I-T. Chiu
R. H. Pletcher
J. L. Steger

P-156

# On Computations of the Integrated Space Shuttle Flowfield Using Overset Grids

July 1990

Submitted to
National Aeronautics and Space Administration
Ames Research Center
Moffett Field, Calif. 89055
NASA Cooperative Agreement No. NCC2-476

(NASA-CR-186927) ON COMPUTATIONS OF THE INTEGRATED SPACE SHUTTLE FLOWFIELD USING OVERSET GRIDS Final Report (Iowa State Univ. of Science and Technology) 156 p
N90-27734
Unclas
CSCL 22B G3/16 0302643

Final Report

report

College of Engineering
Iowa State University

HTL-53, CFD-23
Project 1964
ISU-ERI-Ames-91040

I-T. Chiu
R. H. Pletcher
J. L. Steger

# On Computations of the Integrated Space Shuttle Flowfield Using Overset Grids

## Final Report

*R. H. Pletcher, Principal Investigator*
Project: Numerical Modeling of Propulsive
Effects in Complex Flow Fields
6/1/87–9/30/90

Supported by
National Aeronautics and Space Administration
Ames Research Center
NASA Cooperative Agreement No. NCC2-476

Heat Transfer Laboratory
Department of Mechanical Engineering
Computational Fluid Dynamics Center

engineering
research institute
iowa state university

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

Numerical simulations using the thin-layer Navier-Stokes equations and chimera (overset) grid approach were carried out for flows around the integrated space shuttle vehicle over a range of Mach numbers. Body-conforming grids were used for all the component grids. Testcases include a three-component overset grid — the external tank (ET), the solid rocket booster (SRB) and the orbiter (ORB), and a five-component overset grid — the ET, SRB, ORB, forward and aft attach hardware, configurations. The results were compared with the wind tunnel and flight data.

In addition, a Poisson solution procedure (a special case of the vorticity-velocity formulation) using primitive variables was developed to solve three-dimensional, irrotational, inviscid flows for single as well as overset grids. The solutions were validated by comparisons with other analytical or numerical solutions, and/or experimental results for various geometries. The Poisson solution was also used as an initial guess for the thin-layer Navier-Stokes solution procedure to improve the efficiency of the numerical flow simulations. It was found that this approach resulted in roughly a 30% CPU time savings as compared with the procedure solving the thin-layer Navier-Stokes equations from a uniform free stream flowfield.

# ACKNOWLEDGMENTS

The authors gratefully acknowledge the support provided for this study by the NASA Ames Research Center through the Cooperative Agreement No. NCC2-476. The authors also wish to thank members of the Shuttle Simulation Group — Dr. Pieter G. Buning, Dr. Shigeru Obayashi and Dr. Yehia M. Rizk, of NASA Ames Research Center for their help in providing the grids and making the plots and suggestions in this study.

# NOMENCLATURE

## Roman Symbols

| | |
|---|---|
| $a$ | speed of sound |
| $C_p$ | pressure coefficient $\left(\frac{p-p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2}\right)$ |
| $c_p$ | specific heat at constant pressure |
| $c_v$ | specific heat at constant volume |
| $e$ | internal energy |
| $E_t$ | total energy per unit volume |
| $\vec{f}$ | heat flux |
| $H$ | total enthalpy |
| $h$ | enthalpy |
| $h$ | time step |
| $I$ | identity matrix |
| $i, j, k$ | indices in tensor notation |
| $\vec{i}, \vec{j}, \vec{k}$ | unit vector in the $x, y, z$ direction respectively |
| $J$ | Jacobian |
| $k$ | thermal conductivity |
| $L$ | linear operator |
| $M$ | Mach number |
| $N$ | number of cycles |
| $p$ | pressure |
| Pr | Prandtl number $\left(\frac{\mu c_p}{k}\right)$ |
| $\vec{Q}$ | velocity vector |
| $R$ | right hand side operator |
| $R$ | gas constant |
| Re | Reynolds number |
| $s$ | entropy |
| $T$ | temperature |

[redacted]

| | |
|---|---|
| $U$ | $\xi$ direction contravariant velocity |
| $u$ | x component of velocity |
| $V$ | $\eta$ direction contravariant velocity |
| $\vec{V}$ | velocity vector |
| $v$ | y component of velocity |
| $W$ | $\zeta$ direction contravariant velocity |
| $w$ | z component of velocity |
| $x$ | streamwise coordinate direction |
| $y$ | spanwise coordinate direction |
| $z$ | normal coordinate direction |

## Greek Symbols

| | |
|---|---|
| $\gamma$ | ratio of specific heats |
| $\delta_{ij}$ | Kronecker delta function |
| $\delta$ | central difference operator |
| $\overline{\delta}$ | mid-point central difference operator, subscript gives direction to difference, e.g., $\overline{\delta}_x u = u_{i+1} - u_{i-1}$ |
| $\delta^b$ | backward difference operator |
| $\delta^f$ | forward difference operator |
| $\xi$ | transformed coordinate |
| $\eta$ | transformed coordinate |
| $\zeta$ | transformed coordinate |
| $\kappa$ | coefficient of thermal conductivity |
| $\lambda \lambda_1 \lambda_2$ | relaxation parameters |
| $\mu$ | viscosity |
| $\rho$ | density |
| $\vec{\mu}$ | viscous stress tensor |
| $\vartheta$ | compressibility source term |
| $\vec{\omega}$ | vorticity vector |
| $\omega_1, \omega_2, \omega_3$ | vorticity components |
| $\omega$ | overrelaxation parameter |

## Subscripts

| | |
|---|---|
| $j$ | x or $\xi$ index |

[redacted]

| | |
|---|---|
| $k$ | y or $\eta$ index |
| $l$ | z or $\zeta$ index |
| $n$ | normal direction |
| $r$ | reference |
| $t$ | tangential direction |
| $x$ | derivative in x direction |
| $y$ | derivative in y direction |
| $z$ | derivative in z direction |
| $\xi$ | derivative in $\xi$ direction |
| $\xi$ | $\xi$ direction operator |
| $\eta$ | derivative in $\eta$ direction |
| $\eta$ | $\eta$ direction operator |
| $\zeta$ | derivative in $\zeta$ direction |
| $\zeta$ | $\zeta$ direction operator |
| $\infty$ | free stream value |

## Superscripts

| | |
|---|---|
| $n$ | time level |

## Other Symbols

| | |
|---|---|
| $\nabla$ | gradient operator |
| $\nabla^2$ | Laplacian operator |
| $\nabla$ | backward difference operator, subscript gives direction to difference, e.g., $\nabla_x u = u_i - u_{i-1}$ |
| $\Delta$ | forward difference operator, subscript gives direction to difference, e.g., $\Delta_x u = u_{i+1} - u_i$ |

# 1. INTRODUCTION

## 1.1 Objective

Over the years, many researchers have been searching for ways to simulate flows around complex geometries, such as full aircraft configurations (Jameson et al., 1986; Flores et al., 1987), and the integrated space shuttle configuration (Szema et al., 1988; Buning et al., 1988, 1989). This goal demands the combined use of different technologies developed for computational fluid dynamics and other fields, like grid generation, flow solver, and flow visualization through the use of dedicated graphics workstations. The final results must be capable of capturing the significant flow features, and be verified with experimental data. This report describes the computation of the flow around the launch configuration of the integrated space shuttle vehicle as well as ways to improve the efficiency of numerical methods for flow simulations. The primary components of the configuration consist of the external tank (ET), the solid rocket booster (SRB), and the orbiter (ORB); and the secondary parts include the forward and aft attach hardware which link the external tank and the orbiter. The computed solutions were generally in good agreement with the wind tunnel tests and the flight data. With this achieved, the flow simulation around the space shuttle geometry can be further refined to include the parts purposely excluded in the current research, for example, the ORB vertical tail, the ET/SRB attach ring, and the like. These were excluded because of limitations on modeling capability and computational resources. With increasingly faster computers and continuous development of computational fluid dynamics, it will not be long before the quasi-static numerical simulation of the flow around complex geometries through different flow regimes (or Mach numbers) becomes a routine job for the major aircraft manufacturers and is integrated into the design loop. Hopefully, this technology will someday advance to a level that real-time simulation becomes a reality and the information rendered by the computer can help in real-time control on increasingly complicated flying machines.

The primary objective of computer simulations of the space shuttle is to supplement the available experimental and flight data which suffer from inadequacies due to scaling effects, wind tunnel wall-interference effects, sting interference effects, instrumentation limitations and the difficulty of safely obtaining valid flight data. Since the computer simulations are quite flexible, in that they allow for easy reconfiguration of the shuttle geometry, subject to the limitations of the flow solver and the difficulties of gridding, the simulations can easily be carried out for different geometries and different flow conditions. In this study, the simulation had been carried out from a grid consisting only of the ET, SRB and ORB to a more refined model with the addition of the forward and aft attach hardware, and Mach numbers ranging from 0.6 to 2.0. The results from the simulations have already helped in the diagnostics of the damaged thermal protection system on the ORB surface on one of the shuttle missions (Li, 1989). Li reported using PLOT3D (Buning and Steger, 1985), a graphics tool developed by Buning of NASA Ames Research Center, to draw particle traces based on the computed solutions from the damaged area to determine the possible debris path during the launching period. Other possible applications from the numerical simulations are to predict the aerodynamic behavior during the emergency abort maneuvers or to determine the escape path for the astronauts. All of these applications are either impossible, impractical or extremely expensive to evaluate in the wind tunnel or in real flights; while for computer simulations, once the solution is obtained, the complete three-dimensional flowfield can be analyzed with the aid of modern graphics software and hardware to provide additional insight that might have otherwise been neglected, difficult, or impossible to capture in an experimental setup. The computer simulation can also be used as an aid to check the validity of the aerodynamic data base for the space shuttle; for example, the computed wing load from the experimental data may be different from what was observed in the real flight; thus, the numerical solution provides a third check for the data base. In the design or modification of the shuttle geometry, the relatively low cost computer simulation (as compared with the cost of the wind tunnel test) allows room for designers to conduct various numerical experiments with a reasonable cost as is currently being done at Rockwell International for the ET/SRB attach ring.

## 1.2 Approach

This research was carried out as part of the space shuttle flow simulation project in the Applied Computational Fluids Branch (RFA) at NASA Ames Research Center. In one phase of the work, the thin-layer Navier-Stokes equations were solved using an implicit approximately factored finite-difference procedure (Steger et al., 1986; Ying et al., 1986) for the flows around all the components of the integrated space vehicle during its ascent mode for various nominal and abort flight conditions. The Baldwin-Lomax turbulence model was used to calculate the turbulent eddy viscosity.

Due to the complexity of the geometry, the chimera (Steger et al., 1983; Benek et al., 1985, 1987; Dougherty et al., 1985) composite-grid approach was chosen to manage data communication between different grids. For each component (the external tank, the solid rocket booster, the orbiter and the forward and aft attach hardware) the body-conforming grid was generated independently and overlaid with each other to form the composite grid for the shuttle launch configuration. More details on the grid will be given in the chapter on the chimera approach.

Since the geometry of the entire space vehicle is very complicated, it is extremely difficult, if not impossible, to model every possible detail of the entire vehicle. Thus, some simplification was made on the geometry, such as the elimination or idealization of the ET/ORB forward and aft attach hardware, the elimination of the ET/SRB attach ring, ET sway braces, the main fuel and oxidizer feed lines, etc. In addition, certain engineering approximations were made. For instance, stings have been used to represent plumes. This is a reasonable approximation for supersonic free stream flow due to the limited upstream influence; while at lower Mach numbers, the shuttle is in the lower, more dense atmosphere, and the plume expansions tend to be greatly reduced.

In the first stage, a coarse grid (about 250,000 points for the whole shuttle configuration) was used for a flow at a free stream Mach number of 2.0. Since this was a supersonic flow, the required computer time was relatively small as compared to a subsonic flow due to the lack of upstream signal propagation, and was a proper testcase for testing the feasibility of the entire numerical procedure.

In the second stage, some modifications were made in the geometry. For example, the ET/ORB attach hardware was added with some idealization of the geometry, and the ET sting was removed. Various Mach numbers and different angles of elevon deflection for the ORB were computed and compared with the experimental and flight data. All the calculations in the first two stages were carried out on the NASA Numerical Aerodynamic Simulation (NAS) Cray 2 computer.

Though it is possible to simulate flows around complex geometries such as the one presented in this research, the required computer time (6 to 20 hours on the Cray 2 computer for the shuttle geometry) is still inhibitively high even for large aircraft manufacturers. This prompted the research on a faster procedure for the numerical simulation over complex geometries. At this point, a vorticity-velocity formulation was implemented to obtain a rough estimate of the flowfield[1] and the solution was then fed back to a more accurate solver for further calculations. The reason that this procedure may reduce the overall computing effort is that most flow solvers utilize a great amount of computer time in settling out the transient state; therefore, if a rough estimate of the flowfield is available, it is possible to reduce the computer time in obtaining the flow solution. The other possible use of the formulation is to evaluate a grid without putting in great effort just to find out that the grid is not adequate for the intended flow geometry. The solution of this formulation was carried out for several generic geometries (sphere, ellipsoid, and the external tank) to verify the validity of the formulation. Finally a solution for the entire shuttle vehicle was presented. All solutions from this formulation were compared with either the analytical solution or the solution obtained from the thin-layer Navier-Stokes equations as mentioned above.

---

[1] Only inviscid flows were solved using the vorticity-velocity formulation.

## 1.3 Literature Review

Navier-Stokes equations or its simplified approximations, such as the Euler equations which neglect the viscous terms, have been used to solve for flows in many engineering applications. Depending on the choice of the dependent variables, the mathematical formulations of the Navier-Stokes equations (or its simplified approximations) may be divided into the following categories (Guj and Stella, 1988),

- vorticity/stream-function (Fromm and Harlow, 1963; Benjamin and Denny, 1979), which in three-dimensional flows extends to vorticity-vector potential (Mallinson and de Vahl Davis, 1973; Richardson and Cornish, 1977)
- vorticity/velocity
- primitive variables.

The focus here will be on the formulations suitable for solving three-dimensional flows. Thus, the review on solving the two-dimensional flows using the vorticity/stream-function method is left out; the reader is referred to Roache (1972) for a good review on the method using this formulation.

### 1.3.1 Vorticity/velocity-vector potential

The vorticity/stream-function formulation has been one of the popular approaches used for two-dimensional flow simulations. While the three-dimensional counterpart of the stream-function, referred to as the vector potential, or velocity-vector potential, has been known for over a century (Helmholtz's decomposition theorem, 1858, 1867), it was not successfully implemented in computations until 1967 (Aziz and Hellums, 1967). The reasons for such a long delay were the difficulty in finding the proper boundary conditions for the vector potential and the demands on computer resources since this formulation requires solutions for more variables than the primitive variable formulation. Aziz and Hellums (1967) in their implementation utilized boundary conditions similar to those formulated for a general hydrodynamic flowfield by Hirasaki and Hellums (1968) to study three-dimensional laminar natural convection in enclosures. The study of Aziz and Hellums showed that the vorticity/vector potential approach can lead to faster and more stable convergence than for the comparable primitive variable formulation. Although the boundary conditions presented in Hirasaki and Hellums (1968) were fairly complete, their complexity in treating through-flows rendered the method useless in these situations. Later, Hirasaki and Hellums (1970) proposed a simplified boundary condition with the introduction of a scalar potential to account for the through-flow velocities. This formulation, termed "dual potential" (Chaderjian and Steger 1983), has been used to solve three-dimensional natural convection in enclosures (Mallinson and de Vahl Davis, 1973; Ozoe et al., 1976, 1977, 1979, 1985) and three-dimensional flows in ducts (Aregbesola and Burley, 1977; Wong and Reizes, 1984, 1986).

Due to the added variable, i.e., the scalar potential, in Hirasaki and Hellums (1970), the vorticity/vector potential approach suffered from extra demands on computer time and storage. Wong and Reizes (1984) used the normal component of the specified inlet velocity vector in place of the scalar potential to reduce the requirement of computer storage. Calculations using this formulation for flows in a constant cross section duct were performed for a wide range of Reynolds numbers. Later, Wong and Reizes (1986) extended this approach to flows in multiply connected regions.

Efforts have also been made to extend the vorticity/vector potential formulation to deal with compressible flows. Hafez and Lovell (1981, 1983, 1988) devised an entropy and vorticity correction procedure for the potential/stream-function formulation. They showed details of the treatment of shocks and wakes, and the solution was compared with Euler solutions. Rao et al. (1987, 1989) combined the boundary-layer equations with the vorticity/vector potential formulation to do viscous-inviscid interaction. The vorticity was injected from the boundary-layer edge into the potential flow region to obtain the viscous effect. Gegg (1989) extended the vorticity/vector potential method to solve through-flow problems with heat transfer.

### 1.3.2 Vorticity/velocity

Only a handful of research projects have been carried out using the vorticity/velocity formulation for flow simulations. The earliest calculation using this formulation was reported in Fasel (1976). At each time step, Fasel solved two Poisson equations, derivable from the definition of vorticity, for the components of the velocity vector for a two-dimensional flow. This approach was applied to the study of the stability of boundary-layers. Dennis et al. (1979) extended the two-dimensional vorticity/velocity approach to three-dimensional steady flows by solving the Navier-Stokes equations in a cubical driven box. His approach bore some similarity to that of Aziz and Hellums (1967). The main difference was that three equations connecting the velocity and vorticity components were employed instead of the vector potential. These three equations, together with the three vorticity transport equations, form six simultaneous second-order partial differential equations to be solved. Gatski et al. (1982) applied compact finite-difference schemes to the vorticity-velocity form of the two-dimensional Navier-Stokes equations. The numerical solutions were obtained for driven cavity flows. Fasel and Booz (1984) investigated the axisymmetric supercritical Taylor vortex flow for a wide gap. Farouk and Fusegi (1985) studied the natural and forced convection and heat transfer in a two-dimensional annulus. A coupled solution procedure was used for solving simultaneously the dependent variables using a block tridiagonal matrix inversion algorithm. The formulation was found to be fairly stable over a large range of Reynolds and Rayleigh numbers. Orlandi (1987) solved high-Re flows using a block ADI method which strongly coupled field equations and boundary conditions and satisfied the continuity equation without requiring an iterative procedure. Driven cavity and backward facing step flows were computed and verified with other numerical solutions and/or experimental results. Osswald et al. (1987) solved the three-dimensional unsteady Navier-Stokes equations using a direct inversion procedure for a shear driven viscous flow within a cubical box. The three-dimensional vorticity transport equation in their procedure was solved using an approximate factorization method which required the inversion of only scalar tridiagonal matrices, rather than the usual block-tridiagonal systems. Guj and Stella (1988) computed two-dimensional incompressible flows in driven cavity and over a backward-facing step using a scalar ADI method. Speziale (1987) elaborated on the advantage of the vorticity/velocity formulation in a non-inertia coordinate system. He showed that the non-inertia effects, arising from both the rotation and translation of the frame of reference relative to an inertia frame, only enter into the equation through the implementation of initial and boundary conditions. This is in contrast to the primitive variable formulation, where non-inertia effects appear directly in the momentum equations in the form of Coriolis and Eulerian accelerations which may give rise to a variety of numerical problems (Williams, 1969). Considering the relatively short list of references for this formulation, it is evident that the vorticity/velocity approach needs further study. The major reason for the lack of research on this formulation may be attributed to the extra variables needed which in turn translate into extra demand on computer storage. For three-dimensional flows, memory requirements tend to govern the feasibility of implementing a particular numerical scheme. It is therefore understandable that numerical schemes which deal with a smaller number of variables have been strongly favored.

Of all the studies mentioned above for the vorticity/velocity formulation, none were conducted for compressible flows. In the current study, the Crocco relationship was used in place of the vorticity transport equations; however, it is valid only for inviscid flows. Bernoulli's equation for compressible, non-isentropic flows was used to account for density changes. Although only irrotational flow was computed in the current research, the proposed scheme has the potential of treating inviscid rotational flows. For the computation of viscous flows, the vorticity/velocity procedure implemented in the current research can be coupled with the boundary-layer (as in Rao et al., 1987, 1989) or Navier-Stokes equations to solve the viscous flow. The viscous effects in the inviscid region are accounted for through the injection of vorticity from the boundary-layer edge or the computational domain of the Navier-Stokes equations.

### 1.3.3 Primitive variables

Numerous methods using primitive variables have been developed to solve the Navier-Stokes equations. However, only methods dealing with improving the efficiency of the numerical flow simulation and techniques to treat complex geometries will be given attention here. Reviews of other related numerical methods for solving the Navier-Stokes equations can be found in Shang (1985) and Holst (1987). An outlook for computational aerodynamics was given by Chapman (1979).

With the debut of increasingly powerful computers, the quest for numerical flow simulations around real-world geometries, e.g., a complete aircraft, is within reach for those with access to supercomputers. Jameson and Baker (1987) computed an Euler solution around a Boeing 747-200 using an unstructured grid consisting of tetrahedral meshes. Obayashi (1987) carried out a Navier-Stokes solution for the ONERA M-5 model (a wing-fuselage-tail geometry) using a single grid system. Flores et al. (1987) reported a zonal approach to obtain a transonic flow solution to the Navier-Stokes equations for a fighter-like configuration, while Buning et al. (1988) used an overset grid approach to solve the thin-layer Navier-Stokes equations for the integrated space vehicle launching configuration over a range of Mach numbers. Of those reported calculations for complex geometries, the grid systems used can roughly be divided into the following types:

- single grid
- multiple grids — can further be divided into
    1. zonal (patched) grid
    2. overset grid
- adaptive grid.

Often, a single grid generated for a complex geometry contains overly skewed meshes which in turn give rise to inaccurate solutions. Thus, a significant amount of effort is usually needed to modify the existing grid generater to yield acceptable grids. In addition, the single grid approach does not have the flexibility of selective grid refinement and may require more points to resolve the flow than a comparable multiple-grid method. While the multiple grid approach is flexible, it is not without its own problems. For the patched grid, each subgrid is generated subject to boundary constraints placed by the neighboring grids, and multiple grids must be interfaced and managed (Steger and Benek, 1986). Rai (1986a, 1986b) developed a scheme such that the zonal boundaries were treated in a conservative manner so that the discontinuities could move freely across these boundaries. Many calculations were carried out using patched grids to resolve gradients, treat moving boundaries (Rai, 1985), and complex geometries (Eberle and Misegades, 1986).

For the overset grids, the grid does not need common boundaries between subgrids, but rather, a common or overlap region is required to allow ways for matching the solutions across boundary interfaces. Usually, interpolation is used for the solution matching among subgrids; however, this will not ensure conservation of flux quantities, and inaccuracies can occur in shock capturing. Each subgrid in this approach is generated independently, which, in turn, reduces a complex grid generation problem into a series of simple ones. Atta and Vadyak (1982) devised this approach to solve the potential equation for two- and three-dimensional flows. Steger et al. (1983) and Benek et al. (1983) developed a chimera scheme to solve the two-dimensional Euler equations. Subsequently, the scheme was extended to treat three-dimensional flows (Benek et al., 1985). Buning et al. (1988) used the scheme to solve for flows around the integrated space shuttle vehicle. A related application by Wedan and South (1983) employed a Cartesian mesh in which the body was embedded.

The adaptive grid method allows the mesh to evolve with the solutions and does not need, for the initial mesh, to anticipate accurately the large gradient regions. An advantage of this approach is that the grid points are efficiently used and human intervention is not needed to place the grid points in regions of large gradients. Gnoffo (1982) modeled the mesh as a network of springs with the spring constants determined by the gradient of flow variables. Ghia et al. (1983) coupled the grid-evolution equation to the flow equation by requiring that the coefficients of the convective terms be minimized. Brackbill (1982) and Saltzman and Brackbill (1982) used a variational technique to produce grid-evolution equations. Berger and Oliger (1984) developed a dynamic refinement method which embeds finer and finer grids to resolve flow gradients.

Although it is feasible to carry out flow simulations on the current generation of supercomputers for complex geometries like a complete aircraft, the work still demands a significant amount of computer resources. Therefore, it is still necessary to improve the rate of convergence for the numerical algorithms. Van Dalsem and Steger (1985) implemented a "fortified" Navier-Stokes approach, in which solutions to the subset equations, e.g., the boundary-layer equations, were used to add forcing terms to the Navier-Stokes algorithm in the proper flow regions. This approach was found to improve the efficiency as well as the accuracy of a given Navier-Stokes algorithm. Van Dalsem and Steger (1986) solved the boundary-layer equations on a fine grid near the wall to resolve the viscous gradients near the wall. The boundary-layer solution was then used as a forcing function and interpolated to the coarse grid solved by a Navier-Stokes algorithm. They reported a 20-fold increase for the rate of convergence in their testcases.

Another approach is to use different sets of equations for different flow regions, e.g., the boundary-layer equations for the boundary-layer region and the potential equation for the potential flow far from the body. Various combinations of equations are possible. Whitfield et al. (1981) used the Euler equations to obtain the inviscid flow solution and boundary-layer equations in the viscous layer. Halim and Hafez (1984) developed a scheme in which the stream function was used to calculate the inviscid flow and the partially parabolized Navier-Stokes equations were used for the near wall shear layer. The third and the most commonly used approach is the boundary-layer equations for the viscous layer and the full-potential equations for the inviscid flow. The two solutions are matched by iterating for the displacement thickness. Most of these have concentrated on solving the two sets of equations simultaneously to improve the rate of convergence (Lee and Pletcher, 1986). Although the approach has taken into account the physics at different flow regions and can possibly save a significant amount of computer time if each flow region is resolved properly, the complexity of treating multiple solution algorithms and domain interfaces usually makes the coding more difficult.

# 2. CHIMERA APPROACH AND GRID TOPOLOGY

## 2.1 Introduction

For complex geometries, generating the grid for the flow solver is itself a difficult task. Though it is possible to generate a single grid for a complex geometry, the resultant grid is most often overly skewed in one direction or another, or doesn't have the needed clustering to resolve the flowfield in regions of rapid change. A natural way to overcome this difficulty is to divide the complex shape into several simple ones and generate the grid about these simple shapes, then either patch them together, the so-called patch grid, or overlay one on top of the others, as an overset grid. Combined with inter-grid communication in the flow solver, the patched grid or overset grid can be used to treat complex geometries. Since this research was done entirely with overset grids, the details of the gridding are described only for the overset grid. The overset grid approach used in this research was first devised by Steger et al. (1983) and given the name, the chimera approach, after the Greek legendary creature that was compounded of incompatible parts which signifies that the chimera approach can take incompatible grids (i.e., no common boundaries between different grids) and "glue" them together to be solved by the flow solver.

Although drawbacks exist in the composite grid, patched or overset, like difficulties in accurately passing boundary data in-between sub-grids as well as finding the interfacing information at the boundaries that separate different component grids, several advantages of applying the chimera approach outweigh the concerns of the drawbacks mentioned. First, the chimera approach does not require common boundaries between component meshes. Due to this characteristic, the component grid can be generated separately, thus degenerating the complexity of grid generation into a combination of a series of simple ones. Furthermore, changes of some component grids do not usually involve changes of other component grids, thus allowing more flexibility than other approaches in constructing a grid for a complex geometry. This approach also saves the effort of gridding for a complex geometry since the chimera approach allows for arbitrarily adding or subtracting component grids. For instance, the gridding for the complex geometry can start with a simple one for initial testing and gradually be refined to a more accurate one by adding more component grids without regenerating from scratch the whole grid for the complete configuration. Besides, each component grid can be tested individually and added later when it is good enough or needed. The second feature of the chimera approach is that the flow simulation is done in sequence for component grids. This approach offers a savings in memory usage for solving flows around a complex geometry since it only requires memory enough to handle the largest component grid. The nature of solving each component grid in sequence also suggests the possibility of using different schemes, e.g., different set of equations, different time steps, etc., for different components. This opens many possibilities for enhancing the rate of convergence. For example, in the case of steady flow, it is possible to carry out more iterations for the component grid having the slower rate of convergence or using different time steps for different component grids. The chimera approach is also readily available for multitasking if sizes of the component grids are roughly the same due to its "separate and conquer" approach. Or when the complex geometry consists of one large grid and several much smaller grids, it is possible to carry out the computation for the large grid on one processor and the rest of the smaller grids in sequence on another processor or processors.

## 2.2 Chimera Approach

Several important concepts and implementation details underlying the chimera scheme are briefly described in this section. For more details the reader is referred to Steger et al. (1983), and Benek et al. (1983, 1985–1987). The chimera scheme involves the composite of the overlapping grids (generated individually), and the intergrid communications. As each mesh is generated individually, some grid points in one mesh will inevitably fall within the body boundary of another grid or grids, thus creating one or more "holes" in the mesh. The ET grid shown in Figure 2.8a depicts the holes created by the presence of the ORB and the SRB; the points in the ET grid surrounding the ORB and the SRB are called the hole boundary points. The values of the flow variables at these points are interpolated from solutions on either the ORB or the SRB grid, thus creating a link between the ET grid and grids of the ORB or the SRB. All the points within the hole boundary (including the boundary itself) will not enter into the solution process via a flag (will explain later) in the flow solver to differentiate them from the field points. In Figures 2.8b and 2.8c, not only the hole boundaries but also the outer boundaries of the ORB and the SRB grids are used to establish links to other grids. The composite of the overlapped grids and the interpolation data at the interface boundaries among component grids are created by a code, named PEGASUS, developed at CALSPAN of AEDC (Arnold Engineering Development Center).

### 2.2.1 Hole creation

As explained above, the holes in a mesh are due to the presence of the solid bodies embedded in the mesh. To find the hole points as well as the hole boundary, the current implementation basically involves a two-step procedure. First, by introducing an imaginary rectangular box enclosing the embedded body, all the points that fall outside of the sphere with diameter equal to the diagonal of this rectangular box are considered field points. This method is fast, though somewhat crude, and cheap as compared with the method (will be explained below) used to find the hole points. Thus, it is used to filter out most of the points from the hole searching procedure. If the points tested fall within the sphere, they may be inside the embedded body. A more accurate method is needed to tell whether or not a point is a hole point. To clarify the basic idea underlying the hole searching procedure, a two-dimensional instead of a three-dimensional case is presented to avoid unnecessary confusion. In Figure 2.1, after the point, $P$, is tested and found to fall within the sphere mentioned above, the point nearest to $P$ is found on the surface of the embedded body, say $P_C$, and from this point, an outward normal, $\vec{N}$, is constructed. If the dot product, $\vec{N} \cdot \vec{R}_P < 0$, $P$ lies within the hole and a flag variable, IBLANK, is set to zero; otherwise, P is outside the hole and IBLANK is set to 1. The IBLANK variable is used by the flow solver to determine whether a point should enter into the solution process or not as illustrated by the equation below.

$$A \Delta Q = IBLANK * RHS \tag{2.1}$$

where A is the coefficient matrix, $\Delta Q$ the change in flow solution, and RHS the source term. Thus, for the hole points, the above equation reduces to

$$\Delta Q = 0 \tag{2.2}$$

and the values of the variables at the hole points are not changed in the solution process. After the hole points are found, the hole boundary points can easily be located by searching the IBLANK values of neighboring points. If any neighboring points have a zero IBLANK value, they are defined as hole boundary points and also assigned zero as their IBLANK values since they are updated from the embedded mesh and should not enter into the solution process. Figure 2.2 illustrates the searching procedure for the hole boundary points. Note that the updating procedure for the hole boundary and the outer boundary is explicit and may somewhat affect the convergence and stability of the numerical scheme.

### 2.2.2 Interpolation points

Values of the flow variables at the hole boundary points are interpolated from the embedded mesh. Thus, it is necessary to find the interpolation points on the embedded mesh from which the hole boundary values are interpolated. The procedure involves locating the point in the embedded mesh closest to each hole boundary point, and once the closest point is found, its pointers (array indices) are added to a list of such points to be used by the flow solver to update the variables at the hole boundary points. To reduce the effort in finding the closest point, each search is started from the point found to be the nearest in the previous search. The interpolation points for the outer boundary are found with the same procedure.

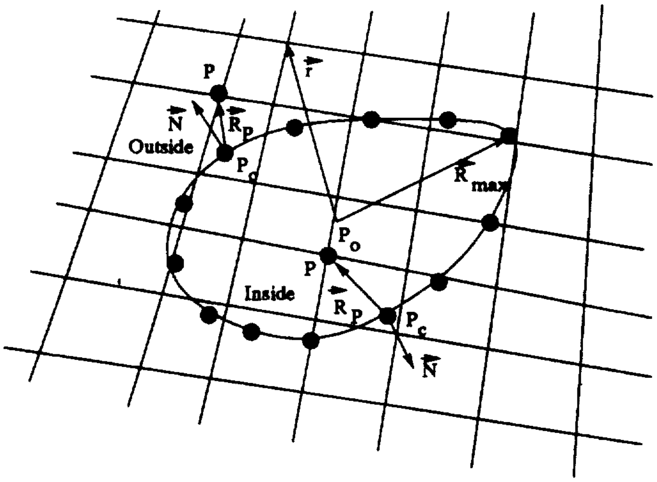


Figure 2.1: Method for locating points within a hole

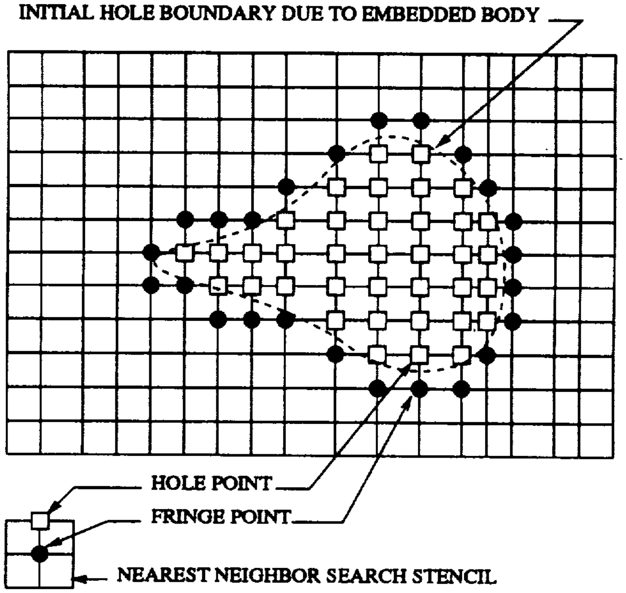


Figure 2.2: Hole boundary point construction

### 2.2.3 Interpolation

The transfer of information for the overset grid is through the hole and outer boundaries. The accuracy of the interpolation procedure influences the accuracy of the solution. Steger et al. (1983) reported a significant amount of mismatch in the vicinity of the shock/grid boundary intersection for an airfoil using single and two-grid configurations. The interpolation scheme, which was based on a Taylor series expansion, was suspected to be the cause. Mastin and McConnaughey (1984) showed that bilinear interpolation in two dimensions is better than Taylor series expansions when higher order derivatives of the solution are not important. In the current research, the trilinear interpolation of the form:

$$\phi = a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi\eta + a_5\xi\zeta + a_6\eta\zeta + a_7\xi\eta\zeta \tag{2.3}$$

was used. In the above equation, $0 \leq \xi, \eta, \zeta \leq 1$ are the coordinates of the point to be interpolated and $a_0$ to $a_7$ are computed based on the values at the points forming the interpolation stencil.

## 2.3 Grid Generation

Since the real geometry of the integrated vehicle is very complicated, it is impossible to include all the details in the computer flow simulation with the current state of technology and limited computer resources. As evidenced in Figure 2.3, the ET fuel feed lines, the ET/SRB attach ring, the ORB vertical tails, and the space shuttle main engine (SSME) are all clearly visible and may influence the surrounding flow. However, to demonstrate the feasibility of the numerical model, the geometry of the integrated space shuttle vehicle was simplified and idealized to a certain degree in the early stage of the research. Specifically, all three major components, ET, SRB, and ORB, were modeled with stings at the back extending to the outflow boundary and the ORB without the vertical tail as shown in Figure 2.4. The first calculation was for a free stream Mach number of 2.0. For this flow, the upstream influence was small, so that it was possible to capture meaningful flow phenomena despite the geometric simplifications. Later, the attach hardware, forward and aft, were added and the ET sting was removed to more accurately model the real geometry as illustrated in Figure 2.5. The stings behind the SRB and the ORB were still kept to mimic the effect of the plume. The elevons of the ORB were also deflected to the wind tunnel testing or real flight position. The surface definition of the integrated shuttle vehicle was provided by Ben-Shmuel of Rockwell International.

Body-conforming grids were used for all the component grids and the grid lines were clustered near the body surface to resolve the high gradients in the boundary layer. Not only does the use of body-conforming grids make the boundary conditions simple to implement, but it also facilitates the clustering of the grid points in the boundary layer. In general, the grid is mapped onto a uniformly spaced computational domain, ($\xi$, $\eta$, $\zeta$), with $\xi$ aligned with the major flow direction, $\eta$ in the circumferential direction, and $\zeta$ away from the body. The orbiter grid is generated using a three-dimensional hyperbolic grid generator developed by Steger and Rizk (1985). The hyperbolic grid generator basically solves three simultaneous partial differential equations — two orthogonality relations between $\xi$ and $\zeta$ and between $\eta$ and $\zeta$,

$$\vec{r}_\xi \cdot \vec{r}_\zeta = 0 \tag{2.4}$$

$$\vec{r}_\eta \cdot \vec{r}_\zeta = 0 \tag{2.5}$$

and a user specified volume constraint,

$$\left|\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}\right| = \Delta V \tag{2.6}$$

where $\vec{r}$ is the position vector, $(x,y,z)^t$. The grid is obtained by first defining a surface grid, then using a hyperbolic grid generator which marches in the outward normal direction from the given surface distribution through the constraint of the two orthogonal relations and the user specified spacings (volume). Due to the nature of the time like marching in the outward normal direction, the location of the outer boundary can not be specified. However, for external flows, the location of the outer boundary does not have to be fixed at a predetermined location as it does for internal flows. Details of the hyperbolic grid generation procedure can be found in Steger and Rizk (1985), while the specific details related to the ORB grid generation are given in Rizk and Ben-Shmuel (1985) and Rizk et al. (1985). Figure 2.6 shows the different views of the ORB grid.

For the ET and SRB, the grids were generated using a two-dimensional hyperbolic grid generator since the geometries were axisymmetric and could be spun around 360° to obtain the three-dimensional grids. The SRB grid was not axisymmetric because points were clustered in the small clearance between the ET and the SRB as shown in Figure 2.8c.

The composite grid, consisting of the ET, SRB and ORB grids, is shown in Figure 2.7 at the plane of symmetry. The ET grid is the major grid and extends all the way to the far field boundary, while the ORB and the SRB grids are smaller computational domains with stings extending to the outflow boundary. Also visible in the figure is the hole boundary in the ET mesh cut out by the ORB. In Figure 2.8, the cross sectional view for all three component grids are presented at a constant $\xi$

location. The hole boundary in each mesh, ET, SRB and ORB, is clearly visible and the hole points are removed due to the existence of the solid bodies. The values of the flow variables on the hole boundaries of the ET mesh are provided by the ORB and the SRB grids, while the flow variables on the hole boundaries of the ORB is updated by information from both the ET and the SRB grids, depending on the locations of the boundary points. The hole boundary of the SRB is entirely updated by the ET grid since the hole is cut out due to the presence of the ET only. The outer boundary update of the ORB grid, for the most part, comes from the ET grid, with a small portion near the SRB coming from the SRB grid. The outer boundary of the SRB grid is similarly updated — partly by the ET grid and partly by the ORB grid. Figure 2.9 shows the three-dimensional view of the hole cut out in the ET grid due to the ORB and the SRB.

