Summer 1990

Navier-Stokes Simulations of Flows About Complex Configurations Using Domain Decomposition Techniques

Kamran Fouladi-Semnani
Old Dominion University

Recommended Citation

Fouladi-Semnani, Kamran. "Navier-Stokes Simulations of Flows About Complex Configurations Using Domain Decomposition Techniques" (1990). Doctor of Philosophy (PhD), dissertation, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/s0cb-9869
https://digitalcommons.odu.edu/mae_etds/234

NAVIER-STOKES SIMULATIONS OF FLOWS ABOUT COMPLEX CONFIGURATIONS USING DOMAIN DECOMPOSITION TECHNIQUES

by

Kamran Fouladi-Semnani
B.S., April 1982, Florida International University, Miami, Florida
M.E., May 1986, Old Dominion University, Norfolk, Virginia

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 1990

Approved by:

Oktay Baysal (Director)

Osama A. Kandil

Robert L. Ash

David S. Miller

James L. Thomas

ABSTRACT

NAVIER-STOKES SIMULATIONS OF FLOWS ABOUT COMPLEX CONFIGURATIONS USING DOMAIN DECOMPOSITION TECHNIQUES

Kamran Fouladi-Semnani
Old Dominion University, 1990
Director: Dr. Oktay Baysal

An algorithm is developed to obtain numerical simulations of flows about complex configurations composed of multiple and nonsimilar components with arbitrary geometries. The algorithm uses a hybridization of the domain decomposition techniques for grid generation and to reduce the computer memory requirement. Three dimensional, Reynolds-averaged, unsteady, compressible, and complete Navier-Stokes equations are solved on each of the subdomains by a fully-vectorized, finite-volume, upwind-biased, approximately-factored, and multigrid method. The effect of Reynolds stresses is incorporated through an algebraic turbulence model with several modifications for interference flows. The present algorithm combines the advantages of an efficient, geometrically conservative, minimally and automatically dissipative algorithm with advantages and flexibility of domain decomposition techniques. The algorithm is used to simulate supersonic flows over two-dimensional profiles and a body of revolution at high angles of attack. This study is performed to examine the suitability of the baseline solution algorithm and gain a better understanding of this class of flows. The grid overlapping is tested by obtaining the solution of a supersonic flow over a blunt-nose-cylinder at high angles of attack using a composite of overlapped grids. This solution compares very well with the solution of the same flowfield obtained with no overlapping and the experimental data.

The multigrid algorithm used for this case shows substantial savings in the computational time. To accomplish one of the main objectives of this study, the algorithm is then applied to simulate the supersonic flow over an ogive-nose-cylinder near and inside a cavity. The cylinder is attached to an offset L-shaped sting when placed above the cavity opening. The results of the time-accurate computations depict these complex flows and help understanding interference effects. The unsteady nature of these flowfields and the interaction of the cavity shear layer with the cylinder are simulated. These cases illustrate two significantly different and important interference characteristics for a store separating from its parent body. Unsteadiness of the cavity flow has a more pronounced effect on the normal forces acting on the cylinder when the cylinder is placed inside the cavity. A clearer understanding of the flow between the base of the cylinder and the cavity rear face is gained by eliminating the offset sting when the cylinder is inside the cavity. The time averaged surface pressures compare favorably with the wind tunnel data, despite the averaging time period for the computations being three orders of magnitude smaller than that of the experimental measurements. The results of the present computations contribute to the much needed database for the internal store carriage and separation.

TABLE OF CONTENTS

APPENDICES

ACKNOWLEDGEMENTS

I wish to express my highest appreciation to my advisor Dr. Oktay Baysal for his invaluable guidance, support, and friendship during the entire course of this study.

Special thanks are also extended to members of my dissertation committee for their valuable suggestions. I also wish to thank my research group partners V. R. Lessard and G. W. Yen for all their help.

Further, I would like to express my love, gratitude and appreciation to my parents and other members of my family for their continuous support and encouragement. I am grateful to them forever.

This work was supported by NASA Langley Research Center under Grant No. NAG-1-664.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	Flux Jacobian
a,b	bilinear interpolation constants
a	speed of sound
BNC	blunt-nose-cylinder
C_A	axial force coefficient
C_N	normal force coefficient
C_f	skin friction coefficient
C_p	pressure coefficient
CFD	computational fluid dynamics
CPU	central processing unit
CSC	cylinder-sting-cavity configuration
C_1,C_2	coarser level cells
D	cavity depth
DDT	domain decomposition technique
d	cylinder diameter
E	relative truncation error
e	energy unit per volume
F,G,H	inviscid flux vectors
F_v,G_v,H_v	viscous flux vectors
F	cavity floor
F(y)	function in turbulence model
FAS	full approximation solution
FD	finite difference
FDS	flux-difference splitting
FF	cavity front face
FV	finite volume
f_1,f_2	psuedo fine level cells

G_L	grid at hierarchy level L
h	spatial step size
I	unit diagonal matrix
I_i^{i+1}	volume-weighted restriction operator from level i+1 to i
$\hat{I}_i^{i+1}$	restriction operator from level i+1 to level i
IFLAG	flag indicating computed, fringe, hole cells
IFLAGM	flag indicating illegal zone for coarser level M
i,j,k	component of unit vector
J	Jacobian matrix
L	reference length
	cavity length
	linear operator
M	Mach number
	coarser grid level
	metric terms
N	outward normal
n	unit normal
ONC	ogive-nose-cylinder
P	static pressure
P_t	stagnation pressure
P_o	origin of search circle
Pr	Prandtl number
Pr_t	turbulent Prandtl number
p	order of differential equation
R	Residual
	Reimann invariants
	search radius
RF	cavity rearface
R_e	Reynolds number
r	order of accuracy
Q	conserved variables
q	primitive variable
$\dot{q}$	heat flux

S	surface
	limiter control function
SW	side wall
s	arc length
	Sutherland coefficient
T	temperature
	diagonalization matrix
TDP	two-dimensional profile
T_t	stagnation temperature
t	time
t_{c1}	first time characteristic
t_{c2}	second time characteristic
U,V,W	contravariant velocity components
u,v,w	velocity components
V	volume
x,y,z	Cartesian coordinates

<u>Greek Symbols</u>

α	angle of attack
α,β,γ	coordinates of fringe cell relative to target cell
γ	ratio of specific heat
Δ	change
δ	Kronecker delta
λ	coefficient of bulk viscosity
$\lambda_1,\lambda_2,\lambda_3\lambda_4,\lambda_5$	eigenvalues of Jacobian matrix
μ	coefficient of molecular viscosity
μ_t	turbulent coefficient of molecular viscosity
ξ,η,ζ	curvilinear coordinates
ρ	density
Σ	summation
τ	viscous stress terms

<u>Superscripts</u>

L,R	left and right of cell interface
n	time level

$\sim$	dimensional
$\sim$	Roe-averaged value
$\rightarrow$	vector
$\pm$	positive and negative flux and eigenvalue contribution forward and backward spatial differencing
*,**	denote intermediate values
<u>Subscripts</u>	
blp	boundary layer profile value
e	effective
i,j,k	cell indices
L	hierarchy level
L,R	left and right of the interface
w	wall value
∞	freestream value
<u>Symbols</u>	
grad(ξ)	gradient of ξ
$\lvert x \rvert$	magnitude of x
δ	difference operator
Δ	forward difference operator
∇	backward difference operator
∂	denotes partial derivative

Chapter 1
INTRODUCTION

1.1 Rationale

Rapid advances in computer hardware and architecture have made it possible to solve complex flow problems using computational fluid dynamics (CFD). The objective of CFD is to solve the system of partial differential equations which govern the fluid flow, using discrete methods. An essential element of CFD solutions is the construction of a grid (mesh) on which the flow equations are solved in a finite form. A major limitation in CFD is the generation of suitable grids for a complex three-dimensional configuration. Configurations, such as, a wing and a fuselage, a wing and a store, a wing and a nacelle, a store and a cavity, a complete stage of a turbomachine, an external-internal nozzle, or a complete aircraft are recognized as complex. These configurations consist of multiple bodies with nonsmooth, joint or disjoint components. A suitable grid has to accurately represent the boundaries of the configuration and conform to the expected flow. Frequently, constructing a single grid about a complex configuration is either a difficult task or it may result in a skewed mesh, which in turn results in an erroneous solution. Other difficulties may also arise when clustering the grids for the regions where the flow variables and their gradients change rapidly. Several different approaches have been used by researchers to reduce this complexity and ease the grid generation efforts. They include domain decomposition, unstructured grids, and grid adaption.

