved using a very refined grid near the wall. If the boundary layer solution returns the same wall values computed from the Navier Stokes equations, then the Navier Stokes solution is adequately resolved. (Because the boundary layer will likely be run in an inverse mode if the flow is separated, the check might be whether or not the boundary layer solution returns the same surface pressure distribution.)

For complex three dimensional flows it will also be difficult to provide code verification. Comparison with linear theory may be inadequate and experimental data may not be available or reliable. One possiblility is to compare the Navier Stokes computed results to those obtained from the solution of a simplified set of equations that are solved over the entire field, but in which rotational and perhaps even compressibility effects are directly taken from the Navier Stokes solution. As a quite simple illustration, the Poisson equation for the incompressible streamfunction ψ

$$\psi_{xx} + \psi_{yy} = -\omega$$

should reproduce an incompressible two-dimensional Navier Stokes result if the vorticity, ω, is taken directly from the Navier Stokes solution. Because this is a simple scalar equation, it should be possible to solve for it very quickly and using a more refined grid than what was used for the Navier Stokes simulation. If the refined grid Poissson calculation can return the Navier Stokes solution then considerable validation of the code is obtained. Carrying out such a process for complex configurations in three dimensions will be difficult but useful.

CONCLUDING REMARKS

Evolution of existing finite difference schemes should provide reliable and efficient single mesh codes for nonlinear three dimensional flow field simulation. To treat viscous flow about highly complex body configurations, however, composite meshes will be needed and considerable work in interfacing these schemes into multiple grid codes must yet be carried out. Just as the task of optimizing the flow solvers has been accelerated by the availability of large fast computers, so too the task of developing interface schemes will be accelerated with the availablity of forthcoming supercomputers, provided these machines are easy to use.

Cost considerations are leading to supercomputers that achieve more and more of their performance advantage by use of repetitive processors. Many of these machines are more difficult to use than machines that achieve higher performance by having a faster single instruction time. Hopefully these future supercomputers will not be a machines of 'insurmountable opportunity'.

It is generally accepted within the CFD community that three dimensional data bases are so large that new ways of displaying and interacting with the data is necessary. It must be emphasized that hardware and graphics software provides only part of the solution. If the computed solutions are to be properly analyzed and understood, extensive new algorithm developments in flow visualization and diagnostics will be needed, and these algorithms will require the same kind of creativity that went into generating the flow simulation procedures themselves.

REFERENCES

1. Beam, R. M. and Warming, R. F., *An Implicit Finite - Difference Algorithm for Hyperbolic Systems in Conservation - Law - Form*, J. Comp. Phys. **22** (Sept. 1976,), p. 87-110.
2. Warming, R. F. and Beam, R. M., *On the Construction and Application of Implicit Factored Schemes for Conservation Laws*, Symposium of CFD, New York, **11** (April 16-17, 1977, SIAM-AMS Proceedings).
3. Steger, J. L., *Implicit Finite Difference Simulation of Flow About Arbitrary Two Dimensional Geometries*, AIAA J. **16, No. 7** (July 1978).
4. Pulliam, T. H. and Steger, J. L., *On Implicit Finite Difference Simulations of Three Dimensional Flows*, AIAA J. **18** (Feb. 1980).
5. Pulliam, T. H., *Euler and Thin Layer Navier Stokes Codes : ARC2D, ARC3D, Notes for Computational Fluid Dynamics User's Workshop*, The University of Tennessee Space Institute, Tullahoma, Tenn. (March 12-16, 1984).
6. Pulliam, T. H. and Steger, J. L., *Recent Inprovements in Efficiency, Accuracy, and Convergence of an Implicit Approximate Factorization Algorithm*, AIAA Paper 85-0360 (Jan. 1985).
7. Viviand, H., *Conservative Forms of Gas Dynamics Equations*, La Recherche Aerospatiale **1** (1978).
8. Thomas, P. D. and Lombard, C. K., *Geometric Conservation Law and Its Application to Flow Computations on Moving Grids*, AIAA Journal **17** (1979).
9. Steger, J. L., Pulliam, T. H. and Chima, R. V., *An Implicit Finite Difference Code for Inviscid and Viscous Cascade Flow*, AIAA paper 80-1427 (1980).
10. Pulliam, T. H., *Characteristic Boundary Conditions for the Euler Equations*, NASA Publication 2201, Numerical Boundary Condition Procedures (Oct. 1980).
11. Chakravarthy, *Euler Equations - Implicit Schemes and Implicit Boundary Conditions*, AIAA 82-0228 (Jan. 11-14, 1982).
12. Schiff, L. B. and Steger, J. L., *Numerical Simulation of Steady Supersonic Viscous Flow*, AIAA Journal **18** (Dec. 1980).
13. Nicolet, W. E., Shanks, S., Srinivasan, G. and Steger, J. L., *Flowfield Predictions About Lifting Entry Vehicles*, AIAA Paper 82-0026 (Jan. 11-14, 1982).
14. Lomax, H. and Pulliam, T. H., *A Fully Implicit Factored Code for Computing Three - Dimensional Flows on the ILLIAC IV*, Parallel Computations, G. Rodrigue, Ed. Academic Press, New York (1982).
15. Pulliam, T. H. and Chaussee, D. S., *A Diagonal Form of an Implicit Approximate - Factorization Algorithm*, J. Comp. Phys. **39, no. 2** (Feb. 1981), p. 347.
16. Chaussee, D. S. and Pulliam, T. H., *A Diagonal Form of an Implicit Approximate Factorization Algorithm with Application to a Two Dimensional Inlet*, AIAA J. **19, no. 2** (Feb. 1981), p. 153.
17. Barth, T. J. and Steger, J. L., *An Efficient Approximate Factorization Implicit Scheme for the Equations of Gasdynamics*, NASA TM 85957 (June 1984).
18. Eberhardt, D. S., *A Study of Multiple Grid Problems on Concurrent Processing Computers*, Stanford University Dissertation, Department of Aeronautics and Astronautics, Stanford, CA. (Sept., 1984).

19. Reddy, K. C., *Pseudospectral Approximation in a Three - Dimensional Navier - Stokes Code*, AIAA J. **21, no. 8** (Aug. 1983).
20. Yee, H. C., Warming, R. F. and Harten, A., *Implicit Total Variation Diminishing (TVD) Schemes for Steady - State Calculations*, NASA TM 84342 (March 1983).