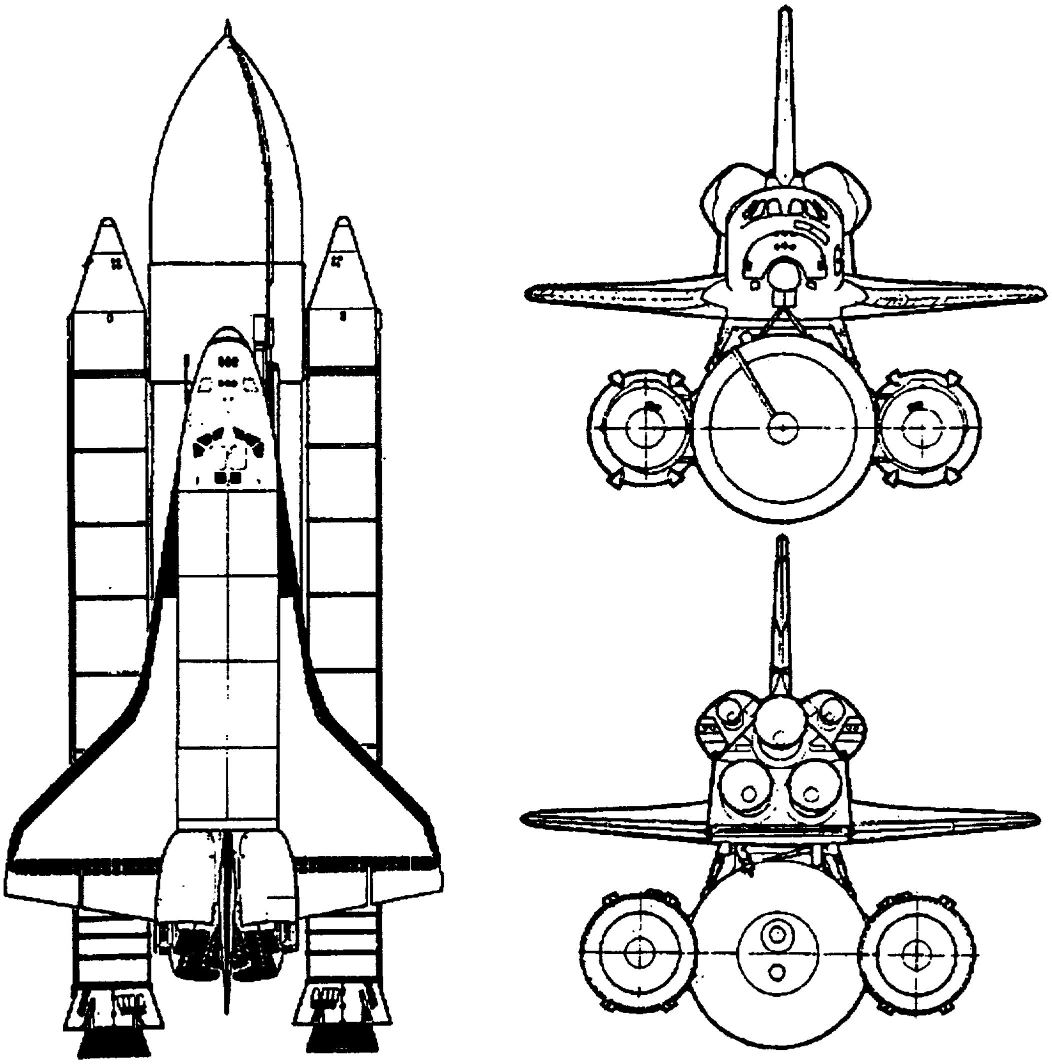


Figure 2.3: Detail views of the space shuttle vehicle

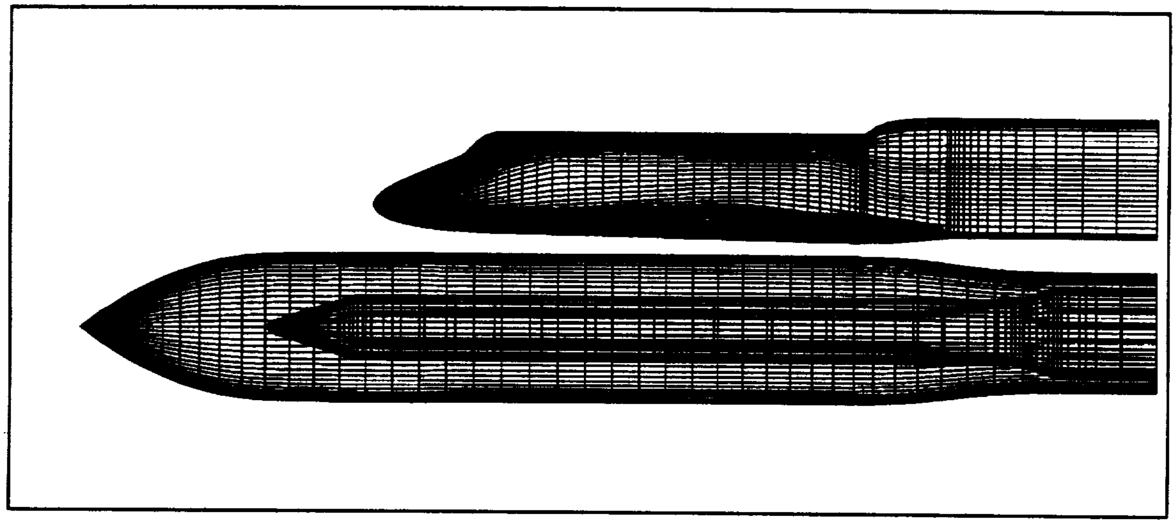


Figure 2.4: Simplified configuration and surface grid point distributions for preliminary supersonic flow calculations

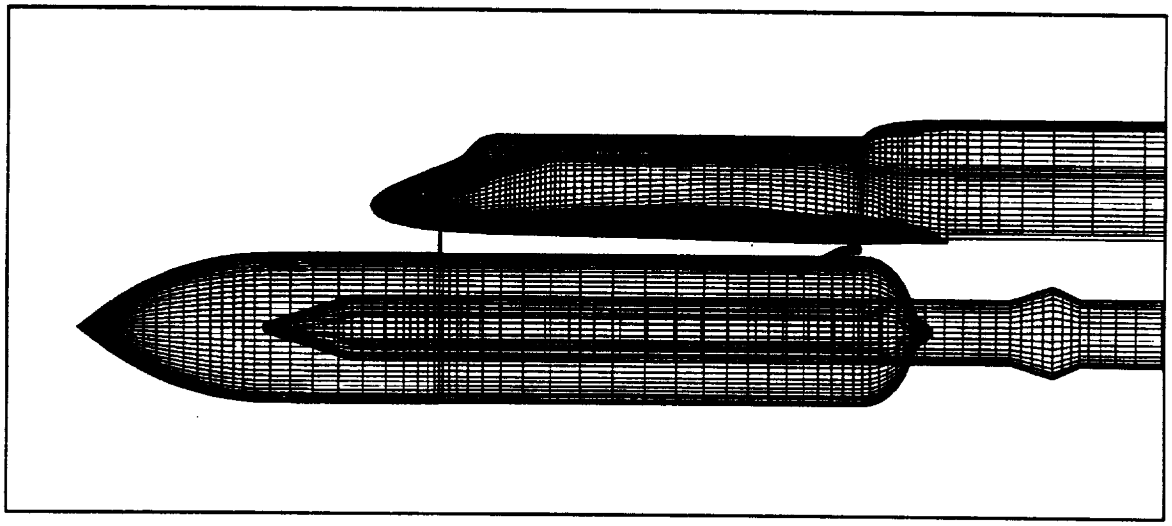


Figure 2.5: Improved configuration and surface grid point distributions

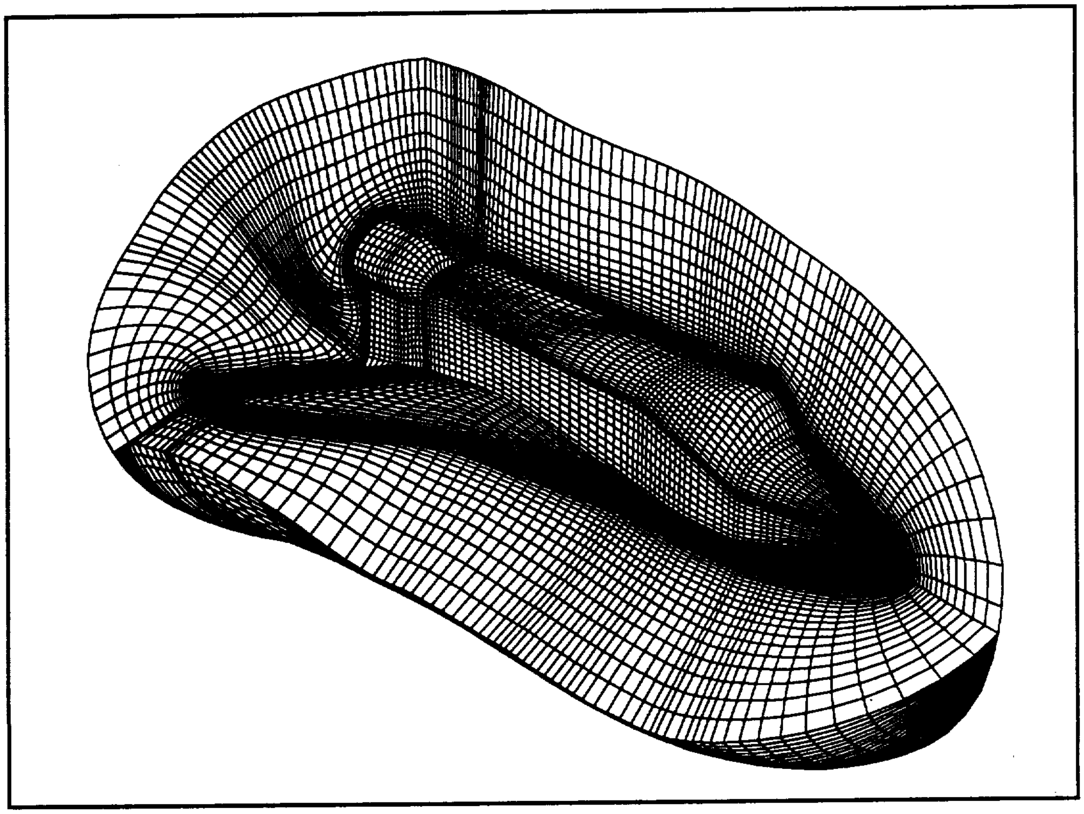


Figure 2.6: Various computational planes of the orbiter grid

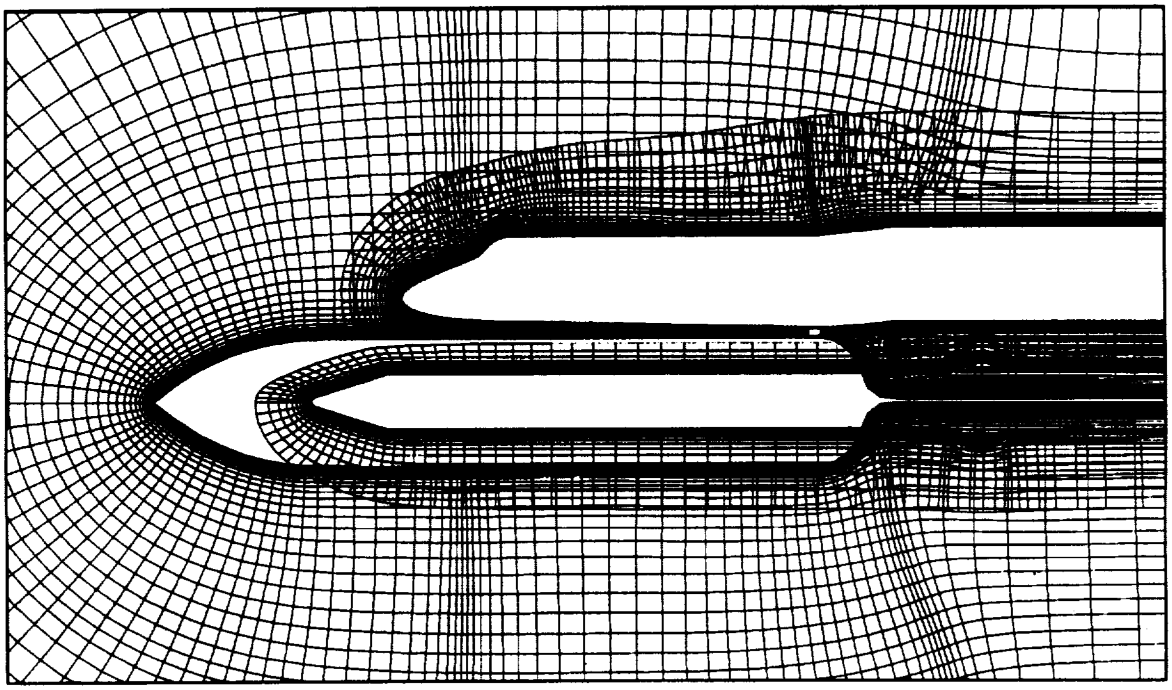
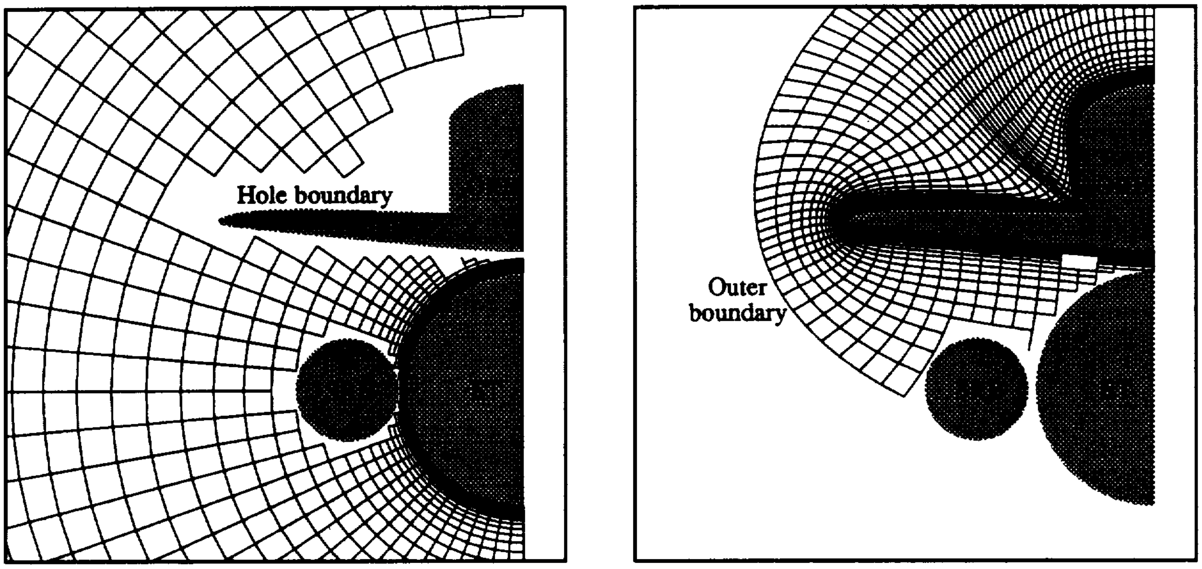
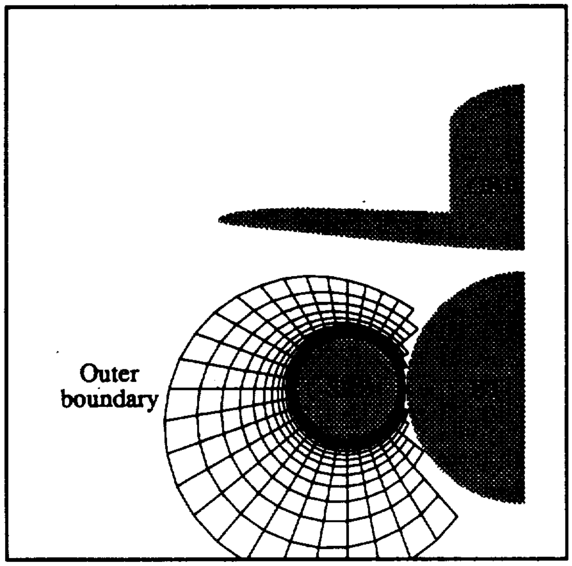


Figure 2.7: Symmetry planes of all grids



(a) ET grid (b) ORB grid



(c) SRB grid

Figure 2.8: Grid cross-section showing holes

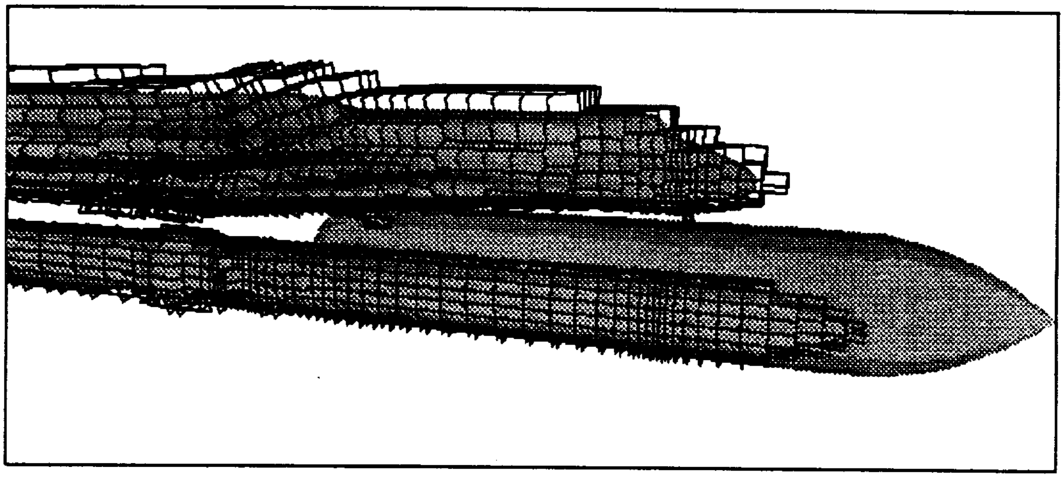


Figure 2.9: Hole boundaries of ET grid

# 3. GOVERNING EQUATIONS AND NUMERICAL METHODS

## 3.1 Introduction

The thin-layer Navier-Stokes, Euler, and Poisson[1] equations were used to solve for the flowfields considered in this research. First used by Pulliam and Steger (1978), the thin-layer Navier-Stokes equations are based on the observation that for high Reynolds number flows, the viscous effects are confined to a thin-layer near rigid boundaries. The gradients in this layer vary very rapidly only in the direction normal to the surface. Thus, all the viscous terms in the other two directions are dropped in this approximation. Coupled with the use of body-conforming grids and clustering of the grid lines near the body surface, the thin-layer Navier-Stokes equations can be used to properly resolve the flowfield around the body surface at high Reynolds numbers. The Euler equations are derived from the same set of equations by dropping all the viscous terms. Finally, the Poisson equations are used to obtain a rough estimate of the flowfield. This solution is then fed back into the Navier-Stokes equation solver as the initial guess to improve the overall rate of convergence. It is believed that a great deal of CPU time is spent in damping out the initial transient for the flow solver. Thus, if a good initial guess is available, the overall convergence will be improved by

---

[1] The Poisson equation referred to throughout this report is a special case of the vorticity-velocity algorithm.

reducing the time in the transient state. In the following sections, the Navier-Stokes equations will first be presented followed by the thin-layer Navier-Stokes equations, and the vorticity-velocity algorithm.

## 3.2 Navier-Stokes Equations

The three-dimensional unsteady Navier-Stokes equations in the Cartesian coordinates can be given as (Peyret and Viviand 1975):

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} + \frac{\partial G_v}{\partial z} \tag{3.1}$$

where Q is the vector of the flow variables, $E$, $F$, and $G$ represent the inviscid fluxes and $E_v$, $F_v$ and $G_v$ correspond to the viscous fluxes.

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \tag{3.2}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{bmatrix}, \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{bmatrix}, \quad G = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{bmatrix} \tag{3.3}$$

$$E_v = Re^{-1}\begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ \beta_x \end{bmatrix}, \quad F_v = Re^{-1}\begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ \beta_y \end{bmatrix}, \quad G_v = Re^{-1}\begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ \beta_z \end{bmatrix} \tag{3.4}$$

with

$$\tau_{xx} = \lambda\left(u_x + v_y + w_z\right) + 2\mu u_x \tag{3.5}$$

$$\tau_{yy} = \lambda\left(u_x + v_y + w_z\right) + 2\mu v_y \tag{3.6}$$

$$\tau_{zz} = \lambda\left(u_x + v_y + w_z\right) + 2\mu w_z \tag{3.7}$$

$$\tau_{xy} = \tau_{yx} = \mu\left(u_y + v_x\right) \tag{3.8}$$

$$\tau_{xz} = \tau_{zx} = \mu\left(u_z + w_x\right) \tag{3.9}$$

$$\tau_{yz} = \tau_{zy} = \mu\left(v_z + w_y\right) \tag{3.10}$$

$$\beta_x = \gamma\kappa Pr^{-1}\partial_x e_I + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \tag{3.11}$$

$$\beta_y = \gamma\kappa Pr^{-1}\partial_y e_I + u\tau_{yx} + v\tau_{yy} + w\tau_{yz} \tag{3.12}$$

$$\beta_z = \gamma\kappa Pr^{-1}\partial_z e_I + u\tau_{zx} + v\tau_{zy} + w\tau_{zz} \tag{3.13}$$

$$e_I = e\rho^{-1} - 0.5\left(u^2 + v^2 + w^2\right) \tag{3.14}$$

The Cartesian velocity components $u$, $v$, and $w$ are nondimensionalized by the free stream speed of sound, $a_\infty$, whereas the density, $\rho$, and the total energy, $e$, are nondimensionalized by the free stream density, $\rho_\infty$, and $\rho_\infty a_\infty^2$, respectively. Pressure can be obtained from the perfect gas law:

$$p = (\gamma - 1)\left[e - 0.5\rho\left(u^2 + v^2 + w^2\right)\right] \tag{3.15}$$

The ratio of the specific heat, $\gamma$, is set equal to 1.4. $\kappa$ is the coefficient of thermal conductivity, $\mu$ the dynamic viscosity, and $\lambda$ from the Stokes hypothesis is $-2\mu/3$. The Reynolds number is $Re$ and the Prandtl number is $Pr$.

**3.2.0.1 Generalized coordinates** Body-fitted coordinates are employed in the numerical simulation to simplify the treatment of arbitrary geometries, especially the imposition of boundary conditions. The flowfield is mapped onto a uniformly spaced computational domain, and the transformed equations are maintained in strong conservation law form for the purpose of shock capturing. The generalized coordinate transformation is defined by

$$\begin{aligned} \tau &= t \\ \xi &= \xi\left(x, y, z, t\right) \\ \eta &= \eta\left(x, y, z, t\right) \\ \zeta &= \zeta\left(x, y, z, t\right) \end{aligned} \tag{3.16}$$

The transformed Navier-Stokes equations are given by:

$$\partial_\tau \widehat{Q} + \partial_\xi \left(\widehat{E} - \widehat{E}_v\right) + \partial_\eta \left(\widehat{F} - \widehat{F}_v\right) + \partial_\zeta \left(\widehat{G} - \widehat{G}_v\right) = 0 \tag{3.17}$$

$$\widehat{Q} = J^{-1} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \widehat{E} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)\, U - \xi_t p \end{bmatrix} \tag{3.18}$$

$$\widehat{F} = J^{-1} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p) V - \eta_t p \end{bmatrix}, \quad \widehat{G} = J^{-1} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p) W - \zeta_t p \end{bmatrix} \tag{3.19}$$

and

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{3.20}$$

U, V, and W are contravariant velocity components. The viscous terms are:

$$\widehat{E}_v = J^{-1} Re^{-1} \begin{bmatrix} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} + \xi_z \tau_{xz} \\ \xi_x \tau_{yx} + \xi_y \tau_{yy} + \xi_z \tau_{yz} \\ \xi_x \tau_{zx} + \xi_y \tau_{zy} + \xi_z \tau_{zz} \\ \xi_x \beta_x + \xi_y \beta_y + \xi_z \beta_z \end{bmatrix} \tag{3.21}$$

$$\widehat{F}_v = J^{-1} Re^{-1} \begin{bmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} + \eta_z \tau_{xz} \\ \eta_x \tau_{yx} + \eta_y \tau_{yy} + \eta_z \tau_{yz} \\ \eta_x \tau_{zx} + \eta_y \tau_{zy} + \eta_z \tau_{zz} \\ \eta_x \beta_x + \eta_y \beta_y + \eta_z \beta_z \end{bmatrix} \tag{3.22}$$

$$\hat{G}_v = J^{-1} Re^{-1} \begin{bmatrix} 0 \\ \zeta_x \tau_{xx} + \zeta_y \tau_{xy} + \zeta_z \tau_{xz} \\ \zeta_x \tau_{yx} + \zeta_y \tau_{yy} + \zeta_z \tau_{yz} \\ \zeta_x \tau_{zx} + \zeta_y \tau_{zy} + \zeta_z \tau_{zz} \\ \zeta_x \beta_x + \zeta_y \beta_y + \zeta_z \beta_z \end{bmatrix} \tag{3.23}$$

where the components of the shear-stress tensor are given in Eqs. (3.5–3.10). The respective Cartesian derivative terms are expanded according to the chain rule of partial differentiation; for example,

$$u_x = \xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta \tag{3.24}$$

The metric terms are obtained from the chain-rule expansion of $x_\xi$, $y_\xi$, $z_\xi$, etc., and solved for $\xi_x$, $\xi_y$, $\xi_z$, etc., to give

$$\begin{aligned} \xi_x &= J\left(y_\eta z_\zeta - z_\eta y_\zeta\right) & \eta_x &= J\left(z_\xi y_\zeta - z_\zeta y_\xi\right) \\ \xi_y &= J\left(z_\eta x_\zeta - x_\eta z_\zeta\right) & \eta_y &= J\left(x_\xi z_\zeta - x_\zeta z_\xi\right) \\ \xi_z &= J\left(x_\eta y_\zeta - y_\eta x_\zeta\right) & \eta_z &= J\left(y_\xi x_\zeta - y_\zeta x_\xi\right) \\ \zeta_x &= J\left(z_\eta y_\xi - z_\xi y_\eta\right) & \xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \\ \zeta_y &= J\left(z_\eta x_\xi - x_\xi z_\eta\right) & \eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z \\ \zeta_z &= J\left(y_\eta x_\xi - y_\xi x_\eta\right) & \zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z \end{aligned} \tag{3.25}$$

and the Jacobian of the coordinate transformation is given as

$$J = \left(x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi\right)^{-1} \tag{3.26}$$

The metric terms, $\xi_x$, $\xi_y$, etc., are all differenced using central differencing for the interior points, and second-order one-sided differencing for the boundary points. However, in rare cases when the spacing between $\zeta = 2$ and $\zeta = 1$ is much smaller than that between $\zeta = 3$ and $\zeta = 2$, the second-order one-sided differencing in the $\zeta$ derivatives causes the Jacobian to become negative. In these cases, switching to first-order one-sided differencing remedies this problem.

### 3.3 Thin-Layer Navier-Stokes Equations

The numerical procedure used for the thin-layer Navier-Stokes equations in this research was developed by Steger et al. (1986) in a program called F3D. For high Reynolds number flows, the viscous terms, $\widehat{E}_v$ and $\widehat{F}_v$, can generally be neglected based on the same arguments used for the boundary-layer approximation. The cross derivative terms in $\widehat{G}_v$ are also dropped for the same reasons. However, in this research, the viscous term, $\widehat{F}_v$, was treated the same way as the viscous term, $\widehat{G}_v$, so that viscous effects could be accounted for in either the $\xi$ or $\eta$ directions or both. The remaining of the viscous terms, $\widehat{F}_v$ and $\widehat{G}_v$, were collected into the right-hand-side, i.e., $Re^{-1}\left(\partial_\zeta \widehat{S} + \partial_\eta \widehat{R}\right)$, of the thin-layer Navier-Stokes equations as shown below.

$$\partial_\tau \widehat{Q} + \partial_\xi \widehat{E} + \partial_\eta \widehat{F} + \partial_\zeta \widehat{G} = Re^{-1}\left(\partial_\eta \widehat{R} + \partial_\zeta \widehat{S}\right) \tag{3.27}$$

$$\widehat{R} = J^{-1}\begin{bmatrix} 0 \\ \mu\left(\eta_x^2+\eta_y^2+\eta_z^2\right)u_\eta + (\mu/3)\left(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta\right)\eta_x \\ \mu\left(\eta_x^2+\eta_y^2+\eta_z^2\right)v_\eta + (\mu/3)\left(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta\right)\eta_y \\ \mu\left(\eta_x^2+\eta_y^2+\eta_z^2\right)w_\eta + (\mu/3)\left(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta\right)\eta_z \\ \Big\{\left(\eta_x^2+\eta_y^2+\eta_z^2\right) \\ \times\left[0.5\mu\left(u^2+v^2+w^2\right)_\eta + \mu Pr^{-1}(\gamma-1)^{-1}\left(a^2\right)_\eta\right] \\ +(\mu/3)\left(\eta_x u + \eta_y v + \eta_z w\right)\left(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta\right)\Big\} \end{bmatrix} \tag{3.28}$$

$$\widehat{S} = J^{-1} \begin{bmatrix} 0 \\ \mu\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right) u_\zeta + (\mu/3)\left(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta\right)\zeta_x \\ \mu\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right) v_\zeta + (\mu/3)\left(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta\right)\zeta_y \\ \mu\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right) w_\zeta + (\mu/3)\left(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta\right)\zeta_z \\ \left\{\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)\right. \\ \times\left[0.5\mu\left(u^2 + v^2 + w^2\right)_\zeta + \mu Pr^{-1}(\gamma - 1)^{-1}\left(a^2\right)_\zeta\right] \\ \left. + (\mu/3)\left(\zeta_x u + \zeta_y v + \zeta_z w\right)\left(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta\right)\right\} \end{bmatrix} \tag{3.29}$$

In the implementation of the thin-layer Navier-Stokes equations, the following equation was used instead of Eq. (3.27).

$$\begin{aligned} &\partial_\tau\left(\widehat{Q} - \widehat{Q}_\infty\right) + \partial_\xi\left(\widehat{E} - \widehat{E}_\infty\right) + \partial_\eta\left(\widehat{F} - \widehat{F}_\infty\right) + \partial_\zeta\left(\widehat{G} - \widehat{G}_\infty\right) \\ &= Re^{-1}\left(\partial_\eta \widehat{R} + \partial_\zeta \widehat{S}\right) \end{aligned} \tag{3.30}$$

The "free stream subtraction" used in the above equation is to avoid the errors introduced from the approximations in computing the metric terms $x_\xi$, $y_\xi$, etc. The $\widehat{R}_\infty$ and $\widehat{S}_\infty$ are neglected since they are small for high Reynolds number flows. An implicit two-factor approximate factorization scheme, shown below, that uses central differencing in the $\eta$ and $\zeta$ directions and upwind differencing in the $\xi$ direction is used for the above equation.

$$\begin{aligned} &\left[I + h\delta_\xi^b(\widehat{A}^+)^n + h\delta_\zeta \widehat{C}^n - hRe^{-1}\bar{\delta}_\zeta J^{-1}\widehat{M}^n J - D_i|_\zeta\right] \\ &\times\left[I + h\delta_\xi^f(\widehat{A}^-)^n + h\delta_\eta \widehat{B}^n - hRe^{-1}\bar{\delta}_\eta J^{-1}\widehat{N}^n J - D_i|_\eta\right]\Delta\widehat{Q}^n = \\ &-\Delta t\{\delta_\xi^b(\widehat{E}^+)^n + \delta_\xi^f(\widehat{E}^-)^n + \delta_\eta \widehat{F}^n + \delta_\zeta \widehat{G}^n - Re^{-1}\bar{\delta}_\eta \widehat{R}^n - Re^{-1}\bar{\delta}_\zeta \widehat{S}^n\} \\ &-(D_e|_\eta + D_e|_\zeta)\widehat{Q}^n \end{aligned} \tag{3.31}$$

where $\delta^b$ and $\delta^f$ are the backward and forward difference operators, $\delta$ the central-difference operator, $\bar{\delta}$ the mid-point central difference operator for the viscous terms and $D_i$ and $De$ the implicit and explicit smoothing operators used in the centrally differenced directions, $\eta$ and $\zeta$. The scheme is capable of achieving first-order ($h = \Delta t$) or second-order time accuracy ($h = \Delta t/2$). The matrices $\widehat{A}^{\pm}$, $\widehat{B}$, $\widehat{C}$, $\widehat{M}$, and $\widehat{N}$ are the results from the local linearization of the fluxes, $\widehat{E}^{\pm}$, $\widehat{F}$, $\widehat{G}$, $\widehat{R}$, and $\widehat{S}$ about the previous time level, e.g.,

$$\widehat{F}^{n+1} = \widehat{F}^n + \widehat{B}^n \Delta \widehat{Q}^n + O\left(h^2\right) \tag{3.32}$$

The implicit approximately factored scheme, Eq. (3.31), is solved by marching in the $\xi$ direction using two sequential sweeps of the block tridiagonal inversion procedure, one in the $\eta$ and the other in the $\zeta$ direction, at each constant $\xi$ plane.

In the chimera scheme, the flow solver developed for a single curvilinear grid has to be modified to account for the hole points (including the hole boundary points) introduced in the overset grid. At such points, the values should be kept unchanged since these points are either at the hole boundary or within a body (or a user specified boundary zone), and are updated by other grids or assume no meaningful values. An array of values $i_b$, 1 at the regular points, and 0 at the hole points, is thus introduced into the approximately factored scheme to turn off the finite differencing at the hole or hole boundary points. The following shows how $i_b$ is used in the differencing scheme.

$$\begin{aligned}
&\left[I + i_b \left(h\delta_\xi^b(\widehat{A}^+)^n + h\delta_\zeta \widehat{C}^n - hRe^{-1}\bar{\delta}_\zeta J^{-1}\widehat{M}^n J - D_i|_\zeta\right)\right] \\
&\times \left[I + i_b \left(h\delta_\xi^f(\widehat{A}^-)^n + h\delta_\eta \widehat{B}^n - hRe^{-1}\bar{\delta}_\eta J^{-1}\widehat{N}^n J - D_i|_\eta\right)\right] \Delta \widehat{Q}^n = \\
&-i_b \left[\Delta t \delta_\xi^b(\widehat{E}^+)^n + \delta_\xi^f(\widehat{E}^-)^n + \delta_\eta \widehat{F}^n + \delta_\zeta \widehat{G}^n\right. \\
&\left. -Re^{-1}\bar{\delta}_\eta \widehat{R}^n - Re^{-1}\bar{\delta}_\zeta \widehat{S}^n + (De|_\eta + De|_\zeta)\widehat{Q}^n\right]
\end{aligned} \tag{3.33}$$

Thus, for hole points, $i_b = 0$, and the above equation reduces to

$$\Delta \widehat{Q}^n = 0 \tag{3.34}$$

and $\widehat{Q}$ remains constant. If only three-point central differencing is used, the scheme requires no further modifications. However, when differencing the points adjacent to a hole boundary, finite-difference operators that require information beyond the adjacent points on either side of the differenced point will need to be modified since only the hole boundary points are updated from other grids, and the hole points do not contain meaningful data. Therefore, difference operators adjacent to a hole boundary will need to be modified such that data from the hole points are excluded from the calculation. The same $i_b$ mentioned above can be used to achieve this purpose. For example, the right hand side dissipation term in the $\zeta$ direction is currently implemented as

$$\begin{aligned}
D_e|_\zeta \widehat{Q} = &\left(\tfrac{\epsilon}{2}\right)\left(\frac{\varrho_{l+1}}{J_{l+1}} + \frac{\varrho_l}{J_l}\right) \times \\
&\left[\left(\frac{\nabla\Delta Q_{l+1} - \nabla\Delta Q_l}{1+\alpha_{l+1}+\alpha_l}\right) - \left(\frac{\alpha_{l+1}+\alpha_l}{2}\right)(Q_{l+1} - Q_l)\right] \\
&-\left(\tfrac{\epsilon}{2}\right)\left(\frac{\varrho_l}{J_l} + \frac{\varrho_{l-1}}{J_{l-1}}\right) \times \\
&\left[\left(\frac{\nabla\Delta Q_l - \nabla\Delta Q_{l-1}}{1+\alpha_l+\alpha_{l-1}}\right) - \left(\frac{\alpha_l+\alpha_{l-1}}{2}\right)(Q_l - Q_{l-1})\right]
\end{aligned} \tag{3.35}$$

with $\epsilon = O(.1)$, $\nabla\Delta$ a second-order differencing, and

$$\alpha = \left(\frac{1 + M_\infty^\gamma}{16}\right)\left(\frac{p_{l+1} - 2p_l + p_{l-1}}{p_{l+1} + 2p_l + p_{l-1}}\right) \tag{3.36}$$

Here $\varrho$ is a modified spectral radius of the matrix $\widehat{C}$ (see Eq. 3.31),

$$\varrho = |\zeta_x u + \zeta_y v + \zeta_z w| + \sqrt{(\zeta_x{}^2 + \zeta_y{}^2 + \zeta_z{}^2)\bar{c}^2 + .01} \tag{3.37}$$

with $\bar{c}^2 = \frac{\gamma p}{\rho}(1-\beta) + \frac{\beta(u^2+v^2+w^2)}{M_\infty^2}$ and $0 \le \beta \le 1$ where the choice $\beta = 1$ reduces the smoothing in the boundary layer. The parameter $\alpha$ determined from the pressure gradient is used to switch from second-order to fourth-order smoothing. To avoid using data from the hole points, the differencing is modified as

$$\begin{aligned} De|_\zeta \hat{Q} = {} & \left(\tfrac{\epsilon}{2}\right)\left(\frac{\rho_{l+1}}{J_{l+1}} + \frac{\rho_l}{J_l}\right) \times \\ & \left[\left(\frac{i_b|_{l+1}\nabla\Delta Q_{l+1} - i_b|_l \nabla\Delta Q_l}{1+\alpha_{l+1}+\alpha_l}\right) - \left(\frac{\alpha_{l+1}+\alpha_l}{2}\right)(Q_{l+1}-Q_l)\right] \\ & -\left(\tfrac{\epsilon}{2}\right)\left(\frac{\rho_l}{J_l} + \frac{\rho_{l-1}}{J_{l-1}}\right) \times \\ & \left[\left(\frac{i_b|_l \nabla\Delta Q_l - i_b|_{l-1}\nabla\Delta Q_{l-1}}{1+\alpha_l+\alpha_{l-1}}\right) - \left(\frac{\alpha_l+\alpha_{l-1}}{2}\right)(Q_l - Q_{l-1})\right] \end{aligned} \tag{3.38}$$

Thus the fourth-order differencing is reduced to an uncentered second-order differencing adjacent to a hole boundary point.

**3.3.0.2 Boundary conditions** Explicit boundary conditions were used for the thin-layer Navier-Stokes equations. Each boundary condition was coded in a separate subroutine; thus, additional boundary conditions can be conveniently added as need arises. The boundary conditions implemented include inviscid/viscous wall conditions, far-field conditions, axis conditions, wing cut conditions, symmetry plane conditions, periodic conditions, and overset grid hole and outer boundary conditions.

**3.3.0.3 Wall conditions** For viscous flows, the no-slip boundary condition is enforced by setting the velocities on the wall boundary to be zero. For inviscid flows, the tangency boundary condition is implemented by setting the contravariant velocities $W_{\zeta=1} = 0$; in other words, the fluid flow is not allowed to go through the wall boundary. The values of the contravariant velocities U and V are extrapolated from the interior points at $\zeta = 2$ and $\zeta = 3$. The Cartesian component of the velocities can be found by solving Equation ( 3.20). The density on the wall is obtained through zeroth-order extrapolation from the interior points at $\zeta = 2$, i.e., $\rho_{\zeta=1} = \rho_{\zeta=2}$. This is a reasonable approximation in the current study since the grids used are usually clustered near the surface and the grid spacing there is very small.

The pressure on the surface is obtained from a normal momentum relation found by combining the three transformed momentum equations (Pulliam and Steger 1978),

$$\begin{aligned} &\left(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z\right)p_\xi + \left(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z\right)p_\eta \\ &+ \left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)p_\zeta = \\ &-\rho U\left(\zeta_x u_\xi + \zeta_y v_\xi + \zeta_z w_\xi\right) - \rho V\left(\zeta_x u_\eta + \zeta_y v_\eta + \zeta_z w_\eta\right) \end{aligned} \tag{3.39}$$

The above equation is solved using central differencing for the $\xi$ and $\eta$ derivatives, second-order one-sided differencing for the $\zeta$ derivatives, and values from the interior points at $\zeta = 2$ and $\zeta = 3$. For viscous flows, $U = V = 0$ is used in the above equation.