A domain decomposition technique (DDT) subdivides the flow domain into simpler subdomains which accept easily constructed grids with appropriate grid refinement. Nonsimilar grid topologies, generally independent of each other, can be generated and used to improve grid deficiencies. In addition a DDT allows employment of different solution methods for different subdomains. For example, the physical domain can be divided into two subdomains as nearfield and farfield. The viscous flow equations can be used to investigate the flowfield near the body and the inviscid flow equations for the subdomain far from the body. This may result in savings of both computer time and storage. There are various approaches in subdividing the flow domain and establishing the communication among these subdomains. Multiblock grids, zonal grids, and overlapped grids are widely used methods of domain decomposition.

The multiblock grid method allows any number of blocks to be employed to fill an arbitrary three-dimensional region. Any block can be linked to any other block subject to the constraint that the grid lines are contiguous between blocks, one-to-one continuity in the grid lines. The union of these blocks fills the entire flowfield without holes or overlaps.

The zonal method, also known as grid patching, requires the domain to be divided into subdomains with simple and easily generated grids. These grids are then patched together along common boundaries or surfaces to create a global grid. The grid lines in adjacent grids are not aligned with each other at the grid interface.

In contrast to the zonal and the multiblock methods, the grid overlapping method does not require common boundaries between subdomains. Matching the solution across the boundaries is provided by a common or overlapped region between subdomains. The necessary intergrid communication is maintained through an interpolation procedure within this overlap region. This scheme allows each subdomain to be gridded independently with no restrictions on the continuity of the grid lines across the interfaces. Hence, the efforts in grid generation are reduced immensely. Subdomain grids can be of nonsimilar topologies to match the needs of each subdomain.

Unstructured grids discretize the flow domain by arbitrary, but generally triangular or quadrilateral, elments. The nodes are placed at the vertices of the elements. An unstructured mesh possesses more flexibility in constructing grids for complicated regions of the flowfield than does a structured mesh. However, it requires more computer time and storage, as well as increased efforts in handling the data structure.

In grid adaptive schemes, a dynamically-adaptive grid continually adjusts to the flow and follows the developing gradients in the physical solution. Grid adaption reduces the errors associated with inadequate resolution at regions of high flow gradients and oscillations (such as shocks and boundary layers). This technique combined with domain decomposition may prove to be an important tool in solving complex flow problems.

A DDT increases the flexibility of grid generation for a complex configuration and this in turn enhances the ability to simulate and analyze the flowfield with complicated features. For example, domain decomposition methods make it possible to solve the flow about a store in the proximity of its parent body, in which various shapes and models can be considered for the store and the parent body. This flexibility also allows continued study of the same flowfield as the store moves away from its parent body. However, the biggest contribution of domain decomposition may be the possibility of analyzing and understanding the interference flowfield that exists between multiple bodies. This is important since these types of flows pose enormous challenges for experimental flow techniques. In many cases, for example, Schlieren photographs are not available and vapor screen photographs are also limited in the information that they provide.

A numerical algorithm, which accurately simulates complex flow problems, must be developed with regard to types of geometries and characteristics of flows being considered. The algorithm must possess the ability and the flexibility to consider any arbitrary complex configuration without further burden on available computer resources. The governing equations being solved, the discretization method of these equations, and the difference methods for them are also essential elements in developing of a solution algorithm. Efficiency of the solution algorithm is another desirable feature which must be addressed. The following discussions describe the techniques, equations, and the methods deemed necessary for such an algorithm.

The simplicity of grid generation with domain decomposition techniques allows one to represent the critical regions of a flowfield with very fine meshes. This subsequently increases the computational time required to obtain the solution on these meshes. In addition, the logarithm of the spectral radius for a typical scheme generally varies linearly with the mesh step size, making it impractical to solve the flowfield on very fine grids. The convergence acceleration schemes, such as the multigrid technique (Sec. 3.3), are important tools in offsetting the additional work required due to increased mesh sizes. The multigrid scheme can be very successful in accelerating the rate of convergence particularly for steady state solutions. It is also desirable since it maintains the high level of robustness and accuracy obtained through methods such as upwind differencing.

The multigrid scheme can easily be extended to both multiblock grid generation and the zonal methods. However, it requires special consideration when incorporated into the overlapping technique. These considerations are discussed in detail in Chap. 3. Nevertheless, it is well worth the efforts to implement the multigrid scheme into the overlapping technique due to advantages gained by using a multigrid scheme.

For many of the configurations noted earlier, solving the Euler equations or the potential flow equations might be sufficient for design purposes. However, for detailed analyses of these flows where diffusion effects are not negligible, one is required to solve the Navier-Stokes equations. A notable example is the flow past a cavity. Viscous effects are significant due to the solid boundaries in all directions normal and parallel to the flow stream. The flow is also unsteady, fully turbulent, and vortical. Hence, time dependent, Reynolds-averaged, compressible, complete Navier-Stokes equations are required for an accurate solution.

The viscous effects on the aerodynamics of stores or missiles had been ignored in calculations until recently because of the difficulties involved. The existence of shock waves in the vicinity of the forebody, the development of the asymmetric boundary layer with consequent shedding of vortices, and finally, the formation of asymmetric and turbulent wake due to the base contribute to the complications encountered. However for these cases where the wall exists only in one direction, the thin-layer approximation to the Navier-Stokes equations is proven efficient and effective. In this approximation, the viscous terms containing derivatives parallel to the body surface are neglected.

Compressible flows, where various types of shocks may exist in the flow domain, make up a large portion of the recent fluid dynamics research. Upwind difference methods have been shown to be useful in obtaining a crisp resolution of shocks. These methods using characteristic theory reflect the predominant nature of the governing equations by determining in which direction the information is propagating through the grid cells. They have the advantage of being naturally dissipative. Hence, there is no need for separate spatial dissipation terms to overcome oscillations or instabilities arising in regions of high gradients.

It is also beneficial to discretize the Navier-Stokes equations using a finite volume method. The finite volume formulation begins with the integral equations and satisfies the integral conservation laws. Thus, it remains valid in the presence of large gradients in the flowfield. It should be noted that solution schemes such as the one used here are of shock capturing type and treat the shocks as the regions of high flow gradients. The finite volume discretization allows more flexibility with regard to the geometry and it is tolerant of grid singularities.

1.2 Literature Survey

The following section deals with the pioneering works on domain decomposition methods and their applications to various flow problems. The earlier computational studies on interference flows between multiple bodies are reviewed in Sec. 1.2.2.

1.2.1 Domain Decomposition Methods

A multiblock grid generation code was developed by Thompson [1]* to overcome the difficulty faced by complex geometries. This grid code can operate as an elliptic or as an algebraic grid generation system. The code has been shown to be successful in constructing grids about complex configurations such as three mutually finned bodies, a wing-pylon-store configuration, and a general aircraft configuration. Using multiblock grid generation, Belk and Whitfield [2] solved an unsteady flow about oscillating airfoils and wings. Janus [3] also used this method to investigate the flowfield within rotating geometrical environment. Other works based on this method, such as, Chow and Marconi [4] and Arabshahi and Whitfield [5] are also noted here but will be discussed in Sec. 1.2.2.

Early works on zonal methods were conducted by Hessenius and Pulliam [6] who developed a technique to solve the Euler equations for one– and two–dimensional flows using an implicit scheme. Their scheme required continuity of the mesh in point and slope near the interfacing region. A scheme which does not require the continuity at the interface was later developed by Rai [7]. This scheme is more flexible and uses conservative treatment of zonal boundaries. Later, Hessenius and Rai [8] developed a three-dimensional conservative boundary scheme. This scheme was used to solve flows about a wing-cannard combination using two interfacing patched grids.

The zonal boundary scheme of Ramshaw [9] has been shown to be fully conservative across surfaces with planar interfaces. However, this conservation characteristic has not been shown for surfaces with high curvature. This technique has no restrictions on grid slope and density across the grid interfaces. In a later study, Kathong et al. [10] analyzed the feasibility of the Ramshaw technique with application to a realistic three-dimensional configuration. They showed that global conservation can be maintained for these complex configurations.