21. Chow, L. J., Pulliam, T. H. and Steger J. L., *A General Perturbation Approach for the Equations of Fluid Dynamics*, AIAA Paper 83-1903 (July 13-15, 1983).
22. Anderson, W. K., Thomas, J. L. and Rumsey, C. L., *Application of Thin Layer Navier - Stokes Equations Near Maximum Lift*, AIAA Paper 84-0049 (Jan. 1984).
23. Barth, T. J., Pulliam, T. H. and Buning, P., *Navier - Stokes Computations for Exotic Airfoils*, AIAA Paper 85-0109 (Jan. 1985).
24. Barton, J. T. and Pulliam, T. H., *Airfoil Computations at High Angles of Attack, Inviscid and Viscous Phenomena*, AIAA Paper 84-0524 (1984).
25. Srinivasan , G. R., McCroskey, W. J. and Kutler, P., *Numerical Simulation of the Interaction of a Vortex with Stationary Airfoil in Transonic Flow*, AIAA Paper 84-0254 (Jan. 1984).
26. Nietubicz, C. J., *Navier Stokes Computations for Conventional and Hollow Projectile Shapes at Transonic Velocities*, AIAA Paper 81-1262 (June 1981).
27. Sahu, J., Nietubicz, C. J. and Steger, J. L., *Numerical Computation of Base Flow for a Projectile at Transonic Speeds*, AIAA 82-1358 (Aug. 9-11, 1982).
28. Kutler, P., Pedelty, J. A. and Pulliam, T. H., *Supersonic Flow over Three - Dimensional Ablated Nosetips Using an Unsteady Implicit Numerical Procedure*, AIAA Paper 80-0063 (1980).
29. Mansour, N. N., *Numerical Simulation of the Tip Vortex Off a Low - Aspect Ratio Wing at Transonic Speed*, AIAA Paper 84-0522 (Jan. 1984).
30. Diewert, G. S., *Topological Analysis of Computed Three -Dimensional Viscous Flow Fields*, In 'Recent Contributions to Fluid Mechanics', W. Haase, ed. Springer-Verlag, Berlin (1982).
31. Chaussee, D. S., Rizk, Y. M. and Buning, P. G., *Viscous Computation of a Space Shuttle Flow Field*, NASA TM 85977 (June 1984).
32. Baldwin, B. S. and Lomax, H.,, *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*, AIAA Paper 78-257 (Jan. 1978).
33. Steger, J. L. and Warming, R. F., *Flux Vector Splitting of the Inviscid Gasdynamics Equations with Applications to Finite Difference Methods*, J. of Comp. Physics 40 (1981).
34. Buning, P. G., *Computation of Inviscid Transonic Flow Using Flux Vector Splitting in Generalized Coordinates*, Stanford University Dissertation, Department of Aeronautics and Astronautics, Stanford, CA (1983).
35. Anderson, W. K., Thomas, J. L. and Van Leer, B., *A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations*, AIAA Paper 85-0122 (Jan. 1985).
36. Chakravarthy, S. R. and Osher, S., *Numerical Experiments with the Osher Upwind Scheme for the Euler Equations*, AIAA Paper 82-0975 (June 1982).
37. Chakravarthy, S. R. and Osher, S., *High Resolution Applications of the Osher Upwind Scheme for the Euler Equations*, AIAA Paper 83-1943 (1983).

38. Clarke, D. K., Hassan, H. A. and Salas, M. D., *Euler Calculations for Multi-element Airfoils Using Cartesian Grids*, AIAA Paper 85-0291 (Jan. 1985).
39. Lasinski, T. A., et. al., *Computation of the Steady Viscous Flow Over a Tri-Element Augmenter Wing Airfoil*,, AIAA Paper 82-0021 (Jan. 1982).
40. Rai, M. M., *A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations*, AIAA Paper 84-0164 (1984).
41. Hessenius, K. A. and Rai, M. M., *Applications of a Conservative Zonal Scheme to Tansient and Geometrically Complex Problems*, AIAA Paper 84-1532 (June 1984).
42. Atta, E. H. and Vadyak, T., *A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows About Aircraft Configurations*, AIAA Paper 82-1017 (June 1982).
43. Steger, J. L., Dougherty, F. C. and Benek, J. A., *A Chimera Grid Scheme*, ASME Mini-Symposium on Advances in Grid Generation, Houston, Texas, (June 1983).
44. Benek, J. A., Steger, J. L. and Dougherty, F. C., *A Flexible Grid Embedding Technique with Application to the Euler Equations*, AIAA 83-1944 (July 13-15, 1983).
45. Berger, M. J., *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, STAN-CS-82-924, Stanford University (Aug. 1982).
46. Dougherty, F. C., *Development of a Chimera Grid Scheme with Applications to Unsteady Problems*, Stanford University Dissertation, Department of Aeronautics and Astronautics, Stanford, CA (Oct. 1984).

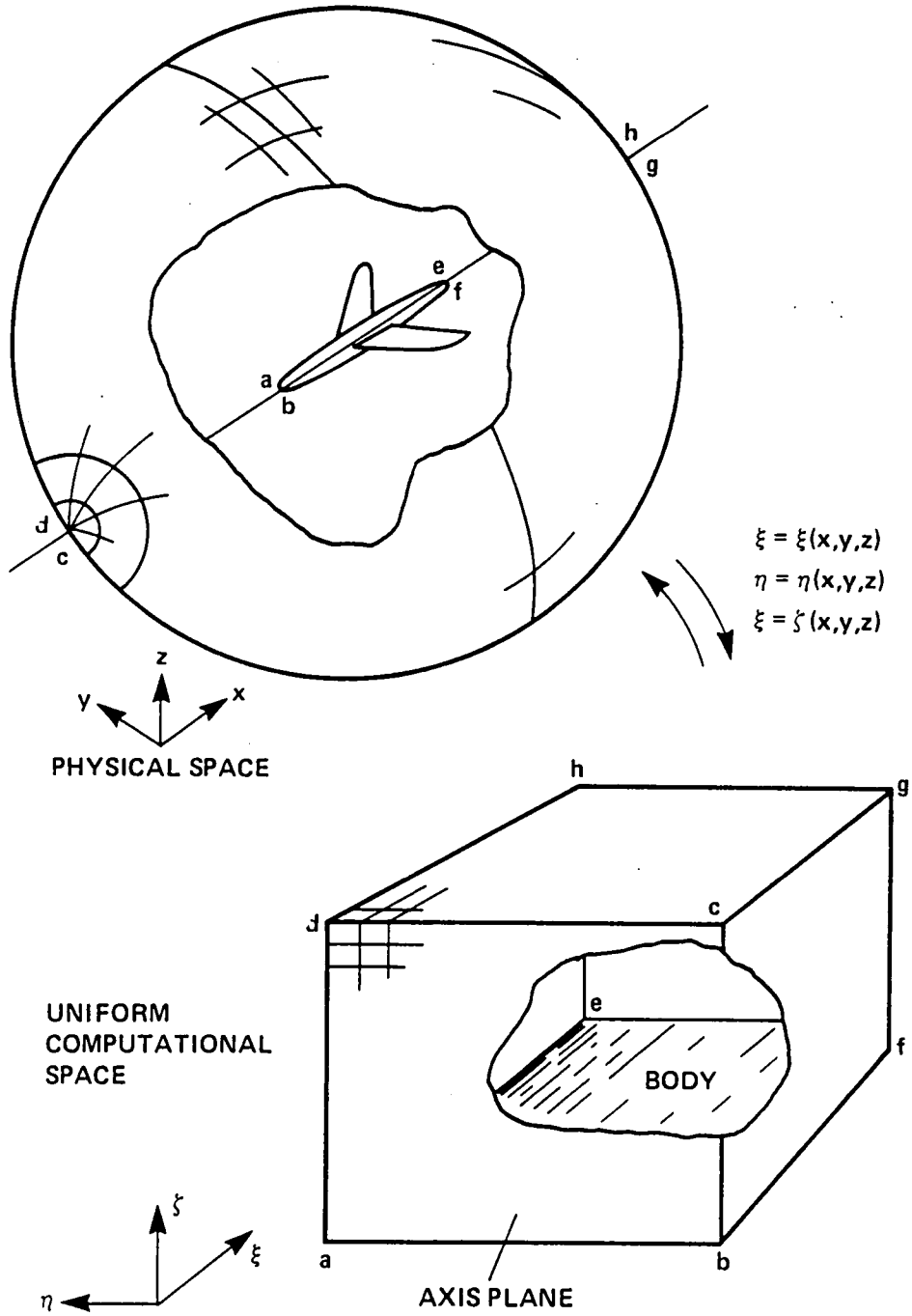


Fig. 1. Sketch showing mapping of physical space to computational space for a well-ordered warped spherical grid.