**3.3.0.4 Far-field conditions** At the far-field boundary, two different boundary conditions were implemented. The first one is the Dirichlet boundary condition which simply sets the free stream values for points on the $\zeta = \zeta_{max}$ surface. Generally, this boundary condition requires the largest computational domain among the possible far-field boundary conditions, and is valid only when the far-field boundary is far enough from the solid boundary.

$$Q_{far-field} = Q_\infty \tag{3.40}$$

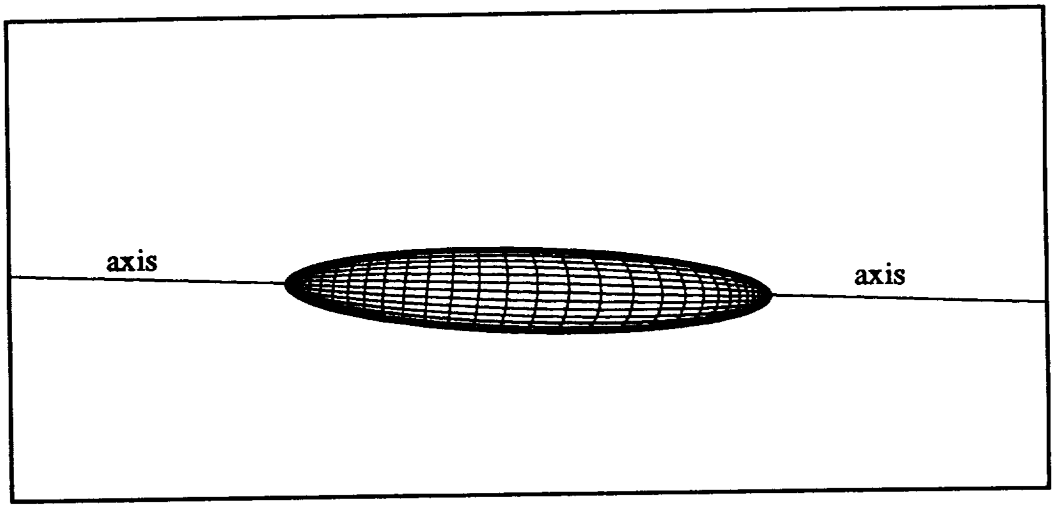


Figure 3.1: Surface grid for an ellipsoid

The other far-field boundary condition implemented is the outflow boundary condition. It assumes zero gradient at the downstream boundary. Of those grids implemented with this boundary condition are the ET, SRB and ORB grids.

$$Q_{\xi_{max}} = Q_{\xi_{max-1}} \tag{3.41}$$

**3.3.0.5 Axis conditions** For the grid extending from both ends of an ellipsoid (Figure 3.1) or from the ET nose, the grid points fall onto a single line which is referred to as an axis for convenience. The values of the variables on the axis are evaluated from the average of the variables at the neighboring points surrounding the axis. This condition was implemented for $\xi = constant$ and $\zeta = constant$ axes.

$$q_{axis} = \frac{1}{n} \sum_{i=1}^{n} q_i \tag{3.42}$$

Note that this averaging should not include hole points since they contain no meaningful data. Other geometries, like the SRB, ORB, ellipsoid, and sphere, also used this boundary condition at the axis.

**3.3.0.6 Wing cut conditions** The orbiter grid was generated with a wing sting extending from the trailing edge to the outflow boundary. For the flow simulation to more accurately mimic the real flowfield, a condition was introduced to allow the flow to go through the wing sting without regenerating a new sting-less orbiter grid. What this condition did was to set the flow variables at the top and bottom surfaces (at the same spanwise location) to equal values. Since the thickness of the sting is very small, this is considered a reasonable approximation.

**3.3.0.7 Symmetry plane conditions** To save computer time, a symmetry plane condition was used as the situation permits, as with the ET and ORB calculations. This condition assumes no penetration of fluid flow through the symmetry plane and sets values for other variables at points on opposite sides of the symmetry plane equal.

**3.3.0.8 Periodic conditions** This condition ensures that flow variables at points on the planes at $\eta = 1$ ($0^\circ$) and $\eta = \eta max$ ($360^\circ$) are of equal values. This condition was used for the SRB grid, where the flow had to be solved for a complete $360^\circ$ circle.

$$Q_{\eta=1} = Q_{\eta=max} \tag{3.43}$$

**3.3.0.9 Overset grid hole and outer boundary conditions** As was explained in the chapter on "Chimera Approach and Grid Topology", the interpolation is required to transfer information (through the hole and outer boundaries) among the different grids of the composite geometry. For this study, trilinear interpolation was used.

### 3.3.1 Solution Procedure

As described in the chapter on "Chimera Approach and Grid Topology", the major advantage of the chimera approach is the embedded flexibility in the chimera scheme which allows one to blend different grids, different flow solvers, etc. Here, a flow solver developed for a single curvilinear grid can easily be tailored for the chimera scheme by just adding a control loop outside the major flow simulation loop to update the grid interface boundaries and provide ways to blank out the hole and hole boundary points.

In the current implementation, data from each grid and the boundary interface arrays are brought in sequentially from external storage (high speed disk or CRAY solid state device, SSD) at each time step. The flow variables on the hole and outer[2] boundaries are first updated by the values stored in the boundary interface arrays. Then the solution is updated by the flow solver and the imposed boundary conditions. The boundary interface data that the current grid sends to other grids are then loaded into the boundary interface arrays and all the arrays are sent back to the external storage. The next grid is then brought in and so on.

## 3.4 Vorticity-Velocity Formulation

The vorticity-velocity formulation consists of:

1. Poisson equations — derived from the continuity equation.
2. Crocco relations — derived from the momentum equations for inviscid flows and the Gibbs function, a thermodynamic relation based on the first

---

[2] Outer boundary is updated only if the current grid is embedded in other grids.

and second laws of thermodynamics.

3. Convection of entropy and stagnation enthalpy

4. Bernoulli equation — derived from Crocco relations and the equation of state.

5. Vorticity consistency condition — a vector identity which enforces the conservation of vorticity in the flowfield.

The details of each equation will be described in the following sections in the order listed above.

The vorticity-velocity formulation is a set of weakly coupled equations. Hence, it can be solved more efficiently than the Navier-Stokes or Euler equations since only scalar tridiagonal (rather than block tridiagonal) systems of equations are involved in the computation. However, this algorithm won't be as accurate as the traditional flow solver using Navier-Stokes equations because viscous terms are omitted. Currently, the algorithm is limited to subsonic flows and is intended to be used as a diagnostic tool for testing grids (especially, in the case of complex geometry consisting of overset grids) or to offer an approximate solution as an initial guess for the more accurate but more expensive solvers using the Navier-Stokes equations. In the future, the current algorithm may be extended to treat viscous flows by combining with the boundary-layer or Navier-Stokes equations.

### 3.4.1 Poisson equations

The Poisson equations are derived from the continuity equation. First, by writing the continuity equation in non-conservative form,

$$\nabla \cdot \vec{V} + \vartheta = 0 \tag{3.44}$$

where

$$\vartheta = \frac{\nabla \rho}{\rho} \cdot \vec{V} \tag{3.45}$$

and then via differentiation of Eq. (3.44) and the use of the following vector identity,

$$\nabla \times (\nabla \times \vec{V}) = \nabla(\nabla \cdot \vec{V}) - \nabla^2 \vec{V} \tag{3.46}$$

the vector form of the Poisson equations are obtained.

$$\nabla^2 \vec{V} + \nabla \times (\nabla \times \vec{V}) + \nabla \vartheta = 0 \tag{3.47}$$

where $\nabla \times \vec{V}$ can be replaced by the vorticity vector, $\vec{\omega}$,

$$\vec{\omega} = (w_y - v_z)\vec{i} + (u_z - w_x)\vec{j} + (v_x - u_y)\vec{k} \tag{3.48}$$

and thus

$$\begin{aligned} \nabla \times (\nabla \times \vec{V}) &= \nabla \times \vec{\omega} \\ &= (\omega_{3y} - \omega_{2z})\vec{i} + (\omega_{1z} - \omega_{3x})\vec{j} + (\omega_{2x} - \omega_{1y})\vec{k} \end{aligned} \tag{3.49}$$

For three-dimensional flows, the Poisson equations can be written as

$$\nabla^2 u + \vartheta_x + \omega_{3y} - \omega_{2z} = 0 \tag{3.50}$$

$$\nabla^2 v + \vartheta_y + \omega_{1z} - \omega_{3x} = 0 \tag{3.51}$$

$$\nabla^2 w + \vartheta_z + \omega_{2x} - \omega_{1y} = 0 \tag{3.52}$$

where $\nabla^2$ is the Laplacian operator, $\nabla^2 = \partial_{xx} + \partial_{yy} + \partial_{zz}$. For irrotational flow, the vorticity terms can be dropped from the Poisson equations, resulting in the following equations:

$$u_{xx} + u_{yy} + u_{zz} + \vartheta_x = 0 \tag{3.53}$$

$$v_{xx} + v_{yy} + v_{zz} + \vartheta_y = 0 \tag{3.54}$$

$$w_{xx} + w_{yy} + w_{zz} + \vartheta_z = 0 \tag{3.55}$$

**3.4.1.1 Generalized coordinates** The Poisson equations in the generalized coordinate system are derived using the chain rule of partial differentiation and the vector identities resulting from the coordinate transformation. Only the results are shown here; for details of the derivation, see Appendix A.

$$\begin{aligned} &\left(\frac{\alpha_{\xi\xi}u_\xi + \alpha_{\xi\eta}u_\eta + \alpha_{\xi\zeta}u_\zeta}{J}\right)_\xi + \left(\frac{\alpha_{\xi\eta}u_\xi + \alpha_{\eta\eta}u_\eta + \alpha_{\eta\zeta}u_\zeta}{J}\right)_\eta + \\ &\qquad \left(\frac{\alpha_{\xi\zeta}u_\xi + \alpha_{\eta\zeta}u_\eta + \alpha_{\zeta\zeta}u_\zeta}{J}\right)_\zeta + \frac{f_u}{J} = 0 \end{aligned} \tag{3.56}$$

$$\begin{aligned} &\left(\frac{\alpha_{\xi\xi}v_\xi + \alpha_{\xi\eta}v_\eta + \alpha_{\xi\zeta}v_\zeta}{J}\right)_\xi + \left(\frac{\alpha_{\xi\eta}v_\xi + \alpha_{\eta\eta}v_\eta + \alpha_{\eta\zeta}v_\zeta}{J}\right)_\eta + \\ &\qquad \left(\frac{\alpha_{\xi\zeta}v_\xi + \alpha_{\eta\zeta}v_\eta + \alpha_{\zeta\zeta}v_\zeta}{J}\right)_\zeta + \frac{f_v}{J} = 0 \end{aligned} \tag{3.57}$$

$$\begin{aligned} &\left(\frac{\alpha_{\xi\xi}w_\xi + \alpha_{\xi\eta}w_\eta + \alpha_{\xi\zeta}w_\zeta}{J}\right)_\xi + \left(\frac{\alpha_{\xi\eta}w_\xi + \alpha_{\eta\eta}w_\eta + \alpha_{\eta\zeta}w_\zeta}{J}\right)_\eta + \\ &\qquad \left(\frac{\alpha_{\xi\zeta}w_\xi + \alpha_{\eta\zeta}w_\eta + \alpha_{\zeta\zeta}w_\zeta}{J}\right)_\zeta + \frac{f_w}{J} = 0 \end{aligned} \tag{3.58}$$

where

$$f_u = \xi_x\vartheta_\xi + \eta_x\vartheta_\eta + \zeta_x\vartheta_\zeta + \xi_y\omega_{3\xi} + \eta_y\omega_{3\eta} + \zeta_y\omega_{3\zeta} - \xi_z\omega_{2\xi} - \eta_z\omega_{2\eta} - \zeta_z\omega_{2\zeta} \tag{3.59}$$

$$f_v = \xi_y\vartheta_\xi + \eta_y\vartheta_\eta + \zeta_y\vartheta_\zeta + \xi_z\omega_{1\xi} + \eta_z\omega_{1\eta} + \zeta_z\omega_{1\zeta} - \xi_x\omega_{3\xi} - \eta_x\omega_{3\eta} - \zeta_x\omega_{3\zeta} \tag{3.60}$$

$$fw = \xi z \vartheta_{\xi} + \eta z \vartheta_{\eta} + \zeta z \vartheta_{\zeta} + \xi x \omega_{2\xi} + \eta x \omega_{2\eta} + \zeta x \omega_{2\zeta} - \xi y \omega_{1\xi} - \eta y \omega_{1\eta} - \zeta y \omega_{1\zeta} \tag{3.61}$$

and

$$\alpha_{\xi\xi} = \xi x \xi x + \xi y \xi y + \xi z \xi z \tag{3.62}$$

$$\alpha_{\eta\eta} = \eta x \eta x + \eta y \eta y + \eta z \eta z \tag{3.63}$$

$$\alpha_{\zeta\zeta} = \zeta x \zeta x + \zeta y \zeta y + \zeta z \zeta z \tag{3.64}$$

$$\alpha_{\xi\eta} = \xi x \eta x + \xi y \eta y + \xi z \eta z \tag{3.65}$$

$$\alpha_{\xi\zeta} = \xi x \zeta x + \xi y \zeta y + \xi z \zeta z \tag{3.66}$$

$$\alpha_{\eta\zeta} = \eta x \zeta x + \eta y \zeta y + \eta z \zeta z \tag{3.67}$$

**3.4.1.2 Finite-difference formulations** The cross derivative terms ($Q_{\eta\xi}$, $Q_{\zeta\xi}$, $Q_{\xi\eta}$, $Q_{\zeta\eta}$, $Q_{\xi\zeta}$, and $Q_{\eta\zeta}$) are all lagged at the previous time level; thus the finite-difference formulation of the Poisson equations can be written as

$$\begin{aligned}
&\left[\delta_{\xi}\left(\frac{\alpha_{\xi\xi}}{J}\delta_{\xi}\right) + \delta_{\eta}\left(\frac{\alpha_{\eta\eta}}{J}\delta_{\eta}\right) + \delta_{\zeta}\left(\frac{\alpha_{\zeta\zeta}}{J}\delta_{\zeta}\right)\right]\vec{Q}^{\,n+1} - \frac{\vec{Q}^{\,n+1} - \vec{Q}^{\,n}}{h} \\
&= -\left[\delta_{\xi}\left(\frac{\alpha_{\xi\eta}}{J}\delta_{\eta} + \frac{\alpha_{\xi\zeta}}{J}\delta_{\zeta}\right) + \delta_{\eta}\left(\frac{\alpha_{\xi\eta}}{J}\delta_{\xi} + \frac{\alpha_{\eta\zeta}}{J}\delta_{\zeta}\right)\right. \\
&\left.+\delta_{\xi}\left(\frac{\alpha_{\xi\zeta}}{J}\delta_{\xi}\frac{\alpha_{\eta\zeta}}{J}\delta_{\eta}\right)\right]\vec{Q}^{\,n} - \frac{\vec{f}}{J}
\end{aligned} \tag{3.68}$$

where $\delta$ designates the central difference operator and h the time step size.

Note that an additional time derivative term, $(\vec{Q}^{\,n+1} - \vec{Q}^{\,n})/h$, is added to the Poisson equations to allow the use of an approximately factored scheme, which will be explained later. This added time derivative term should not affect the solution when convergence is reached, since $\vec{Q}^{\,n} \rightarrow \vec{Q}^{\,n+1}$ at convergence. The above equation can be rewritten in delta-form (treating the change $(\vec{Q}^{n+1} - \vec{Q}^{n})$ as the unknown vector) as shown below.

$$\begin{aligned}
&\left\{I - h\left[\delta_\xi\left(\frac{\alpha_{\xi\xi}}{J}\delta_\xi\right) + \delta_\eta\left(\frac{\alpha_{\eta\eta}}{J}\delta_\eta\right) + \delta_\zeta\left(\frac{\alpha_{\zeta\zeta}}{J}\delta_\zeta\right)\right]\right\}\left(\vec{Q}^{\,n+1} - \vec{Q}^{\,n}\right) \\
&= h\omega\left[\delta_\xi\left(\frac{\alpha_{\xi\xi}\delta_\xi + \alpha_{\xi\eta}\delta_\eta + \alpha_{\xi\zeta}\delta_\zeta}{J}\right) + \delta_\eta\left(\frac{\alpha_{\eta\eta}\delta_\eta + \alpha_{\xi\eta}\delta_\xi + \alpha_{\eta\zeta}\delta_\zeta}{J}\right)\right. \\
&\left. + \delta_\zeta\left(\frac{\alpha_{\zeta\zeta}\delta_\zeta + \alpha_{\xi\zeta}\delta_\xi + \alpha_{\eta\zeta}\delta_\eta}{J}\right)\right]\vec{Q}^{\,n} + h\omega\frac{\vec{f}}{J} \\
&= RHS
\end{aligned} \tag{3.69}$$

The relaxation parameter (time step), $h$, varies with the Jacobian of the grid point as shown in the equation below; since the Jacobian varies from point to point, values of $h$ also vary at different locations. This procedure is called local time stepping since the time step size is determined by the local grid size. However, this method is only applicable if a steady state solution is desired as in the present study.

$$h = \frac{\Delta T(1 + 0.005J)}{1 + J} \tag{3.70}$$

A geometric sequence was also used to determine the relaxation parameter, $h$.

$$h = \left[\lambda_1\left(\frac{\lambda_2}{\lambda_1}\right)^{\frac{i-1}{N-1}}\right]^{-1}, \qquad i = 1, 2, 3 \ldots N \tag{3.71}$$

A second overrelaxation parameter, $\omega$, for the source term, $\frac{\vec{f}}{J}$, was used to speed up the rate of convergence. Values of $\lambda_1 = 0.02$ to 1, $\lambda_2 = 0.005$ to 0.01, $N = 4$, and $\omega = 1.0$ and 1.8 have been used in the course of the study. If $\lambda_1$ and $\lambda_2$ are chosen properly, the geometric sequence method for determining the relaxation parameter, $h$, can work just as efficiently as the local time stepping.

The three-step approximately factored scheme, as shown below, was used to solve the Poisson equations. The scheme is second-order accurate in space.

$$\begin{aligned}&\left[I - h\delta_\xi\left(\frac{\alpha_{\xi\xi}}{J}\right)\delta_\xi\right]\left[I - h\delta_\eta\left(\frac{\alpha_{\eta\eta}}{J}\right)\delta_\eta\right]\times\\&\left[I - h\delta_\zeta\left(\frac{\alpha_{\zeta\zeta}}{J}\right)\delta_\zeta\right](\vec{Q}^{\,n+1} - \vec{Q}^{\,n}) = RHS + O(h^2)\end{aligned} \tag{3.72}$$

Central differencing is used for all the terms in the above equation. For chimera overset grids, the hole and hole boundary points should be left unchanged[3]; thus, the blanking array, $i_b$[4], needs to be introduced into the above equation to shut off the differencing scheme for these points. Simply by replacing $h$ with $i_b h$, and $RHS$ with $i_b RHS$, the Poisson equations become

$$(\vec{Q}^{\,n+1} - \vec{Q}^{\,n}) = 0 \tag{3.73}$$

for hole or hole boundary points ($i_b = 0$). Thus, values of flow variables for these points remain unchanged.

The following shows the finite-difference formulation used for the Poisson equations. Unless stated otherwise, indices, $j$, $k$, and $l$, used in finite-difference expressions refer to grid points in the $\xi$, $\eta$ and $\zeta$ directions, respectively, and throughout this report, a subscript in a finite-difference expression is not shown unless it varies, e.g., $Q_{j+1} = Q_{j+1,k,l}$.

$$\begin{aligned}\left(\frac{\alpha_{\xi\xi}Q_\xi}{J}\right)_\xi = &\ \frac{\left(\frac{\alpha_{\xi\xi}}{J}\right)_{j+1} + \left(\frac{\alpha_{\xi\xi}}{J}\right)_j}{2}\,\frac{(Q_{j+1} - Q_j)}{\Delta\xi}\\&-\frac{\left(\frac{\alpha_{\xi\xi}}{J}\right)_j + \left(\frac{\alpha_{\xi\xi}}{J}\right)_{j-1}}{2}\,\frac{(Q_j - Q_{j-1})}{\Delta\xi}\end{aligned} \tag{3.74}$$

---

[3] Hole points are points which lie within a body or user specified boundary zone and should not be solved, while hole boundary points, considered as part of the boundary, are updated by other grids and should not be solved either.

[4] See Section 3.3 "Thin-Layer Navier-Stokes Equations" for details.

The finite-difference expressions for $\left(\frac{\alpha_{\eta\eta}Q_\eta}{J}\right)_\eta$ and $\left(\frac{\alpha_{\zeta\zeta}Q_\zeta}{J}\right)_\zeta$ can be obtained by replacing the $j$ index above with the corresponding $k$ and $l$ indices. For the cross derivative terms, the finite-difference expressions are given in the form:

$$\begin{aligned}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_\xi &= \frac{1}{2\Delta\xi}\left[\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{j+1} - \left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{j-1}\right] \\ &= \frac{1}{2\Delta\xi}\left[\left(\frac{\alpha_{\xi\eta}}{J}\right)_{j+1,k}\frac{\left(Q_{j+1,k+1}-Q_{j+1,k-1}\right)}{\Delta\eta} - \right. \\ &\qquad \left.\left(\frac{\alpha_{\xi\eta}}{J}\right)_{j-1,k}\frac{\left(Q_{j-1,k+1}-Q_{j-1,k-1}\right)}{\Delta\eta}\right]\end{aligned} \tag{3.75}$$

Again by replacing the above $j$ and $k$ indices with the corresponding indices for other cross derivative terms, $\left(\frac{\alpha_{\xi\zeta}Q_\zeta}{J}\right)_\xi$, $\left(\frac{\alpha_{\xi\eta}Q_\xi}{J}\right)_\eta$, $\left(\frac{\alpha_{\eta\zeta}Q_\zeta}{J}\right)_\eta$, $\left(\frac{\alpha_{\xi\zeta}Q_\xi}{J}\right)_\zeta$, and $\left(\frac{\alpha_{\eta\zeta}Q_\eta}{J}\right)_\zeta$, the respective finite-difference expressions can be obtained. For the chimera overset grids, the blanking array, $i_b$, needs to be incorporated into the finite-difference expressions for points involving the flow variables at the hole points. Only the cross derivatives need to be modified since only those terms involve flow variables at hole points. Given below is the modified finite-difference expression for $\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_\xi$.

$$\begin{aligned}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_\xi &= \frac{1}{i_{b\,j+1,k+1}+i_{b\,j+1,k-1}+i_{b\,j-1,k+1}+i_{b\,j-1,k+1}} \\ &\left[i_{b\,j+1,k+1}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{\xi_{j+\frac{1}{2},k+\frac{1}{2}}} + i_{b\,j+1,k-1}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{\xi_{j+\frac{1}{2},k-\frac{1}{2}}}\right. \\ &\left. + i_{b\,j-1,k+1}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{\xi_{j-\frac{1}{2},k+\frac{1}{2}}} + i_{b\,j-1,k+1}\left(\frac{\alpha_{\xi\eta}Q_\eta}{J}\right)_{\xi_{j-\frac{1}{2},k-\frac{1}{2}}}\right]\end{aligned} \tag{3.76}$$

Central differencing was used for all the Q derivatives in the above equation. For other cross derivative terms, the finite-difference expressions can be similarly obtained. The resulting finite-difference expressions for the Poisson equations can be given in the form:

$$L_\xi L_\eta L_\zeta (Q^{n+1} - Q^n) = R \tag{3.77}$$

The solution procedure for the above equation involves three sequential ADI sweeps in the $\xi$, $\eta$ and $\zeta$ directions. The following illustrates the step-by-step procedure for this algorithm.

$$\begin{aligned} L_\xi \Delta Q^* &= R \\ L_\eta \Delta Q^{**} &= \Delta Q^* \\ L_\zeta \Delta Q^{n+1} &= \Delta Q^{**} \\ Q^{n+1} &= Q^n + \Delta Q^{n+1} \end{aligned} \tag{3.78}$$

### 3.4.2 Crocco relations

For inviscid flows, the momentum equation can be written as

$$\rho(\vec{V} \cdot \nabla)\, \vec{V} = -\nabla p \tag{3.79}$$

Substituting the vector identity below

$$(\vec{V} \cdot \nabla)\, \vec{V} = \nabla \left( \frac{V^2}{2} \right) - \vec{V} \times \vec{\omega} \tag{3.80}$$

into equation(3.79), the inviscid momentum equation can then be rewritten as

$$\nabla \left( \frac{V^2}{2} \right) - \vec{V} \times \vec{\omega} = -\frac{\nabla p}{\rho} \tag{3.81}$$

Crocco's equation is then obtained from the above equation by expressing the pressure gradient in terms of an entropy and an enthalpy gradients using the Gibbs equation

(3.82), which is a thermodynamic relation derived from the first and second laws of thermodynamics.

$$T\nabla s = \nabla h - \frac{\nabla p}{\rho} \tag{3.82}$$

For a steady, adiabatic flow ($H = h + \frac{1}{2}V^2 =$ constant), the Crocco relation can thus be written as

$$\overrightarrow{V} \times \vec{\omega} = -T\nabla s \tag{3.83}$$

### 3.4.3 Convection of entropy and stagnation enthalpy

By taking the dot product of the velocity vector and the Crocco equation (3.83), and assuming constant total enthalpy, the convection of entropy is found to be

$$\overrightarrow{V} \cdot \nabla s = 0 \tag{3.84}$$

or in Cartesian coordinates

$$us_x + vs_y + ws_z = 0 \tag{3.85}$$

By applying the chain rule of differentiation to the above equation, and the definition of contravariant velocities in equations (3.20), it can be shown that in the computational domain, the entropy convection can be expressed as

$$Us_\xi + Vs_\eta + Ws_\zeta = 0 \tag{3.86}$$

Similar equations can be obtained for convection of stagnation enthalpy by substituting stagnation enthalpy for entropy.

**3.4.3.1 Finite-difference formulations** To illustrate the differencing used with the convection equations, consider the entropy convection equation in generalized coordinates, equation (3.86). Three-point second-order central differencing is used in $\eta$ and $\zeta$ and upwind differencing is used in $\xi$. Thus $Us_\xi$ is differenced as

$$Us_\xi = \frac{U + |U|}{2}\delta_\xi^b s + \frac{U - |U|}{2}\delta_\xi^f s \tag{3.87}$$

Depending on the values of the contravariant velocity, U, either forward or backward upwind differencing is used. The convection equation is then solved using the approximate factorization algorithm used by Bridgeman et al. (1982).

$$\begin{aligned}(I + h\tfrac{U+|U|}{2}\delta_\xi^b + hW\delta_\zeta)(I + h\tfrac{U-|U|}{2}\delta_\xi^f + hV\delta_\eta)(s^{n+1} - s^n) &= \\ -h(\tfrac{U+|U|}{2}\delta_\xi^b + \tfrac{U-|U|}{2}\delta_\xi^f + V\delta_\eta + W\delta_\zeta)s^n\end{aligned} \tag{3.88}$$

where $h$ is a relaxation parameter, $h > 0$. Adding second-order numerical dissipation in the $\eta$ and $\zeta$ directions to the above equation gives

$$\begin{aligned}&(I + hU^+\delta_\xi^b + hW\delta_\zeta - h|W|\Delta\nabla|_\zeta)\times \\ &(I + hU^-\delta_\xi^f + hV\delta_\eta - h|V|\Delta\nabla|_\eta)(s^{n+1} - s^n) = \\ &-h\left[U^+\delta_\xi^b + U^-\delta_\xi^f + V\delta_\eta + W\delta_\zeta + |W|\Delta\nabla|_\eta + |V|\Delta\nabla|_\zeta\right]s^n\end{aligned} \tag{3.89}$$

where $U^+ = \frac{U+|U|}{2}$ and $U^- = \frac{U-|U|}{2}$.

### 3.4.4 Bernoulli equation

By assuming that the flow is steady, inviscid, adiabatic, and using the perfect gas relations, the Bernoulli equation can be derived from the Crocco relations and perfect gas relations (Anderson et al., 1984):

$$\frac{\rho}{\rho_\infty} = \left[1 + \frac{\gamma - 1}{2}\left(M_\infty^2 - \frac{u^2 + v^2 + w^2}{a_\infty^2}\right)\right]^{\frac{1}{\gamma-1}} e^{-(s-s_\infty)/R} \tag{3.90}$$

For irrotational flow, the entropy correction term, $e^{-(s-s_{\infty})/R}$, can be dropped.

To avoid expensive exponential operations, the exponential term was expanded using the binomial expansion:

$$(1 \pm \alpha)^n = 1 \pm \alpha n + \frac{n(n-1)}{2}\alpha^2 \pm \frac{n(n-1)(n-2)}{2 \times 3}\alpha^3 + \ldots \tag{3.91}$$

where

$$\alpha = \frac{\gamma - 1}{2}\left(M_{\infty}^2 - \frac{u^2 + v^2 + w^2}{a_{\infty}^2}\right) \tag{3.92}$$

Written for chaining

$$(1 + \alpha)^n = 1 + \alpha n(1 + \alpha\frac{(n-1)}{2}(1 + \alpha\frac{(n-2)}{3}(1 + \alpha\frac{(n-3)}{4}\ldots))) \tag{3.93}$$

or

$$\begin{aligned} &(1 + \alpha)^n = \\ &1 + \alpha(n + \alpha(\frac{n(n-1)}{2} + \alpha(\frac{n(n-1)(n-2)}{6} + \alpha\frac{n(n-1)(n-2)(n-3)}{12}))) \end{aligned} \tag{3.94}$$

or

$$(1 + \alpha)^n = 1 + \alpha(c_1 + \alpha(c_2 + \alpha(c_3 + \alpha c_4))) \tag{3.95}$$

where

$$c_1 = \frac{1}{\gamma - 1} \tag{3.96}$$

$$c_2 = \frac{c_1}{2}(\frac{1}{\gamma - 1} - 1) \tag{3.97}$$

$$c_3 = \frac{c_2}{3}(\frac{1}{\gamma - 1} - 2) \tag{3.98}$$

$$c_4 = \frac{c_3}{4}(\frac{1}{\gamma - 1} - 3) \tag{3.99}$$

### 3.4.5 Vorticity consistency condition

From the vector identity

$$\nabla \cdot (\nabla \times \vec{V}) = \nabla \cdot \vec{\omega} = 0 \tag{3.100}$$

the consistency relation is obtained

$$\partial_x \omega_1 + \partial_y \omega_2 + \partial_z \omega_3 = 0 \tag{3.101}$$

This consistency condition is used with the tangency boundary condition to form the boundary condition imposed for the solid boundary.

### 3.4.6 Boundary conditions

Since the boundary conditions coded for the thin-layer Navier-Stokes equations are of the modular form, they are readily available for the vorticity-velocity formulation. The only boundary condition implemented specifically for the vorticity-velocity formulation is the tangency/vorticity consistency condition. This boundary condition was implemented implicitly to overcome the slow convergence found in the course of this research.

**3.4.6.1 Tangency/vorticity consistency boundary condition** The tangency (or no-flow-through) condition was imposed on a solid boundary by setting the contravariant velocity to zero in the direction normal to the solid boundary. This was combined with the definition of vorticity to form the boundary condition. The following shows the derivation of this boundary condition. From the vorticity definition

$$\zeta_y w_\zeta - \zeta_z v_\zeta = f_1 = \omega_1 - (\xi_y w_\xi + \eta_y w_\eta - \xi_z v_\xi - \eta_z v_\eta) \tag{3.102}$$

$$\zeta_z u_\zeta - \zeta_x w_\zeta = f_2 = \omega_2 - (\xi_z u_\xi + \eta_z u_\eta - \xi_x w_\xi - \eta_x w_\eta) \tag{3.103}$$

$$\zeta_x v_\zeta - \zeta_y u_\zeta = f_3 = \omega_3 - (\xi_x v_\xi + \eta_x v_\eta - \xi_y u_\xi - \eta_y u_\eta) \tag{3.104}$$

differenced on the wall as

$$\zeta_y w - \zeta_y w^* - \zeta_z v + \zeta_z v^* = -\Delta\zeta f_1 \tag{3.105}$$

$$\zeta_z u - \zeta_z u^* - \zeta_x w + \zeta_x w^* = -\Delta\zeta f_2 \tag{3.106}$$

$$\zeta_x v - \zeta_x v^* - \zeta_y u + \zeta_y u^* = -\Delta\zeta f_3 \tag{3.107}$$

where $u^*, v^*$ and $w^*$ are values of $u, v$, and $w$ at $\Delta\zeta$ above the wall. The tangency condition on the $\zeta = 0$ surface is given by

$$\zeta_x u + \zeta_y v + \zeta_z w = 0 \tag{3.108}$$

Rewrite the above equations in matrix form:

$$\underbrace{\begin{bmatrix} 0 & -\zeta_z & \zeta_y \\ \zeta_z & 0 & -\zeta_x \\ -\zeta_y & \zeta_x & 0 \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}_{q} - \underbrace{\begin{bmatrix} 0 & -\zeta_z & \zeta_y \\ \zeta_z & 0 & -\zeta_x \\ -\zeta_y & \zeta_x & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u^* \\ v^* \\ w^* \end{bmatrix}}_{q^*} = -\Delta\zeta \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}}_{F} \tag{3.109}$$

Solving these four equations for the three unknowns[5] $u$, $v$, and $w$ on the surface by a generalized inverse (or using the vorticity relations to remove the other components

---

[5] The first three equations in Eq. (3.109) defined by the vorticity definition are not linearly independent; therefore, Eq. (3.109), in essence, consists of three equations and three unknowns.

from the tangency relation, or by multiplying the above equation by the transpose of matrix A) gives the vorticity/tangency relations:

$$
\begin{aligned}
l^2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \begin{bmatrix} l^2 - \zeta_x^2 & -\zeta_x\zeta_y & -\zeta_x\zeta_z \\ -\zeta_x\zeta_y & l^2 - \zeta_y^2 & -\zeta_y\zeta_z \\ -\zeta_x\zeta_z & -\zeta_y\zeta_z & l^2 - \zeta_z^2 \end{bmatrix} \begin{bmatrix} u^* \\ v^* \\ w^* \end{bmatrix} \\
= -\Delta\zeta \begin{bmatrix} \zeta_z f_2 - \zeta_y f_3 \\ -\zeta_z f_1 + \zeta_x f_3 \\ \zeta_y f_1 - \zeta_x f_2 \end{bmatrix}
\end{aligned}
\qquad (3.110)
$$

where $l^2 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2$

The tangency boundary condition was solved implicitly in the $\zeta$ direction along with the Poisson equations to overcome the tendency for slow convergence due to the use of an extremely fine grid spacing near the solid boundary. The details of this procedure are described in Appendix B.

### 3.4.7 Solution procedure

**3.4.7.1 Irrotational flow** Since the Poisson equations are weakly coupled, the Thomas algorithm can be used to invert each tridiagonal system of equations sequentially without resorting to the use of the more expensive block tridiagonal solver. At each grid point, an initial guess for u,v,w, and $\rho$ is made. The following procedure is then implemented to obtain the flow solution.

1. **Velocities u,v,w** Sequentially update u, v, w from the Poisson equations.

2. **Density, $\rho$** Update density from Bernoulli equation using the most recently calculated values of u,v and w.

The above procedure is repeated until convergence. The solution was considered converged when the L2-norm dropped three orders of magnitude.

**3.4.7.2 Rotational flow** At each grid point, an initial guess for u, v, w, $\rho$, s, H, $\omega_1$, $\omega_2$, and $\omega_3$ is made. The following procedure is then implemented to obtain the flow solution.

1. **Velocities u,v,w** Sequentially update u,v,w from the Poisson equations for assumed values of $\rho$,$\omega_1$,$\omega_2$, and $\omega_3$.
2. **Entropy, s, and stagnation enthalpy, H** Update s and H from the corresponding convection equations using updated values of u,v and w. Currently, H is assumed a constant.
3. **Density, $\rho$** Update density from the Bernoulli equation using the most recently calculated values of u,v,w and s.
4. **Vorticities, $\omega_1$,$\omega_2$, and $\omega_3$** Vorticity components are calculated using the Crocco relations and the consistency condition for vorticity.

This procedure is repeated until convergence is reached.

# 4. RESULTS AND DISCUSSION

The thin-layer Navier-Stokes solutions for the integrated space shuttle vehicle are presented first in this chapter followed by the solutions from the vorticity-velocity formulation for several geometries.

## 4.1 Thin-Layer Navier-Stokes Solutions

The grids used in the flow simulation for the integrated space shuttle vehicle were generated using a hyperbolic grid generator (Steger and Rizk, 1985). The ET grid extends all the way to the far-field boundary where free stream values were assumed while the SRB and ORB grids only fill a much smaller portion of the computational domain so that duplicated effort in numerical calculations can be avoided. The holes cut out in each grid as seen in Figure 2.8 are usually larger than the dimensions of the embedded body and preferably larger than the dimensions of the boundary layer so that the chimera interpolation won't be carried out in regions of high gradients. However, it may not always be possible to carry out the interpolation outside the boundary layer if two grids are very close to each other, like inside the clearance between the ET/SRB and ET/ORB. Thus, more points are needed in these regions to make the interpolation accurate or possible. The computational domain extended to the outflow boundary where zero gradients of the flow variables were implemented for all three component grids (ET, SRB and ORB). In the circumferential direction, symmetry plane boundary conditions were used for the ET, ORB and the forward and aft attach grids while periodic boundary conditions were employed for the SRB grid. The no-slip wall boundary conditions were applied on the body surface for all grids. The axis boundary condition was implemented for axes extending upstream from the nose of the respective grid to the far-field boundary. Initially, a smaller grid of 289,212 points was used in the calculation for a flow of free stream Mach number 2.0, and gradually the grid size was increased to about 750,000 points including treatment of the idealized attach hardware between the ET and ORB for a transonic flow of Mach number 1.05. Other than the grid points used for the additional grids (forward and aft attach grids), the added grid points were concentrated in regions where the important flow physics was expected to occur, e.g., regions around shock waves, the recirculation region at the back end of the ET, and the regions near the attach hardware where the flow separates due to the obstruction of the attach hardware.

Before proceeding to compute the flowfield around the integrated space shuttle vehicle, the Pegasus code (Benek et al., 1985) was used to obtain the chimera interpolation data between the component grids. Then the process of obtaining the flow solution started from a uniform free stream flow with the wall velocities gradually reduced to zero in 30 iterations to minimize the effect of possible oscillations resulting from setting the wall velocities to zero too abruptly. The flow on the ET grid was computed first with the necessary boundary conditions — far-field, wall, outflow, and chimera boundary conditions (the flow variables on the hole fringe assume free stream values initially). The solution on the ET grid was then used to update the outer and the hole boundaries of the SRB and ORB grids using the interpolation information provided by Pegasus. Then the solutions on the ET grid were written to a high speed solid state device (SSD) or on external disk. The solution on the SRB grid was then computed using the updated outer and hole boundary values and other necessary boundary conditions. As with the ET grid, the solution on the SRB grid was used to update the ORB outer and hole boundaries and the ET hole boundaries (no need to update the ET outer boundary since it was outside the SRB grid and assumed free stream values) before sending it back to SSD or external disks. A similar process was carried out for the ORB grid and other grids, if there were any. This completed an iteration for the entire overset grid. The next iteration was then repeated beginning with the ET grid and so forth.

Flows of different free stream Mach numbers — 0.6, 0.9, 1.05, 1.55 and 2.0 — were calculated. The Mach number at which the maximum pressure loading on the space shuttle occurs during ascent is usually close to this range. The Reynolds numbers used in the computation were taken from the wind tunnel tests, and the angles of attack from the actual flight data. To assess the feasibility of the chimera approach for the integrated space shuttle vehicle configuration, a supersonic flow testcase, $M_\infty = 2.0$, was chosen first since it is cheaper and easier to compute than the more difficult subsonic flow case. A coarse grid of $289,212$ points was used for the entire overset grid. Many simplifications were made for this grid as can be seen in Figure 2.4. The stings behind the SRB and ORB were used to simulate the plume effects while the ORB vertical tails, the ET/SRB attach ring, and the ET/ORB attach hardware were all missing from this grid. Since the supersonic flow has only limited upstream influence, even with these simplifications, the numerical solution was expected to capture some important flow phenomena. In Figure 4.1 the simulated oil flow obtained from PLOT3D (Buning and Steger, 1985) for a constant $\zeta$ plane is compared with the oil flow from the wind tunnel test for the integrated space shuttle vehicle at $M_\infty = 2.0$ and $\alpha = -4^\circ$. The Reynolds number was not correctly modeled at the time of this simulation due to the lack of wind tunnel test data. Nonetheless, the simulated oil flow patterns show reasonable agreement with the wind tunnel oil flow results for portions of the geometry. The differences can be attributed to inaccurate modeling of the geometry, especially in regions near the ET/ORB attach hardware, the protuberances on the SRB surface, and the back end of the ET where the wind tunnel model does not have a sting. A quantitative comparison of the surface pressure coefficients for the orbiter is presented in Figure 4.3a. The flight data (Rockwell International, 1983) in the figure were taken on a constant angle station ($\phi = 70^\circ$) along the side of the orbiter fuselage. As noted in the figure, there are discrepancies in the angle of attack and the elevon deflection; however, the computation does show a trend similar to the flight data.