* The number in brackets indicate references.

Thomas et al. [11] have also used an implicit, upwind-biased algorithm along with a zonal method for the flow around an F-18 aircraft. In their analysis, it was shown that the difference between the spatial-flux approach and time-flux approach, to maintain conservation across the zonal interfaces were minimal. The time-flux approach appears to be more flexible and can be extended to handle more complex situations, such as overlapped grids.

Unstructured grids are also flexible in constructing grids for complex configurations. However, two major drawbacks of the unstructured grids are their prohibitive need for computer memory and their deficiency in solving the Navier-Stokes equations on highly stretched grids. In an effort to overcome this deficiency, a hybrid method made up of the finite-difference (FDM) and the finite-element (FEM) methods has been developed by Nakahashi and Obayashi [12]. The FDM and the FEM are combined together by using a grid patching technique where the FDM is applied only to the viscous region near the body surface. Other hybridization techniques, such as the adaptive unstructured grid, are also shown to be useful for complex configurations. A survey of grid adaptive schemes in the numerical solution of partial differential equations is given by Thompson [13].

One of the early grid overlapping methods was presented by Atta [14]. He developed a finite difference method to solve the full potential equations and obtained a solution for transonic flow about a configuration with multiple components. This technique was later extended to three-dimensional flows by Atta and Vadyak [15]. This full potential, implicit, approximate factorization algorithm was applied to a wing-pylon-nacelle configuration. Separate grids were employed for each component and the grid adaptive approach was employed to optimize each grid. Each region required a separate implementation of the algorithm with the relevant boundary conditions. The solution procedure iterated several times on a given subdomain, updated other grid boundaries contained in that subdomain and proceeded to the next grid.

Steger et al. [16] developed a grid overlapping technique called “CHIMERA” grid scheme. This early study was restricted to finite difference solution of two-dimensional, linearized flows. However, the subsequent studies were extended to three-dimensional Euler [17] and thin-layer Navier-Stokes [18] solutions of complex flows. The scheme involved the automatic connection of multiple, overlaid grids, and the use of different solution procedures for different subdomain grids. In this technique, a major grid covers the entire flow region, and minor grids are then overlaid on the major grid so as to solve secondary features of the configuration, such as flaps, nacelles, stores, etc. The minor grids are fully or partially overlapped without the restriction that grids be joined in any special way. The minor grids create holes in the major grid which are subsequently excluded from the solution in the major grid. The information between the grids are communicated through trilinear interpolation within the overlap regions. The preliminary studies proved the Chimera grid scheme to be very flexible and successful on various configurations. However, it was shown that difficulties can arise when a shock crosses grid boundaries. The mismatch of solution in the neighborhood of the expansion preceding the shock was attributed to several factors such as the nonconservative interpolation procedure, the reflecting boundary conditions imposed at the overlap, and the meager extent of the overlap in the vicinity of the shock [19].

Another notable study using the Chimera grid scheme is the work of Suhs [20] on three-dimensional cavities. Despite the use of the thin-layer Navier–Stokes equations for cavities, where this assumption is not valid, this study further exemplifies the versatility of the Chimera scheme for complex flow problems. Other works by Dougherty et al. [21,22] and Meakin and Suhs [23] are of some importance to the current study and will be discussed in the following section.

The studies cited in this survey indicate the various approaches that researchers have taken in dealing with complex configurations. The multiblock method has been shown to be conservative at grid interfaces where there is one-to-one continuity of grid lines normal to those interfaces. The zonal method has also been proven to be conservative except across surfaces with high curvature. Ensuring conservation across these surfaces is not theoretically impossible, but it is rather difficult to obtain. The overlapping technique appears to be more flexible for most of complex configurations where objects with nonsimilar geometries exist. It has been noticed that these studies have used only one type of DDT to solve their problem. A better approach can be obtained if the advantages of different DDT's are judiciously combined.

1.2.2 Interference Flows Between Nonsimilar Geometries

The earlier computational studies on interference flows of multiple bodies are presented in this section. These studies are presented in an order corresponding to the governing equations that they have considered.

A fast and inexpensive computer code to predict transonic flowfields about three-dimensional configurations was developed by Kafyeke [24]. This code, which uses a grid embedding technique along with a small perturbation formulation, was utilized to study the flow about a generic wing-body-pylon-store model. The pressure data on each component was compared with the experimental data and the results promised that sensible predictions can be made in the future for more realistic configurations. Although, this code compromises the physics by its underlying assumptions, it provides a cost-effective alternative to predict complex flowfields at design stages.

Löhner [25] solved the problem of a two-dimensional object falling into a supersonic freestream using an adaptive remeshing technique. A finite element grid was used to represent the mesh around the object placed in a cavity. Adaptive remeshing was performed every 100 iterations after a steady-state solution was reached. Use of grid adapting proved beneficial in showing the changes in the strength and the location of shocks due to the motion of the object. In this study, the inviscid flow equations, which are inappropriate for cavity flows, are solved resulting in inaccurate solutions. Also for multiple shock reflections, as in this case, shock-wave/boundary-layer interaction becomes critical and viscous effects dominate in the region behind the first shock reflection.

Dougherty et al. [21] performed computations for flows, where two elliptic surfaces in transonic flow using prescribed paths were simulated. A two-dimensional, inviscid, unsteady code was used in this preliminary study to show the feasibility of the Chimera scheme for a future application to the problem of aircraft/store interaction and separation. In this study, a minor grid was allowed to move with respect to the major grid. The solution obtained indicated that allowing one mesh to move with respect to another does not affect adversely the time accuracy of an unsteady solution despite strong shock interaction between the major and minor bodies. Dougherty and Kuan [22] extended this study to the three-dimensional analysis of an elliptic body near a flat plate. A 10% thick ellipsoid of chord length of unity was placed 15% of chord below a flat plate, both following prescribed flight paths. A steady-state computation of the store and the flat plate was initially performed. After reaching the steady state convergence, the solution was used as the initial condition for the time accurate calculations. A simple downward translation and a pitch up and roll rotation were investigated.

Fox and Allee [26] conducted an experimental/computational study of a transonic aircraft with stores. Flowfield measurement and aircraft surface static pressure measurement were made experimentally on three different configurations to find aerodynamic forces on a pylon-mounted store. The Euler calculation using the grid overlapping methods of Refs. [16–19] were performed to support the wind tunnel measurements. Both experimental and computational tests concluded that the dominant effect on the store is the gross sidewash of the vehicle distributed along the length of the store, causing yawing forces in the direction that would move the nose of the store away from the centerline of the fuselage. In addition, this study proved that a solution of the Euler equations over such a complex configuration has become practical, both for pre-test prediction and post-test analysis.

Arabshahi and Whitfield [5] utilized the multiblock grid generation of Ref. [1] to develop a numerical approach to solve the unsteady Euler equations for three-dimensional flowfields around complex configurations. This approach used an implicit, approximately factored, finite volume algorithm to solve the Euler equations about a wing-pylon-store configuration at 0° angle of attack. The study first compared the surface pressure coefficient obtained using the steady-state solution with the experimental data. Then, unsteady dynamic multiblock solutions were demonstrated by computing the flow about the complete multibody configuration as the store moved away from the parent wing-pylon configuration through a prescribed vertical launch trajectory.

Meakin and Suhs [23] developed a prototype method for time-accurate simulation of multiple aerodynamic bodies in relative motion. This method featured the unsteady Chimera domain decomposition technique and an implicit, approximately factored, finite difference procedure to solve the time dependent thin-layer Navier-Stokes equations. Several steady and unsteady configurations were considered as test cases and two configurations were used to demonstrate the prototype method. First, a body was released from its position below a wing. Then, the separation of an integrated space shuttle from its solid rocket booster was considered. The converged steady solution for both configurations were obtained and used as the initial conditions. In both cases, the released objects were moved in prescribed paths.

Chow and Marconi [4] obtained a three-dimensional Navier-Stokes solution of hypersonic flow past a cone-cylinder-cone in the close proximity of a flat plate. The interference flowfield between these bodies was analyzed in detail. The study used an implicit, approximate factorization, central difference algorithm. They performed a steady-state prediction of the flowfield at each instant of the separation process. (They assumed that the flowfield adjusts quickly during the separation making the steady-state computations at each instant acceptable.)