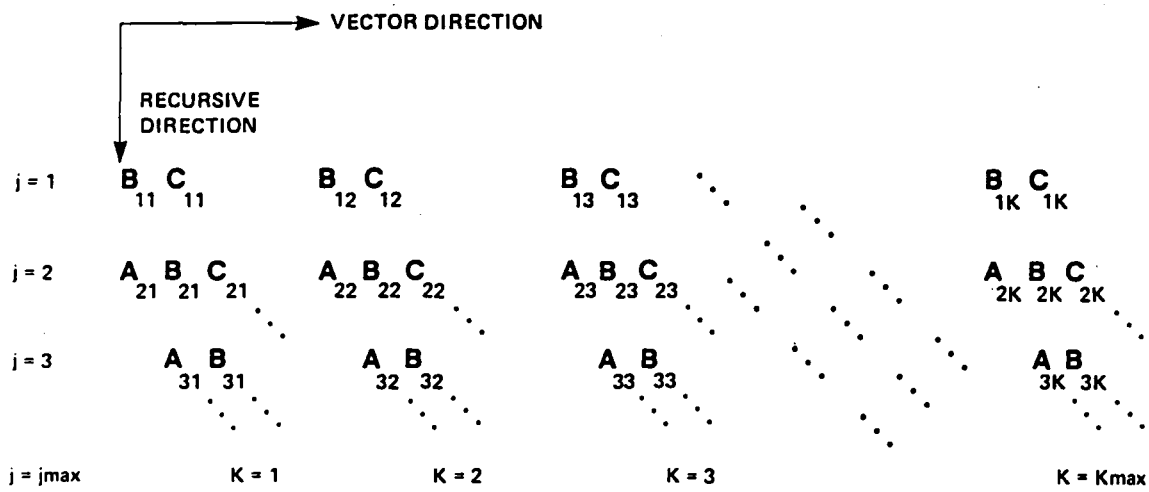
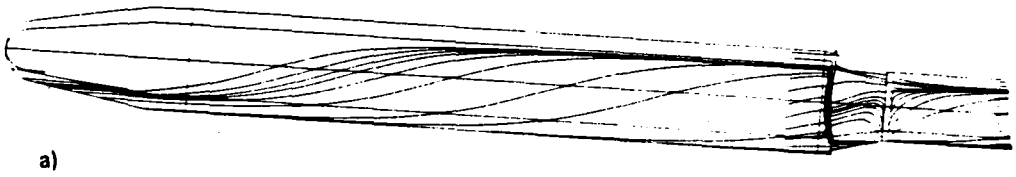
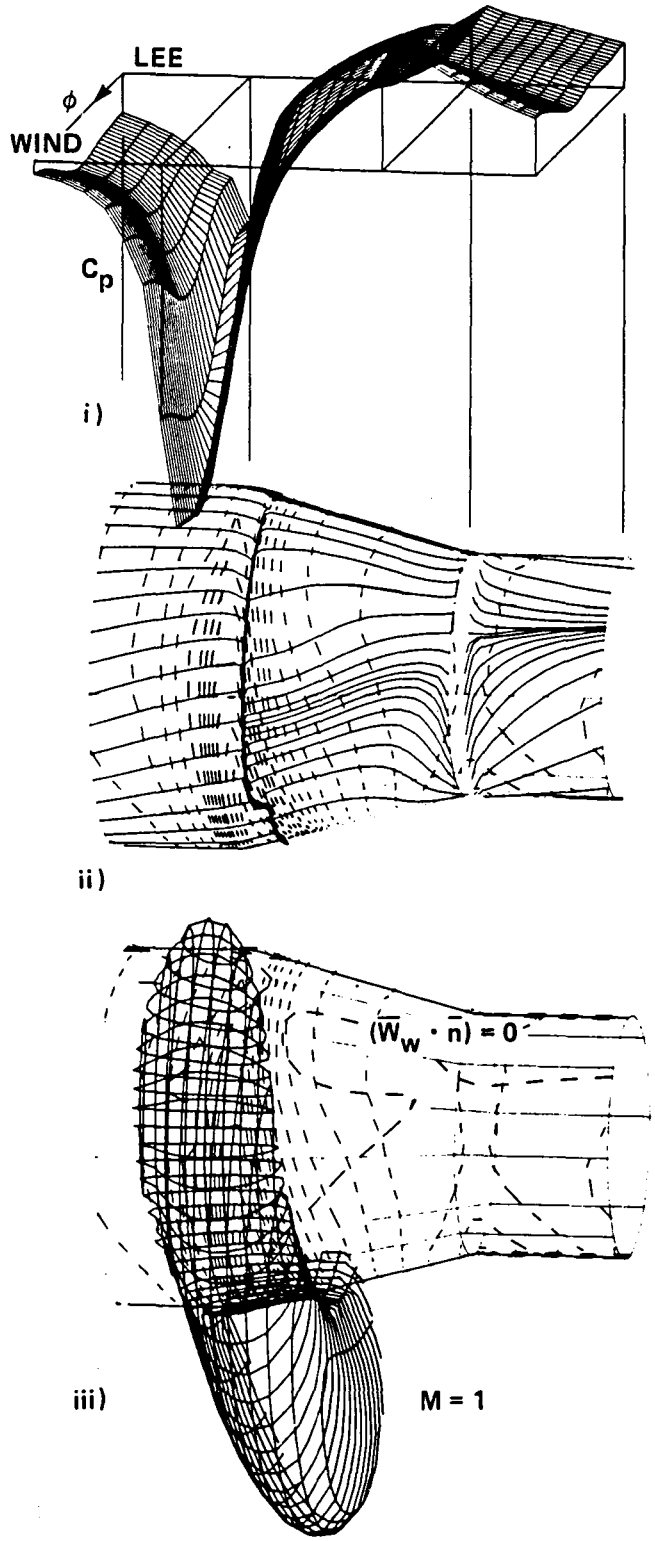


Fig. 2. Simultaneous inversion of block tridiagonals to obtain a vector length.



a). Streamline pattern for afterbody model

Fig. 3. Transonic flow over a body of revolution with conical afterbody, at 8° angle of attack, $M_\infty = 0.9$ and $Re_d = 3 \times 10^6$.



b). Lateral perspective of flow features for conical afterbody showing (i) surface pressures, (ii) isobars (dashed line) and surface shear (solid line), and (iii) sonic surface.

Fig. 3. Concluded.