For the $M_\infty = 1.55$ case, the same grid was used but with the correct wind tunnel Reynolds number, $Re = 3.2 \times 10^6/ft$, and the angle of attack, $\alpha = -6^\circ$. Figure 4.2 shows the surface pressure coefficient comparisons between the computation and the wind tunnel tests (Spangler, 1981). The 3% wind tunnel model of the integrated space shuttle vehicle was equipped with 1538 pressure taps which provided enough data to allow extraction of meaningful pressure contours from the experimental data. The wind tunnel test was carried out by Rockwell International, Inc. For convenience of reproduction, gray scale contours, instead of color contours, of the surface pressure coefficient are given in Figure 4.2. Unlike the color contours, the gray scale contours

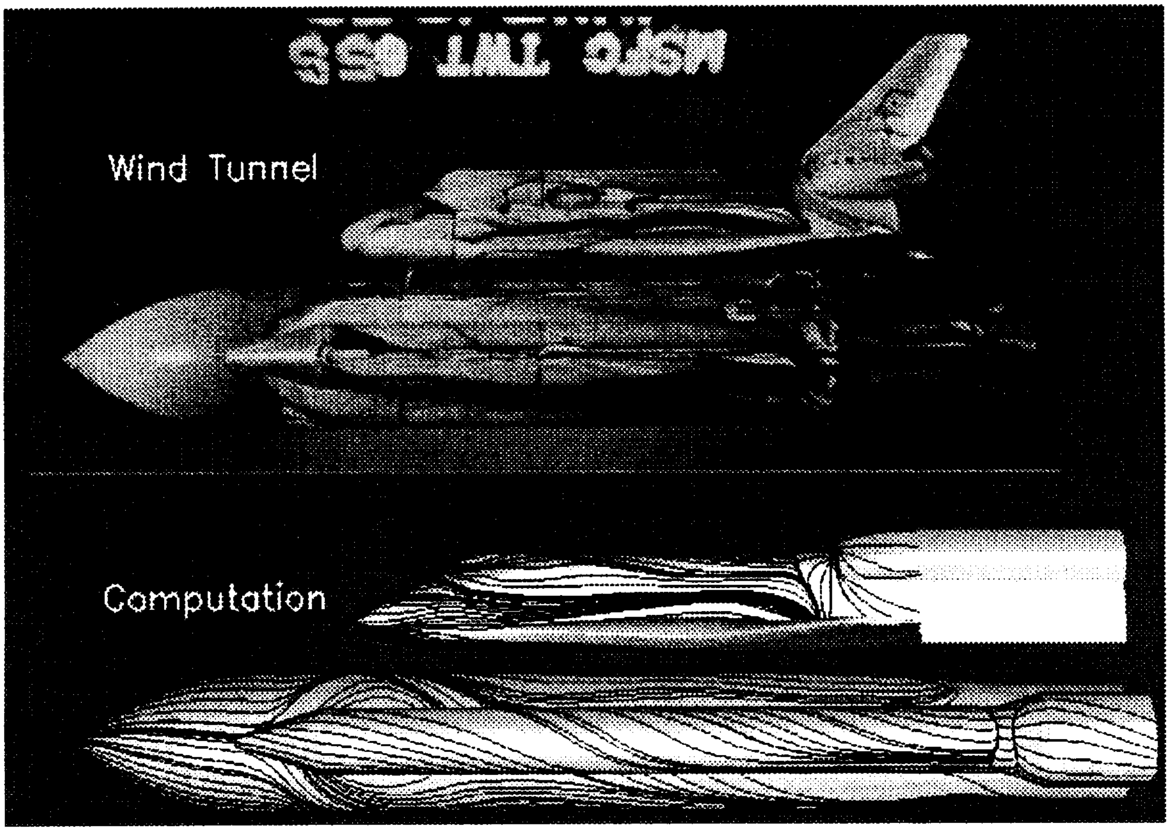
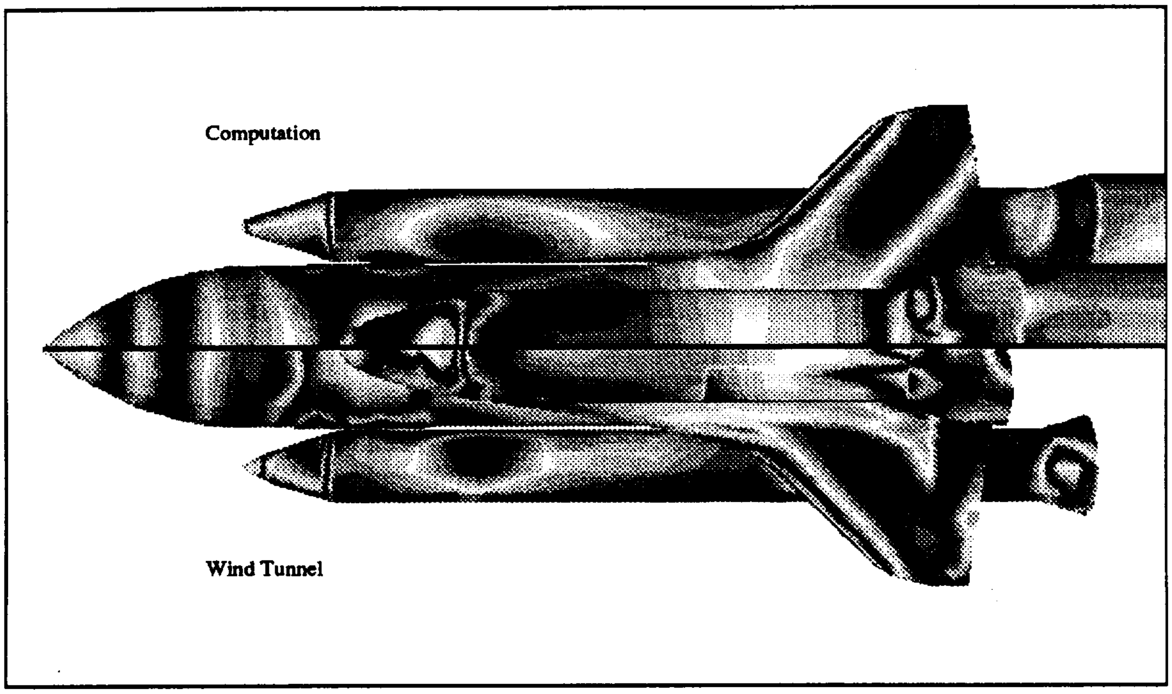


Figure 4.1: Wind tunnel and simulated surface oil flow for the integrated vehicle at $M_\infty = 2.0$ and $\alpha = -4^\circ$

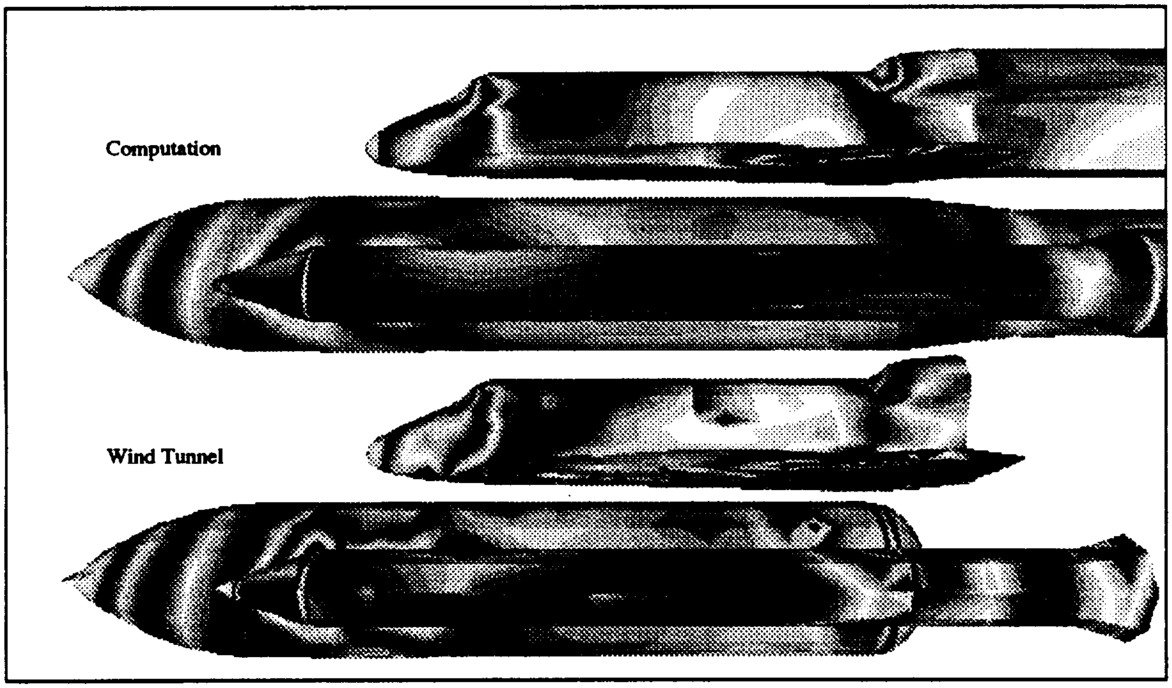
of the surface pressure coefficient in Figure 4.2 do not show the magnitude of the pressure coefficient. However, the variational change does offer insight on the extent of the expansion and compression regions. Overall, the computation does agree reasonably well with the experimental results except where the computational model failed to model the real geometry, e.g., the lack of the ET/ORB attach hardware and the ET/SRB attach ring and the redundant ET sting. In Figure 4.3b, the comparison of surface pressure coefficients for the orbiter along the $\phi = 70^\circ$ line shows that the numerical solution is generally in agreement with the experimental and flight data. The major discrepancy is at the trailing edge of the orbiter wing where the numerical model did not correctly model the elevon deflection.

For the flows of $M_\infty = 0.6$ and 0.9, a grid of 400,902 points was used. This grid still contains only the three major components of the integrated space shuttle vehicle, i.e., the ET, SRB and ORB. The added points (as compared with the one used for $M_\infty = 1.55$ and 2.0) were concentrated in regions where rapid changes of flow were expected like regions near shock waves, and the rear of the external tank (the sting was removed from the ET). Figures 4.3d and 4.3e show the surface $C_p$ comparisons for the orbiter at the $\phi = 70^\circ$ line, and as indicated in the figures, the computations did not use the correct elevon deflection. That is the possible cause for the disagreement in the $C_p$ comparisons at the trailing edge of the orbiter wing. Other than the orbiter wing trailing edge region, the numerical solutions are generally in better agreement with the wind tunnel data than with the flight data.

At $M_\infty = 1.05$, a refined grid of 771,033 points was used which includes the ET/ORB forward and aft attach grids in addition to the ET, SRB, and ORB grids. The Reynolds number, $Re = 4.0 \times 10^6/ft$, was taken from the wind tunnel test, and

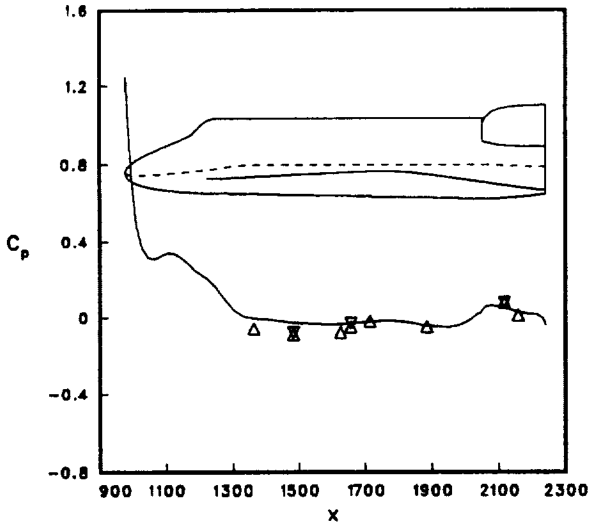


(a) Top view

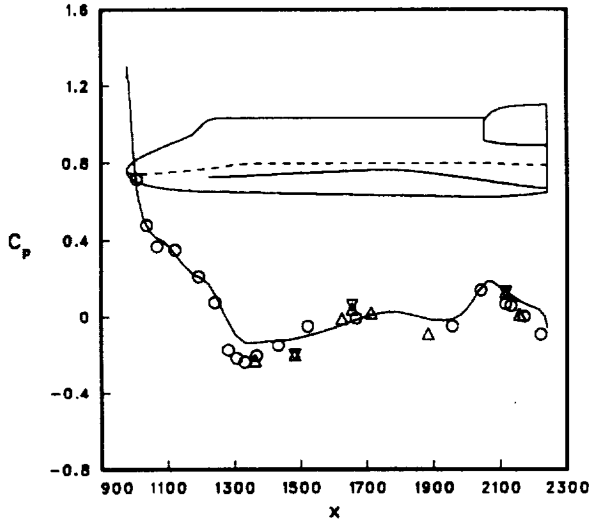


(b) Side view

Figure 4.2: Comparison of computational and wind tunnel surface pressure coefficient at $M_\infty = 1.55$, $\alpha = -6^\circ$, and $Re = 3.2 \times 10^6/ft$ (3% model)

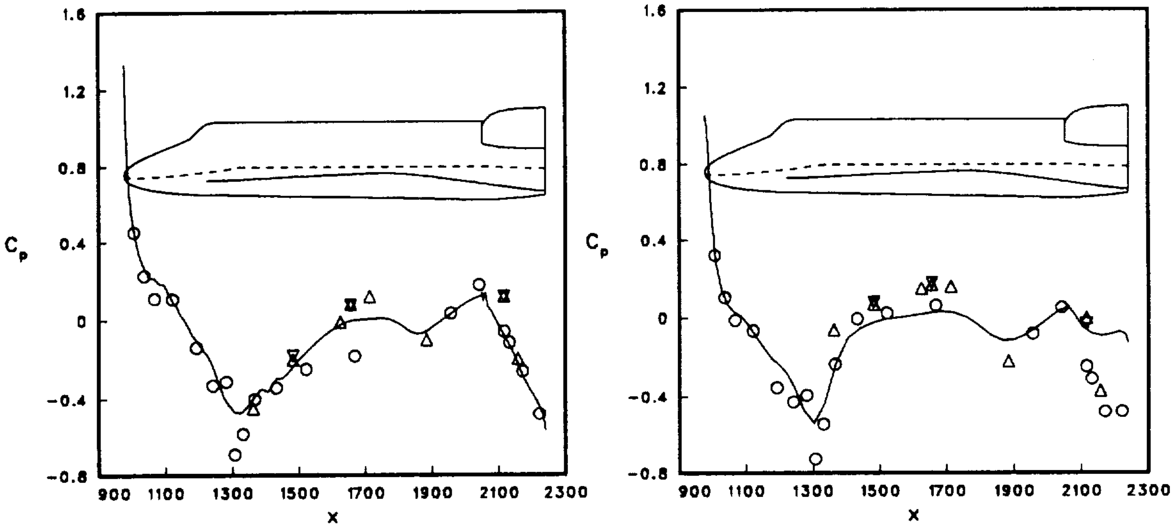


(a) $M_\infty = 2.0$ and $\alpha = -4^\circ$; (computation: no elevon deflection; flight: $8^\circ/-5^\circ$ inboard/outboard elevon deflection, $\alpha = -2^\circ$)



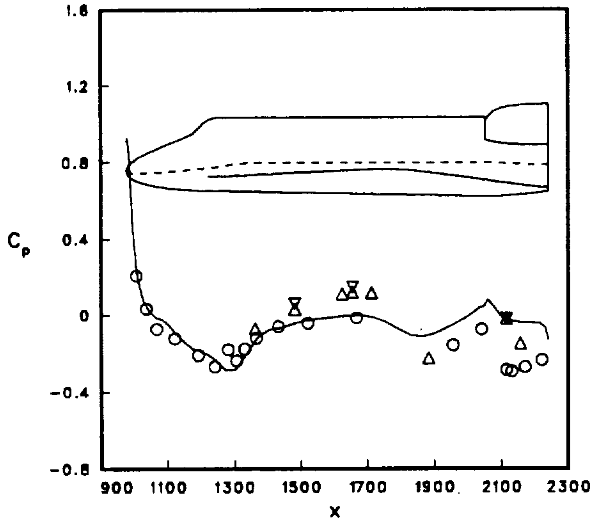
(b) $M_\infty = 1.55$ and $\alpha = -6^\circ$, $Re = 3.2 \times 10^6/ft$; (computation: no elevon deflection; wind tunnel: $10^\circ/-7^\circ$ elevon deflection; flight: $8^\circ/-5^\circ$ inboard/outboard elevon deflection, $\alpha = -2^\circ$).

Figure 4.3: Comparison of $C_p$ from computation (–), wind tunnel (○), and flight ($\nabla$ right side, $\Delta$ left side) along the $\phi = 70^\circ$ line of the orbiter fuselage at $M_\infty = 2.0$, 1.55, 1.05, 0.9 and 0.6 (x in inches)



(c) $M_\infty = 1.05$ and $\alpha = -3^\circ$, $Re = 4.0 \times 10^6/ft$; (all: $10^\circ/9^\circ$ inboard/outboard elevon deflection).

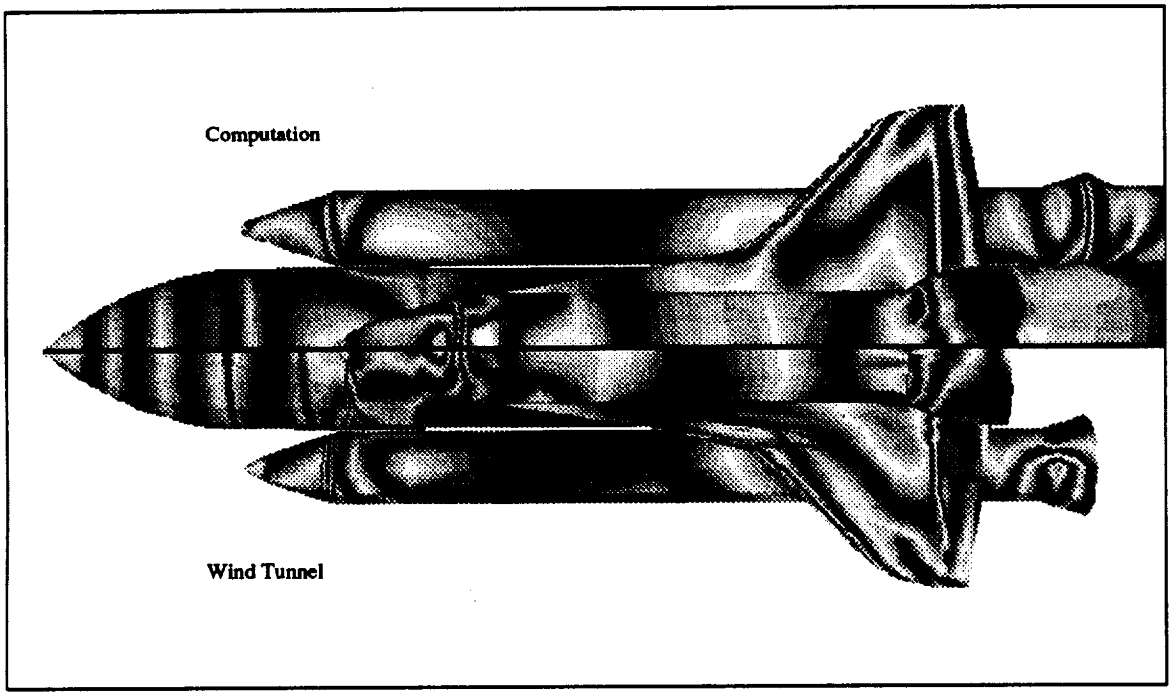
(d) $M_\infty = 0.9$ and $\alpha = -3^\circ$, $Re = 4.0 \times 10^6/ft$; (computation: no elevon deflection; wind tunnel and flight: $10^\circ/9^\circ$ inboard/outboard elevon deflection).



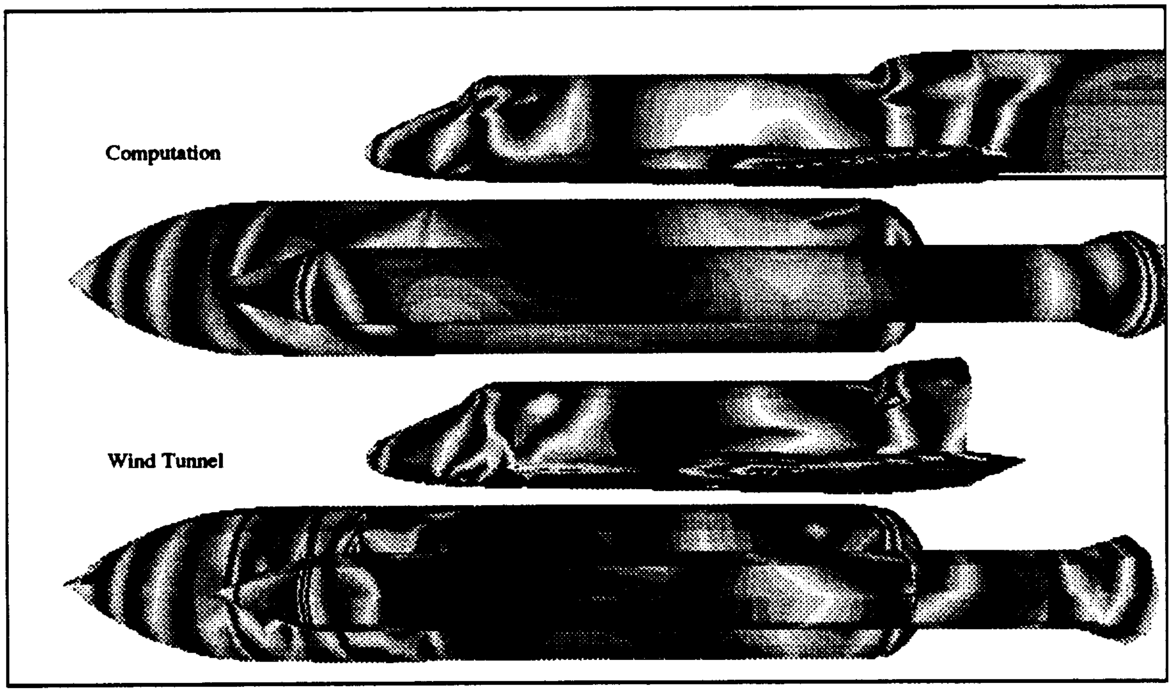
(e) $M_\infty = 0.6$ and $\alpha = -3^\circ$, $Re = 4.0 \times 10^6/ft$; (computation: no elevon deflection; wind tunnel and flight: $10^\circ/9^\circ$ inboard/outboard elevon deflection).

Figure 4.3 (Continued)

the angle of attack, $\alpha = -3^{\circ}$, and the elevon deflection, $10^{\circ}/9^{\circ}$ (inboard/outboard), were those of the wind tunnel and flight data. As observed from the surface pressure coefficients for the orbiter along the fuselage $\phi = 70^{\circ}$ line in Figure 4.3c, the $C_p$ at the orbiter trailing edge is in better agreement than those of other Mach numbers. Overall, the computation is in better agreement with the flight data than with the wind tunnel data for most part of the fuselage — most likely due to the wall interference in the wind tunnel test. Additional surface pressure comparisons between the numerical solutions and the wind tunnel data are presented in Figures 4.4 and 4.5. The shaded surface pressure coefficient comparisons in Figure 4.4 show a similar variation in the pressure contours for both the computational and experimental results. The quantitative comparisons of $C_p$ at various constant angle lines for the ET, SRB and ORB are shown in Figure 4.5. For the most part, the computation is in good agreement with the wind tunnel data. The discrepancy near the back of the orbiter and the external tank, as observed in Figures 4.5a and 4.5b, can in part be attributed to the inadequate modeling of the ET/ORB attach hardware which only accounts for 50% of the blockage incurred from the real attach hardware and the fuel feed lines. The other cause for the discrepancy is that the wind tunnel model only had stings attached to the SRB nozzles while the numerical model had a sting extending from the back of the orbiter in addition to the SRB stings. For the $C_p$ of the SRB in Figure 4.5c, the absence of the ET/SRB attach ring in the SRB grid is thought to be the major reason for the discrepancy between the numerical solutions and the wind tunnel data.

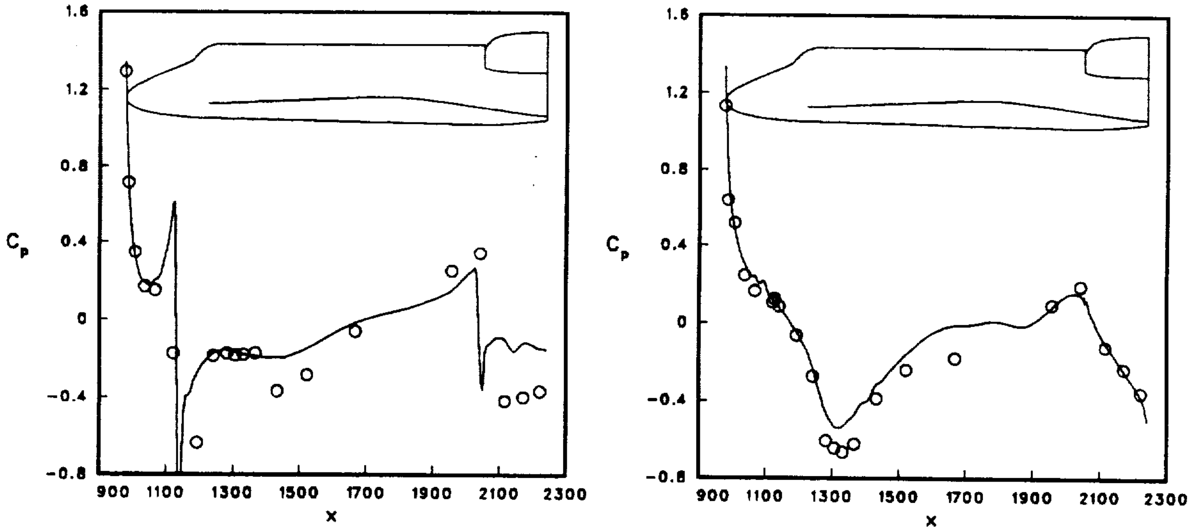


(a) Top view



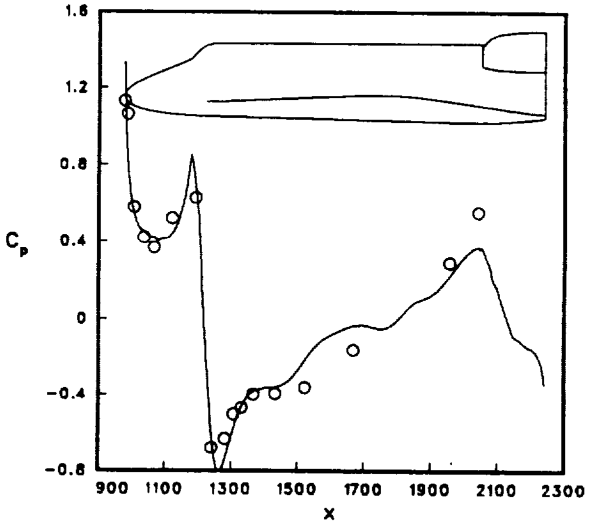
(b) Side view

Figure 4.4: Comparison of computational and wind tunnel surface pressure coefficient at $M_\infty = 1.05$, $\alpha = -3^\circ$, and $Re = 4.0 \times 10^6/ft$ (3% model)



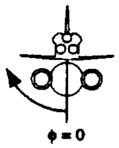
(1) $\phi = 0^{\circ}$ (bottom)

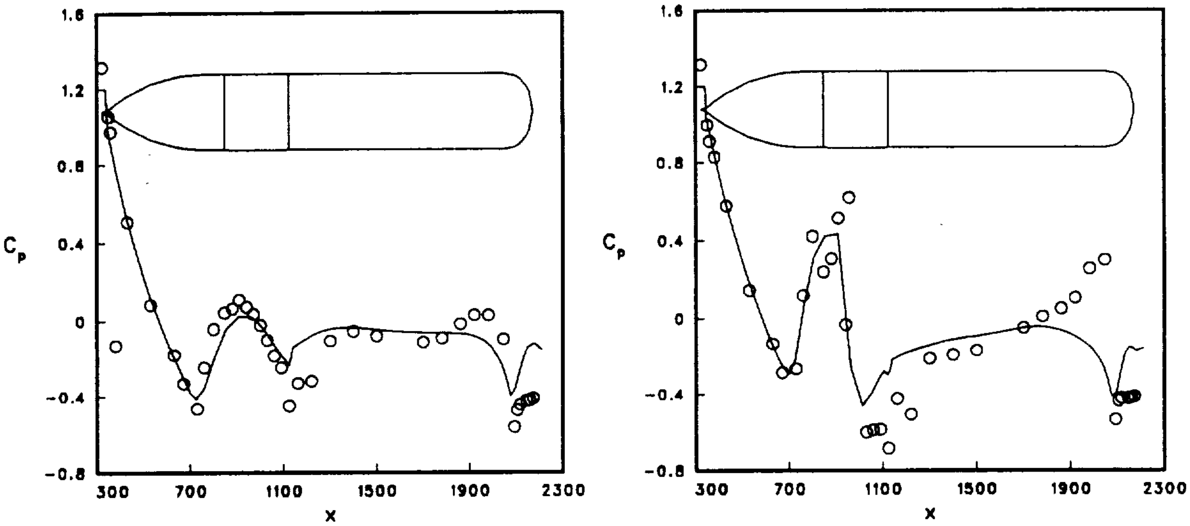
(2) $\phi = 90^{\circ}$ (side)



(3) $\phi = 180^{\circ}$ (top)

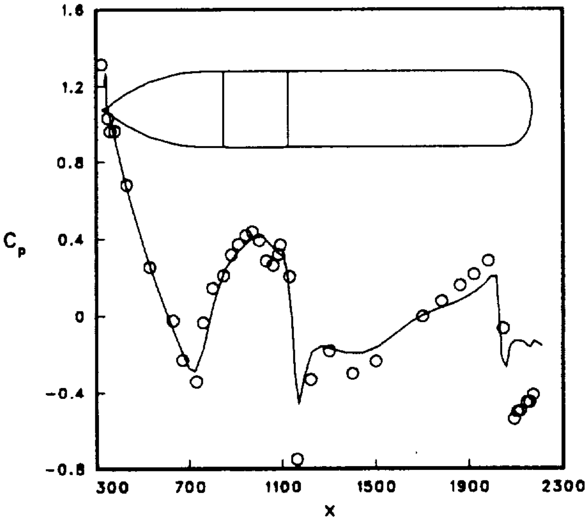
Figure 4.5: Comparison of $C_p$ from computation (–) and wind tunnel (o) for various lines along the orbiter for $M_\infty = 1.05$, $\alpha = -3^{\circ}$, and $Re = 4.0 \times 10^6/ft$ (x in inches)





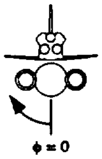
(1) $\phi = 0^\circ$ (bottom)

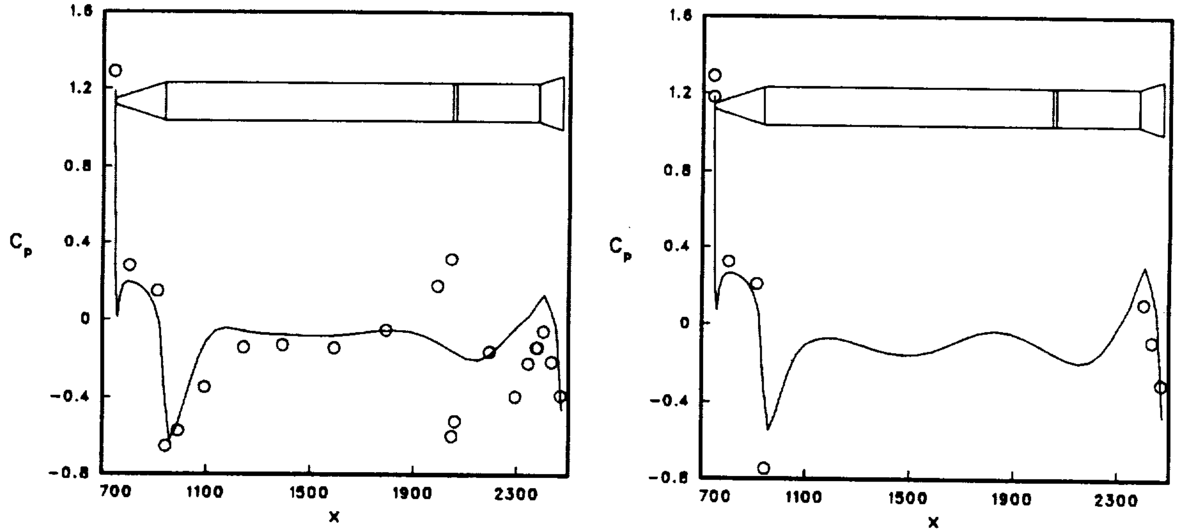
(2) $\phi = 90^\circ$ (side)



(3) $\phi = 180^\circ$ (top)

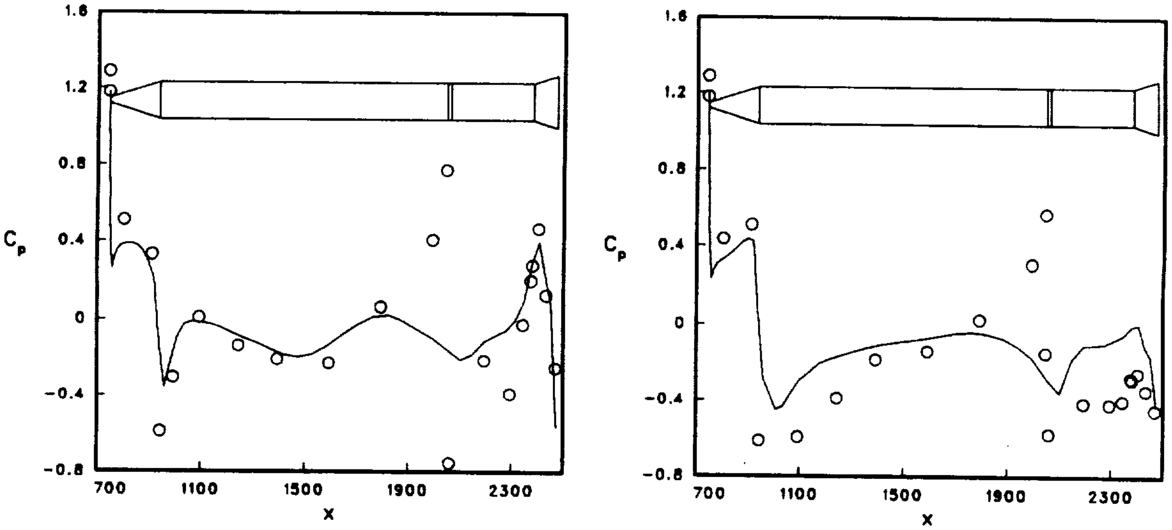
Figure 4.6: Comparison of $C_p$ from computation (-) and wind tunnel (o) for various lines along the external tank for $M_\infty = 1.05$, $\alpha = -3^\circ$, and $Re = 4.0 \times 10^6/ft$ (x in inches)





(1) $\phi = 0^{\circ}$ (bottom)

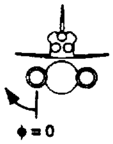
(2) $\phi = 90^{\circ}$ (outer side)



(3) $\phi = 180^{\circ}$ (top)

(4) $\phi = 270^{\circ}$ (inner side)

Figure 4.7: Comparison of $C_p$ from computation (–) and wind tunnel (○) for various lines along the solid rocket booster for $M_{\infty} = 1.05$, $\alpha = -3^{\circ}$, and $Re = 4.0 \times 10^6/ft$ (x in inches)



## 4.2 Vorticity-Velocity Solutions

Numerical solutions were obtained for several geometries including a sphere, an ellipsoid, the ET, and the integrated space shuttle vehicle and compared with experimental results, exact solutions, and numerical solutions of the thin-layer Navier-Stokes and Euler equations. Like the grids used for the thin-layer Navier-Stokes calculation, the grids used for the vorticity-velocity calculation were all generated using a hyperbolic grid generator (Steger and Rizk 1985). The sphere test cases are presented first to illustrate various aspects of the numerical scheme, such as the accuracy of the vorticity-velocity formulation, grid refinement effects, and effects of the outer boundary location on the solution. Then the ellipsoid testcases are presented for further validation of the numerical scheme. A comparison with experimental results is included for the ellipsoid at 10° angle of attack. The solution of the ET alone is then presented followed by the solution of the integrated space shuttle vehicle.

### 4.2.1 Sphere

Since the analytical solution for a potential flow around a sphere is available for comparison, and the sphere grid is relatively easy to generate, the flow past a sphere was chosen as the first test case to verify the vorticity-velocity scheme. The flow was assumed inviscid and irrotational and the calculations were carried out for both the single as well as the chimera overset grids to verify the vorticity-velocity formulation in the chimera approach. Grids of different sizes (or different number of points) were used to investigate the effect of grid refinement. Since the far-field boundary condition used for the sphere is of the Dirichlet type (i.e., values of the flow variables are specified at the far-field boundary — in this case, free stream values were used), it is important to set the far-field boundary far enough away from the surface to obtain an accurate solution. Thus, the effect of the far-field boundary location on the solution was evaluated. Comparisons of the Poisson solution with the analytical (Kaplan, 1940) and Euler solutions were made to evaluate the accuracy of the scheme. Flow conditions were set at Mach number 0.57 and zero angle of attack for a unit sphere. The symmetry plane boundary condition was used to save computer time. The tangency/vorticity consistency boundary condition was imposed on the body surface and the axis averaging boundary condition was used for the two axes extending from both poles of the sphere to the far-field boundary. For the Euler solution, the tangency boundary condition (explained previously in the chapter on "Governing Equations and Numerical Methods") was used on the surface and the rest of the boundary conditions were the same as those used for the vorticity-velocity algorithm.

The analytical solutions of a compressible, irrotational flow past a circular cylinder and a sphere were first calculated by Janzen (1913) and Rayleigh (1916). Their method added a correction term, which involved only the square of the Mach number, to the incompressible flow solution. Kaplan (1938) and Imai (1938) extended the calculations by including the terms involving the fourth power of the Mach number for a circular cylinder.

$$\phi = \phi_0 + \phi_1 M^2 + \phi_2 M^4 + \cdots \tag{4.1}$$

where $\phi$ is the velocity potential and $\phi_0$ the solution of the Laplace equation, $\nabla^2\phi_0 = 0$, for an incompressible flow. By inserting Eq. 4.1 into the continuity equation,

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = \frac{1}{2c^2}\left(\frac{\partial v^2}{\partial x}\frac{\partial\phi}{\partial x} + \frac{\partial v^2}{\partial y}\frac{\partial\phi}{\partial y} + \frac{\partial v^2}{\partial z}\frac{\partial\phi}{\partial z}\right) \tag{4.2}$$

(where $v$ is the fluid velocity and can be expressed in terms of velocity potential, $\phi$) and equating the coefficients of the same powers of Mach number on both sides of the equation, $\phi_0$, $\phi_1$, and $\phi_2$ can be determined. Kaplan (1940) used this method to solve for the irrotational, compressible flow past a sphere. His solution for the velocities on the surface of the sphere is

$$\begin{aligned}\frac{v}{v_\infty} &= \frac{2}{3}\sin\theta + \frac{1}{7040}(989\sin\theta - 1215\sin 3\theta)M^2 \\ &+ (0.10572\sin\theta - 0.16008\sin 3\theta + 0.06434\sin 5\theta)M^4 \\ &+ (\gamma - 1)(0.01168\sin\theta - 0.02475\sin 3\theta + 0.02582\sin 5\theta)M^4\end{aligned} \tag{4.3}$$

where $\gamma$ is the ratio of the specific heats. For a flow at Mach number of 0.57, and $\gamma = 1.408$, the surface velocities on a sphere are:

$$\frac{v}{v_\infty} = 1.55731\sin\theta - 0.07404\sin 3\theta + 0.00791\sin 5\theta \tag{4.4}$$

**4.2.1.1 Single grid calculation** As illustrated in Figure 4.8, the sphere grid is clustered near the surface and quickly stretched out to the far-field boundary. The reason for using a viscous grid (grids with very fine grid spacings near the wall boundary) for the inviscid flow solver used here is to test whether the current scheme would work on such a grid since one of the major objectives for developing the vorticity-velocity formulation is to see if its solution can be used to provide better initial guess for a viscous flow solver such as one using the Navier-Stokes equations. If different grids are used for the vorticity-velocity algorithm and the viscous flow solver, say for the thin-layer Navier-Stokes equations, the error introduced from the interpolation could be large, especially in the boundary layer where most of the grid points reside. For the chimera overset grid with holes embedded, the interpolation will be much more complicated since the hole points contain no meaningful data and

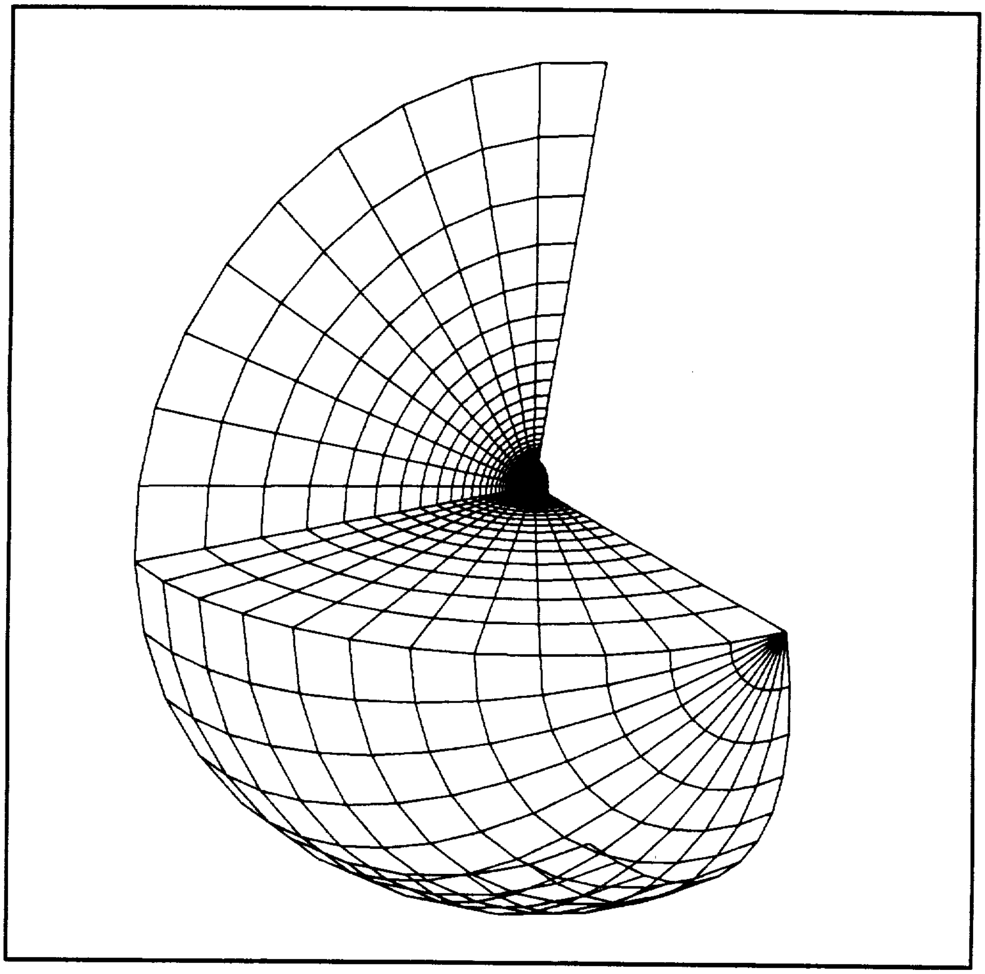


Figure 4.8: Geometry of the computational domain for a sphere

the interpolation has to exclude such data from being used. Thus it is desirable to use the same grid for both the inviscid and viscous flow solvers.