The earlier computational investigations on interference flows have been reviewed in this section. However, one experimental study should also be included here since it is of importance to the present work. The experimental study of Stallings et al. [27] is used for code validation purposes herein. In addition, the result of this experimental study provides useful information to understand the complex fluid phenomena encountered in an internal store carriage and separation. This experimental study was performed to measure the effects of a store when placed inside or near a cavity. Wall surface pressures on the store and the cavity, and the aerodynamic forces and moments on the store were measured. The study considered different locations of the store relative to the floor as well as different depths for the cavity. The test matrix also included various freestream Mach numbers. Surface pressure measurements on the store and the cavity were obtained using electronically scanned pressure transducers, referenced to a vacuum. Hence, a steady measurement technique was used for this unsteady flow by time averaging over 2 to 3 seconds. The measurements were taken on one half of the cavity-store configuration assuming symmetrical pressure gradients with respect to the centerline of the configuration. This assumption is based on the findings of the previous experimental study by Stallings and Wilcox [28]. Reference [28] is the report on the experimental investigation of supersonic flows about clean cavities, i.e. without the store, with various length-to-depth ratios. The study showed the flow about a deep cavity is nearly symmetric with respect to the center plane of the cavity. This conclusion is also the impetus for the reflecting boundary conditions assumption described in Sec. 2.6 of the present study.

After reviewing relevant literature, it can easily be said that the interference flows have not been studied extensively. These investigations generally bear very restrictive assumptions. Hence, there appears to be an urgent need for further and more comprehensive investigations, using both experimental and computational methods. In particular, the interference flows encountered in internal store carriage and separation have not been computationally investigated to this date. The present study represents the first successful Navier-Stokes simulation of the flow past a cylinder-sting-cavity configuration. The improvement to the overlapping technique, hybridization of DDT's, and the results presented in this study, should provide a clearer path to the efforts in simulating the unsteady and complex flow problem of internal store carriage and separation.

1.3 Present Work

The primary objectives of the present work are:

(A) to develop a CFD algorithm which is capable of obtaining solutions for flows about complex configurations,

(B) to simulate the interference flow past a cylinder-sting-cavity (CSC) assembly.

Objective A is accomplished by:

(1) using domain decomposition techniques to decompose the flow domains into simpler subdomains,

(2) solving the Navier-Stokes equations on each subdomain,

(3) implementing an implicit, approximately-factored, upwind-biased, finite volume scheme modified to accommodate the overlapped grids,

(4) incorporating a multigrid scheme modified to serve on the overlapped grids.

Objective B is accomplished by:

(1) following the findings of Refs. [29–33] for a complete analysis of flows past clean cavities,

(2) investigating the flows over cylindrical profiles and a body-of-revolution at various angles of attack,

(3) testing the use of overlapped grids on a relatively simpler geometry, namely a blunt-nose-cylinder,

(4) choosing two different configurations, specifically a cylinder near and inside a cavity, to represent two distinct interference flows which are encountered in an internal store carriage and separation problem.

The organization of this dissertation is as follows. Chapter 2 conveys the basic formulation. The governing equations and corresponding initial and boundary conditions are given in this chapter. The baseline solution algorithm and the turbulence model are also discussed in Chap. 2. The grid overlapping technique is described in Chap. 3. Improvements and the modifications done to the baseline solution algorithm are explained in this chapter. A discussion on the multigrid convergence acceleration scheme for the overlapped grids is also included in Chap. 3. The results are discussed in Chap. 4. The conclusions and the recommendations are given in Chap. 5.

Chapter 2

BASIC FORMULATION

2.1 Governing Equations

The governing equations are the time dependent, Reynolds-averaged, compressible, full Navier-Stokes equations written in conservative form and generalized curvilinear coordinates (ξ, η, ζ) as [34],

$$\frac{\partial Q}{\partial t}+\frac{\partial(F-F_v)}{\partial\xi}+\frac{\partial(G-G_v)}{\partial\eta}+\frac{\partial(H-H_v)}{\partial\zeta}=0 \tag{2.1}$$

where the conserved variables of vector Q are defined as:

$$Q=\frac{1}{J}\left\{\begin{array}{c}\rho\\ \rho u\\ \rho v\\ \rho w\\ e\end{array}\right\} \tag{2.2}$$

The inviscid flux vectors F, G, and H are:

$$F=\frac{1}{J}\left\{\begin{array}{c}\rho U\\ \rho Uu+\xi_x P\\ \rho Uv+\xi_y P\\ \rho Uw+\xi_z P\\ (e+P)U\end{array}\right\} \tag{2.3}$$

$$G=\frac{1}{J}\left\{\begin{array}{c}\rho V\\ \rho Vu+\eta_x P\\ \rho Vv+\eta_y P\\ \rho Vw+\eta_z P\\ (e+P)V\end{array}\right\} \tag{2.4}$$

$$H = \frac{1}{J}\left\{\begin{array}{c} \rho W \\ \rho W u + \zeta_x P \\ \rho W v + \zeta_y P \\ \rho W w + \zeta_z P \\ (e+P) W \end{array}\right\} \tag{2.5}$$

where J is the transformation Jacobian,

$$J = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} \tag{2.6}$$

and U, V, and W are the contravariant velocities defined as

$$U = \xi_x u + \xi_y v + \xi_z w \tag{2.7}$$

$$V = \eta_x u + \eta_y v + \eta_z w \tag{2.8}$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w \tag{2.9}$$

The viscous flux vectors F_v, G_v, H_v of the governing equations are defined as

$$F_v = \frac{1}{J}\left\{\begin{array}{c} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} + \xi_z \tau_{xz} \\ \xi_x \tau_{xy} + \xi_y \tau_{yy} + \xi_z \tau_{yz} \\ \xi_x \tau_{xz} + \xi_y \tau_{yz} + \xi_z \tau_{zz} \\ \xi_x b_x + \xi_y b_y + \xi_z b_z \end{array}\right\} \tag{2.10}$$

$$G_v = \frac{1}{J}\left\{\begin{array}{c} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} + \eta_z \tau_{xz} \\ \eta_x \tau_{xy} + \eta_y \tau_{yy} + \eta_z \tau_{yz} \\ \eta_x \tau_{xz} + \eta_y \tau_{yz} + \eta_z \tau_{zz} \\ \eta_x b_x + \eta_y b_y + \eta_z b_z \end{array}\right\} \tag{2.11}$$

$$H_v = \frac{1}{J}\begin{Bmatrix} 0 \\ \zeta_x\tau_{xx} + \zeta_y\tau_{xy} + \zeta_z\tau_{xz} \\ \zeta_x\tau_{xy} + \zeta_y\tau_{yy} + \zeta_z\tau_{yz} \\ \zeta_x\tau_{xz} + \zeta_y\tau_{yz} + \zeta_z\tau_{zz} \\ \zeta_x b_x + \zeta_y b_y + \zeta_z b_z \end{Bmatrix} \tag{2.12}$$

The shear stress, τ, and the heat flux , $\dot{q}$, terms are written in indicial notations

$$\tau_{x_i x_j} = \frac{M}{Re_l}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{2.13}$$

$$b_{x_i} = u_j \tau_{x_i x_j} - \dot{q}_{x_i} \tag{2.14}$$

$$\dot{q}_{x_i} = \left(\frac{-M_\infty \mu}{Re_l Pr(\gamma - 1)}\right)\frac{\partial c^2}{\partial x_i} \tag{2.15}$$

The pressure is related to the conserved variables through the ideal gas law

$$P = (\gamma - 1)\left[e - \rho\left(u^2 + v^2 + w^2\right)/2\right] \tag{2.16}$$

For the sake of simplicity, molecular viscosity is calculated from the Sutherland's law and Stokes' hypothesis for bulk viscosity, λ+(2/3μ) = 0, is invoked. The Sutherland's law relates μ to the temperature, T, through

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2}\frac{T_\infty + s}{T + s} \tag{2.17}$$

In Eq. (2.1), the conserved variables of vector Q (Eq. (2.2)) are normalized with the freestream values of density and the local speed of sound

$$\rho = \frac{\tilde{\rho}}{\tilde{\rho_\infty}} \; ; \; u = \frac{\tilde{u}}{\tilde{a_\infty}} \; ; \; v = \frac{\tilde{v}}{\tilde{a_\infty}} \; ; \; w = \frac{\tilde{w}}{\tilde{a_\infty}} \; ; \; e = \frac{\tilde{e}}{\tilde{a_\infty}^2\tilde{\rho_\infty}} \; ; \tag{2.18}$$

Also used in the non-dimensionalization are the reference length L and the freestream molecular viscosity, μ_∞.