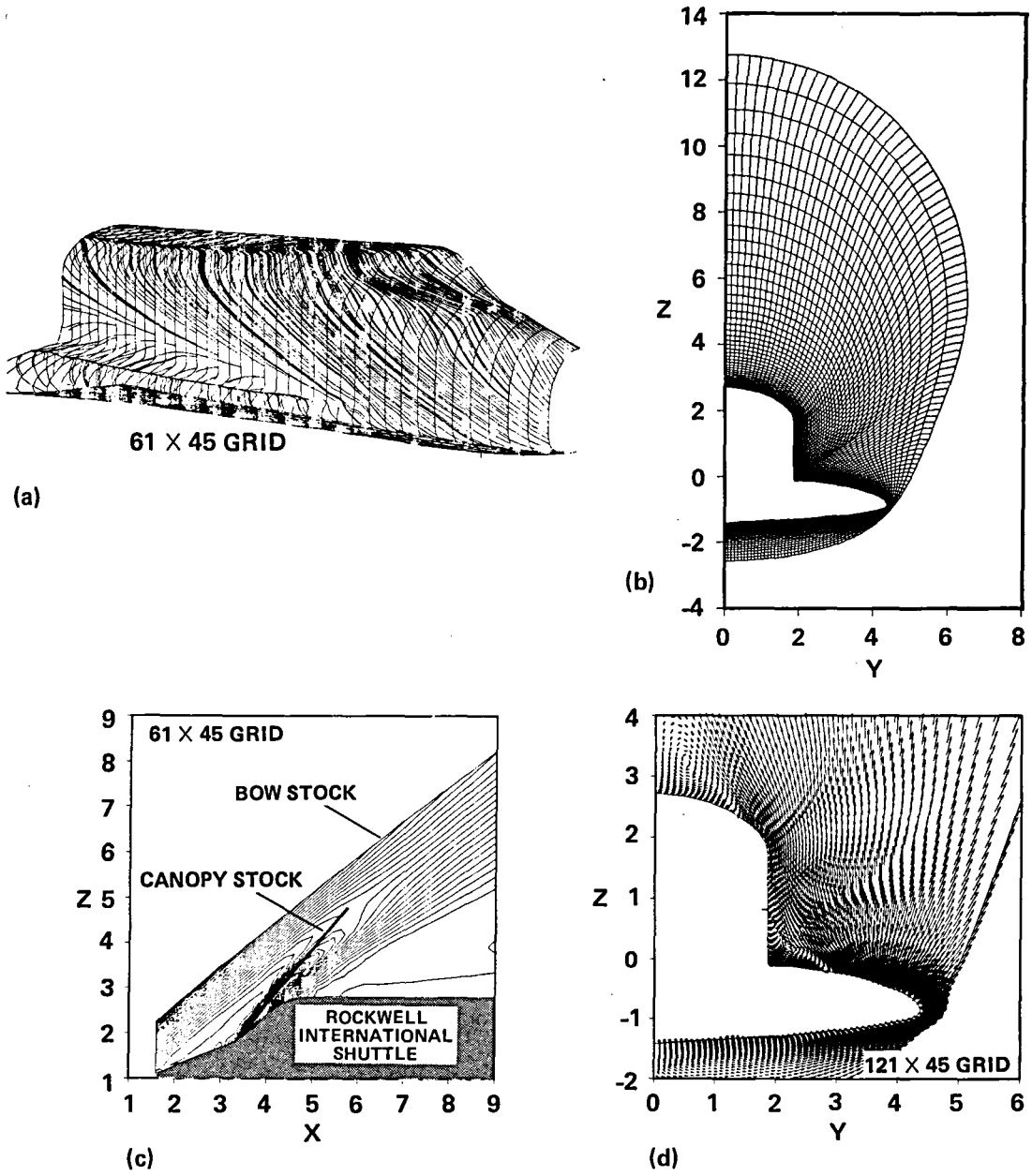


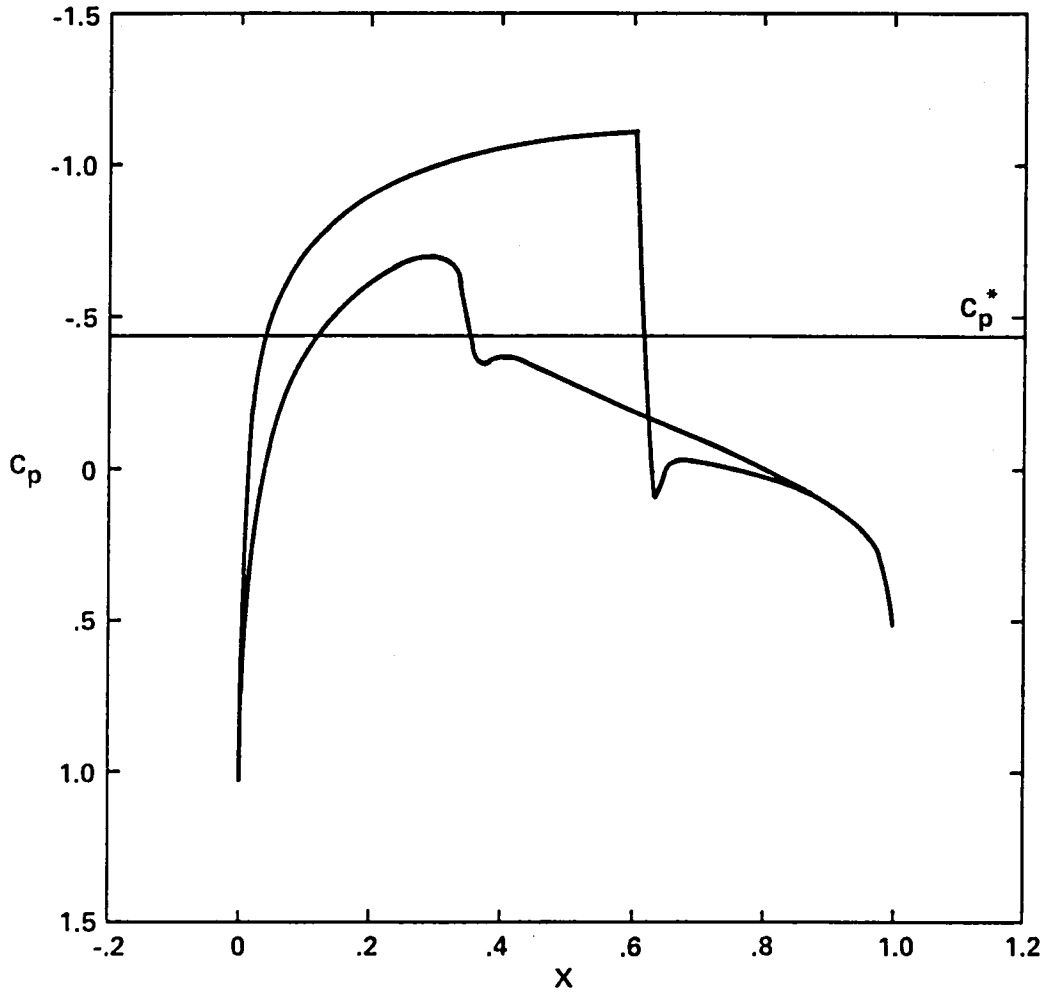
Fig. 4. Various solution features of the space shuttle at $M_\infty = 7.9$, 25° angle of attack and $Re_l = 6.07 \times 10^5$ per inch .

a) Surface streamline pattern.

b) Cross sectional view showing grid slice conforming to outer bow shock.