As indicated in Table 4.1, the flow around a sphere was computed for various far-field boundary locations, from 4 to 16 diameters from the center of the sphere and stretching ratios[1] around 14.5%. The comparison of surface velocities for these cases are shown in Figure 4.9. There appears no appreciable difference in the solutions for different far-field boundary locations. Thus, for other calculations for the sphere geometry, the far-field boundary location was set at 4 diameters from the center of the sphere. Also shown in the same figure is the analytical solution of Kaplan (1940). The Poisson solution predicted lower velocities than that of Kaplan in regions near $\phi = 90^\circ$ (0.38% lower at 90°); however, in general, the two solutions agree well with each other.

Table 4.1: Far-field boundary locations used for flows past a sphere

| Grid # | Grid size ($\xi \times \eta \times \zeta$) | Far-field boundary | Stretching ratio |
|---|---|---|---|
| 1 | 39 × 39 × 47 | 4 dia. | 14.5% |
| 2 | 39 × 39 × 51 | 6 dia. | 14.6% |
| 3 | 39 × 39 × 53 | 8 dia. | 14.4% |
| 4 | 39 × 39 × 58 | 16 dia. | 14.5% |

Table 4.2 lists the size of several grids of different initial spacings ($\Delta\zeta_1$[2] ), ranging from 0.0005 to 0.025. The surface velocity from solutions of these five grids are shown

---

[1] The "stretching ratio" listed in Table 4.1 in percentage actually represents the percentage increase in grid spacing from the body outward (the $\zeta$ direction). That is, for a table entry of 14.5%, the true ratio would be 1.145.

[2] $\Delta\zeta_1$ is the physical distance in the $\zeta$ direction between the surface and the first point out in the flowfield normalized by the diameter of the sphere.

along with the analytical solution (Kaplan, 1940) in Figure 4.10. The solution on the coarsest grid ($21 \times 21 \times 24$) is in total disagreement with the analytical solution. With the second coarsest grid ($27 \times 27 \times 30$), the surface velocities are much closer to the analytical solution, but still somewhat larger than that given by analytical solution. On an even finer grid ($33 \times 33 \times 35$) with an even smaller initial spacing ($\Delta\zeta_1 = 0.005$), the solution generally agrees with the analytical solution except in regions near $90^\circ$ where the Poisson solution predicts somewhat larger velocities than those of the analytical solution. The solutions from the two finest grids ($39 \times 39 \times 47$ and $47 \times 47 \times 52$) are almost identical suggesting that the solution on the finest grid can be considered as the "true" solution of the "finite-difference Poisson equations". Although the solutions from the two finest grids generally under-predict slightly the surface velocity compared to the analytical solution, they are in reasonably good agreement with the analytical solution.

Table 4.2: Grid sizes used for flows past a sphere

| Grid # | Grid size ($\xi \times \eta \times \zeta$) | $\Delta\zeta_1$ | Stretching ratio |
|---|---|---|---|
| 1 | $21 \times 21 \times 24$ | 0.025 | 14.1% |
| 2 | $27 \times 27 \times 30$ | 0.01 | 14.6% |
| 3 | $33 \times 33 \times 35$ | 0.005 | 14.6% |
| 4 | $39 \times 39 \times 47$ | 0.001 | 14.5% |
| 5 | $47 \times 47 \times 52$ | 0.0005 | 14.5% |

An Euler solution was computed for a grid of $39(\xi) \times 39(\eta) \times 47(\zeta)$ points with the initial spacing, $\Delta\zeta_1 = 0.001$, in the radial direction and a 14.5% stretching ratio. Figure 4.11 shows surface velocities from solutions of the Poisson and Euler equations as well as those from the analytical solution for compressible (Kaplan, 1940) and

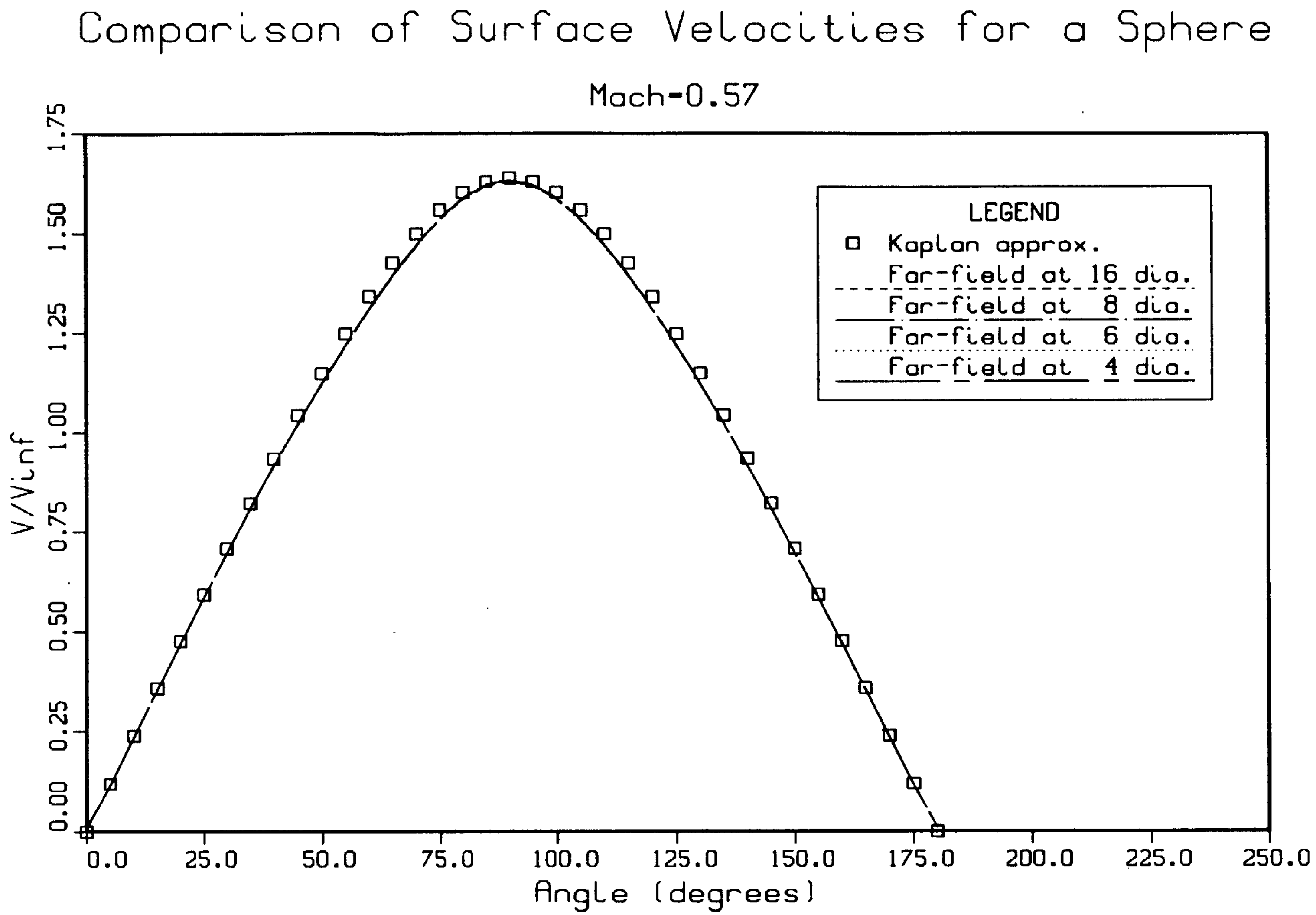


Figure 4.9: The effect of the far-field boundary location on the Poisson solution for a sphere at $M_\infty = 0.57$, and $\alpha = 0^\circ$

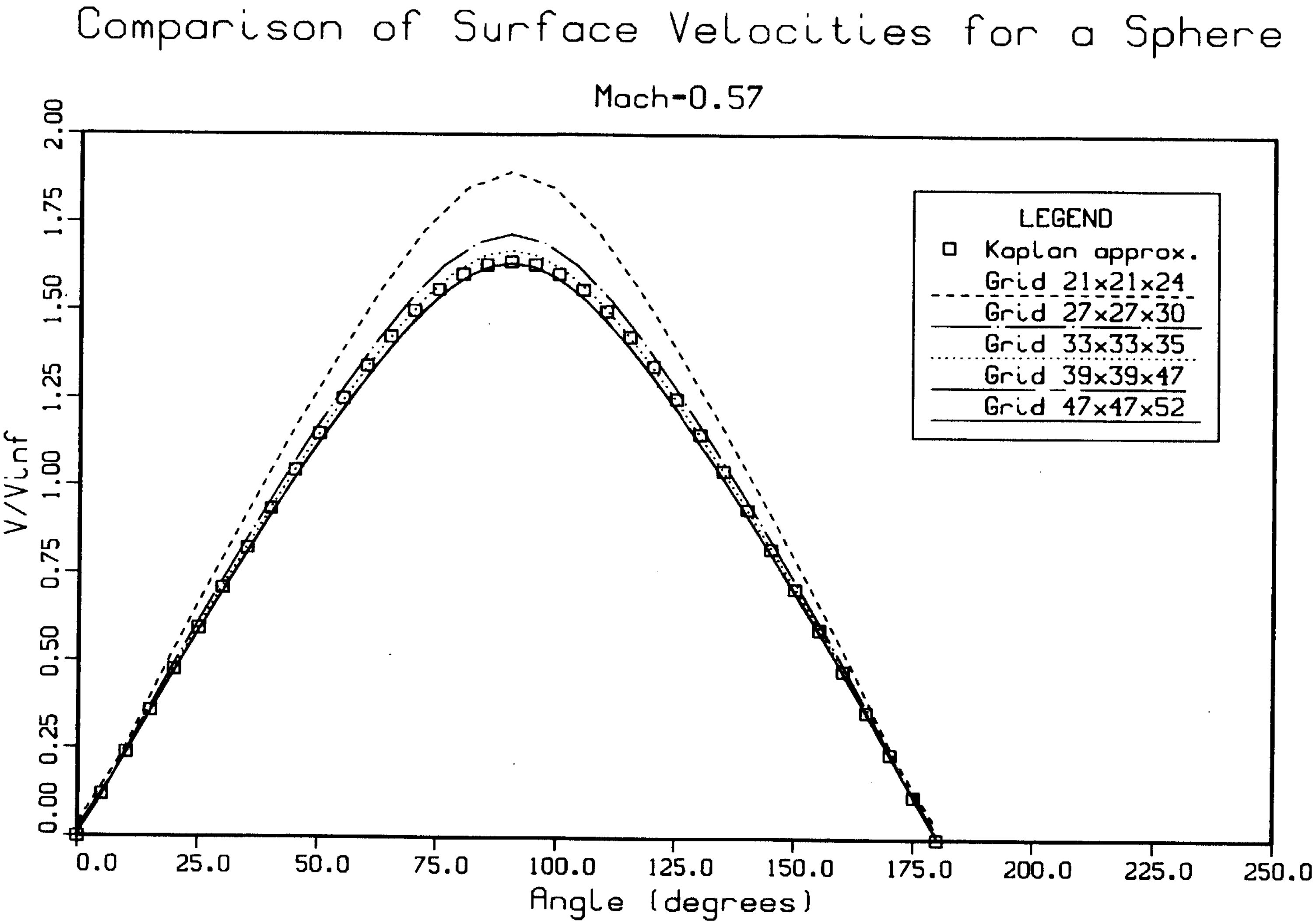


Figure 4.10: The effect of grid refinement on the Poisson solution for a sphere at $M_\infty = 0.57$, and $\alpha = 0^\circ$

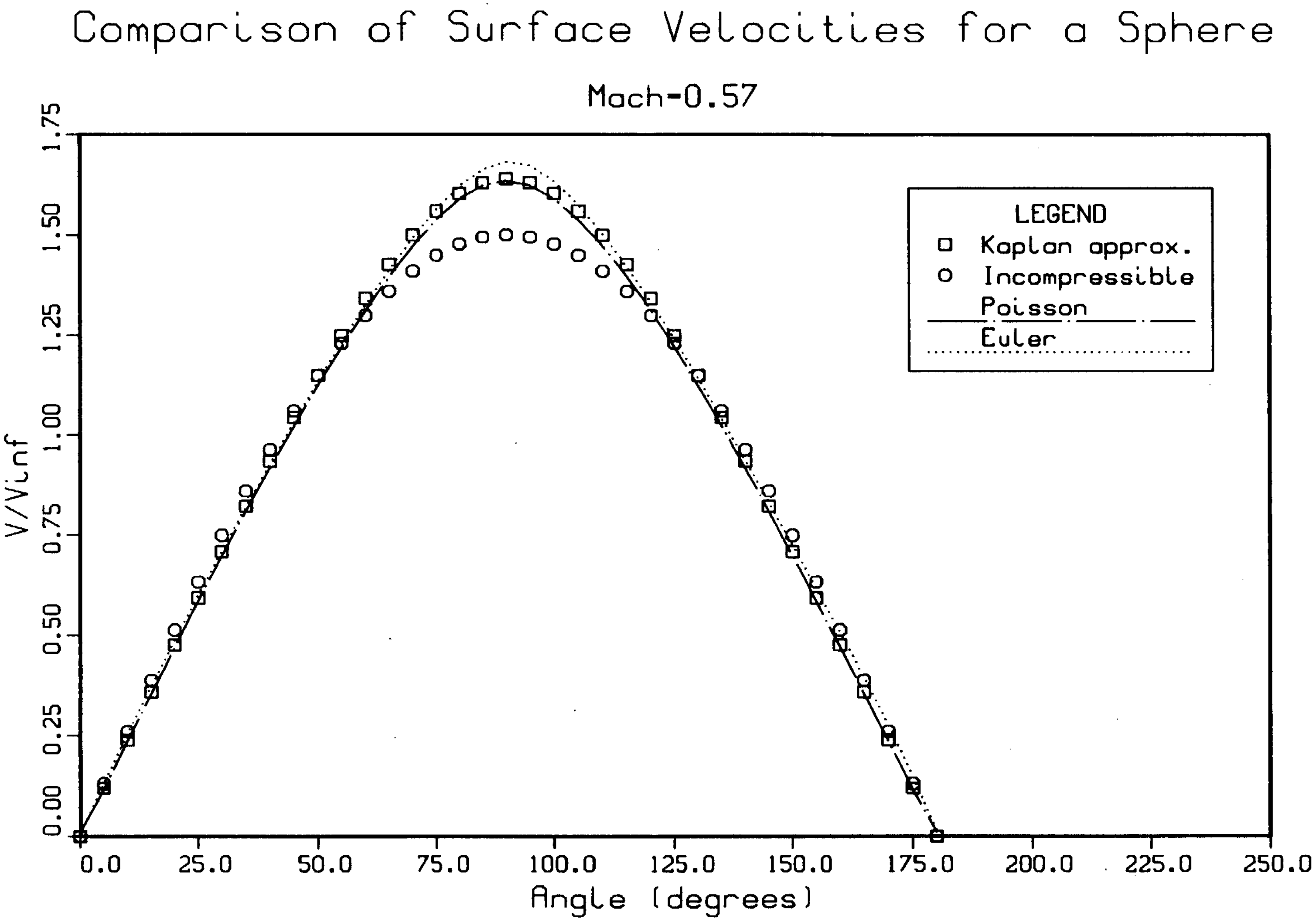


Figure 4.11: Comparison of the Poisson and Euler solutions with the analytical solution for a sphere at $M_\infty = 0.57$, and $\alpha = 0^\circ$

incompressible flows. Overall, the Poisson and Euler solutions agree well with the analytical solution for the compressible flow (Kaplan, 1940). The difference between the Poisson and the Euler solutions is that the Poisson solution under-predicts while the Euler solution over-predicts the analytical solution for the compressible flow near the 90° region. Also observed for the Euler solution is the slight asymmetry in the solution as seen from the minor downstream "shift" from Kaplan's analytical solution. The major discrepancy between the solutions of the compressible and incompressible flows is in regions near 90° where as much as 8.3% difference is observed.

Figure 4.12 shows the typical convergence history of the Poisson solver implemented in the present study. The L2-norm is found to drop very quickly in the early stages of the computation; then the rate of convergence slows down. However, the L2-norm usually drops two orders of magnitude in 100 iterations. The CPU time required for the Poisson solver is about 4.67 $\mu$s per point per iteration on a Cray YMP and requires about 150 to 500 iterations, depending on the size of the grid, to obtain a converged solution. The Euler solver requires about 31.58 $\mu$s per point per iteration and requires about 1000 iterations to converge for the same grid. The convergence criterion is based on the magnitude of the L2-norm. If it is three orders of magnitude smaller than the peak value during the solution process, the solution is considered converged.

#### 4.2.1.2 Chimera grid calculation

For the chimera scheme, two overset grids were computed and compared with the single grid solution. Both the chimera and single grid solutions were expected to be the same or very close to each other. The flow was maintained at a Mach number 0.57, and zero angle of attack. All

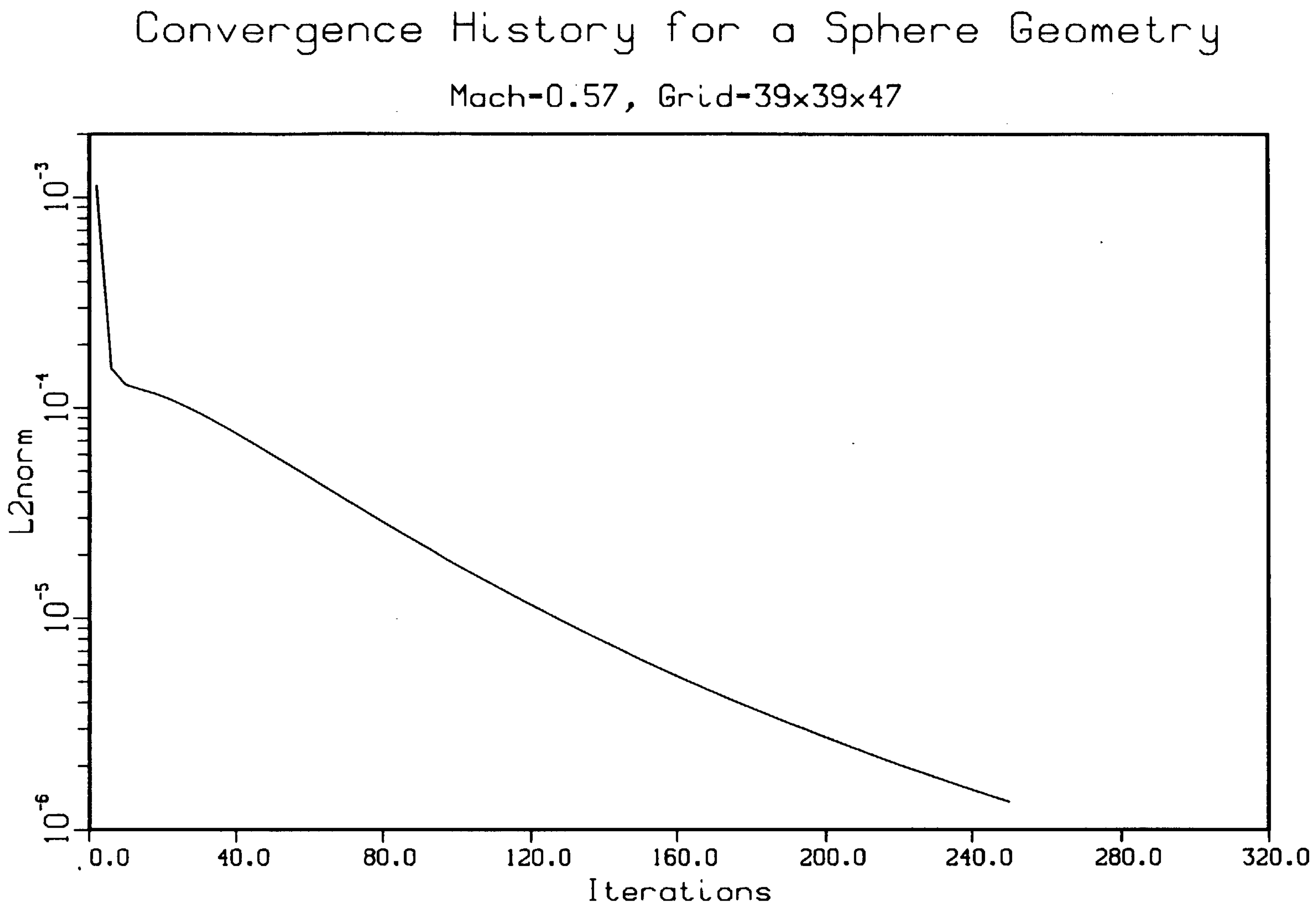


Figure 4.12: Convergence history of the Poisson solution for a sphere at $M_\infty = 0.57$, and $\alpha = 0^\circ$

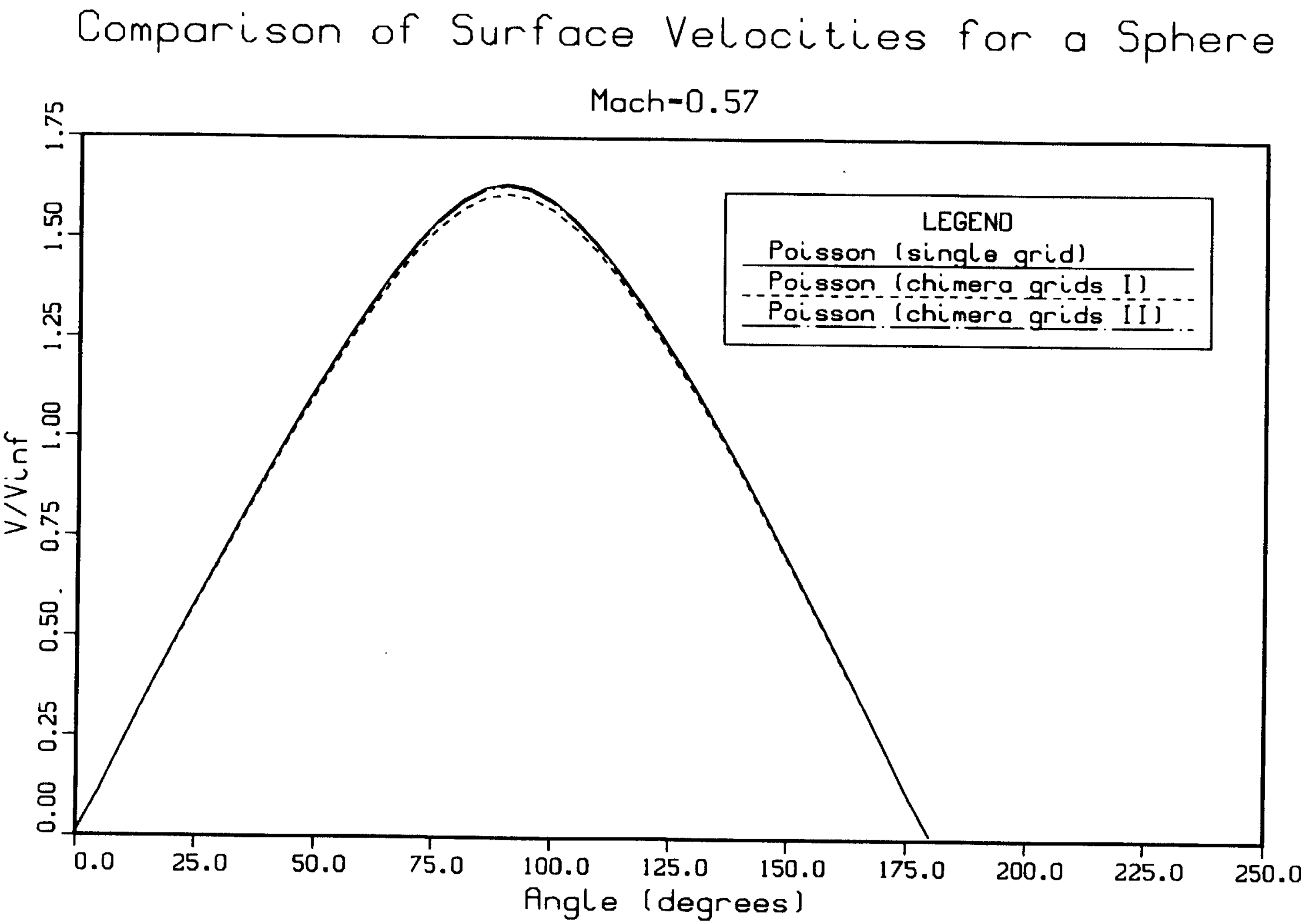


Figure 4.13: Surface velocity comparisons of the single and chimera grid solutions for a sphere at $M_\infty = 0.57$

grids (chimera or single) used were uniformly spaced in the $\xi$ (streamwise) and $\eta$ (circumferential) directions and stretched to the far-field boundary (4 diameters from the center of the sphere) from an initial spacing of 0.001 diameters in the $\zeta$ (radial) direction. The single grid had 39, 39, and 47 points in the $\xi$, $\eta$ and $\zeta$ directions, respectively. The two overset grids were taken from subsets of the single grid to avoid the possible error introduced from the interpolation at the chimera interface boundaries. The first one, designated as chimera grid I in Figure 4.13, consisted of the first thirty ($\zeta = 1$ to 30) and the last twenty-one ($\zeta = 27$ to 47) constant $\zeta$ planes of the single grid while the second one, designated as chimera grid II, consisted of the first twenty-two ($\xi = 1$ to 22) and the last twenty-two ($\xi = 18$ to 39) constant $\xi$ planes. The boundary conditions for the chimera grid were the same as those for the single grid testcase, i.e., free stream values at the far-field boundary, tangency/vorticity consistency boundary conditions on the surface of the sphere, and the axis averaging condition for the axes. Comparison of surface velocities from the single and chimera grids are shown in Figure 4.13. For the chimera grid I (four points overlapped in the $\zeta$ direction), the surface velocity at 90° is 1.57% lower than that of the single grid. This discrepancy is due to the error resulting from the one-sided differencing used in computing the metrics at the chimera interface boundary compounded with the grid stretching in the $\zeta$ direction. While for the chimera grid II, the one-sided differencing used in calculating the metrics at the chimera interface boundaries does not affect the accuracy since the chimera interface boundaries are uniformly spaced $\xi$ planes. This is evidenced by the very good agreement between the solutions of the chimera grid II and the single grid. The minor disagreement (0.3% difference) at 90° is likely due to the different differencing scheme used in the chimera scheme to take into account those points at the chimera interface boundary.

For the cases computed, the chimera scheme required about 4.5 times the CPU time (20.94 $\mu$s vs. 4.67 $\mu$s per point per iteration on a Cray YMP) needed for a corresponding single grid solution due to the extra logic required to identify the hole and the chimera interface boundary points from the field points and the extra disk I/O to bring in and put back the data from/to the disk. The time for disk I/O represented 30% of the total CPU time used and could have been reduced to a bare minimum if the SSD (solid state device) was used. Thus the actual CPU time required per point per iteration, not counting the disk I/O, for the chimera scheme is about 3.1 times that of a single grid. However, due to the explicit type boundary conditions used at the chimera interface boundary, an additional 20% more iterations were required to converge the solution than for the corresponding single grid solution.

### 4.2.2 Ellipsoid

The geometry of the computational domain for the ellipsoid computed is shown in Figure 4.14; the grid is clustered near the wall and the first grid spacing away from the surface is one ten-thousandth of the length of the ellipsoid. Unless stated otherwise, all the cases computed for the ellipsoid in this research used a $39(\xi) \times 21(\eta) \times 51(\zeta)$ grid. The $\xi$ direction is aligned with the incoming flow, $\eta$, the circumferential direction, and $\zeta$, the direction away from the body surface. The aspect ratio of the ellipsoid is 6 to 1 (length vs. diameter of the ellipsoid), and the far-field boundary, a near spherical surface, is at a distance about 5.4 times the length of the ellipsoid from the center. A symmetry plane condition is used to save CPU time. The wall boundary condition is the usual no-slip boundary condition for the thin-layer Navier-Stokes solutions, the tangency (no penetration) boundary condition for the Euler solutions, and the tangency/vorticity consistency boundary condition for the Poisson solutions. The axis boundary condition is used for the two axes extending from both ends of the ellipsoid to the far-field boundary. For details on these boundary conditions, see Chapter 3 on the "Governing Equations and Numerical Methods".

In the $0^{\circ}$ and $-3^{\circ}$ angles of attack testcases, the Mach number was taken as 0.6 and the Reynolds number, 10 million. For the thin-layer Navier-Stokes solution, the flow was assumed turbulent and the Baldwin-Lomax turbulence model was used. Figure 4.15 compares the surface pressure coefficients from the thin-layer Navier-Stokes, Euler and Poisson solutions for an ellipsoid at the plane of symmetry for a flow at $0^{\circ}$ angle of attack. All three solutions show good agreement with each other in most of the flow region except far downstream where the flow separates due to an adverse pressure gradient. The pressure coefficients on the top and bottom surfaces of the ellipsoid are found to be the same for all three solutions as expected for axisymmetric flows. A similar comparison is also presented in Figure 4.16 for a plane perpendicular ($\phi = 90^{\circ}$) to the plane of symmetry. The solution at this plane is essentially the same as the solution on the plane of symmetry due to $0^{\circ}$ angle of attack, and thus the same level of agreement is observed. For the $\alpha = -3^{\circ}$ case, the thin-layer Navier-Stokes solutions were computed and compared in Figure 4.17 for the symmetry plane, and in Figure 4.18 for the $\phi = 90^{\circ}$ plane. The Poisson, Euler, and thin-layer Navier-Stokes solutions agree with each other in most regions of the flowfield although the Poisson solution gives somewhat lower pressure than the other two solutions. The major difference is at the downstream end of the ellipsoid due to the separation resulting from the adverse pressure gradient. Unlike the solution for

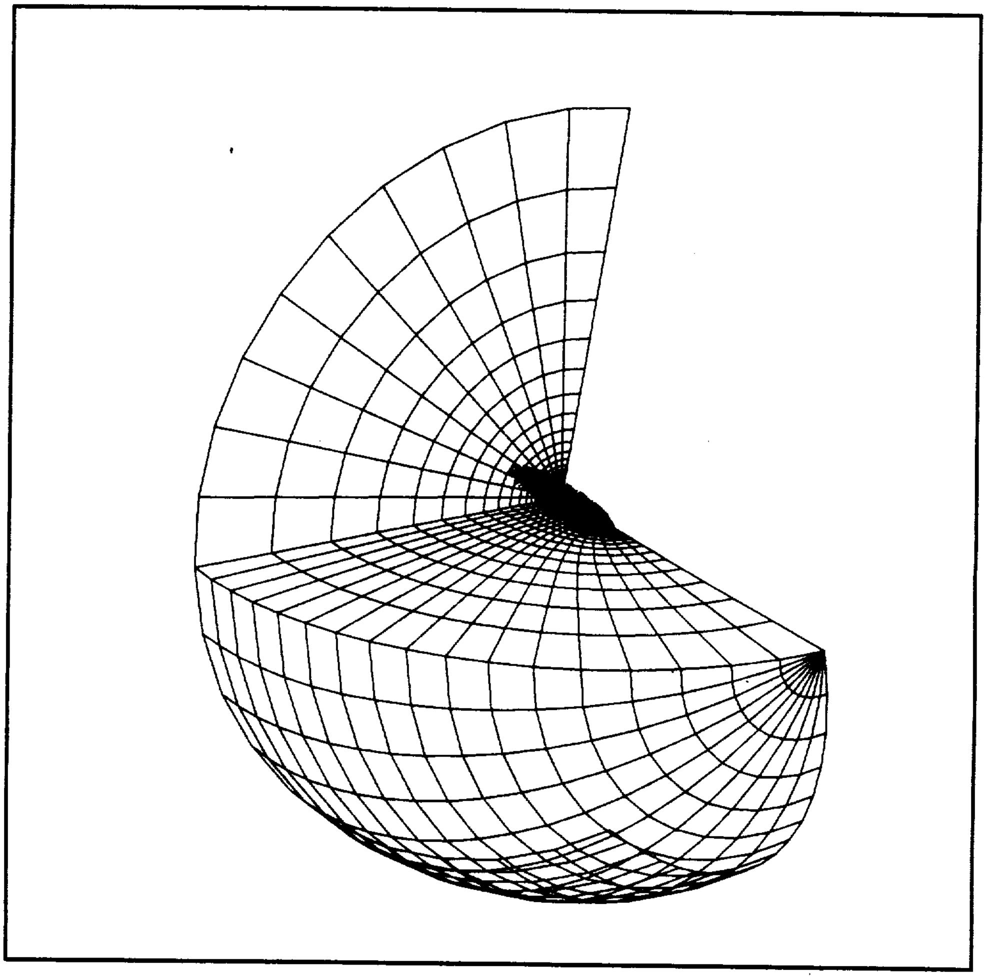


Figure 4.14: Geometry of the computational domain for an ellipsoid

the 0° angle of attack case, the Poisson solution in this case is not symmetric due to the non-zero angle of attack.