2.2 Thin-Layer Approximation

The contribution of some of the terms in Eq. (2.1) in mathematically describing the flow may be marginal for certain types of flows. For example, the viscous terms containing derivatives parallel to the body surface may be neglected for the flows where the wall exists only in one direction. The derivatives parallel to the body surface of the viscous terms yield much smaller values than those normal to it. This formulation is known as the Thin-Layer Approximation. The validity of the thin–layer viscous model for flows with massive crossflow separation has also been assessed in Ref. [35], by comparing the results obtained with and without the circumferential viscous terms and viscous cross-derivative terms. It was found that the inclusion of these terms improved the solution slightly (less than 0.5%), but increased the computation time by 11%.

Hence, an option is built in the computer code to invoke the thin-layer approximation in any direction (ξ, η, or ζ) for such cases. This is particularly beneficial when incorporated with domain decomposition techniques, since different solution algorithms can be used for different subdomains. However, all of the viscous terms in the Navier-Stokes equations are essential when solving complex flows such as a flow over a cavity. The existence of walls (body surfaces) in all directions (ξ, η, and ζ) requires the inclusion of viscous terms and the cross derivative terms in all direction. For such cases, the option for full Navier-Stokes equations is invoked.

2.3 Finite Volume Formulation

Equation (2.1) is solved using the finite volume method. The finite volume (FV) discretization bears some resemblance to the conventional finite difference (FD) discretization. Its difference stencil is that of the FD scheme. However, its formulation begins with the integral equations. The solution obtained using this formulation satisfies the integral conservation law of mass, momentum, and energy. Hence, it remains valid in the presence of large flow gradients in viscous flowfields such as shocks and contact surfaces. In FV discretization, the computational domain is divided into a large number of "non-overlapping" cells. Each computational cell is constructed of eight grid points as its vertex points. The finite volume scheme only requires a cell to be a hexahedron, hence it is well suited for arbitrary configurations. Also, this allows the FV scheme to have a greater tolerance of grid singularities, such as collapsing of a surface to a line or a point. There is no ambiguity at grid singularities since the flow equations are balanced only within the cells of the grid and not at the nodal points. A detailed analysis of finite volume and finite difference formulation is presented in Ref. [36].

Equation (2.1) is expressed in integral form as

$$\frac{\partial}{\partial t}\iiint\limits_{V} Q\,dV + \iint\limits_{S} \vec{F}\,.\,\vec{n}\,dS = 0 \tag{2.19}$$

where $\vec{F} = \left(\vec{F} - \vec{F}_v\right)\vec{i} + \left(\vec{G} - \vec{G}_v\right)\vec{j} + \left(\vec{H} - \vec{H}_v\right)\vec{k}$. Surface integrals are written as the sum of the contributions from six faces of the hexahedron cell. Equation (2.19) is applied to each cell of the computational domain and the resulting semi-discrete finite volume representation is

$$\left(\frac{\partial Q}{\partial t}\right)_{i,j,k} + (F - F_v)_{i+1/2,j,k} - (F - F_v)_{i-1/2,j,k} + \tag{2.20}$$

$$(G - G_v)_{i,j+1/2,k} - (G - G_v)_{i,j-1/2,k} \tag{2.21}$$

$$(H - H_v)_{i,j,k+1/2} - (H - H_v)_{i,j,k-1/2} = 0 \tag{2.22}$$

where $\Delta\xi = \Delta\eta = \Delta\zeta = 1$. It should be noted that $Q_{i,j,k}$ represents the conserved variables at the cell center (cell averaged) and the fluxes are evaluated at the cell surfaces.

In the present method, the ratio of a metric derivative to the transformation Jacobian for a given cell, for example (ξ_x/J) , is taken to be the appropriate projected area of a cell face, and the Jacobian is the reciprocal of the cell volume. The volume of the cell is determined as the sum of six pentahedra. Each pentahedron is defined by one of the six cell faces and the average point in the volume. The magnitude of the directed area and one of its direction cosines at a cell interface are computed, for example at fixed ξ–direction and in the x-direction as $(|grad\,\xi/J|)$ and $(-\xi_x/\,|grad\,\xi|)$, respectively.

2.4 Factorization

Applying the Euler implicit time integration to Eq. (2.1) yields

$$\frac{\Delta Q}{\Delta t} + \delta_\xi (F - F_v)^{n+1} + \delta_\eta (G - G_v)^{n+1} + \delta_\zeta (H - H_v)^{n+1} = 0 \tag{2.23}$$

where n denotes the time level. Fluxes can also be linearized as

$$\delta_\xi (F - F_v)^{n+1} = \delta_\xi (F - F_v)^n + \delta_\xi \frac{\partial (F - F_v)^n}{\partial Q} \Delta Q \tag{2.24}$$

$$\delta_\eta (G - G_v)^{n+1} = \delta_\eta (G - G_v)^n + \delta_\eta \frac{\partial (G - G_v)^n}{\partial Q} \Delta Q \tag{2.25}$$

$$\delta_\zeta (H - H_v)^{n+1} = \delta_\zeta (H - H_v)^n + \delta_\zeta \frac{\partial (H - H_v)^n}{\partial Q} \Delta Q \tag{2.26}$$

Putting Eqs. (2.24–2.26) into Eq. (2.23) results in the following expression

$$\begin{aligned} &\left[\frac{I}{\Delta t} + \delta_\xi \frac{\partial (F - F_v)}{\partial Q} + \delta_\eta \frac{\partial (G - G_v)}{\partial Q} + \delta_\zeta \frac{\partial (H - H_v)}{\partial Q}\right]^n \Delta Q \\ &= - \left[\delta_\xi (F - F_v) + \delta_\eta (G - G_v) + \delta_\zeta (H - H_v)\right]^n = -R^n \end{aligned} \tag{2.27}$$

Equation (2.27) forms a system of algebraic equations. The coefficient matrix is banded and the band width depends on the grid size and choice of spatial differencing method. The left hand side requires an inversion of a very large matrix. The exact inversion of the matrix is very costly due to the large number of operations and computer memory required. An approximate factorization scheme (AF) splits the implicit operator into a sequence of easily inverted equations. The spatial-split, three factor approximate factorization of Beam and Warming [37] applied to Eq. (2.27) yields

$$\left[\frac{I}{\Delta t}+\delta_\xi\frac{\partial(F-F_v)}{\partial Q}\right]^n\times\left[\frac{I}{\Delta t}+\delta_\eta\frac{\partial(G-G_v)}{\partial Q}\right]^n\times$$
$$\left[\frac{I}{\Delta t}+\delta_\zeta\frac{\partial(H-H_v)}{\partial Q}\right]^n\times\Delta Q\simeq -R^n\times(\Delta t)^2=-(R^*)^n \tag{2.28}$$

or

$$\left[\frac{I}{\Delta t}+\delta_\xi\frac{\partial(F-F_v)}{\partial Q}\right]^n\Delta Q^{**} = -(R^*)^n \tag{2.29}$$

$$\left[\frac{I}{\Delta t}+\delta_\eta\frac{\partial(G-G_v)}{\partial Q}\right]^n\Delta Q^{*} = \left(\frac{I}{\Delta t}\right)\Delta Q^{**} \tag{2.30}$$

$$\left[\frac{I}{\Delta t}+\delta_\zeta\frac{\partial(H-H_v)}{\partial Q}\right]^n\Delta Q = \left(\frac{I}{\Delta t}\right)\Delta Q^{*} \tag{2.31}$$

$$Q^{n+1}=Q^n+\Delta Q \tag{2.32}$$

This scheme requires the solution of a 5×5 block tridiagonal system. It has the advantage of being fully vectorizable. The present implementation takes advantage of the large memory available in the current supercomputers by creating long data vectors which, in turn, results in faster processing rates.

2.5 Upwind Differencing

The inviscid fluxes and the time term of the Navier-Stokes equations form a system of hyperbolic equations. Upwind space discretization methods [38–40] reflect the predominant nature of these equations by determining in which direction the information is propagating through the grid cells dictated by the characteristic theory. The dissipative nature of upwind methods have made them a prevailing alternative to the central difference schemes, where artificial dissipation terms are generally needed to overcome oscillations or instabilities arising in regions of high gradients.