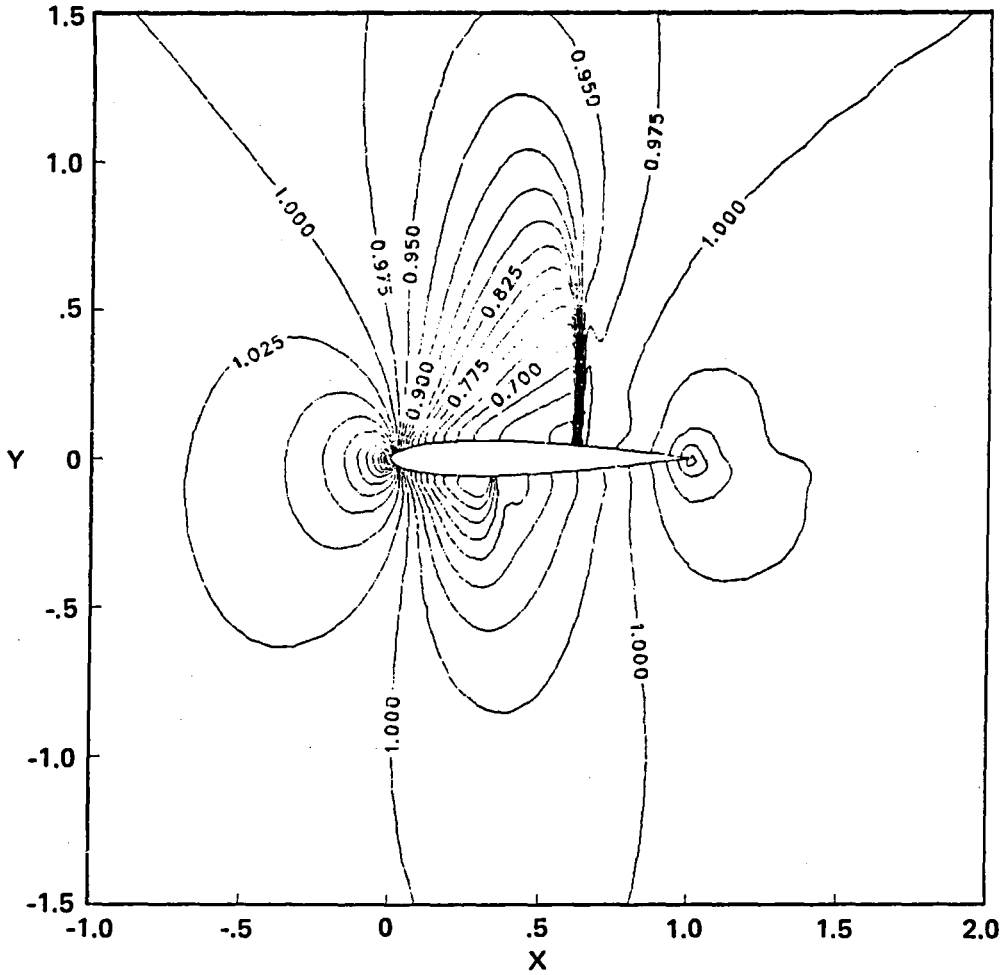
c) Pressure contours in vicinity of canopy region showing embedded bow shock.

d) Cross sectional flow vectors.



a) Surface pressure distribution, C_p.

Fig. 5. Calculation of inviscid transonic flow using flux vector splitting class of finite difference scheme for NACA0012 airfoil at $M_\infty = 0.8$ and 1.25° angle of attack.



b) Mach contours.

Fig. 5. Concluded.

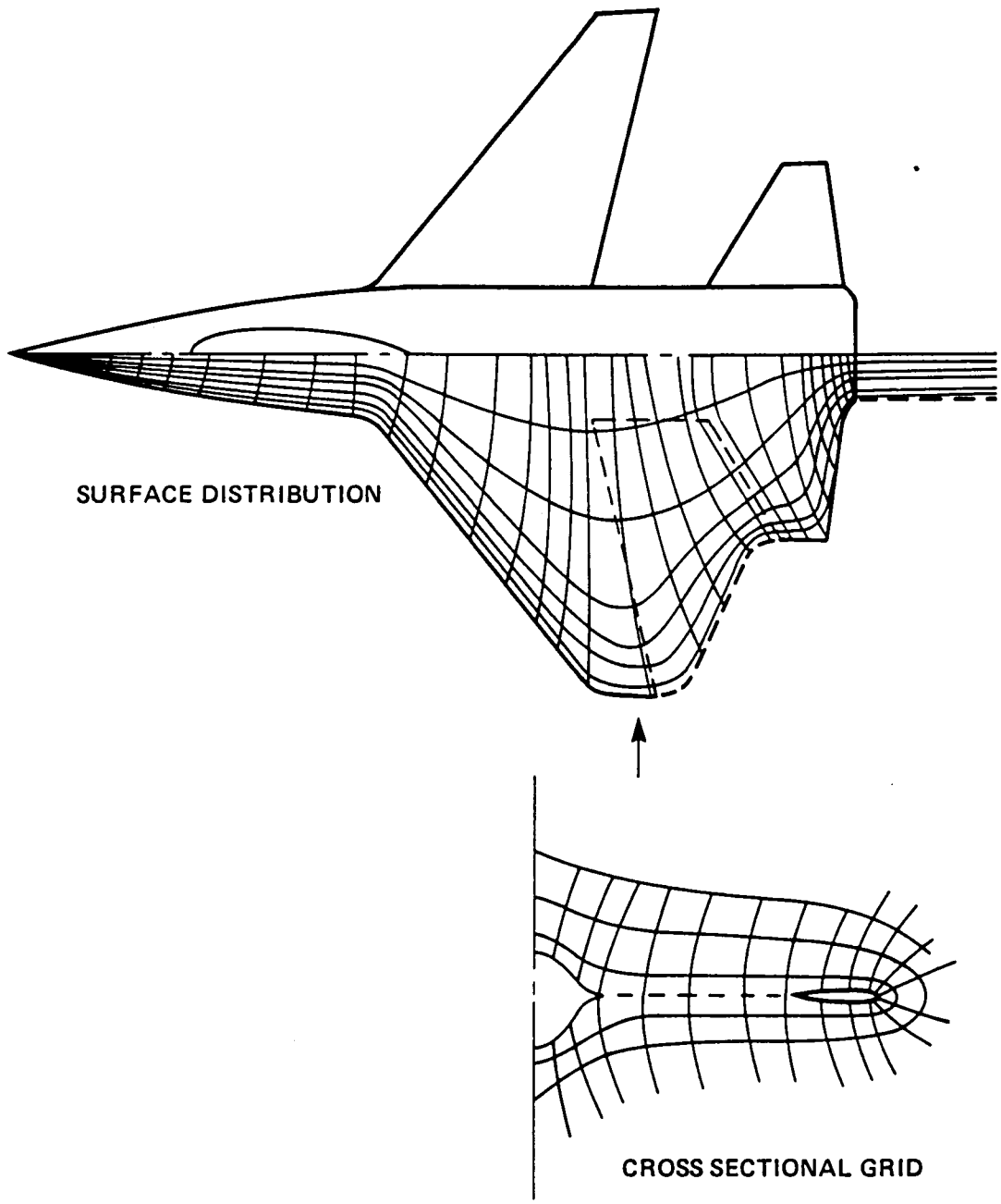


Fig. 6 Sketch illustrating the use of a single body-conforming warped-spherical grid with cuts.