The third case for the ellipsoid geometry is intended to validate the vorticity-velocity algorithm with the experiment done by Kreplin, Vollmers and Meier (1982), and Meier and Cebeci (1985). The Reynolds number based on the free stream conditions and the major axis is $1.6 \times 10^6$, the Mach number, 0.4, and the angle of attack, 10°. Since the experimental study indicated that the flow was nominally laminar, the simulation was carried out for laminar flow. However, unlike the geometry used in the numerical simulation, the real experimental setup had a sting attached to the end of the ellipsoid. Vatsa, Thomas and Wedan (1987) computed a Navier-Stokes solution for this case using upwind-biased and central differencing for the convective and pressure terms respectively, with and without the sting for laminar and transitional flows. Their study showed that the sting mostly affected the flow near the juncture of the sting, and the flow features for the rest of the flowfield were nearly the same with and without the sting. Thus, the current calculation was carried out without the sting. From the comparison of the surface pressure coefficients in Figure 4.19, it can be observed that the Poisson solution generally predicts a lower pressure than was found in Vatsa, Thomas and Wedan (1987) for both the windward and the leeward sides. Nonetheless, the Poisson solution agrees well with their results for most of the leeward side and the first half of the windward side except near the downstream end of the ellipsoid where the flow separated from the surface as illustrated by the particle traces (Figure 4.20) for the thin-layer Navier-Stokes solution. The thin-layer Navier-Stokes solution, like the Poisson solution, predicts lower pressure than the experimental results and is in good agreement with the experiments on the first half

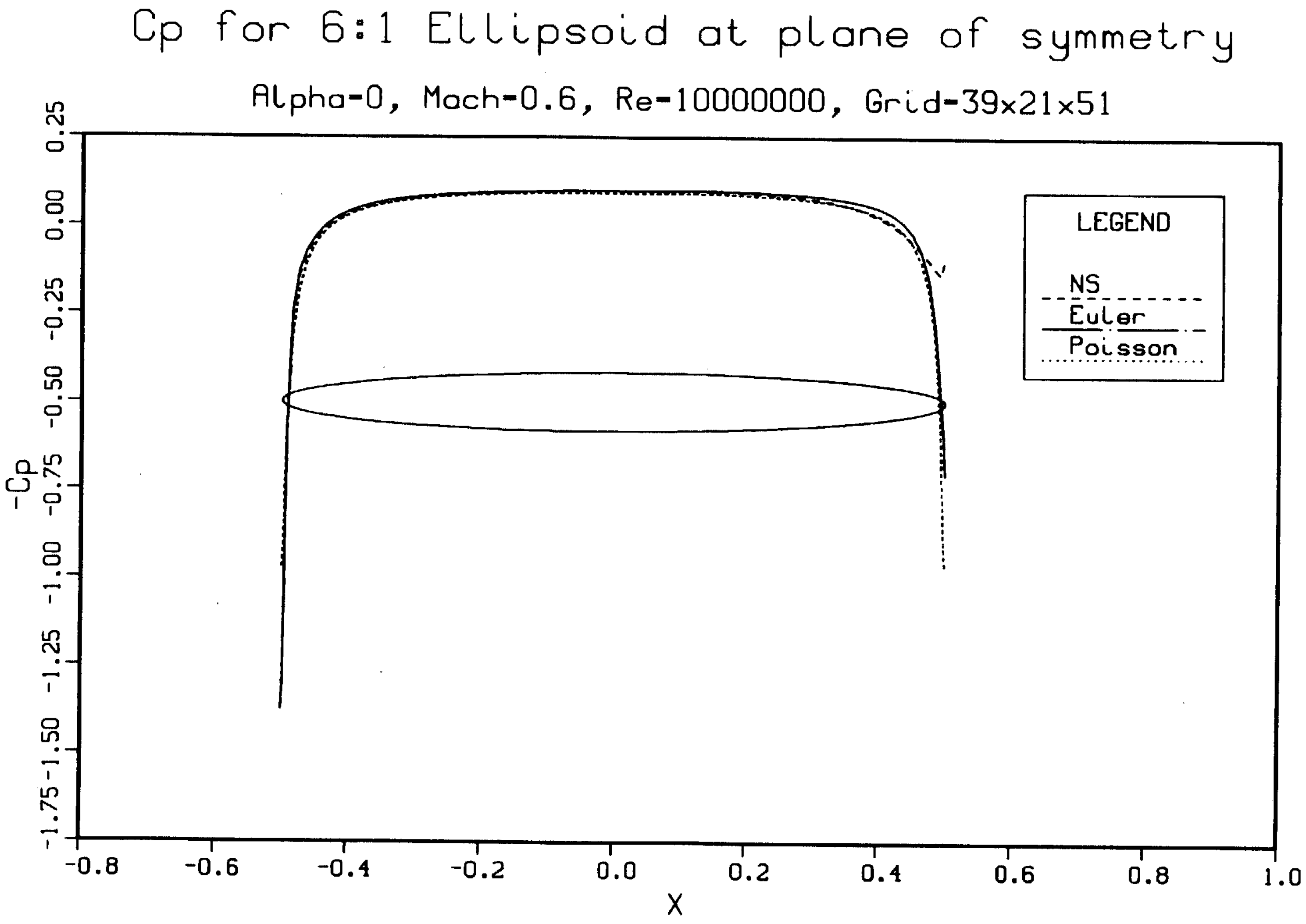


Figure 4.15: Comparison of $C_p$ from the Poisson, Euler and Navier-Stokes solutions at the plane of symmetry for an ellipsoid at $M_\infty = 0.6$, $Re = 10^7$, and $\alpha = 0°$

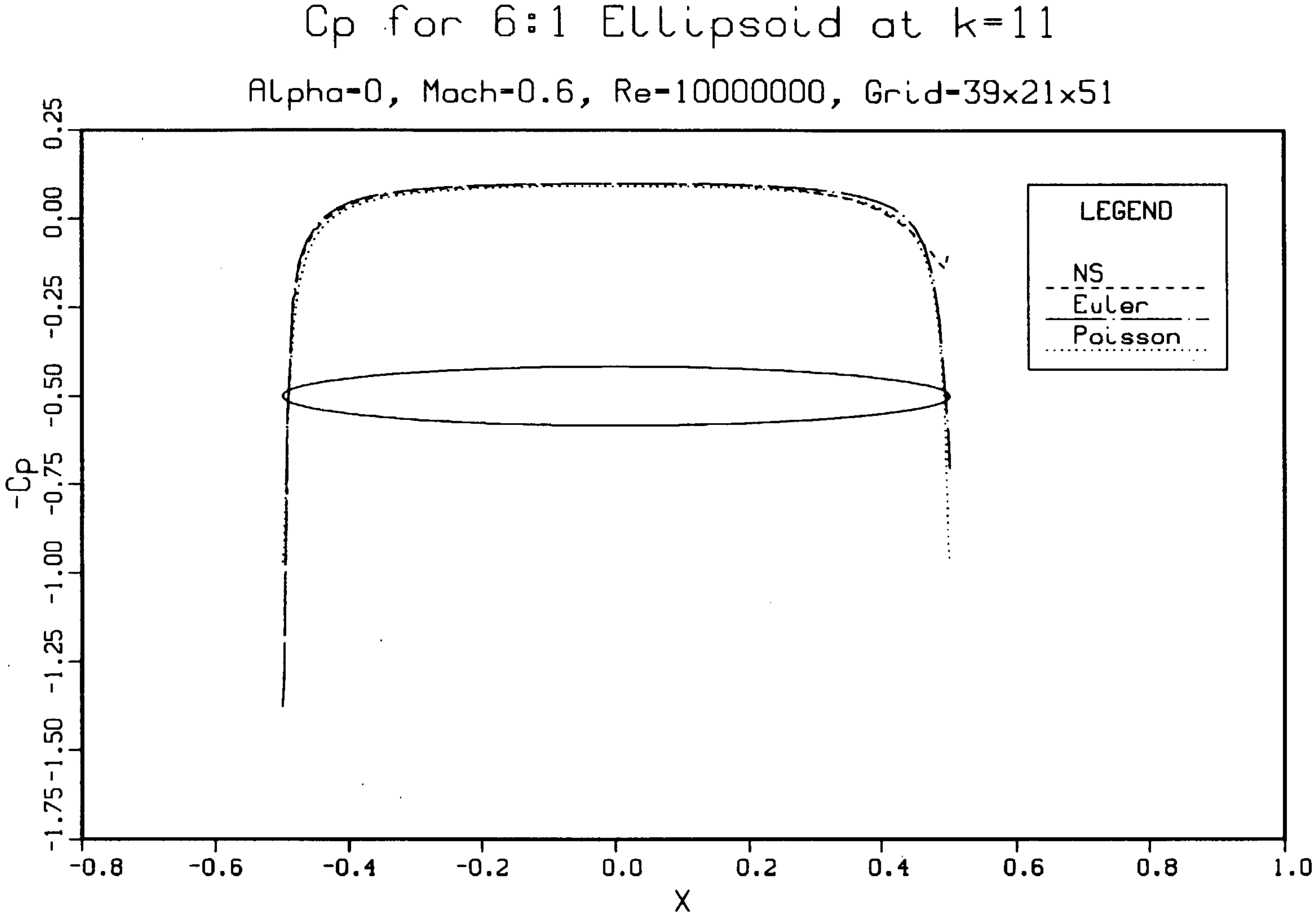


Figure 4.16: Comparison of $C_p$ from the Poisson, Euler and Navier-Stokes solutions at the $\phi = 90^\circ$ plane for an ellipsoid at $M_\infty = 0.6$, $Re = 10^7$, and $\alpha = 0^\circ$

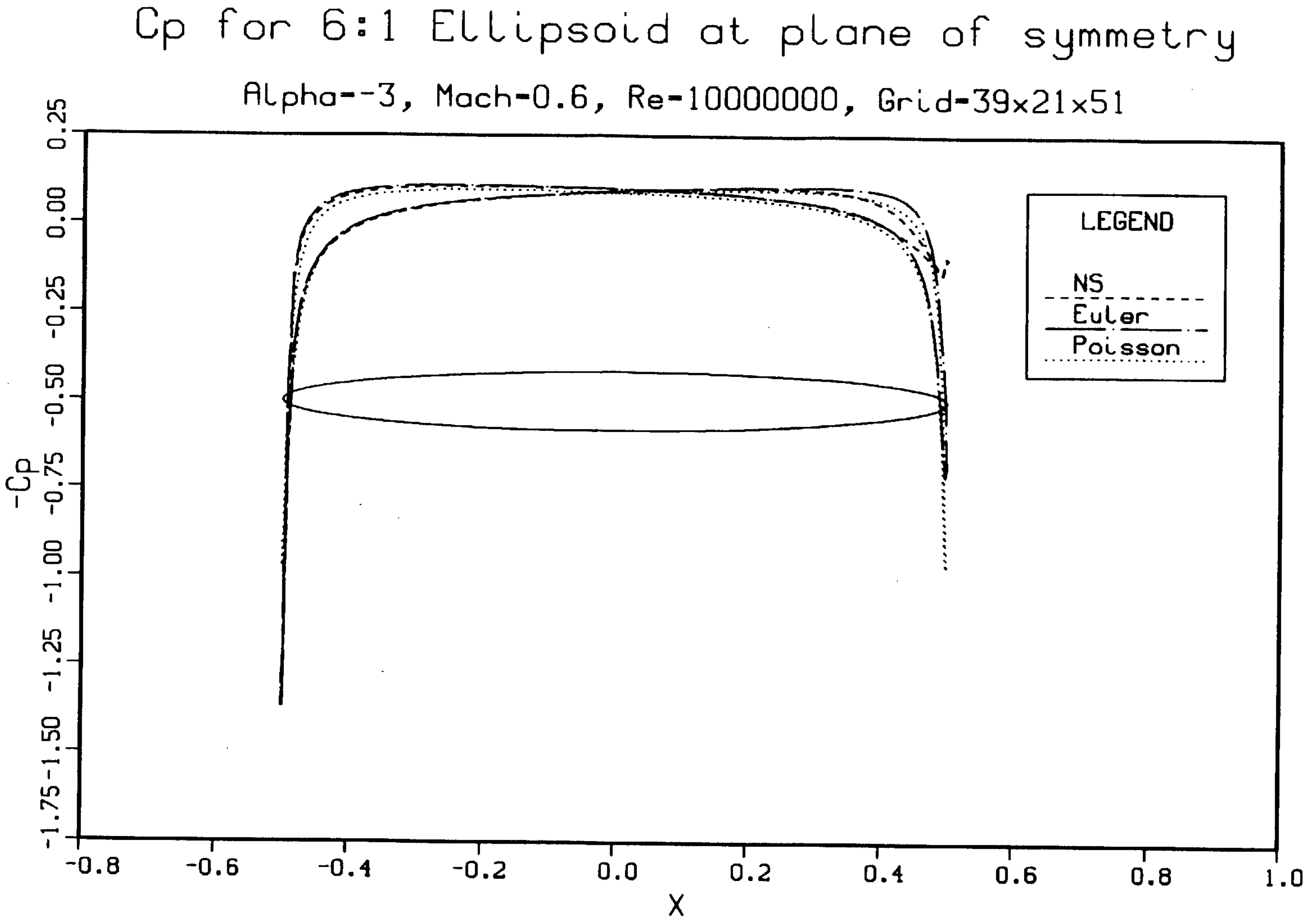


Figure 4.17: Comparison of $C_p$ from the Poisson, Euler and Navier-Stokes solutions at the plane of symmetry for an ellipsoid at $M_\infty = 0.6$, $Re = 10^7$, and $\alpha = -3°$

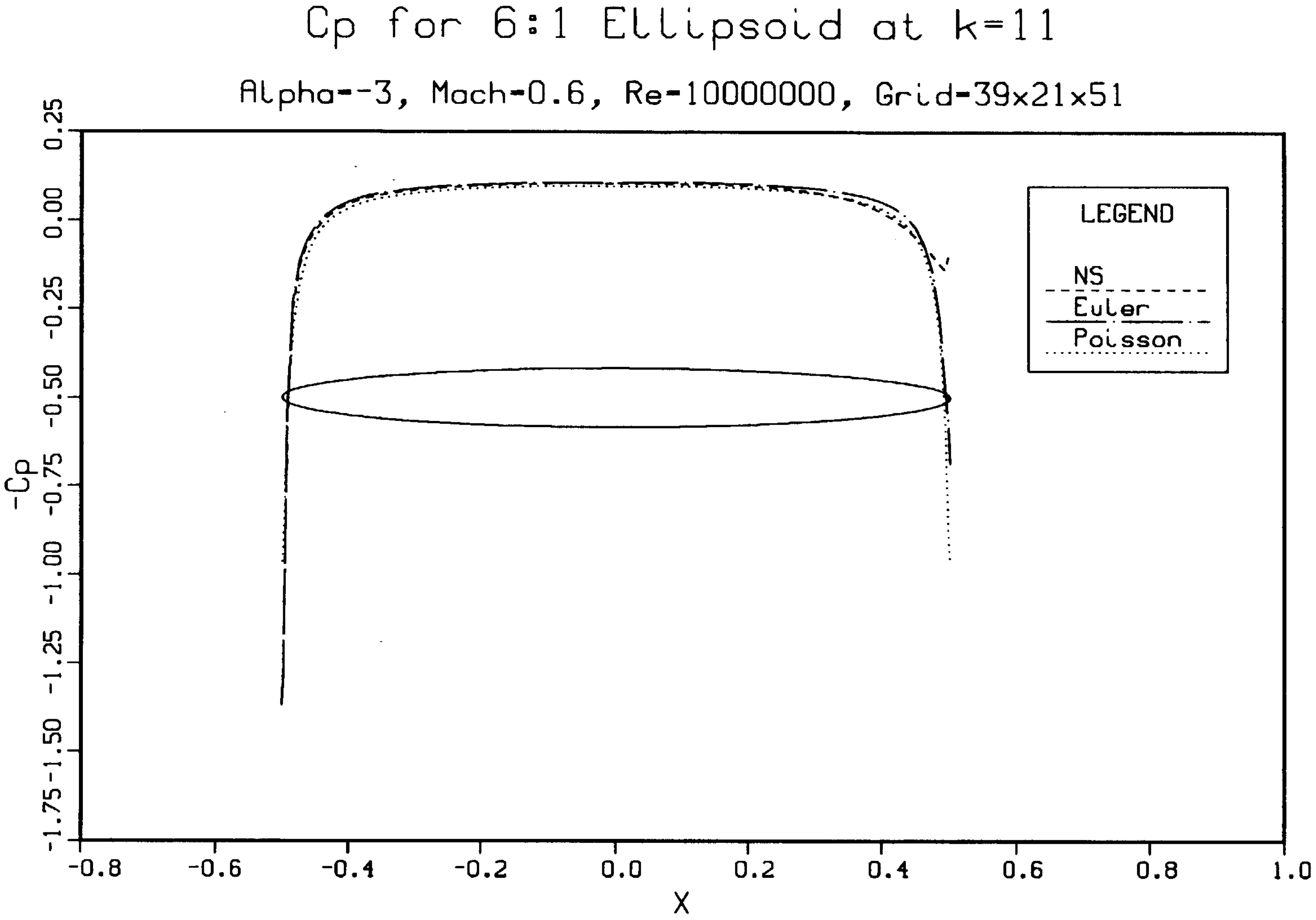


Figure 4.18: Comparison of $C_p$ from the Poisson, Euler and Navier-Stokes solutions at the $\phi = 90^\circ$ plane for an ellipsoid at $M_\infty = 0.6$, $Re = 10^7$, and $\alpha = -3^\circ$

of the windward side and disagrees at the downstream end where the computational model differs from the experimental setup. However, the flow characteristics for the thin-layer Navier-Stokes solution at the downstream end is similar to that of Vatsa, Thomas and Wedan (1987) where the leeward pressure was higher than the windward pressure. Overall, the thin-layer Navier-Stokes solution is quantitatively similar (though not perfect) to the experimental results and the level of agreement can be improved by increasing number of grid points and reducing the grid spacing in the boundary layer to resolve the gradients. In Vatsa, Thomas and Wedan (1987), the grids used were 4.2 times ($73 \times 49 \times 49$) for the case with the sting and 11 times ($97 \times 97 \times 49$) for the sting-less case as many points as used in the current study. Since the primary purpose of the numerical calculation for this case was to demonstrate the capability of the vorticity-velocity algorithm, the thin-layer Navier-Stokes solution was not pursued further to obtain better agreement with the experimental results.

One of the objectives of the current research was to find ways to improve the efficiency of the numerical methods for flow simulations. The proposed method is to first obtain the flow solution from a set of simplified equations (here, the vorticity-velocity formulation is used with vorticity being set to zero, i.e., the Poisson equations) and then either directly feed that solution into the Navier-Stokes solver or "combine" the Poisson solution with the Navier-Stokes equations to obtain an improved solution and then solve the Navier-Stokes equations for the entire flowfield. To combine the Poisson solution with the Navier-Stokes equations means that the Navier-Stokes (or the thin-layer Navier-Stokes) equations are solved for a user specified region which roughly encompasses the boundary layer, using the Poisson solution as the edge

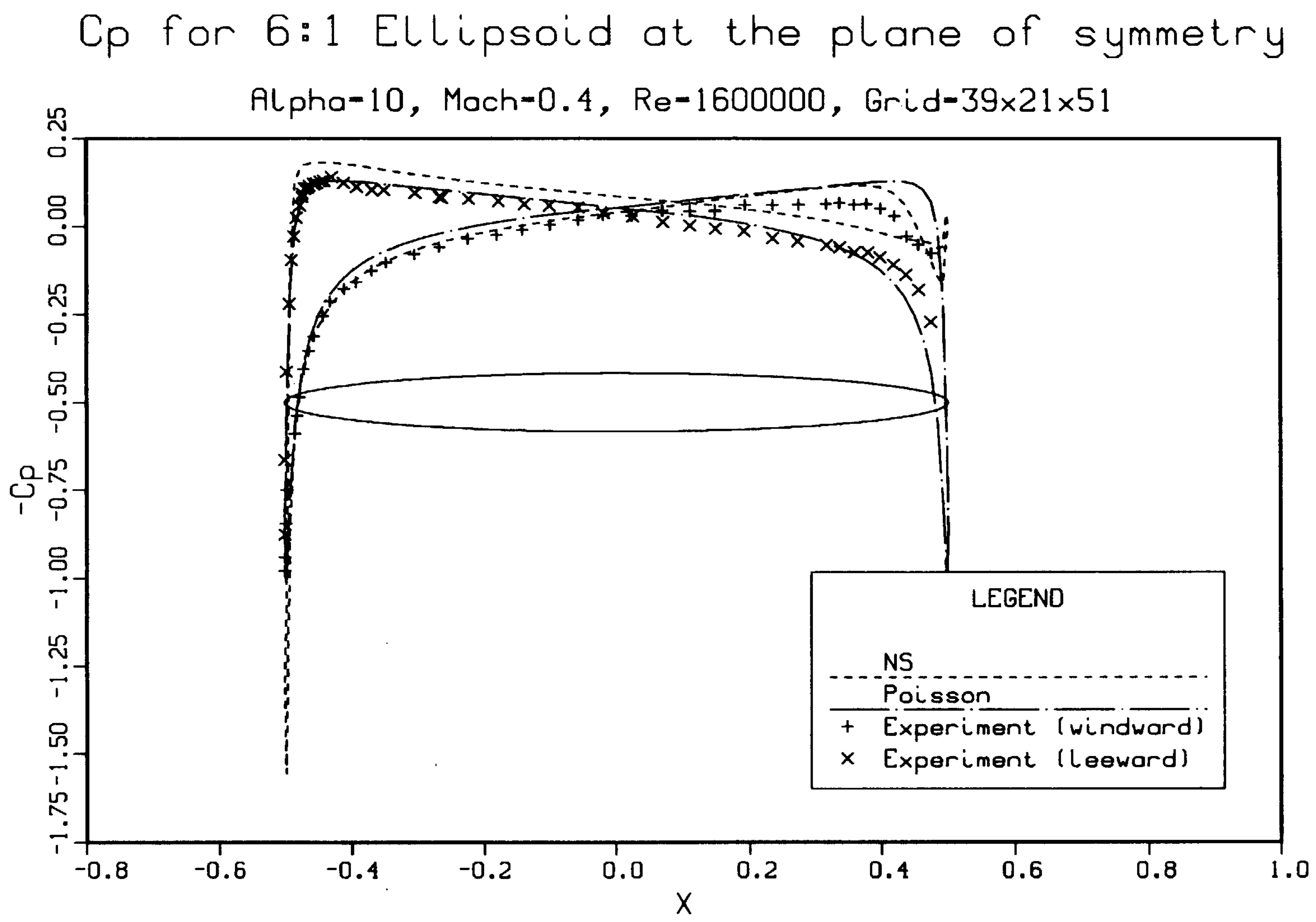
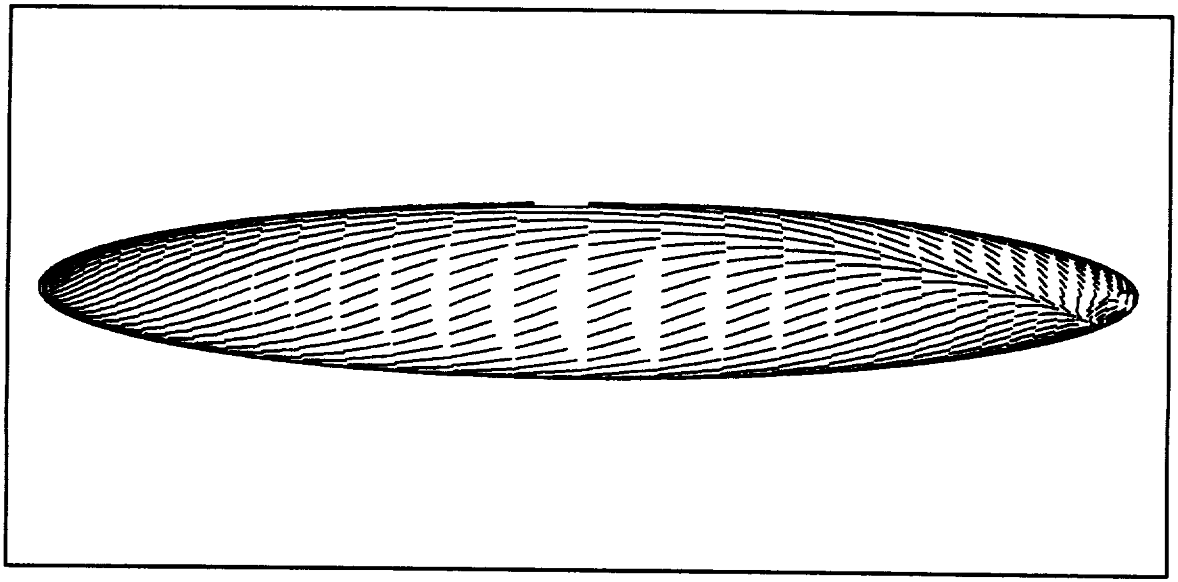
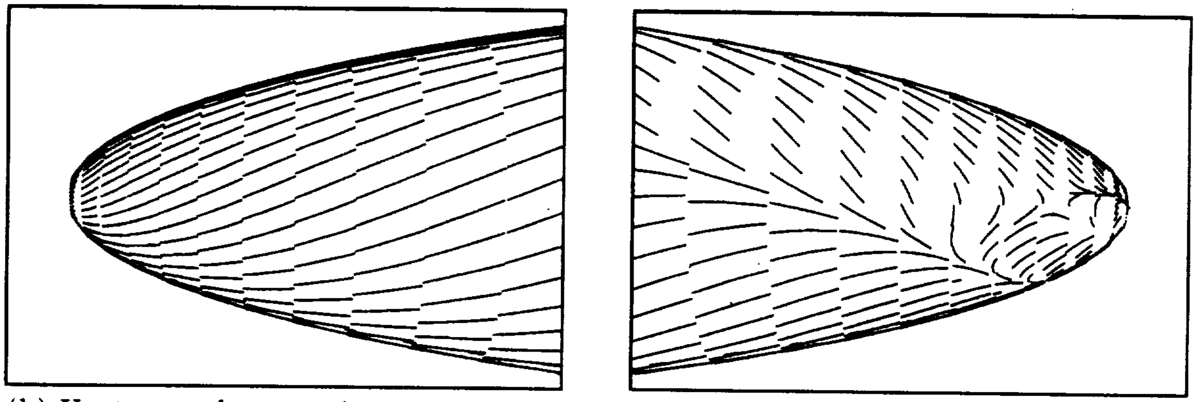


Figure 4.19: Comparison of $C_p$ from the experimental results, the Poisson and Navier-Stokes solutions at the plane of symmetry for an ellipsoid at $M_\infty = 0.4$, $Re = 1.6 \times 10^6$, and $\alpha = 10^\circ$



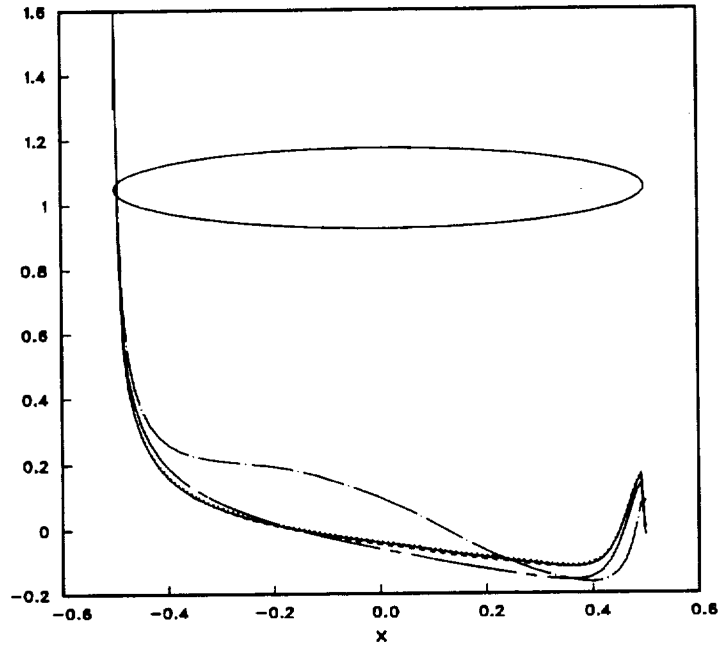
(a) Full view



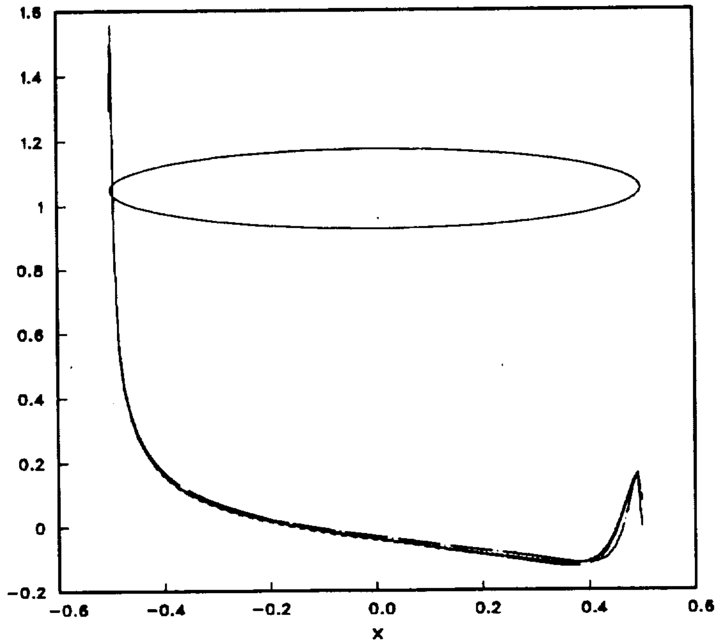
(b) Upstream close-up view

(c) Downstream close-up view

Figure 4.20: Particle traces from the thin-layer Navier-Stokes solutions for an ellipsoid at $M_\infty = 0.4$, $Re = 1.6 \times 10^6$, and $\alpha = 10^\circ$

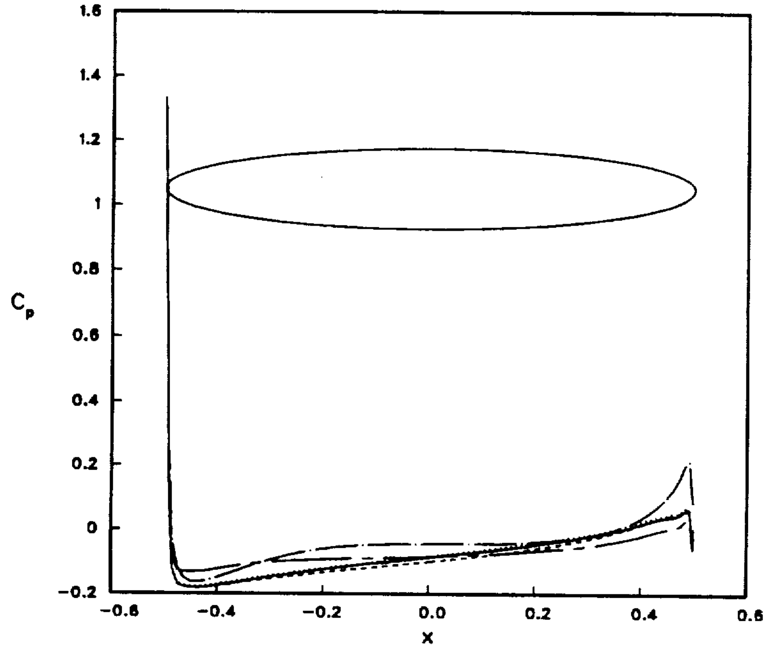


(a) Windward, thin-layer Navier-Stokes solutions.

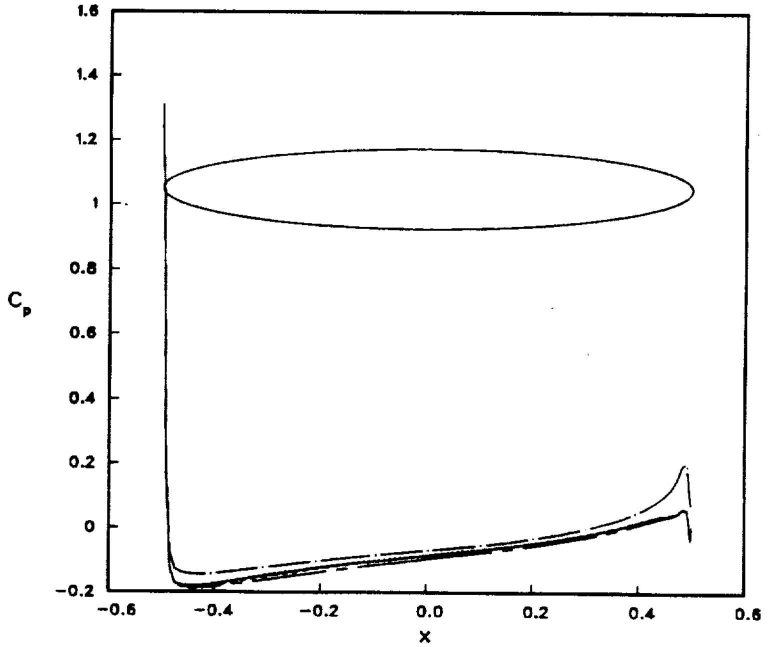


(b) Windward, combined Poisson and thin-layer Navier-Stokes solutions.

Figure 4.21: Convergence history for an ellipsoid at $M_\infty = 0.4$, $Re = 1.6 \times 10^6$, and $\alpha = 10^\circ$ at iteration 190 (—·—), 690 (— – —), 1190 (– –), 1690 (···), 2190 (—)

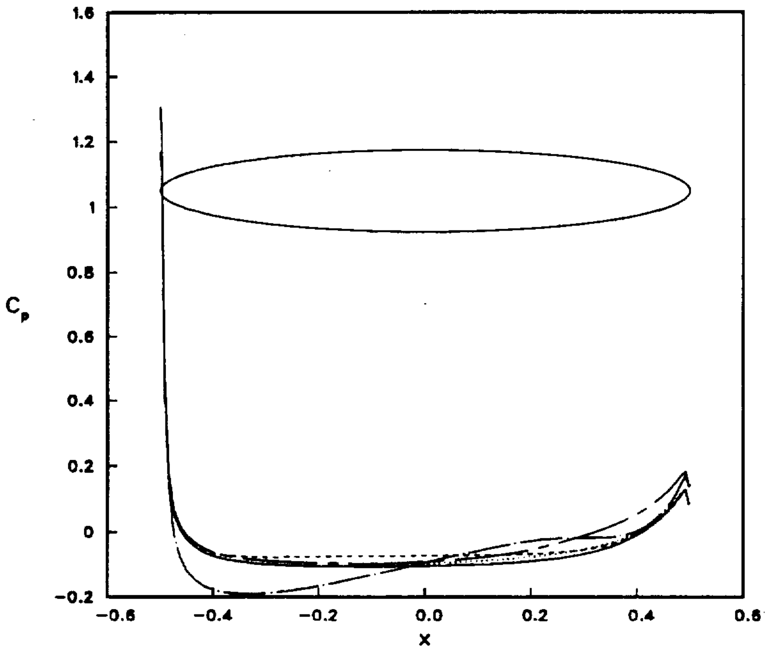


(c) Leeward, thin-layer Navier-Stokes solutions.

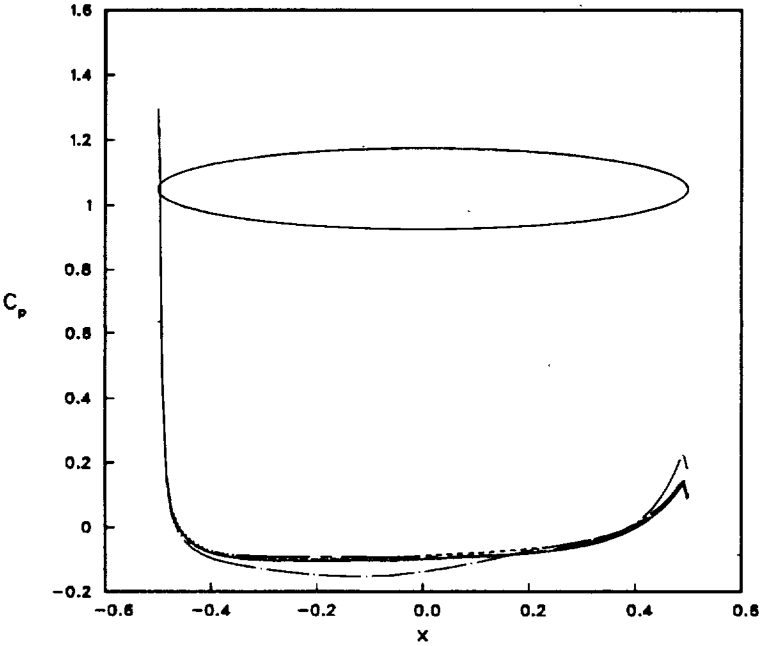


(d) Leeward, combined Poisson and thin-layer Navier-Stokes solutions.

Figure 4.21 (Continued)

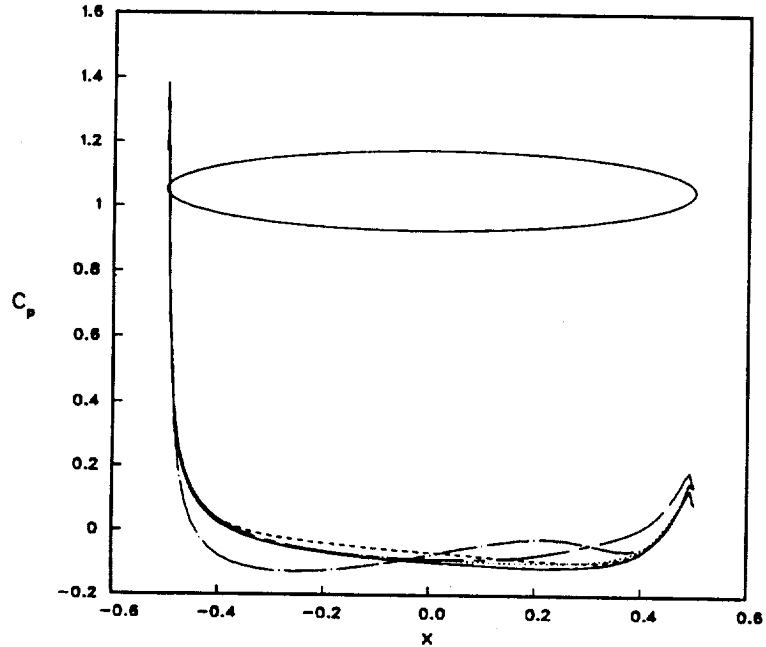


(a) Windward, thin-layer Navier-Stokes solutions.

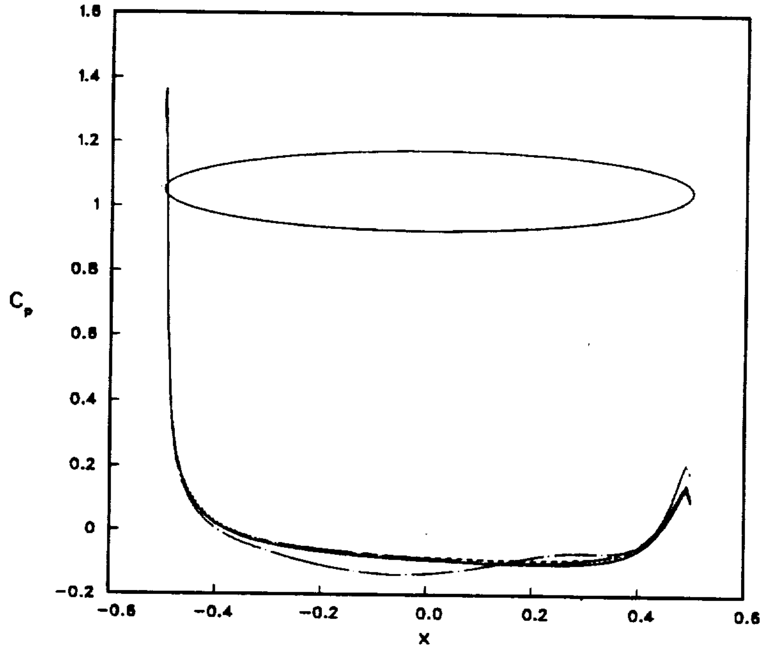


(b) Windward, Poisson and thin-layer Navier-Stokes solutions.

Figure 4.22: Convergence history for an ellipsoid at $M_\infty = 0.6$, $Re = 10^7$, and $\alpha = -3^\circ$ at iteration 190 (—·—), 590 (— - —), 990 (- -), 1390 (···), 1790 (—)



(c) Leeward, thin-layer Navier-Stokes solutions.



(d) Leeward, Poisson and thin-layer Navier-Stokes solutions.

Figure 4.22 (Continued)

conditions. This step need not be carried out until convergence since it is only an intermediate step in obtaining an even better estimate than provided by the Poisson solutions alone. However, from the cases tested, it was found that solving the Navier-Stokes equations for half (or more) of the grid points, the rate of convergence (based on the residual) was an order of magnitude faster than solving the Navier-Stokes equations for the entire flowfield. This clearly shows that the deterioration of convergence is faster than the linear increase of number of grid points used and the larger the grid, the more CPU time will likely be saved by improving the initial guess.

In Figure 4.21, the convergence history of the pressure coefficient is shown for the thin-layer Navier-Stokes solutions using a uniform flow field as the initial guess and also using the combined Poisson and thin-layer Navier-Stokes solutions as the initial guess before solving the entire flowfield using the thin-layer Navier-Stokes equations for the $M_\infty = 0.4$, $Re = 1.6\times10^6$ and $\alpha = 10^\circ$ testcase. Use of the better initial guess appears to significantly reduce the oscillations in the solution process compared to the thin-layer Navier-Stokes results obtained from using a uniform flowfield as the initial guess. Note that the uniform flowfield referred to here was actually implemented with a “slow-start” mechanism which gradually reduced the velocities on the surface from the free stream values to zero to avoid the possible instability resulting from suddenly setting the wall velocities to zero. From here on, the uniform flowfield referred to for the initial guess implies that the “slow start” is used. For both the windward and the leeward sides, the better initial guess required 1073 iterations to converge while the one using the uniform flowfield as the initial guess needed 1790 iterations to achieve the same level of convergence. Adding the CPU time required to obtain the better initial guess, which is equivalent to 145 iterations of the thin-layer Navier-Stokes solutions, the total savings of CPU time is about 32% (or 572 iterations). For the other case, $M_\infty = 0.6$, $Re = 10^7$ and $\alpha = -3^\circ$, the Poisson solution (without using the combined scheme) was used as the initial guess to obtain the thin-layer Navier-Stokes solution and is compared with the thin-layer Navier-Stokes solution computed initially from a uniform flowfield in Figure 4.22. Overall, when the Poisson solution was used as the initial guess, smaller oscillations in the solution process were observed than when the a uniform flowfield was used as the initial guess. With the better initial guess, a converged solution was obtained in 1390 iterations as shown in Figure 4.22 while 2290 iterations was required when the uniform flowfield was used as the initial guess. Adding the CPU time required to obtain the Poisson solution, which is equivalent to 82 iterations of the thin-layer Navier-Stokes solutions, the savings of CPU time is 36%. All the cases presented here were computed on a Cray YMP computer, and the CPU time required for each point per iteration is 43.8 $\mu s$ using the thin-layer Navier-Stokes equations and 9.98 $\mu s$ for the Poisson equations.

### 4.2.3 External Tank (ET)

Since the external tank is one of the major components of the integrated space shuttle vehicle, and its grid was used as the major grid to cover the entire computational domain in the shuttle flow simulation, it was used as yet another testcase before delving into the shuttle flow simulation using the vorticity-velocity algorithm. Two different angles of attack, 0° and $-3^\circ$, were calculated for a flow of $M_\infty = 0.2$, and $Re = 4.0 \times 10^6/ft$. Both the thin-layer Navier-Stokes and Poisson solutions were carried out and compared. As shown in Figure 4.23, the computational domain of the external tank grid starts from three times the body length upstream to 2.7 times the body length downstream of the ET and the far-field boundary is at least three times the body length from the ET center of gravity. Also noticed in Figure 4.23 is a sting approximately the diameter of the ET extending from the back end of the ET to the downstream boundary. The sting was used to circumvent the inability of the Poisson equation to model the separated flow. The grid used 63, 39, and 51 points in the $\xi$ (streamwise), $\eta$ (circumferential), and $\zeta$ (outward-going) directions, respectively, and like the rest of the grids used in this study, most of the points were clustered near the body surface. The initial spacing away from the body surface was $5.936 \times 10^{-5}$ times the ET length ($0.13 ft$). The grid points clustered near the rear end of the external tank in the $\xi$ direction were intended to resolve the flow around the aft attach hardware and were not actually needed here. However, they were used for convenience. The wall boundary conditions used were no-slip and tangency/vorticity consistency conditions for the thin-layer Navier-Stokes and the Poisson solutions, respectively. The axis boundary condition was used for the axis extending upstream from the ET nose to the far-field boundary. Free stream values of flow variables were used at the far-field boundary, and a zero-gradient outflow condition was used at the downstream boundary. A symmetry plane boundary condition was employed in the $\eta$ direction to save CPU time.