Various upwind methods such as the λ scheme [41], split coefficient method [42], flux-vector splitting [43, 44], and flux-difference splitting [45–47] have been developed and widely used. In this study, Roe's flux-difference [47] splitting is used to construct the upwind differences for the convective and pressure terms. However, the diffusive terms are centrally differenced.

2.5.1 Roe Flux-Difference Splitting

According to the upwind differencing proposed by Roe [47], the neighboring cells interact through discrete, finite-amplitude waves. The amplitude and propagation speed of these waves are determined by solving the approximate Reimann problem. The flux-difference splitting (FDS) of Roe seeks the exact solution to this approximate equation and distinguishes between the influence of the forward and backward moving waves. In the FDS method, the Reimann problem is approximated as

$$\frac{\partial Q}{\partial t} + \tilde{A}\left(Q^L, Q^R\right)\frac{\partial Q}{\partial \xi} = 0$$
$$Q(\xi, t=0) = \begin{Bmatrix} Q^L & for & \xi < 0 \\ Q^R & for & \xi > 0 \end{Bmatrix} \tag{2.33}$$

where $\tilde{A}\left(Q^L, Q^R\right)$ is a constant Jacobian matrix, $\frac{\partial F}{\partial Q}$. For flux terms, a monotone upstream centered scheme for conservation laws (MUSCL) approach is used. For example, in the ξ–direction, the spatial derivative is written conservatively as a flux balance across a cell as:

$$\left(\frac{\partial F}{\partial \xi}\right)_i = \left(F_{i+1/2} - F_{i-1/2}\right) / \left(\xi_{i+1/2} - \xi_{i-1/2}\right) \tag{2.34}$$

where, for example, the interface flux at (i + 1/2) is constructed as

$$F_{i+1/2} = 1/2\left[F\left(q^L\right) + F\left(q^R\right) - |\tilde{A}|\left(Q^R - Q^L\right)\right]_{i+1/2} \tag{2.35}$$

The term $|\tilde{A}|\left(Q^R - Q^L\right)$ is the dissipation term contributing to the interface flux and is detailed in Eqs. (2.38–2.52). The cell center and the cell interface are referred to by (i) and (i+1/2), respectively. The parameters Q^L and Q^R denote the state variables to the left and right of the cell interfaces and are determined from upwind-biased interpolations of the primitive variables $q = [\rho, u, v, w, P]^T$ where first-order differencing is given by $(q^L)_{i+1/2} = q_i$ and $(q^R)_{i+1/2} = q_{i+1}$. Higher order accurate differencing is defined by a one-parameter family

$$\left(q^L\right)_{i+1/2} = q_i + \left\{\frac{1}{4}\left[(1-\kappa)\,\nabla + (1+\kappa)\,\Delta\right]\right\}_i \tag{2.36}$$

$$\left(q^R\right)_{i+1/2} = q_{i+1} - \left\{\frac{1}{4}\left[(1+\kappa)\,\nabla + (1-\kappa)\,\Delta\right]\right\}_{i+1} \tag{2.37}$$

where

$$(\Delta)_i = q_{i+1} - q_i \quad , \quad (\nabla)_i = q_i - q_{i-1} \tag{2.38}$$

and

$$\kappa = \begin{cases} 1 & , \; central\ difference \\ \frac{1}{3} & , \; third\text{-}order\ upwind\text{-}biased \end{cases}$$

Some form of flux limiting algorithm can be employed at this point to capture the large flow gradients without the numerical oscillations. The implementation of several different flux limiting algorithms is discussed in Sec. 2.5.3. The dissipation term contributions to the interface flux, $|\tilde{A}|\left(Q^R - Q^L\right)$, can be written as

$$|\tilde{A}|\left(Q^R - Q^L\right) = |\tilde{A}|\;\Delta\,Q = \begin{bmatrix} a_4 \\ \tilde{u}a_4 + k_x a_5 + a_6 \\ \tilde{v}a_4 + k_y a_5 + a_7 \\ \tilde{w}a_4 + k_z a_5 + a_8 \\ \tilde{H}a_4 + Ua_5 + \tilde{u}a_6 + \tilde{v}a_7 + \tilde{w}a_8 - \left(\frac{\tilde{C}^2}{\gamma-1}\right) a_1 \end{bmatrix} \tag{2.38}$$

where

$$a_1 = \left|\frac{grad(\xi)}{J}\right| \left|\tilde{U}\right| \left(\Delta\rho - \frac{\Delta P}{\tilde{C}^2}\right) \tag{2.39}$$

$$a_2 = \frac{1}{2\tilde{C}^2}\left|\frac{grad(\xi)}{J}\right|\left|\tilde{U}+\tilde{C}\right|\left(\Delta P + \tilde{\rho}\tilde{C}\,\Delta\,U\right) \tag{2.40}$$

$$a_3 = \frac{1}{2\tilde{C}^2}\left|\frac{grad(\xi)}{J}\right|\left|\tilde{U}-\tilde{C}\right|\left(\Delta P - \tilde{\rho}\tilde{C}\,\Delta\,U\right) \tag{2.41}$$

$$a_4 = a_1 + a_2 + a_3 \tag{2.42}$$

$$a_5 = \tilde{C}\ (a_2 - a_3) \tag{2.43}$$

$$a_6 = \left|\frac{grad(\xi)}{J}\right|\left|\tilde{U}\right|(\tilde{\rho}\,\Delta\,u - k_x\tilde{\rho}\,\Delta\,U) \tag{2.44}$$

$$a_7 = \left|\frac{grad(\xi)}{J}\right|\left|\tilde{U}\right|(\tilde{\rho}\,\Delta\,v - k_y\tilde{\rho}\,\Delta\,U) \tag{2.45}$$

$$a_8 = \left|\frac{grad(\xi)}{J}\right|\left|\tilde{U}\right|(\tilde{\rho}\,\Delta\,w - k_z\tilde{\rho}\,\Delta\,U) \tag{2.46}$$

The symbol ($\sim$) indicates Roe-averaged evaluations

$$\tilde{\rho} = \sqrt{\rho_L\ \rho_R} \tag{2.47}$$

$$\tilde{u} = \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)\ /\ \left(1+\sqrt{\rho_R/\rho_L}\right) \tag{2.48}$$

$$\tilde{v} = \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)\ /\ \left(1+\sqrt{\rho_R/\rho_L}\right) \tag{2.49}$$

$$\tilde{w} = \left(w_L + w_R\sqrt{\rho_R/\rho_L}\right)\ /\ \left(1+\sqrt{\rho_R/\rho_L}\right) \tag{2.50}$$

$$\tilde{H} = \left(H_L + H_R\sqrt{\rho_R/\rho_L}\right)\ /\ \left(1+\sqrt{\rho_R/\rho_L}\right) \tag{2.51}$$

$$\tilde{C}^2 = (\gamma - 1)\left[H - (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2)/2\right] \tag{2.52}$$

The magnitude of the cell interface directed area is $|grad(\xi)/J|$ and the direction cosines are

$$(k_x, k_y, k_z) = -(\xi_x, \xi_y, \xi_z)/\,|grad(\xi)| \tag{2.53}$$

where U is the contravariant velocity normal to the cell interface.

2.5.2 Inversion of Approximate Diagonalization

Roe's flux-difference splitting scheme allows spatial factors in each direction to be approximated with a diagonal inversion [48]. This results in a considerable saving in the computational time and can be used to advance the solution past the initial numerical transient stage.