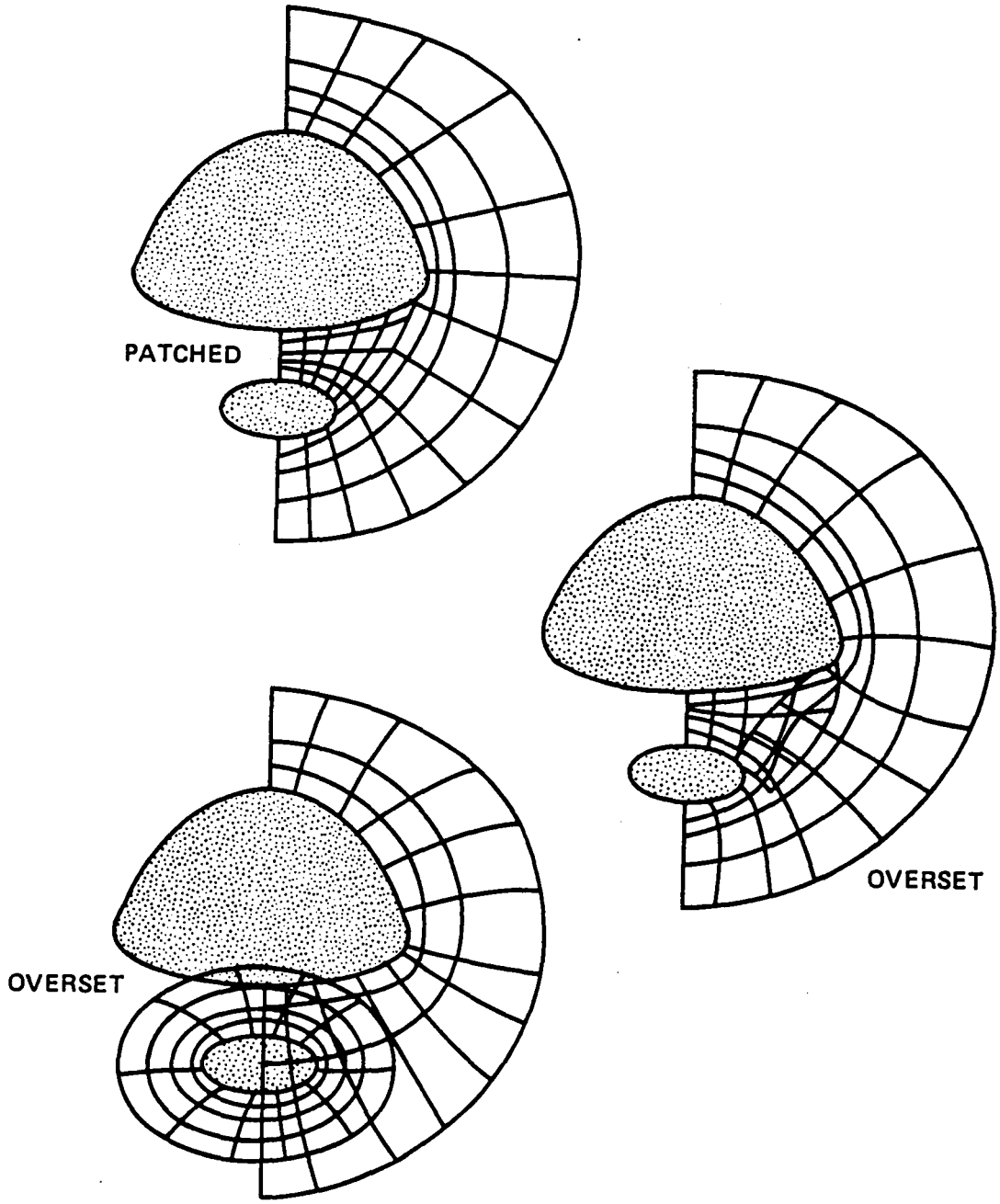
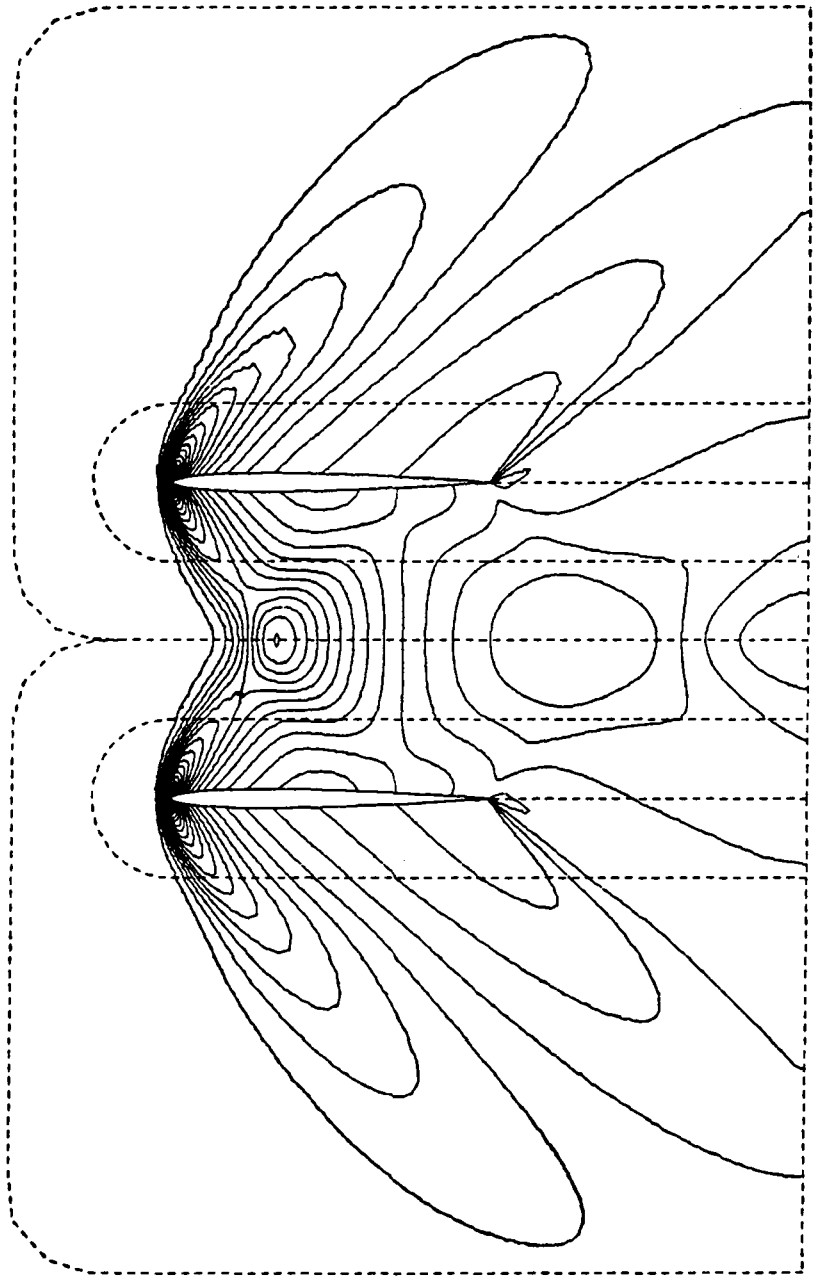
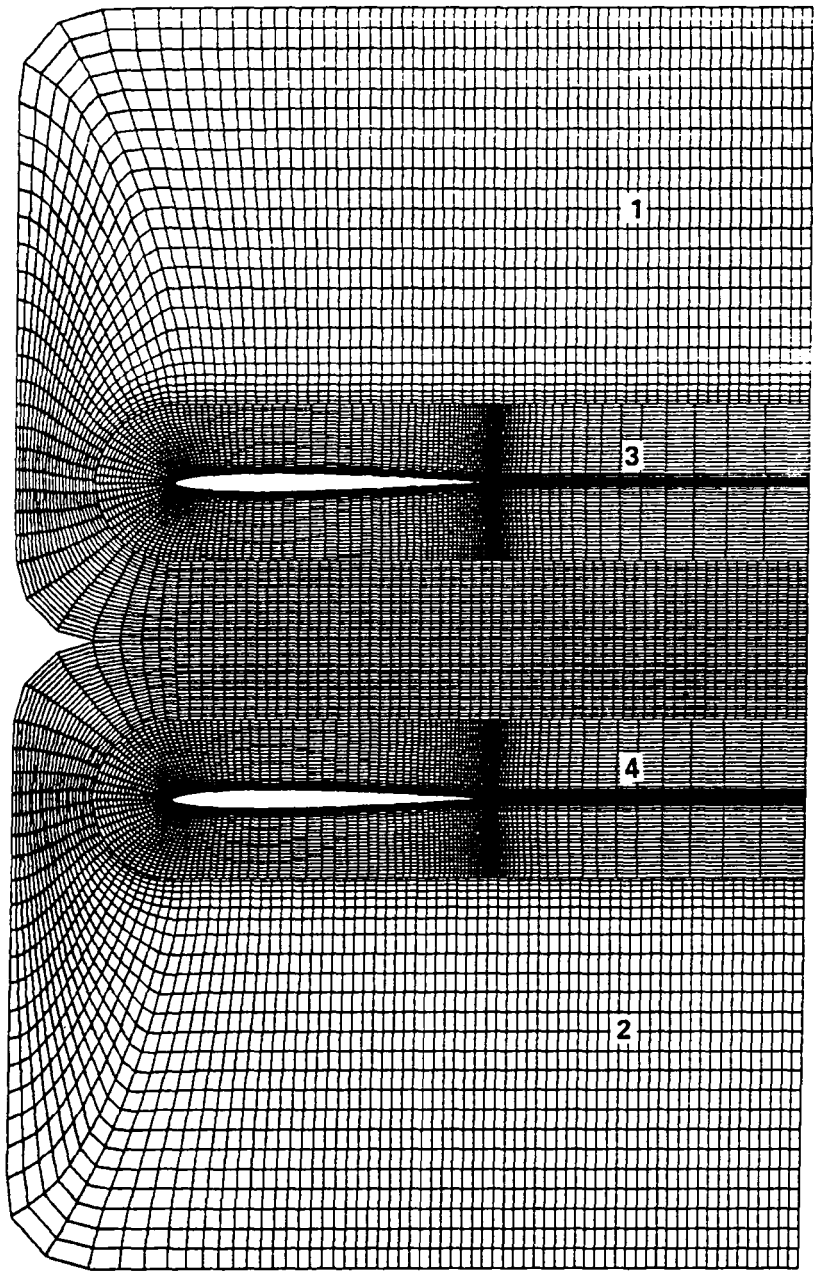