For the $\alpha = 0^\circ$ case, the surface pressure coefficient comparisons are presented in Figure 4.24. The thin-layer Navier-Stokes solutions were computed with a uniform flowfield as the initial guess. Since this flow is axisymmetric due to zero angle of attack, the pressure at the top and bottom of the symmetry plane are the same, as shown in the figure. The thin-layer Navier-Stokes solutions show a sharper expansion

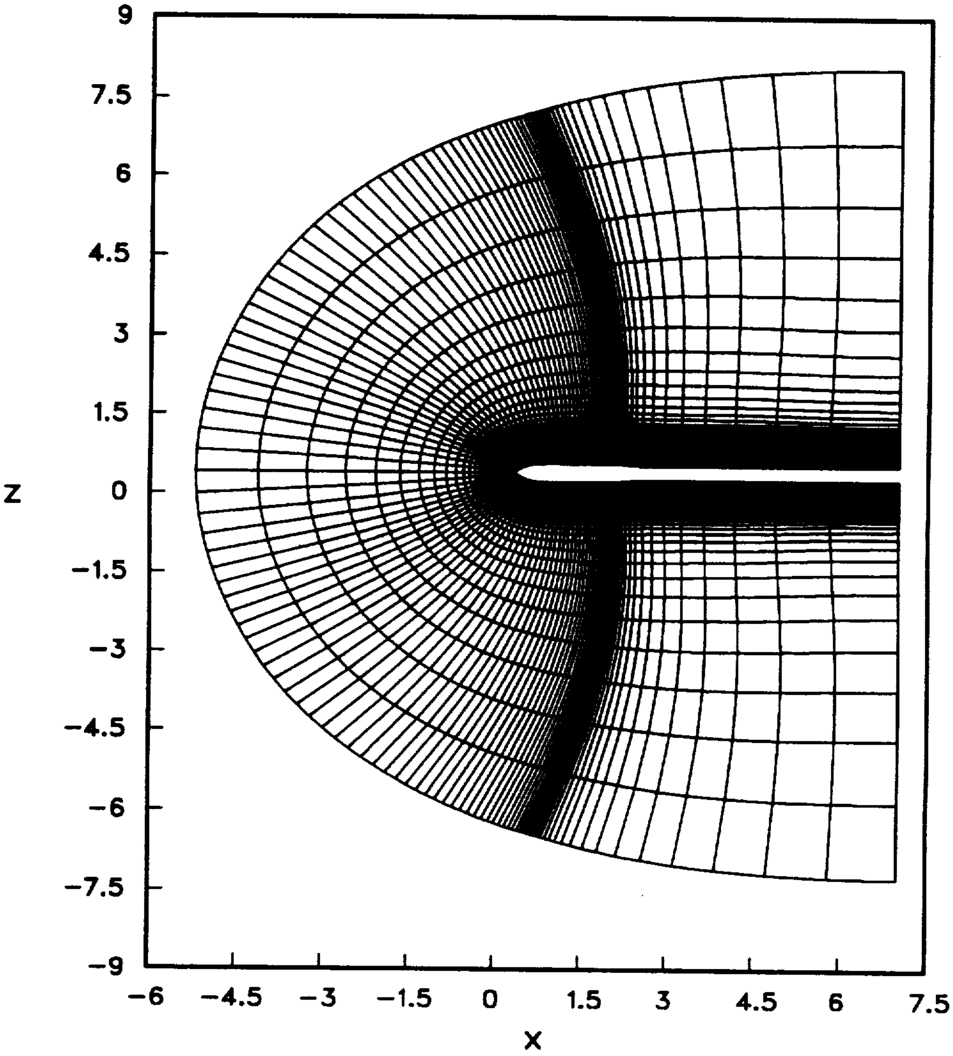


Figure 4.23: Geometry of the computational domain for the external tank (x in $10^3$ inches)

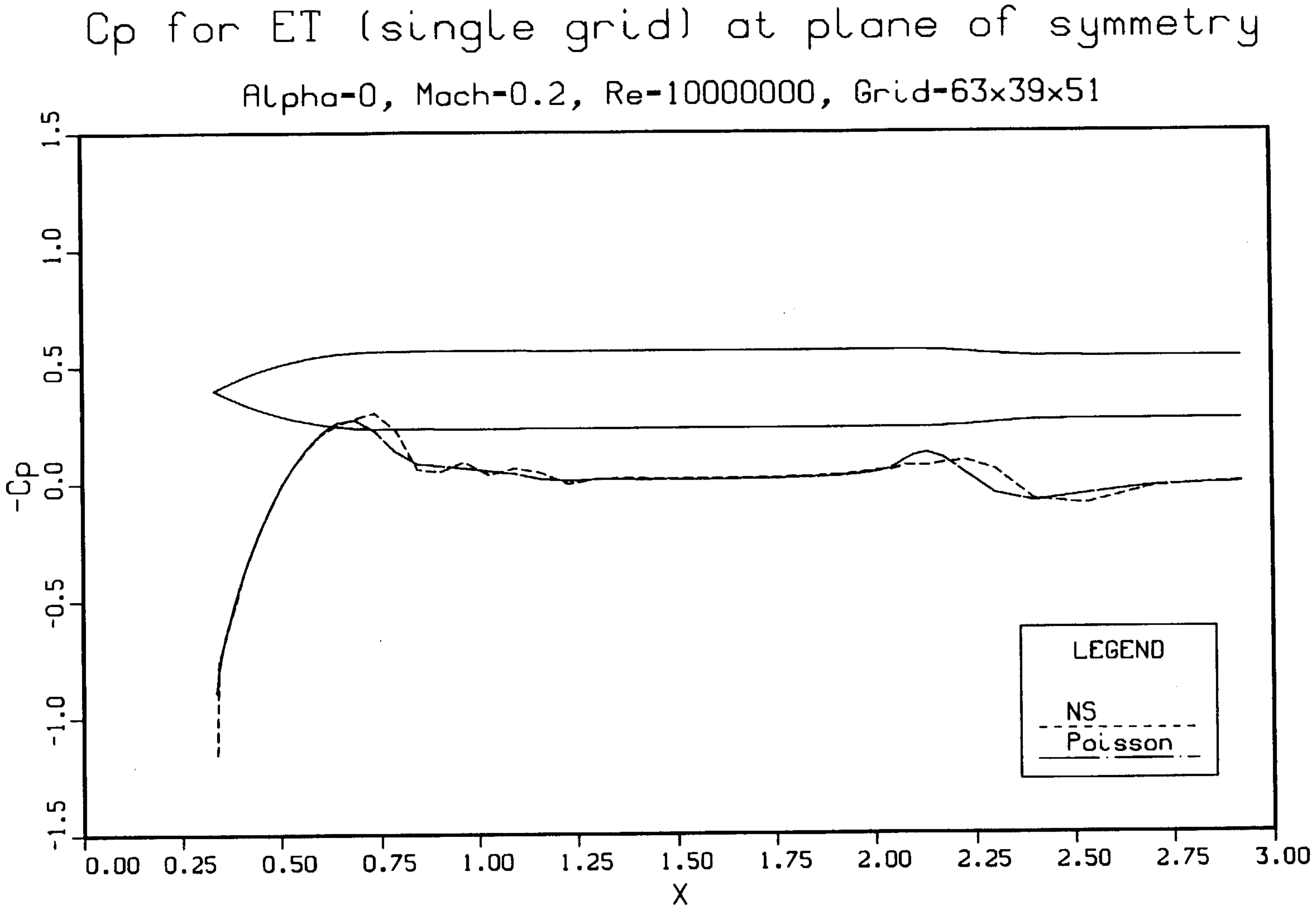


Figure 4.24: Comparison of surface pressure coefficients for the external tank, $M_\infty = 0.2$, $Re = 4.0 \times 10^6/ft$, and $\alpha = 0^\circ$ (x in $10^3$ inches)

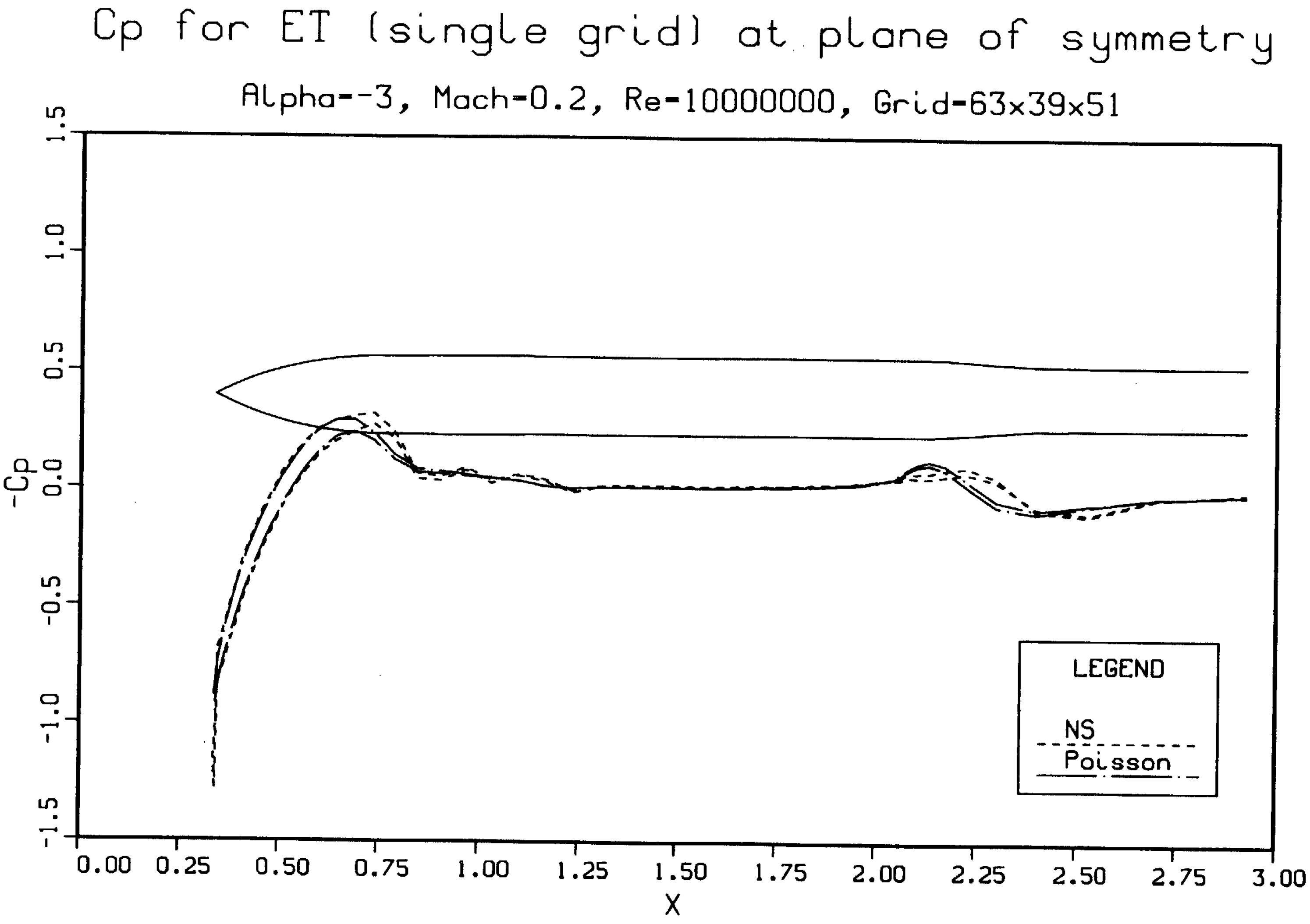
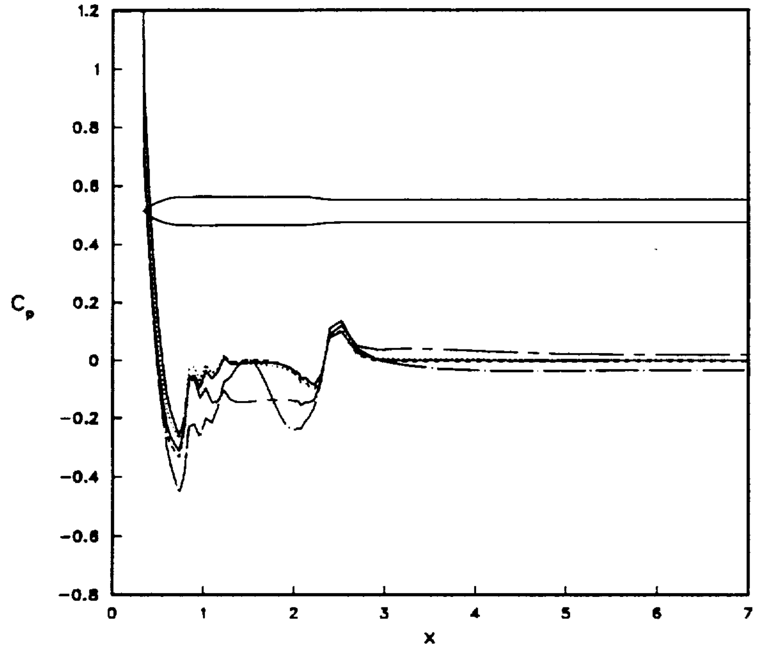
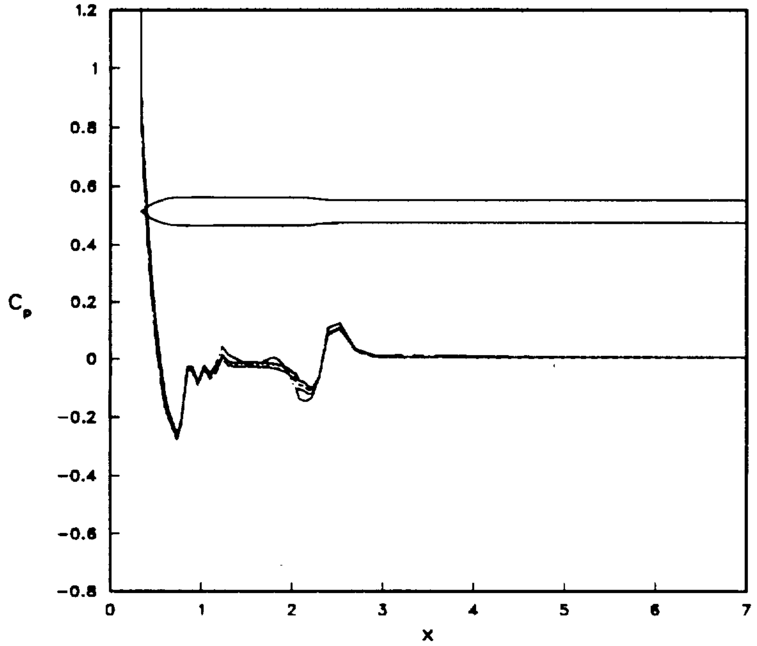


Figure 4.25: Comparison of surface pressure coefficients for the external tank, $M_\infty = 0.2$, $Re = 4.0 \times 10^6/ft$, and $\alpha = -3^\circ$ (x in $10^3$ inches)

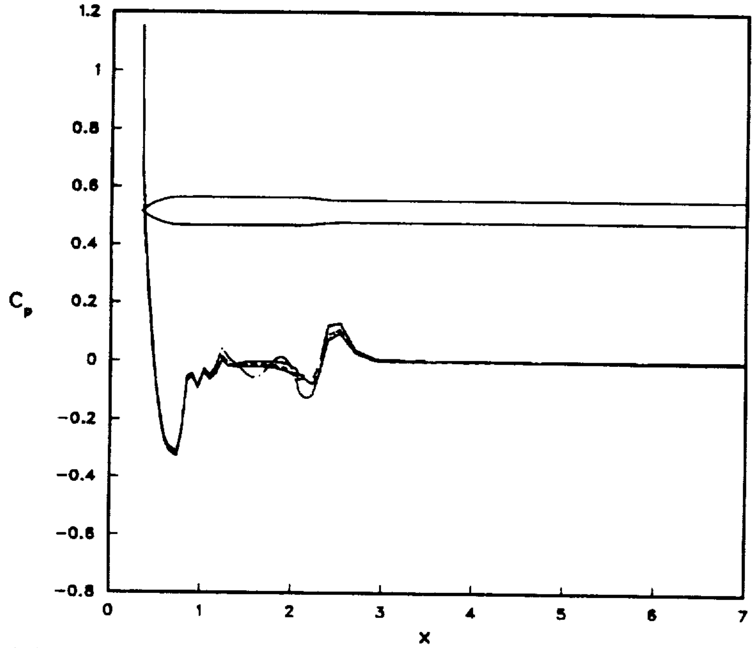


(a) Windward, thin-layer Navier-Stokes solutions, $\alpha = 0^{\circ}$



(b) Windward, combined Poisson and thin-layer Navier-Stokes solutions, $\alpha = -3^{\circ}$

Figure 4.26: Convergence history for the external tank at $M_{\infty} = 0.2$, and $Re = 4.0 \times 10^{6}/ft$ at iteration 190 (—·—), 590 (— - —), 1190 (- -), 1790 (···), 2390 (—), (x in $10^3$ inches)



(c) Leeward, combined Poisson and thin-layer Navier-Stokes solutions, $\alpha = -3^{\circ}$

Figure 4.26 (Continued)

than the Poisson solutions at the front portion of the external tank followed by a wiggle in the solution which is likely due to the lack of grid points to resolve the flow. Then both solutions return back to free stream pressure for the rest of the external tank. At the juncture of the external tank and the sting, the pressure from the Poisson solution shows a sharper variation than that of the thin-layer Navier-Stokes solutions due to lack of viscous terms to smooth out the solutions. Then, both solutions gradually return to the free stream pressure for the rest of the sting. In Figure 4.25, the same trend — sharper expansion for the thin-layer Navier-Stokes solutions in the front portion of the ET and the sharper pressure variation for the Poisson solutions at the juncture of the ET and the sting — can be found for the $\alpha = -3^\circ$ case. However, the top and bottom sides of the symmetry plane have different pressures due to non-zero angle of attack. Generally speaking, for both cases presented here, the Poisson solutions agree with the thin-layer Navier-Stokes solutions for most regions of the external tank and can be considered as a good initial guess for the thin-layer Navier-Stokes solutions if such a solution is desired.

The convergence history for the external tank is shown in Figure 4.26. In Figure 4.26a, the convergence history of the thin-layer Navier-Stokes solution computed from a uniform flow field with a slow start shows large oscillations during the solution process and is not converged yet after 2390 iterations while for the case using the combined Poisson and the thin-layer Navier-Stokes solutions as the initial guess, shown in Figures 4.26b and 4.26c, the level of oscillation is significantly smaller and the solution converged in 1790 iterations.

### 4.2.4 Integrated Space Shuttle Vehicle (ISSV)

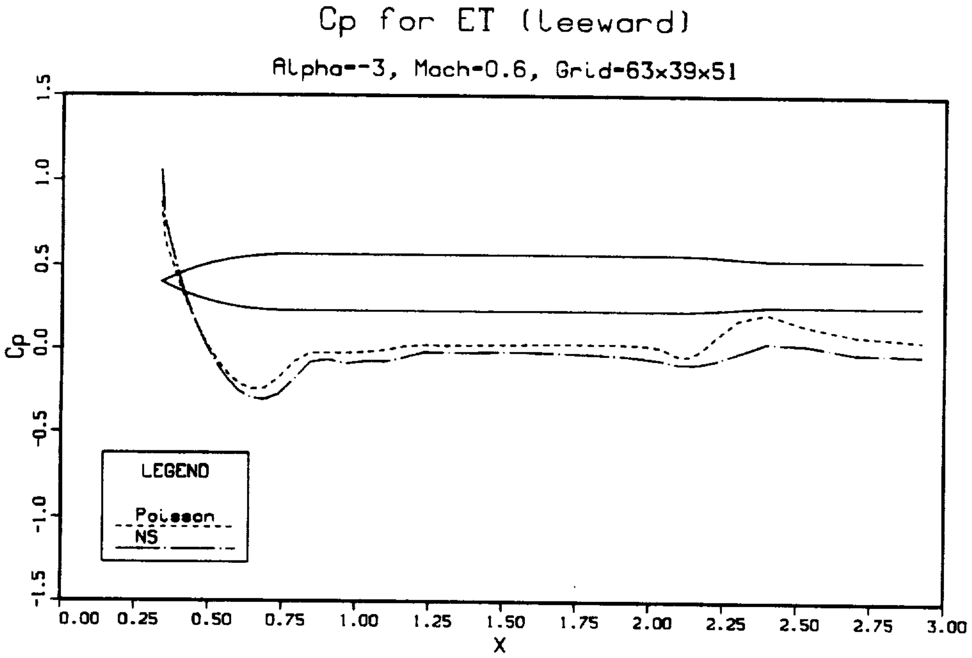
The primary objective of developing the vorticity-velocity algorithm was to be able to economically simulate the flow around the integrated space shuttle vehicle. Since the algorithm essentially involves solving the Poisson equations, it is faster than solving the Navier-Stokes or the thin-layer Navier-Stokes equations, and thus can be used to evaluate the overset grid used in the chimera scheme with only modest use of computing resources. The resulting solution can also be used as an initial guess for more accurate algorithms which solve the Navier-Stokes or the thin-layer Navier-Stokes equations.

Due to limited computing resources, only the accuracy of the vorticity-velocity algorithm is evaluated in the chimera approach. The convergence acceleration using the Poisson solution as the initial guess was not carried out; however, it is believed that the rate of convergence will be improved using the Poisson solution as the initial guess as was found for the ellipsoid and external tank geometries. Like the grids used earlier for the thin-layer Navier-Stokes flow simulation of the integrated space shuttle vehicle, the ET grid was made the major grid which extended all the way out to the far-field boundary while the SRB and ORB grids occupied a much smaller portion of the computational domain. The computational domain started roughly 3 times the ET length upstream to 2.7 times the ET length downstream of the ET with a grid of 383,001 points spread through the three component grids. All three grids were generated in the same manner as described in the section on the thin-layer Navier-Stokes solutions in this chapter. Since the purpose of computing the integrated space shuttle vehicle here was to verify the vorticity-velocity algorithm in the chimera approach, the grids used were not intended to reflect the true space shuttle geometry.

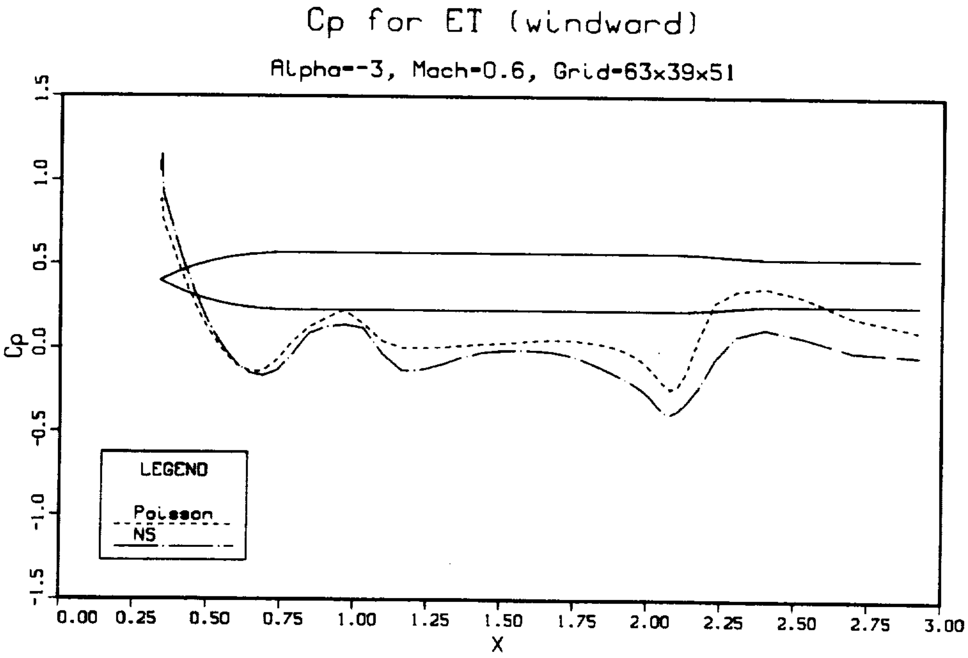
For example, the ET/ORB attach hardware was not modeled in the geometry and all three component grids, ET, SRB and ORB, had stings (including the orbiter wing sting) extending from the back of each component to the downstream boundary.

As with the thin-layer Navier-Stokes equations, uniform free stream flow was assumed initially for the whole computational domain in the Poisson solution procedure. The solution process started first by solving flows on the ET grid with the proper boundary conditions. The flow variables on the hole boundary were still at free stream values at this stage. After the ET solution was obtained, the variables on the outer and hole boundaries of the SRB and ORB grids were then updated by the ET solution. The SRB grid was then solved with the newly updated outer and hole boundary values as well as other necessary boundary conditions. Then this SRB solution was used to update the flow variables on the hole boundaries of the ET and ORB grids and the outer boundary of the ORB grid. Likewise, the solution on the ORB grid was obtained and used to update the other two grids. This process repeated until convergence was reached. Although it is not mentioned above, only one grid was solved at a time and the solution was put back to external disk or SSD before another grid was brought into the computer core memory and solved. Unlike the thin-layer Navier-Stokes solution procedure, the slow start was not required for the Poisson solution procedure since the wall velocities did not suffer from an abrupt change of values as in viscous flows.

Both the thin-layer Navier-Stokes and Poisson solutions were carried out and compared. Free stream values were assumed on the ET far-field boundary. The downstream boundaries of all three grids used a zero-gradient boundary condition which forced the flow variables at the last two downstream stations to be the same.

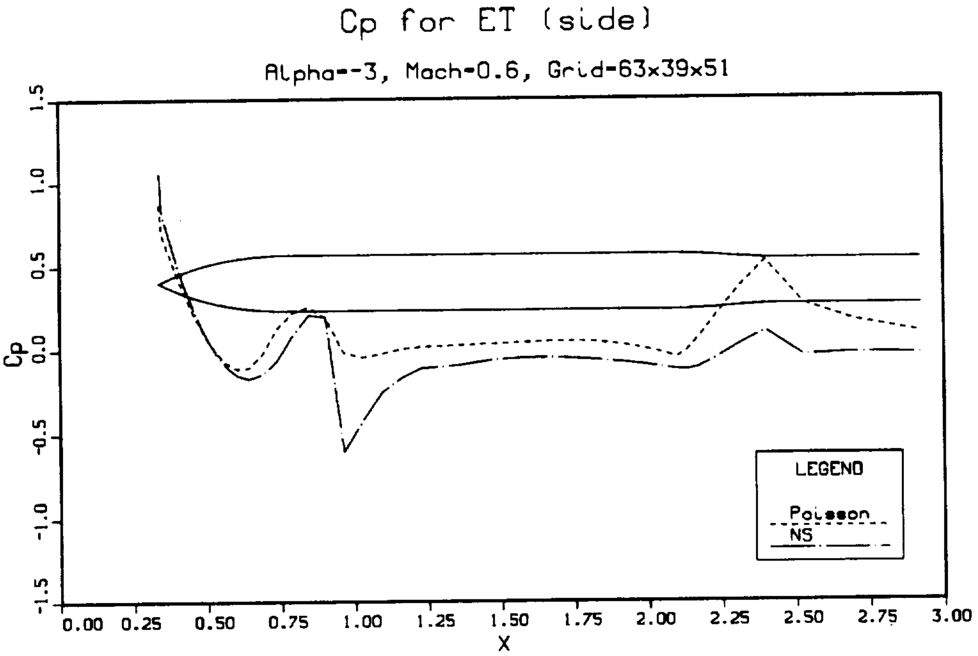


(a) $\phi = 0^{\circ}$ (bottom, leeward)



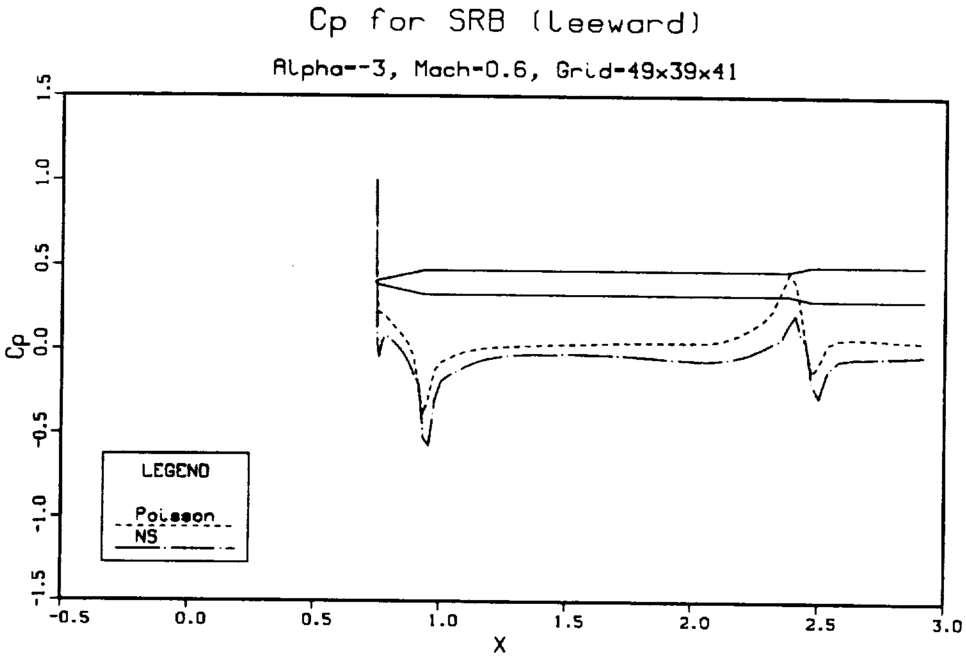
(b) $\phi = 180^{\circ}$ (top, windward)

Figure 4.27: Comparison of surface pressure coefficients from the Poisson and thin-layer Navier-Stokes solutions for the ET in ISSV configuration (x in $10^3$ inches)

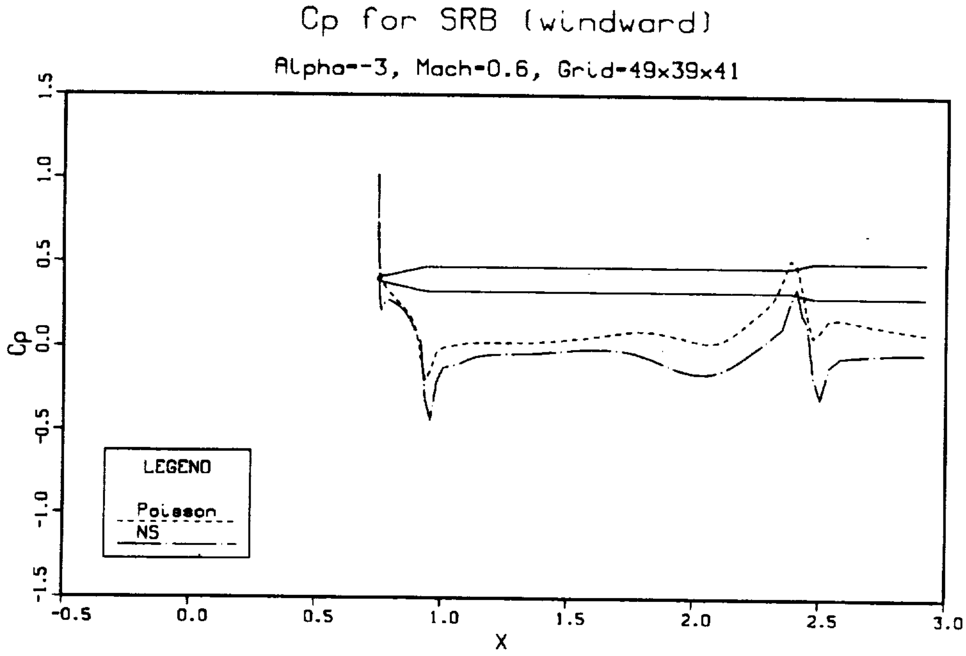


(c) $\phi = 90^\circ$ (side)

Figure 4.27 (Continued)

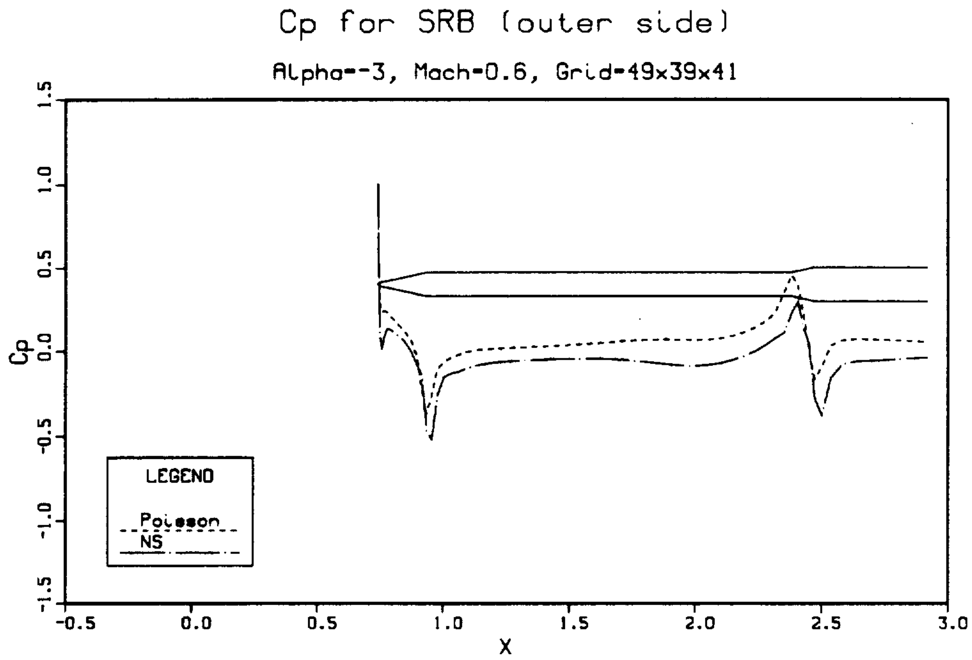


(a) $\phi = 0^{\circ}$ (bottom, leeward)

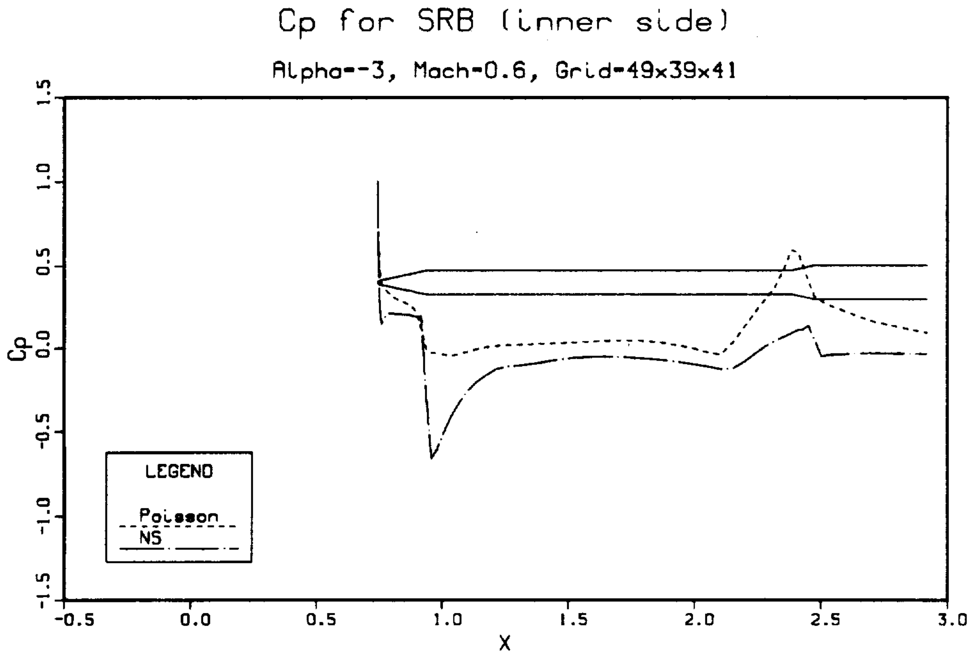


(b) $\phi = 180^{\circ}$ (top, windward)

Figure 4.28: Comparison of surface pressure coefficients from the Poisson and thin-layer Navier-Stokes solutions for the SRB in ISSV configuration (x in $10^3$ inches)

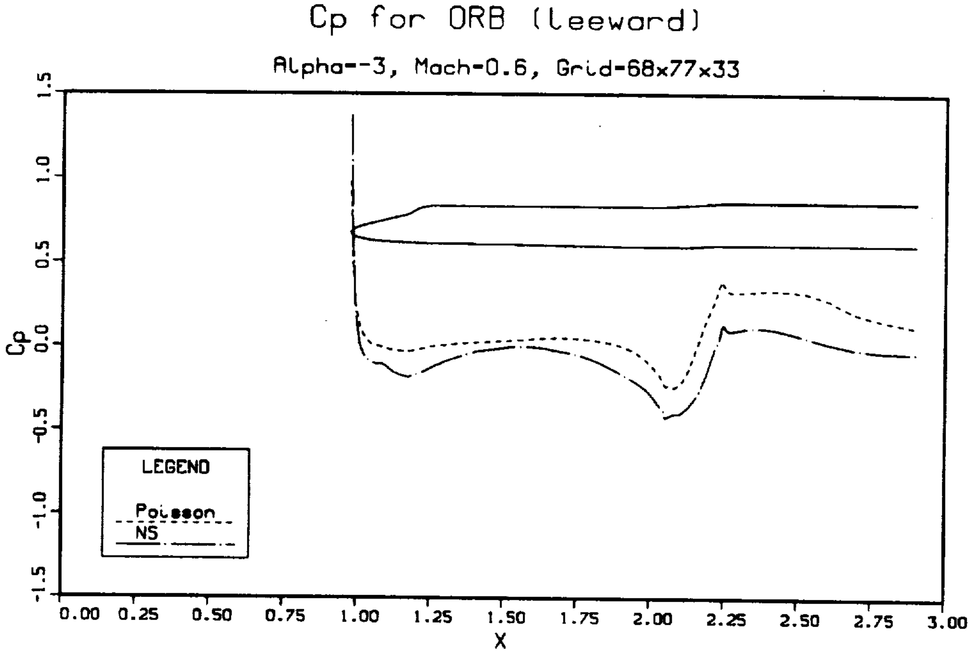


(c) $\phi = 90^{\circ}$ (outer side)

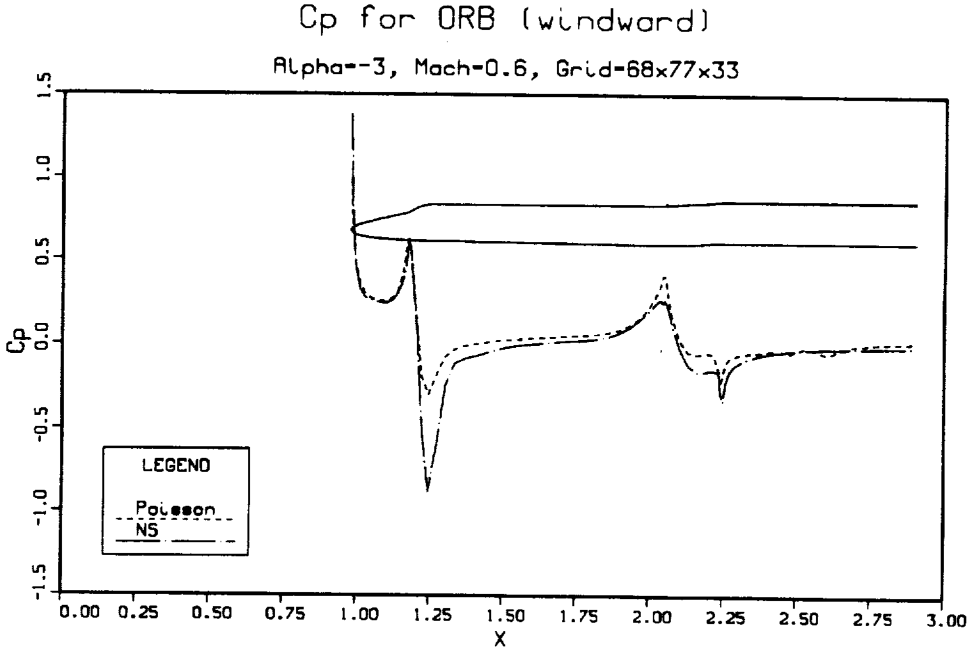


(d) $\phi = 270^{\circ}$ (inner side)

Figure 4.28 (Continued)

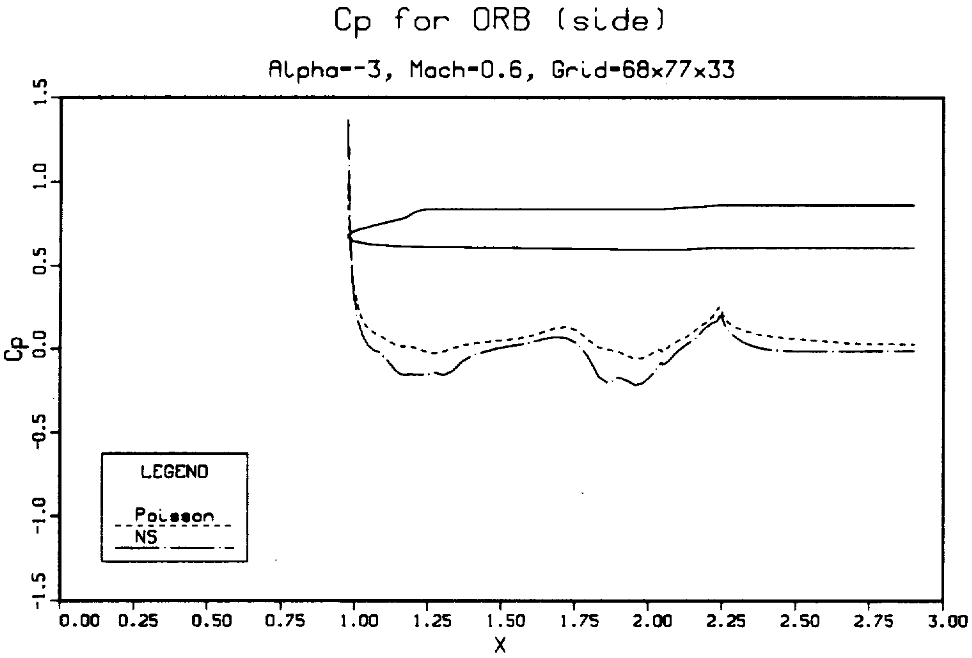


(a) $\phi = 0^{\circ}$ (bottom, leeward)