The Jacobian matrix is split through a similarity transformation as

$$A = \frac{\partial F}{\partial Q} = T\Lambda T^{-1} = T\left(\Lambda^{+} + \Lambda^{-}\right)T^{-1} \tag{2.54}$$

where

$$\Lambda^{\pm} = \frac{\Lambda \pm |\Lambda|}{2} \tag{2.55}$$

T and T^{-1} are the diagonalizing matrices of the right eigenvectors of A and its inverse, respectively. The diagonal matrix Λ is the matrix of eigenvalues of A, i.e.

$$\Lambda = diag\left(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\right) \tag{2.56}$$

$$\lambda_1 = \lambda_2 = \lambda_3 = \frac{\tilde{U}\,|\nabla\xi|}{J} \tag{2.57}$$

$$\lambda_4 = \frac{\left(\tilde{U} + \tilde{a}\right)|\nabla\xi|}{J} \tag{2.58}$$

$$\lambda_5 = \frac{\left(\tilde{U} - \tilde{a}\right) |\nabla \xi|}{J} \tag{2.59}$$

where U is the contravariant velocity normal to the cell interface U = (ξ_x u + ξ_y u + ξ_z w)/$|\nabla \xi|$. The spatial factor, for example in the ξ–direction, can be approximated as

$$\left[\frac{I}{\Delta t} + \delta_\xi \frac{\partial F}{\partial Q}\right]^n \triangle Q^* \approx T \left[\frac{I}{\Delta t} + \delta_\xi^- \Lambda^+ + \delta_\xi^+ \Lambda^-\right] T^{-1} \triangle Q^* \tag{2.60}$$

Then, Eq. (2.29) becomes,

$$\left[\frac{I}{\Delta t} + \delta_\xi^- \Lambda^+ + \delta_\xi^+ \Lambda^-\right] \left(T^{-1} \triangle Q^{**}\right) = -T^{-1} R^* \tag{2.61}$$

Due to the repeated eigenvalues ($\lambda_1 = \lambda_2 = \lambda_3$), only three scalar tridiagonal LU decompositions are required for each line. The tridiagonal matrix can be written as

$$\begin{aligned} &- \Lambda^+ \left(M_{i-1/2}, Q_{i-1}\right) \left(T^{-1} \triangle Q^{**}\right)_{i-1} \\ &+ \left[\frac{I}{\Delta t} \Lambda^+ \left(M_{i+1/2}, Q_i\right) - \Lambda^- \left(M_{i-1/2}, Q_i\right)\right] \left(T^{-1} \triangle Q^{**}\right)_i \\ &+ \Lambda^- \left(M_{i+1/2}, Q_{i+1}\right) \left(T^{-1} \triangle Q^{**}\right)_{i+1} = -T_i^{-1} R_i^* \end{aligned} \tag{2.62}$$

the metric terms M and the state variables Q are evaluated at the cell interfaces and the cell center, respectively. It should be noted that the error introduced by the diagonal approximation is first-order in time. The spatial accuracy of the standard block or diagonalized algorithm for steady-state problems (i.e., where ΔQ goes to zero) is dependent upon the type of differencing used in forming $(R^*)^n$. Since the modification that produces the diagonal algorithm does not affect $(R^*)^n$ both schemes will have the steady-state solution [48].

2.5.3 Flux Limiters

Numerical oscillations such as overshoots and undershoots are expected to appear in the presence of large flow gradients when an upwind-biased scheme is used. A flux limiter can be used to reduce the upwind-biased scheme to a fully one-sided upwind scheme in these regions. This in turn ensures a monotonic interpolation and eliminates the overshoots the undershoots.

The first limiter considered for this study is called the minimum-modulus (min-mod) limiter [49]. The upwind-biased interpolation (Eqs. (2.36), (2.37)) is rewritten as

$$\left(q^L\right)_{i+1/2} = q_i + \left\{\frac{1}{4}\left[(1-\kappa)\bar{\nabla} + (1+\kappa)\bar{\Delta}\right]\right\}_i \tag{2.63}$$

$$\left(q^R\right)_{i+1/2} = q_{i+1} - \left\{\frac{1}{4}\left[(1+\kappa)\bar{\nabla} + (1-\kappa)\bar{\Delta}\right]\right\}_{i+1} \tag{2.64}$$

where

$$\begin{aligned} \bar{\Delta} &= max\left[0, min\left(\Delta sgn\nabla, \beta\, \nabla\; sgn\Delta\right)\right] sgn\Delta \\ \bar{\nabla} &= max\left[0, min\left(\nabla sgn\Delta, \beta\, \Delta\; sgn\nabla\right)\right] sgn\nabla \\ \beta &= \frac{(3-\kappa)}{(1-\kappa)} \end{aligned} \tag{2.65}$$

The min-mod limiter is activated only in regions of high gradients and it has been shown to cause a limit cycle in the residual on the fine grids [50].

The second limiter acts in a continuously differentiable fashion, and it has the advantage of overcoming the convergence problem encountered with the first limiter. This limiter known as a Van Albada [51] type limiter is implemented by rewritting the Eqs. (2.36–2.37) as

$$\left(q^L\right)_{i+1/2} = q_i + \left\{\frac{S}{4}\left[(1-\kappa S)\,\nabla + (1+\kappa S)\,\Delta\right]\right\}_i \tag{2.66}$$

$$\left(q^R\right)_{i+1/2} = q_{i+1} - \left\{\frac{S}{4}\left[(1+\kappa S)\nabla + (1-\kappa S)\Delta\right]\right\}_{i+1} \tag{2.67}$$

where

$$S = \frac{2\nabla\Delta + \epsilon}{\Delta^2 + \nabla^2 + \epsilon} \tag{2.68}$$

and ϵ is a small number to prevent the division by zero in the regions of zero gradient. The quantity S is the limiter control function. Its value reaches unity at the regions of small gradients, thus using no limiter for these regions. However, its value goes to zero for very large gradients regions, hence reducing to a first-order interpolation.

The third limiter considered is due to Spekreijise [52] and it invokes monotonicity at the interpolation stage by rewritting the Eqs. (2.36–2.37) as

$$\left(q^L\right)_{i+1/2} = \left[q + \frac{1}{2}\ \varphi(R)\ \ \phi(R)\ \ \nabla\right]_i \tag{2.69}$$

$$\left(q^R\right)_{i-1/2} = \left[q - \frac{1}{2}\ \varphi\left(\frac{1}{R}\right)\ \ \phi\left(\frac{1}{R}\right)\ \ \Delta\right]_i \tag{2.70}$$

where

$$\varphi(R) = \frac{1}{3} + \frac{2}{3}R\ ;\ \ R = \frac{\Delta q}{\nabla q}\ ;\ \ \phi(R) = \frac{3R}{2R^2 - R + 2} \tag{2.71}$$

The comparison among the aforementioned limiter algorithms and judgment on the merits and performances of each algorithm is outside the objectives of the present study. Hence, no effort is made here to justify one algorithm over the others. However, based on the limited experience of the author on this matter, it is evident that performance of the limiters depend on the characteristics of the flowfield considered. That is, while one limiter may perform well for one case, it may not do as well as the others in another case.

2.6 Initial and Boundary Conditions

The accuracy of a CFD solution is bounded by the proper choices of initial and boundary conditions. The initial conditions should correspond to the physical nature of the flow. In practice, the initial conditions are chosen to be the freestream conditions or the best approximate solution obtained from experiments, empirical relations, theories, or previous computational results. In either case, the initial conditions must be chosen prudently since the proper initial guess will result in considerable savings in computational run time.

For a flow past a simple geometry, such as a cylinder, the flow domain is initialized with the freestream conditions. However, the initialization process is different for a cavity. Since the flow is two-dimensional on the flat plate ahead of a cavity, a two-dimensional, turbulent boundary layer profile is generated to match the experimentally determined thickness. The entire flowfield above a cavity is initialized with this profile. The flow inside a cavity is shown to be slower. Hence, the initial values for the points inside the cavity are set for a very slow streamwise flow.

The initialization process is changed somewhat for configurations with more complex geometries such as a cylinder in the proximity of a cavity. First, a steady state calculation of the flow past a cylinder is performed. Meanwhile, a time-accurate calculation is performed for a clean cavity. The solutions of the cylinder and the cavity are then combined and used as the initial conditions for the flow past a cylinder near a cavity. When the cylinder is connected to an offset sting, a solution for the sting is also obtained, separately. For the flow past the cylinder inside the cavity, all points in the cylinder grid are initialized with values of the nearest grid point from the clean cavity solution.