Fig. 7. Sketches of a possible composite grid formed from patched and overset grids about a typical multiple body combination.



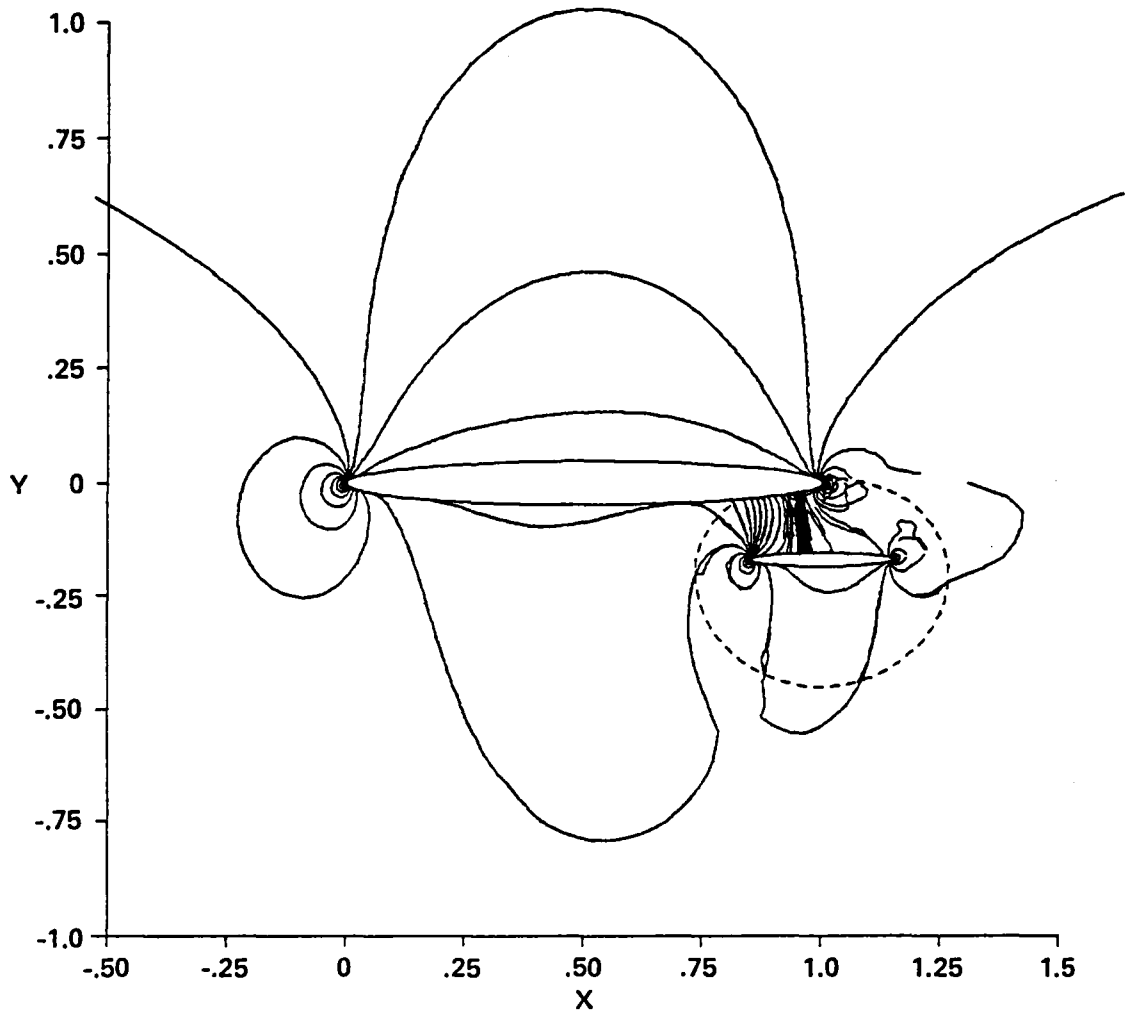
a) Mach contours

Fig. 8. Inviscid supersonic flow about a biplane computed using a first order Osher scheme with patched grids, $M_\infty = 1.5$.