(b) $\phi = 180^{\circ}$ (top, windward)

Figure 4.29: Comparison of surface pressure coefficients from the Poisson and thin-layer Navier-Stokes solutions for the ORB in ISSV configuration (x in $10^3$ inches)

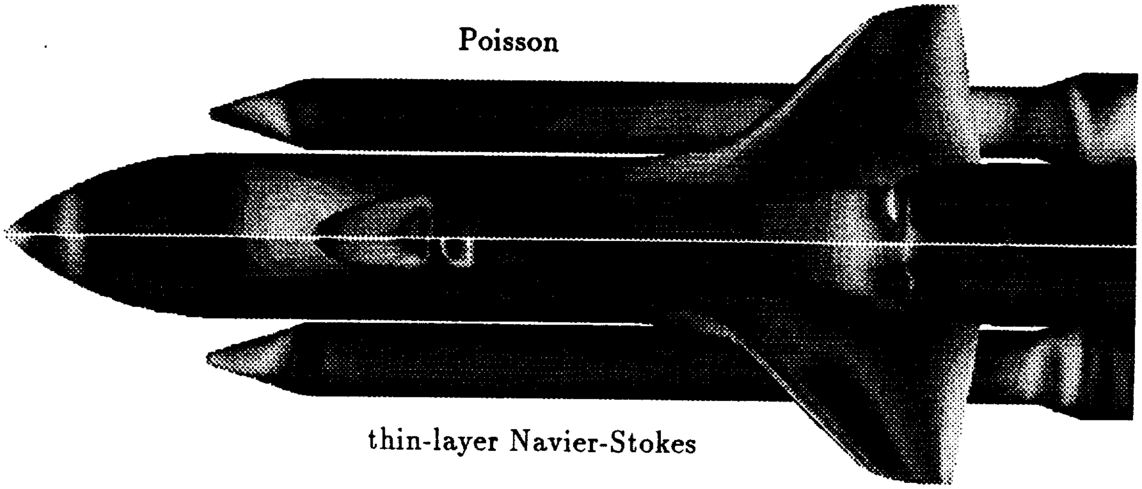


(c) $\phi = 90^{\circ}$ (side)

**Figure 4.29** (Continued)

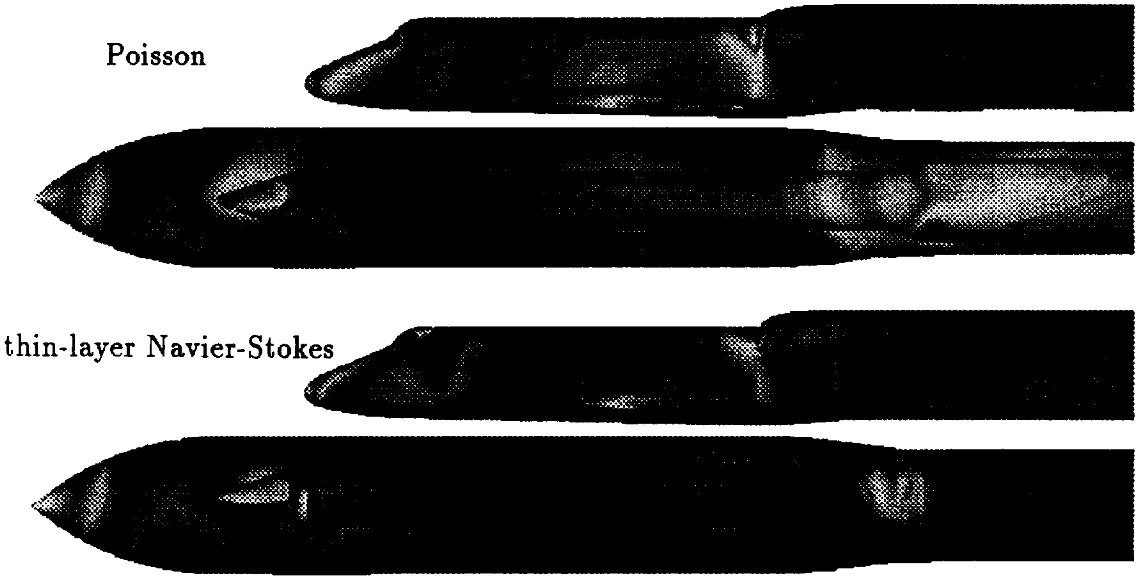
The values on the axes extending from the ET, SRB and ORB noses to the upstream far-field boundary were calculated by taking the average from points surrounding the axes. On the body surface, the tangency/vorticity consistency and the no-slip boundary conditions were employed for the Poisson and thin-layer Navier-Stokes solutions, respectively. The flow was computed for a Mach number of 0.6, and an angle of attack of $-3^{\circ}$. Surface pressure coefficient comparisons for the ET, SRB and ORB at various constant angle lines are presented in Figures 4.27 to 4.29. For the ET, the thin-layer Navier-Stokes solution predicted a sharper expansion than the Poisson solution near the nose region; however, the two solutions are in a reasonable agreement with each other. At $\phi = 180^{\circ}$, $x = 1.25$, and $\phi = 90^{\circ}$, $x = 1.0$, the thin-layer Navier-Stokes solution also shows a sharper expansion than the Poisson solution, due to the sharper expansion predicted by the thin-layer Navier-Stokes solution at the orbiter nose and the SRB nose, respectively. At $\phi = 180^{\circ}$ and $x \approx 2.0$, the Poisson solution predicted higher pressures than the thin-layer Navier-Stokes solution due to the higher pressure predicted by the Poisson equations on the orbiter surface at the corresponding location. At the rear end of the ET ($x \approx 2.4$) for all three constant angle lines ($\phi = 0$, 90, and $180^{\circ}$), the Poisson solution again shows higher pressure than the thin-layer Navier-Stokes solution. This is partly due to the higher pressure predicted by the Poisson equations at the OMS (orbiter maneuvering system) pod of the orbiter and near the nozzle of the SRB, and partly due to the Poisson solution trying to return to the stagnation pressure due to the shrinkage at the rear of the ET. For the SRB and ORB, a sharper expansion is observed for the thin-layer Navier-Stokes solution near the nose regions while higher pressure was predicted by the Poisson equations at the OMS pod of the orbiter and near the nozzle of the SRB ($x \approx 2.4$). For the rest of the flow, the Poisson solution predicted higher pressure than the thin-layer Navier-Stokes solution. Generally, the farther away from regions where two components are near each other, the better is the agreement observed between the Poisson and thin-layer Navier-Stokes solutions, e.g., at $\phi = 0^{\circ}$ (bottom) of the ET, $\phi = 180^{\circ}$ (top) of the ORB and regions near noses of all three grids. This can be attributed to the flow inside the small clearance between the ET/SRB and ET/ORB being largely (or entirely) inside the boundary layer where the potential flow assumption is not valid. Overall, the surface pressure coefficients from the Poisson solution are quite different from those of the thin-layer Navier-Stokes solution. Thus, viscous effects need to be incorporated into the Poisson equations to capture meaningful flow features for the integrated space shuttle vehicle. This can be done through the viscous-inviscid interaction between the Poisson (or the vorticity-velocity) equations and the Navier-Stokes (or the boundary-layer) equations. This part, however, is not carried out in the current research and remains a future research topic.

In Figure 4.30, the gray scale contours of the surface pressure coefficient show the three-dimensional effects of pressure variation for the two solutions. Like the quantitative comparisons of surface pressure coefficient at various constant angle lines, only regions near the noses of all three grids show similar patterns of pressure variations. As for the CPU time usage, the Poisson solution took 6850 seconds and 960 iterations to converge while the thin-layer Navier-Stokes solution took more than 16 hours of Cray-2 time and 2000 iterations to converge.



(a) Top view

Figure 4.30: Comparison of $C_p$ from the Poisson and thin-layer Navier-Stokes solutions for ISSV at $M_\infty = 0.6$, $Re = 4 \times 10^6/ft$, and $\alpha = -3°$



(b) Side view

Figure 4.30 (Continued)

# 5. CONCLUSIONS

Numerical simulations using the thin-layer Navier-Stokes equations and chimera approach for flows around the integrated space shuttle vehicle were carried out. In general, the computed surface pressure from the numerical solution was in good agreement with the available wind tunnel and flight data. However, there were discrepancies due to simplification or omission of geometric features, and improvements are needed for the numerical solution to be of greater use in engineering applications. Among the crucial improvements needed are the refinement in the modeling of the ET/ORB attach hardware and the addition of the ET/SRB attach ring. The idealization or the lack of these parts in the numerical model is believed to have caused the major discrepancies between the numerical solutions and the wind tunnel and/or flight data. In the expansion region, either increasing the grid resolution or using a higher order (more accurate) scheme may improve the accuracy of the numerical solutions. Currently, a joint effort between NASA Johnson Space Center and NASA Ames Research Center to improve the computational model of the space shuttle geometry is underway, e.g., the addition of the orbiter vertical tail and the space shuttle main engines. For the chimera scheme, using a conservative interpolation procedure instead of the current non-conservative trilinear interpolation is likely to help the shock capturing for the transonic or supersonic flows.

A Poisson solution procedure (a special case of the vorticity-velocity algorithm) using primitive variables was implemented to enhance the efficiency of the numerical flow simulation. Both single and chimera overset grids can be used with this Poisson solution procedure. The accuracy of the Poisson solution was validated[1] with the analytical (Kaplan, 1940) and Euler solutions for a sphere geometry. The effects of the far-field boundary location and grid refinement were evaluated. It was found that a far-field boundary placed at 4 diameters from the center of the sphere was sufficient to obtain an accurate solution. The computed results were found to approach fixed values when the grid was refined. In the chimera approach, the solution for a sphere using chimera overset grids was found to be in good agreement with the corresponding single grid solution. However, the chimera overset grid solution required several times more CPU time than a corresponding single grid solution. This can be improved by reducing disk I/O time through the use of the SSD (solid state device) and optimizing the finite-differencing scheme in distinguishing the hole points from the field points.

Additional validation of the Poisson solution was carried out for the ellipsoid and the external tank geometries by comparing it with the thin-layer Navier-Stokes and/or Euler solutions. In general, good agreement with the thin-layer Navier-Stokes and Euler solutions was observed for the Poisson solution. A Poisson solution for an ellipsoid at 10° angle of attack was computed and compared with the thin-layer Navier-Stokes solution as well as experimental results (Kreplin et al., 1982; Meier and Cebeci, 1985). Good agreement was observed for most regions except near the downstream end of the ellipsoid where flow separation occurred. By using the Poisson or the combined Poisson/thin-layer Navier-Stokes solutions as an initial guess for

---

[1] Only surface pressure coefficients were compared.

the thin-layer Navier-Stokes equations, roughly 30% CPU time savings was found as compared with solving the thin-layer Navier-Stokes from a constant free stream flowfield.

Finally, the Poisson equations were solved for the integrated space shuttle vehicle. The Poisson solution in general did not agree well with the thin-layer Navier-Stokes solution. This is believed to be due to the fact that the clearance between different component grids was very small so that the flows in these regions were mostly (if not entirely) inside the boundary layer, which the Poisson equations were incapable of modeling. Thus, some form of viscous-inviscid interaction (e.g., using the Poisson and thin-layer Navier-Stokes equations for viscous-inviscid interaction) is needed to capture the viscous effects in the boundary layer. A simpler way to do the viscous-inviscid interaction without computing the boundary-layer thickness is to solve the thin-layer Navier-Stokes equations for a user specified region and to solve the Poisson equations for the region that lies outside the layer in which the thin-layer Navier-Stokes equations are solved. The vorticity can be added to the Poisson equations to account for the rotational flow.

## 6. REFERENCES

Anderson, D. A., Tannehill, J. C. and Pletcher, R. H. (1984). *Computational Fluid Mechanics and Heat Transfer.* New York: McGraw-Hill.

Aregbesola, Y. A. S. and Burley, D. M. (1977). "The Vector and Scalar Potential for the Numerical Solution of Two- and Three-Dimensional Navier-Stokes Equations." *J. Comput. Phys.*, **24**, 398–415.

Atta, D. H. and Vadyak, T. (1982). "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows about Aircraft Configurations." AIAA Paper 82-1017.

Aziz, K. and Hellums, J. D. (1967). "Numerical Solution of the Three-Dimensional Equations of Motion for Laminar Natural Convection." *Phys. Fluids*, **10**, 314–325.

Baldwin, B. S. and Lomax, H. (1978). "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow." AIAA Paper 78-257.

Benek, J. A., Buning, P. G. and Steger, J. L. (1985). "A 3-D Grid Embedding Technique." AIAA Paper 85-1523.

Benek, J. A., Donegan, T. L. and Suhs, N. E. (1987). "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows." AIAA Paper 87-1126-CP.

Benek, J. A., Steger, J. L. and Dougherty, F. C. (1983). "A Flexible Grid Embedding Technique with Application to the Euler Equations." AIAA Paper 83-1944.

Benek, J. A., Steger, J. L., Dougherty, F. C. and Buning, P. G. (1986). "Chimera: A Grid Embedding Technique." AEDC-TR-85-64, Arnold Air Force Station, TN.

Benjamin, A. S. and Denny, V. E. (1979). "On the Convergence of Numerical Solutions for 2-D Flows in a Cavity at Large Re." *J. Comput. Phys.*, **33**, 340–358.

Berger, M. J. and Oliger, J. (1984). "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations." *J. Comput. Phys.*, **53**, 484–512.

Brackbill, J. U. (1982). "Coordinate System Control: Adaptive Meshes." In *Numerical Grid Generation*, ed. J. F. Thompson, 277–294. New York: North-Holland.

Bridgeman, J. O., Steger, J. L. and Caradonna, F. X. (1982). "A Conservative Finite Difference Algorithm for the Unsteady Transonic Potential Equation in Generalized Coordinates." AIAA Paper 82-1388.

Buning, P. G. and Steger, J. L. (1985). "Graphics and Flow Visualization in Computational Fluid Dynamics." AIAA Paper 85-1507.

Buning, P. G., Chiu, I. T., Obayashi, S., Rizk, Y. M. and Steger, J. L. (1988). "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent." AIAA Paper 88-4359-CP.

Buning, P. G., Chiu, I. T., Martin, F., Meakin, R., Obayashi, S., Rizk, Y. M., Steger, J. L. and Yarrow, M. (1989). "Flowfield Simulation of the Space Shuttle Vehicle in Ascent." Proceedings of the Fourth International Conference on Supercomputing, Santa Clara, California. 20–28.

Chaderjian, N. M. and Steger, J. L. (1983). "A Zonal Approach for the Steady Transonic Simulation of Inviscid Rotational Flow." AIAA Paper 83-1927.

Chakravarthy, S. R. and Szema, K. Y. (1985). "An Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets." AIAA Paper 85-1703.

Chapman, D. R. (1979). "Computational Aerodynamics Development and Outlook." *AIAA J.*, **17**, 1293–1313.

Dennis, S. C. R., Ingham, D. B. and Cook, R. N. (1979). "Finite-Difference Methods for Calculating Steady Incompressible Flows in Three Dimensions." *J.*

*Comput. Phys.*, **33**, 325–339.

Dougherty, F. C., Benek, J. A. and Steger, J. L. (1985). "On Application of Chimera Grid Scheme to Store Separation." NASA TM 88193.

Eberle, A. and Misegades, K. (1986). "Euler Solution for a Complete Fighter Aircraft at Sub- and Supersonic Speed." AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France.

Farouk, B. and Fusegi, T. (1985). "A Coupled Solution of the Vorticity-Velocity Formulation of the Incompressible Navier-Stokes Equations." *Int. J. Numerical Methods Fluids*, **5**, 1017–1034.

Farouk, B. and Guceri, S. I. (1982). "Natural Convection from a Horizontal Cylinder within Confining Walls." *Numerical Heat Transfer*, **5**, 324–341.

Fasel, H. (1976). "Investigation of the Stability of Boundary Layers by a Finite-Difference Model of the Navier-Stokes Equations." *J. Fluid Mechanics*, **78**, 355–383.

Fasel, H. and Booz, O. (1984). "Numerical Investigation of Supercritical Taylor-Vortex Flow over a Wide Gap." *J. Fluid Mechanics*, **138**, 21–52.

Fletcher, C. A. J. (1988). *Computational Techniques for Fluid Dynamics. Volume I – Fundamental and General Techniques.* New York: Springer-Verlag.

Flores, J., Reznick, S. G., Holst, T. L. and Gundy, K. L. (1987). "Transonic Navier-Stokes Solution for a Fighter-like Configuration." AIAA Paper 87-0032.

Fromm, J. E. and Harlow, F. H. (1963). "Numerical Solution of the Problem of Vortex Street Development." *Phys. Fluids*, **6**, 975–982.

Gatski, T. B., Grosch, C. E. and Rose, M. E. (1982). "A Numerical Study of the Two-Dimensional Navier-Stokes Equations in Vorticity-Velocity Variables." *J. Comput. Phys.*, **48**, 1–22.

Gegg, S. G. (1989). "A Dual Potential Formulation of the Navier-Stokes Equations." Ph.D. dissertation, Iowa State University, Ames, Iowa.

Ghia, K., Ghia, U. and Shin, C. T. (1983). "Adaptive Grid Generation for Flows with Local High Gradient Regions." In *Advances in Grid Generation*, ASME FED-5, eds. K. N. Ghia and U. Ghia. New York: ASME.

Gnoffo, P. A. (1982). "A Vectorized, Finite Volume, Adaptive Grid Algorithm for Navier-Stokes." In *Numerical Grid Generation*, ed. J. F. Thompson, 819–836. New York: North-Holland.

Guj, G. and Stella, F. (1988). "Numerical Solutions of High-Re Recirculating Flows in Vorticity-Velocity Form." *Int. J. Numerical Methods Fluids*, **8**, 405–416.

Gunzburger, M. D. and Peterson, J. S. (1984). "On the Finite-Element Approximation of the Stream Function-Vorticity Equations." In *Advances in Computer Methods for Partial Differential Equations V*, eds. R. Vichnevetsky and R. S. Stepleman, 47–56. Brussels: IMACS.

Hafez, M. and Lovell, D. (1981). "Numerical Solution of Transonic Stream Function." AIAA Paper 81-1017.

Hafez, M. and Lovell, D. (1983). "Entropy and Vorticity Corrections for Transonic Flows." AIAA Paper 83-1926.

Hafez, M. and Lovell, D. (1988). "Entropy and Vorticity Corrections for Transonic Flows." *Int. J. Numerical Methods Fluids*, **8**, 31–53.

Halim, A. and Hafez, M. (1984). "Calculations of Separation Bubbles Using Boundary-Layer-Type Equations – Part I & II." AIAA Paper 84-1585.

Helmholtz, H. (1858). "Über Integrale der Hydrodynamischen Gleichungen, welche den Wirbelbewegungen entsprechen." *Crelles J.*, **55**, 25–55.

Helmholtz, H. (1867). "On Integrals of the Hydrodynamical Equations, which express Vortex-Motion." *Phil. Mag.*, Ser. 4, **33**, 485–512.

Hirasaki, G. J. and Hellums, J. D. (1968). "A General Formulation of the Boundary Conditions on the Vector Potential in Three-Dimensional Hydrodynamics." *Q. Appl. Math.*, **26**, 331–342.

Hirasaki, G. J. and Hellums, J. D. (1970). "Boundary Conditions on the Vector and Scalar Potentials in Viscous Three-Dimensional Hydrodynamics." *Q. Appl. Math.*, **28**, 293–296.

Holst, T. L. (1987). "Numerical Solution of the Navier-Stokes Equations about Three-Dimensional Configurations — A Survey." *Supercomputing in Aerospace*, NASA CP 2454, 281–298.

Imai, I. (1938). "On the Flow of a Compressible Fluid past a Circular Cylinder." Proc. Phys., Math. Soc. of Japan, Ser. 3, **20**, 636–645.

Jameson, A. and Baker, T. J. (1987). "Improvements to the Aircraft Euler Method." AIAA Paper 87-0452.

Jameson, A., Baker, T. J. and Weatherill, N. P. (1986). "Calculation of Inviscid Transonic Flow Over a Complete Aircraft." AIAA Paper 86-0103.

Janzen, O. (1913). "Beitrag zu einer Theorie der Stationaren Strömung kompressibler Flüssigkeiten." *Phys. Z.*, 14, Jahrg., Nr. 14, 639–643.

Kaplan, C. (1938). "Two-Dimensional Subsonic Compressible Flow past Elliptic Cylinders." T. R. No. 624, NACA.

Kaplan, C. (1940). "The Flow of a Compressible Fluid past a Sphere." Tech. Notes No. 762, NACA.

Kinsey, D. W. and Barth, T. J. (1984). "Description of a Hyperbolic Grid Generation Procedure for Arbitrary Two-Dimensional Bodies." AFWAL Technical Memorandum, No. 84-191-FIMM.

Kreplin, H. P., Vollmers, H. and Meier, H. U. (1982). "Measurements of the Wall Shear Stress on an Inclined Prolate Spheroid." *Zeitschrift für Flugwissenschaft Weltraumforschung*, **6**, 248–252.

Lee, D-S. and Pletcher, R. H. (1986). "Application of Viscous-Inviscid Interaction Methods to Transonic Turbulent Flows." Iowa State University, Report HTL-42, CFD-16, ISU-ERI-Ames-87055.

Li, C. P. (1989). "Johnson Space Center CFD Overview." Proceedings of NASA Computational Fluid Dynamics Conference, NASA Ames Research Center, Moffett Field, California. Vol. 1, 95–121.

Mallinson, G. D. and de Vahl Davis, G. (1973). "The Method of the False Transient for the Solution of Coupled Elliptic Equations." *J. Comput. Phys.*, **12**, 435–461.

Mastin, C. W. and McConnaughey, H. V. (1984). "Computational Problems on Composite Grids." AIAA Paper 84-1611.

Meier, H. U. and Cebeci, T. (1985). "Flow Characteristics of a Body of Revolution at Incidence." Third Symposium on Numerical and Physical Aspects of Aerodynamics Flows, Long Beach, California, Jan. 21–24.

Obayashi, S. (1987). "Toward the Navier-Stokes Analysis of Transport Aircraft Configurations." AIAA Paper 87-0428.

Obayashi, S. (1988). "Numerical Simulation of Underexpanded Plumes using Upwind Algorithms." AIAA Paper 88-4360.

Orlandi, P. (1987). "Vorticity-Velocity Formulation for High Re Flows." *Computers and Fluids*, **15**, 137–149.

Osswald, G. A., Ghia, K. N. and Ghia, U. (1987). "A Direct Algorithm for Solution of Incompressible Three-Dimensional Unsteady Navier-Stokes Equations." AIAA Paper 87-1139.

Ozoe, H., Yamamoto, K., Churchill, S. W. and Sayama, H. (1976). "Three-Dimensional Numerical Analysis of Laminar Natural Convection in a Confined Fluid Heated from below." *J. Heat Transfer*, Transactions of ASME, **98**, 202–207.

Ozoe, H., Yamamoto, K., Sayama, H. and Churchill, S. W. (1977). "Natural Convection Patterns in a Long Inclined Rectangular Box Heated from below. Part II. Three-Dimensional Numerical Results." *Int. J. Heat Mass Transfer*, **20**, 131–139.

Ozoe, H., Yamamoto, K. and Churchill, S. W. (1979). "Three-Dimensional Numerical Analysis of Natural Convection in an Inclined Channel with a Square Cross Section." *AIChE J.*, **25**, 709–716.

Ozoe, H., Mouri, A., Ohmuro, M., Churchill, S. W. and Lior, H. (1985). "Numerical Calculations of Laminar and Turbulent Natural Convection in Water in Rectangular Channels Heated and Cooled Isothermally on the Opposing Vertical Walls." *Int. J. Heat Mass Transfer*, **28**, 125–138.

Peyret, R. and Viviand, H. (1975). "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations." AGARD-AG-212, 1975.

Pulliam, T. H. and Steger, J. L. (1978). "On Implicit Finite-Difference Simulations of Three-Dimensional Flow." AIAA Paper 78-10.

Rai, M. M. (1985). "A Relaxation Approach to Patched-Grid Calculations with the Euler Equations." AIAA Paper 85-0295.

Rai, M. M. (1986a). "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations." *J. Comput. Phys.*, **62**, 472–503.

Rai, M. M. (1986b). "An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculations." *Computers and Fluids*, **14**, 295–319.

Rao, K. V., Pletcher, R. H., Steger, J. L. and Van Dalsem W. R. (1987). "A Three-Dimensional Dual Potential Procedure with Applications to Wind Tunnel Inlets and Interacting Boundary Layers." Iowa State University, Report HTL-47, CFD-18, ISU-ERI-Ames-87296.

Rao, K. V., Steger, J. L. and Pletcher, R. H. (1989). "Three-Dimensional Dual Potential Procedure for Inlets and Indraft Wind Tunnels." *AIAA J.*, **27**, 876–884.

Rayleigh, Lord. (1916). "On the Flow of Compressible Fluid past an Obstacle." *Phil. Mag.*, Ser. 6, **32**, 1–6.

Richardson, S. M. and Cornish, A. R. H. (1977). "Solution of Three-Dimensional Incompressible Flow Problems." *J. Fluid Mechanics*, **82**, 309–319.

Rizk, Y. M. and Ben-Shmuel, S. (1985). "Computation of the Viscous Flow around the Shuttle Orbiter at Low Supersonic Speeds." AIAA Paper 85-0168.

Rizk, Y. M., Steger, J. L. and Chaussee, D. (1985). "Use of a Hyperbolic Grid Generation Scheme in Simulating Supersonic Viscous Flow about Three-Dimensional Winged Configurations." NASA TM 86776.

Roache, P. J. (1972). *Computational Fluid Dynamics.* New Mexico: Hermosa Publishers.

Rockwell International Internal Letter SAS / AERO 83-094. "STS-5 Postflight Report on Ascent External Aerodynamic Loads." March 1983.

Saltzman, J. and Brackbill, J. U. (1982). "Application and Generation of Variational Methods for Generating Adaptive Systems." In *Numerical Grid Generation*, ed. J. F. Thompson, 865-884. New York: North-Holland.

Shang, J. S. (1985). "An Assessment of Numerical Solutions of the Compressible

Navier-Stokes Equations." *J. Aircraft*, **22**, 353–370.

Spangler, R. H. (1981). "Results of Tests Using a 0.03 Scale Model (47-OTS) of the Space Shuttle Integrated Vehicle in the AEDC 16 Foot Transonic Propulsion Wind Tunnel (IA105A)." Vol. 2 of 3, NASA Contractor Report 160851.

Speziale, C. G. (1987). "On the Advantages of the Vorticity-Velocity Formulation of the Equations of Fluid Dynamics." *J. Comput. Phys.*, **73**, 476–480.

Steger, J. L. (1977). "Implicit Finite Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils." AIAA Paper 77-665.

Steger, J. L. and Benek, J. A. (1987). "On the Use of Composite Grid Schemes in Computational Aerodynamics." *Computer Methods in Applied Mechanics and Engineering*, **64**, 301–320.

Steger, J. L. and Chaussee, D. S. (1980). "Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations." *SIAM J. Sci. Stat. Comput.*, **1**, 431–437.

Steger, J. L. and Rizk, Y. M. (1985). "Generation of Three-Dimensional Body Fitted Coordinates Using Hyperbolic Partial Differential Equations." NASA TM 86753.

Steger, J. L., Dougherty, F. C. and Benek, J. A. (1983). "A Chimera Grid Scheme." In *Advances in Grid Generation*, ASME FED-5, eds. K. N. Ghia and U. Ghia, 59–69. New York: ASME.

Steger, J. L., Ying, S. X. and Schiff, L. B. (1986). "A Partially Flux-Split Algorithm for Numerical Simulation of Compressible Inviscid and Viscous Flow." Proceedings of the Workshop on Computational Fluid Dynamics, Institute of Nonlinear Sciences, University of California, Davis, California.

Szema, K. Y., Chakravarthy, S. R., Riba, W. T., Byerly, J. and Dresser, H. S. (1987). "Multi-Zone Euler Marching Technique for Flow Over Singe and Multi-Body Configurations." AIAA Paper 87-0592.

Szema, K. Y., Chakravarthy, S. R., Pan, D., Bihari, B. L., Riba, W. T., Akdag, V. M. and Dresser, H. S. (1988). "The Application of a Unified Marching Technique for Flow over Complex 3-Dimensional Configurations across the Mach Number Range." AIAA Paper 88-0276.

Thomas, J. L., Taylor, S. L. and Anderson, W. K. (1987). "Navier-Stokes Computations of Vortical Flows over Low Aspect Ratio Wings." AIAA Paper 87-0207.

Van Dalsem, W. R. and Steger, J. L. (1985). "The Efficient Simulation of Separated Three-Dimensional Viscous Flows Using the Boundary-Layer Equations." AIAA Paper 85-4064.

Van Dalsem, W. R. and Steger, J. L. (1986). "The Fortified Navier-Stokes Approach." Proceedings of the Workshop on Computational Fluid Dynamics, Institute of Nonlinear Sciences, University of California, Davis, California.

Vatsa, V. N., Thomas, J. L. and Wedan, B. W. (1987). "Navier-Stokes Computations of Prolate Spheroids at Angle of Attack." AIAA Paper 87-2627-CP.

Vinokur, M. (1986). "An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws." NASA Contractor Report 177416.

Wedan, B. and South, J. C., Jr. (1983). "A Method for Solving the Transonic Full-Potential Equation for General Configurations." AIAA Paper 83-1889.

Whitfield, D. L., Swafford, T. M. and Jacocks, J. L. (1981). "Calculation of Turbulent Boundary Layers with Separation and Viscous-Inviscid Interaction." *AIAA J.*, **19**, 1315–1322.

Williams, G. P. (1969). "Numerical Integration of the Three-Dimensional Navier-Stokes Equations for Incompressible Flow." *J. Fluid Mechanics*, **37**, 727–750.

Wong, A. K. and Reizes, J. A. (1984). "An Effective Vorticity-Vector Potential Formulation for the Numerical Solution of Three-Dimensional Duct Flow Problems." *J. Comput. Phys.*, **55**, 98–114.

Wong, A. K. and Reizes, J. A. (1986). "The Vector Potential in the Numerical Solution of Three-Dimensional Fluid Dynamics Problems in Multiply Connected Regions." *J. Comput. Phys.*, **62**, 124–142.

Ying, S. X., Steger, J. L., Schiff, L. B. and Baganoff, D. (1986). "Numerical Simulation of Unsteady, Viscous, High Angle of Attack Flows Using a Partially Flux-Split Algorithm." AIAA Paper 86-2179.

# 7. APPENDIX A. POISSON EQUATIONS IN GENERALIZED COORDINATES

Poisson equations:

$$\nabla^2 u + \vartheta_x + \omega_{3y} - \omega_{2z} = 0 \tag{7.1}$$

$$\nabla^2 v + \vartheta_y + \omega_{1z} - \omega_{3x} = 0 \tag{7.2}$$

$$\nabla^2 w + \vartheta_z + \omega_{2x} - \omega_{1y} = 0 \tag{7.3}$$

Rewrite the above equations in vector form:

$$\partial_x E + \partial_y F + \partial_z G + H = 0 \tag{7.4}$$

where

$$E = \begin{bmatrix} u_x \\ v_x \\ w_x \end{bmatrix}, F = \begin{bmatrix} u_y \\ v_y \\ w_y \end{bmatrix}, G = \begin{bmatrix} u_z \\ v_z \\ w_z \end{bmatrix} \tag{7.5}$$

and

$$H = \begin{bmatrix} \vartheta_x + \omega_{3y} - \omega_{2z} \\ \vartheta_y + \omega_{1z} - \omega_{3x} \\ \vartheta_z + \omega_{2x} - \omega_{1y} \end{bmatrix} \tag{7.6}$$

In generalized coordinates, the above equation can be written as

$$\xi_x E_\xi + \eta_x E_\eta + \zeta_x E_\zeta + \xi_y F_\xi + \eta_y F_\eta + \zeta_y F_\zeta + \xi_z G_\xi + \eta_z G_\eta + \zeta_z G_\zeta + H = 0 \tag{7.7}$$

Identities resulting from the coordinate transformation:

$$\left(\frac{\xi x}{J}\right)_\xi + \left(\frac{\eta x}{J}\right)_\eta + \left(\frac{\zeta x}{J}\right)_\zeta = 0 \tag{7.8}$$

$$\left(\frac{\xi y}{J}\right)_\xi + \left(\frac{\eta y}{J}\right)_\eta + \left(\frac{\zeta y}{J}\right)_\zeta = 0 \tag{7.9}$$

$$\left(\frac{\xi z}{J}\right)_\xi + \left(\frac{\eta z}{J}\right)_\eta + \left(\frac{\zeta z}{J}\right)_\zeta = 0 \tag{7.10}$$

Scale the Poisson equations by the Jacobian and combine them with the identities from coordinate transformation to form the Poisson equations in conservative form as shown below.

$$\begin{aligned}
&\tfrac{\xi x}{J}E_\xi + \tfrac{\eta x}{J}E_\eta + \tfrac{\zeta x}{J}E_\zeta + E\left[\left(\tfrac{\xi x}{J}\right)_\xi + \left(\tfrac{\eta x}{J}\right)_\eta + \left(\tfrac{\zeta x}{J}\right)_\zeta\right] + \\
&\tfrac{\xi y}{J}F_\xi + \tfrac{\eta y}{J}F_\eta + \tfrac{\zeta y}{J}F_\zeta + F\left[\left(\tfrac{\xi y}{J}\right)_\xi + \left(\tfrac{\eta y}{J}\right)_\eta + \left(\tfrac{\zeta y}{J}\right)_\zeta\right] + \\
&\tfrac{\xi z}{J}G_\xi + \tfrac{\eta z}{J}G_\eta + \tfrac{\zeta z}{J}G_\zeta + G\left[\left(\tfrac{\xi z}{J}\right)_\xi + \left(\tfrac{\eta z}{J}\right)_\eta + \left(\tfrac{\zeta z}{J}\right)_\zeta\right] + \tfrac{H}{J} = 0
\end{aligned} \tag{7.11}$$

From the chain rule of differentiation, the above equation can be rewritten as,

$$\left(\frac{\xi x E + \xi y F + \xi z G}{J}\right)_\xi + \left(\frac{\eta x E + \eta y F + \eta z G}{J}\right)_\eta + \left(\frac{\zeta x E + \zeta y F + \zeta z G}{J}\right)_\zeta + \frac{H}{J} = 0 \tag{7.12}$$

where

$$E = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \xi x \\ \eta x \\ \zeta x \end{bmatrix} \tag{7.13}$$

$$F = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \xi y \\ \eta y \\ \zeta y \end{bmatrix} \tag{7.14}$$

$$G = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \xi z \\ \eta z \\ \zeta z \end{bmatrix} \tag{7.15}$$

$$H = \begin{bmatrix} \xi x\vartheta_\xi + \eta x\vartheta_\eta + \zeta x\vartheta_\zeta + \xi y\omega_{3\xi} + \eta y\omega_{3\eta} + \zeta y\omega_{3\zeta} - \xi z\omega_{2\xi} - \eta z\omega_{2\eta} - \zeta z\omega_{2\zeta} \\ \xi y\vartheta_\xi + \eta y\vartheta_\eta + \zeta y\vartheta_\zeta + \xi z\omega_{1\xi} + \eta z\omega_{1\eta} + \zeta z\omega_{1\zeta} - \xi x\omega_{3\xi} - \eta x\omega_{3\eta} - \zeta x\omega_{3\zeta} \\ \xi z\vartheta_\xi + \eta z\vartheta_\eta + \zeta z\vartheta_\zeta + \xi x\omega_{2\xi} + \eta x\omega_{2\eta} + \zeta x\omega_{2\zeta} - \xi y\omega_{1\xi} - \eta y\omega_{1\eta} - \zeta y\omega_{1\zeta} \end{bmatrix} \tag{7.16}$$

$$J = \begin{bmatrix} \xi x & \eta x & \zeta x \\ \xi y & \eta y & \zeta y \\ \xi z & \eta z & \zeta z \end{bmatrix} \tag{7.17}$$

and

$$\xi x E + \xi y F + \xi z G = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \xi x\xi x + \xi y\xi y + \xi z\xi z \\ \xi x\eta x + \xi y\eta y + \xi z\eta z \\ \xi x\zeta x + \xi y\zeta y + \xi z\zeta z \end{bmatrix} \tag{7.18}$$

$$\eta x E + \eta y F + \eta z G = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \eta x\xi x + \eta y\xi y + \eta z\xi z \\ \eta x\eta x + \eta y\eta y + \eta z\eta z \\ \eta x\zeta x + \eta y\zeta y + \eta z\zeta z \end{bmatrix} \tag{7.19}$$

$$\zeta x E + \zeta y F + \zeta z G = \begin{bmatrix} u_\xi & u_\eta & u_\zeta \\ v_\xi & v_\eta & v_\zeta \\ w_\xi & w_\eta & w_\zeta \end{bmatrix} \begin{bmatrix} \zeta x\xi x + \zeta y\xi y + \zeta z\xi z \\ \zeta x\eta x + \zeta y\eta y + \zeta z\eta z \\ \zeta x\zeta x + \zeta y\zeta y + \zeta z\zeta z \end{bmatrix} \tag{7.20}$$

# 8. APPENDIX B. IMPLICIT PROCEDURE FOR TANGENCY/VORTICITY CONSISTENCY BOUNDARY CONDITION WITH POISSON EQUATIONS

The tangency/vorticity consistency boundary condition for the solid boundary is solved implicitly with the Poisson equations. The equation below illustrates that the finite-difference equations of the computational domain are a system of block tridiagonal matrices. Generally, a block tridiagonal solver is used to solve a block tridiagonal system of equations, and the number of operations required to invert this system of equations is approximately $(5NM^3/3)$ where $N$ is the number of equations and $M$ the blocksize (Fletcher, 1988). However, due to the special structure of the matrix involved here, it is possible to apply the Thomas algorithm (used to invert scalar tridiagonal matrices) to invert the block tridiagonal system of equations. Since the number of operations needed to invert scalar tridiagonal systems is only about $(5NM)$ (Fletcher, 1988), the savings in number of operations is roughly $(M^2/3 - 1)$. For the Poisson equation $(M = 3)$, so only one-third of the operations required to invert the block tridiagonal matrix is needed to invert the following block tridiagonal matrices using the Thomas algorithm.

$$\begin{bmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & B_{n-1} & C_{n-1} \\ & & & A_n & B_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix} \tag{8.1}$$

In the above equation, $A_n$, $B_n$ and $C_n$ are 3 by 3 diagonal matrices, except for $C_1$ which is a full 3 by 3 matrix, and $X_n$ and $D_n$ are vectors of length 3, $n = 1, 2, \cdots, n$. Since all matrices in the above equation are diagonal matrices (except $C_1$), it is easy to eliminate the upper diagonal matrices by Gauss elimination for $n = n - 1, \cdots, 2$. Thus, by applying Gauss elimination, the above equation becomes

$$\begin{bmatrix} B_1 & C_1 & & & \\ A'_2 & B'_2 & & & \\ & \ddots & \ddots & & \\ & & A'_{n-1} & B'_{n-1} & \\ & & & A_n & B_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D'_2 \\ \vdots \\ D'_{n-1} \\ D_n \end{bmatrix} \tag{8.2}$$

with the first rows of matrix elements assuming the form as illustrated below:

$$\left( \left[\begin{array}{ccc} * & & \\ & * & \\ & & * \\ \hline * & & \\ & * & \\ & & * \end{array}\right] \left[\begin{array}{ccc} * & * & * \\ * & * & * \\ * & * & * \\ \hline * & & \\ & * & \\ & & * \end{array}\right] \right) \left( \left[\begin{array}{c} * \\ * \\ * \\ \hline * \\ * \\ * \end{array}\right] \right) = \left( \left[\begin{array}{c} * \\ * \\ * \\ \hline * \\ * \\ * \end{array}\right] \right) \tag{8.3}$$

The solution of the above equation, $X_1$ and $X_2$, can be obtained analytically. Then, a forward substitution can be used to get the solution for $X_n, n = 3, \cdots, n$.