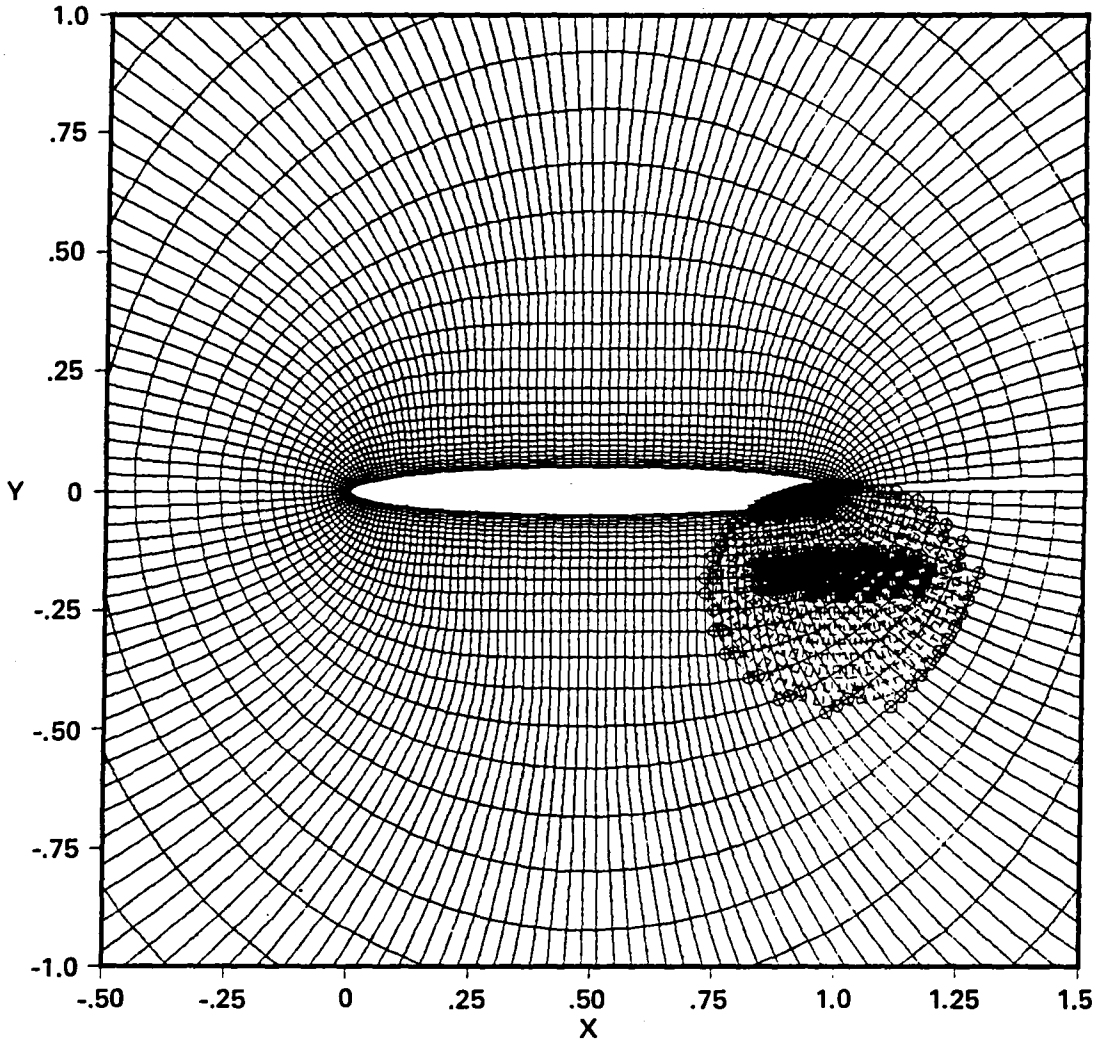
b) Composite grid formed using four patches.

Fig. 8. Concluded.



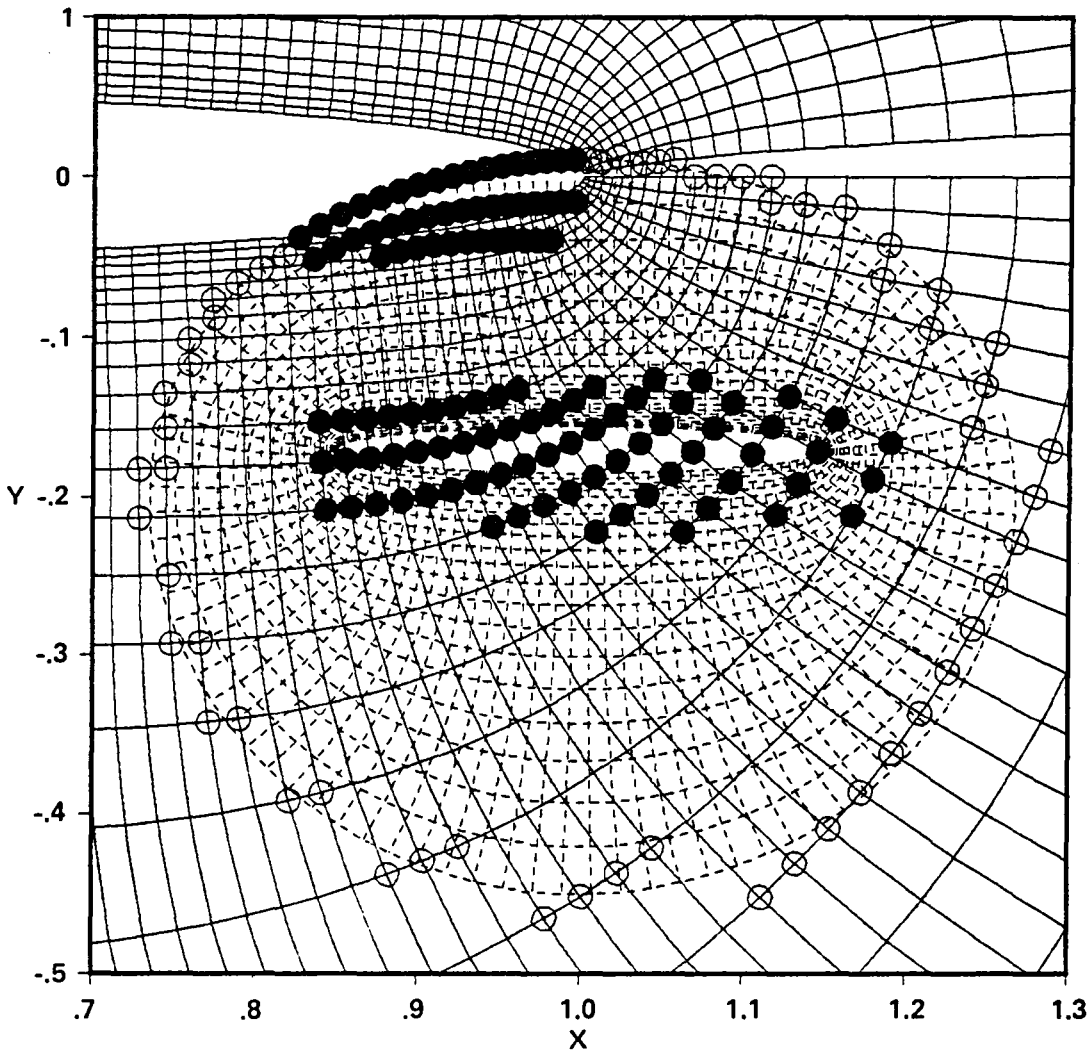
a) Mach contours

Fig. 9. Inviscid transonic flow about a generic airfoil detached flap combination using the implicit algorithm and overset grids, $M_\infty = 0.7$.



b) Composite grid formed using two overset grids.

Fig. 9. Continued.



c) Detail of minor grid showing all blanked points and interpolation points for the outer boundary of the minor grid. (Hole interpolation points not shown).

Fig. 9. Concluded.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-86674		

4. Title and Subtitle	5. Report Date
DEVELOPMENTS IN THE SIMULATION OF COMPRESSIBLE INVISCID AND VISCOUS FLOW ON SUPERCOMPUTERS	February 1985
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Joseph L. Steger and Pieter G. Buning	85121
	10. Work Unit No.
9. Performing Organization Name and Address	
Ames Research Center Moffett Field, Calif. 94035	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code 505-31-01

15. Supplementary Notes

Point of Contact: Pieter G. Buning, Ames Research Center, MS 202A-1, Moffett Field, Calif. 94035
(415) 694-5194 or FTS 464-5194

16. Abstract

In anticipation of future supercomputers, finite difference codes are rapidly being extended to simulate three-dimensional compressible flow about complex configurations. This paper reviews some of these developments. The importance of computational flow visualization and diagnostic methods to three-dimensional flow simulation is also briefly discussed.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computational fluid dynamics Supercomputers	Unlimited Subject Category – 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	28	A03

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

End of Document