The boundary conditions are specified explicitly. The walls are considered to be impermeable and adiabatic. The pressure at the solid surfaces is evaluated using a zeroth-order extrapolation from the interior point value. The density is then calculated by employing the state equation. Consequently

$$u = 0 \, , \, v = 0 \, , \, w = 0 \, , \, \frac{\partial T}{\partial n} = 0 \, , \, \frac{\partial P}{\partial n} = 0 \tag{2.72}$$

The locally one-dimensional characteristic boundary conditions are employed for the farfield boundaries. For each cell, the velocity normal to the boundary and the speed of sound are calculated from the two locally one-dimensional Riemann invariants given by

$$R^{\pm} = U \pm \frac{2}{\gamma - 1} a \tag{2.73}$$

The invariants are constant along the characteristics defined by

$$\left(\frac{d\xi}{dt} \right)^{\pm} = U \pm a \tag{2.74}$$

The local normal velocity at the boundary is calculated by summing the two Riemann invariants and the speed of sound by subtracting the two. The appropriate boundary conditions are specified after the magnitude of the contravariant Mach number (U/a) and the direction of the contravariant velocity at each cell are checked.

$$|M_\xi| > 1 \; ; \; U < 0 \qquad u = u_\infty \; , \; v = v_\infty \; , \; w = w_\infty \; , \; T = T_\infty \; , \; P = P_\infty \tag{2.75}$$

$$|M_\xi| > 1 \; ; \; U > 0 \qquad \frac{\partial u}{\partial \xi} = 0 \, , \, \frac{\partial v}{\partial \xi} = 0 \, , \, \frac{\partial w}{\partial \xi} = 0 \, , \, \frac{\partial T}{\partial \xi} = 0 \, , \, \frac{\partial P}{\partial \xi} = 0 \tag{2.76}$$

where ξ is the streamwise coordinate.

$$|M_\xi| < 1 \; ; \; U < 0 \qquad u = u_\infty \; , \; v = v_\infty \; , \; w = w_\infty \; , \; T = T_\infty \; , \; \frac{\partial P}{\partial \xi} = 0 \tag{2.77}$$

$$|M_\xi| < 1 \; ; \; U > 0 \qquad \frac{\partial u}{\partial \xi} = 0 \, , \, \frac{\partial v}{\partial \xi} = 0 \, , \, \frac{\partial w}{\partial \xi} = 0 \, , \, \frac{\partial T}{\partial \xi} = 0 \, , \, P = P_\infty \tag{2.78}$$

For supersonic inflow, all flow characteristics point from the outside toward the inside of the computational domain. Hence, the inflow boundaries are set equal to the freestream conditions (Eq. 2.75) except for the flat plate and cavity. In these cases, all elements of the primitive variables have been specified by a profile generated using the boundary layer equations. A more detailed description of this condition is given in Ref. [33].

$$u = u_{blp}\ ,\ v = 0\ ,\ w = w_{blp}\ ,\ T = T_{blp}\ ,\ P = P_\infty \tag{2.79}$$

where the subscript (blp) denotes the value given by the boundary layer profile.

In the case of supersonic outflow, the flow signals propagate from the inside toward the outside. Therefore, the flow variables at the outflow boundaries are computed from the interior flow solution by a zeroth-order extrapolation ((Eq. 2.76)).

Reflection boundary conditions are used on the plane of symmetry for the CSC configuration. This symmetric boundary condition is based on the assumption used in Ref. [27] and follows the findings of Ref. [28]. If the plane of symmetry is assumed in constant η-direction, then

$$V = 0 \ \longrightarrow \ \text{determine } u, v, w$$

$$\frac{\partial T}{\partial \eta} = 0\ ,\ \frac{\partial P}{\partial \eta} = 0 \tag{2.80}$$

2.7 Turbulence Model

The Navier-Stokes equations ((Eq. 2.1)) solved in this study are time-averaged, i.e. they are the Reynolds-averaged Navier-Stokes equations. Therefore, they do not contain enough information for turbulence to form a closed set of equations (closure problem). To remedy this problem and simulate the effects of turbulence, researchers are replacing these exact but insolvable equations by approximate but solvable ones. This is called turbulence modeling. Unfortunately, a perfect, efficient, and universally accepted turbulence model is yet to be developed. The major obstacle to turbulence modeling is the existence of various length scales controlling the turbulent kinetic energy. The turbulent Reynolds stress resulting from time averaging is assumed proportional to the laminar stress tensor with the coefficient of proportionality defined as the eddy viscosity, μ_t. Similarly, the turbulent Reynolds heat flux is assumed to be proportional to the laminar heat flux with the coefficient of proportionality defined as $(\gamma\ Pr\ \mu_t)\,/\,[Pr_t\ \ (\gamma-1)]$, where Pr_t is the turbulent Prandtl number. Hence, for turbulent flow computations the laminar flow coefficients are replaced by

$$\mu_e = \mu + \mu_t \tag{2.81}$$

$$\frac{k}{C_p} = \frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \tag{2.82}$$

A two-layer algebraic model proposed by Baldwin and Lomax [53] is chosen for the current study to calculate the local values of the eddy viscosity. The mixing-length model employed in this model divides the shear layer into an inner and outer region, and is patterned after a method developed for attached boundary layers by Cebeci [54]. The viscosity coefficient, μ_t, is given by

$$\mu_t = \{ \begin{matrix} (\mu_t)_{inner} & y \leq y_{crossover} \\ (\mu_t)_{outer} & y > y_{crossover} \end{matrix} \tag{2.83}$$

where y is the normal distance from the wall, and $y_{crossover}$ is the smallest value of y at which the values from the inner and outer formulas are equal.

In the inner region, the Prandtl-Van Driest mixing length formulation is given by

$$(\mu_t)_{inner} = \rho\, l^2 |\omega| \tag{2.84}$$

where $|\omega|$ is the magnitude of the local vorticity vector, and

$$l=ky\left[1-exp\left(y^{+}/A^{+}\right)\right] \tag{2.85}$$

$$y^{+}=\frac{y\sqrt{\rho_w\tau_w}}{\mu_w} \tag{2.86}$$

In the outer region:

$$\left(\mu_t\right)_{outer}=\mathrm{K}\,C_{cp}\,\rho\,F_{wake}\,F_{kleb}\left(y\right) \tag{2.87}$$

with

$$F_{wake}=\;minimum\;\left\{\begin{matrix} y_{max}\,F_{max} \\ C_{wk}\,y_{max}\,u_{diff}^{2}\,/F_{max}\end{matrix}\right\} \tag{2.88}$$

where F_{max} is the maximum value of the function F(y) defined as:

$$F\left(y\right)=y\left|\omega\right|\left[1-exp\left(-y^{+}/A^{+}\right)\right] \tag{2.89}$$

and y_{max} is the value of y at which F_{max} occurs. The difference between the maximum and minimum total velocity at a fixed x (or ξ) station, u_{diff}, is given as

$$u_{diff}=\left(\sqrt{u^2+v^2+w^2}\right)_{max}-\left(\sqrt{u^2+v^2+w^2}\right)_{min} \tag{2.90}$$

The second term in u_{diff} is set to zero (except in wakes). The function $F_{kleb}(y)$ is the Klebanoff intermittency factor, and is defined as

$$F_{kleb}\left(y\right)=\left[1+5.5\left(\frac{C_{kleb}}{y_{max}}\right)^{6}\right]^{-1} \tag{2.91}$$

The constants which appear in Eqs. (2.84–2.91) are given the following values [53]

$$A^{+}=26\;;\;C_{cp}=1.6\;;\;C_{kleb}=0.3\;;\;k=0.4\;;\;K=0.0168\;;\;C_{wk}=0.25 \tag{2.92}$$

One major advantage of this model lies in the fact that the location of the boundary layer edge does not need to be computed, since the length scales are based on the vorticity distribution.

The Baldwin-Lomax model faces difficulty in evaluating the length scale y_{max} and in turn $(\mu_t)_{outer}$ when dealing with vortical flows. This difficulty is apparent by examining the behavior of the function F(y) in a general crossflow separation region. A general curve for F(y) is shown in Fig. 2.1. The profile illustrates a single, well-defined peak which makes it easy to calculate y_{max} for the function F_{max}. However, this task is not so straightforward when there exists a strong overlaying vortex in the profile. As shown in Fig. 2.1b, in addition to a local peak in F(y) in the attached boundary layer at y=a, the vortex structure causes a large peak in F(y) at y=b. It is argued that the choice of the peak at y=b results in a value of F_{wake} and in turn a value of $(\mu_t)_{outer}$ which is much too high [35].

To eliminate this shortcoming, implementation of the Baldwin-Lomax model is modified as proposed by Degani and Schiff [35]. At each computational coordinate the code searches outward, sweeping from wall to freestream, in order to find the first peak in F(y). The search is stopped when the peak is reached. A peak is considered to have been found when the value of F(y) drops to 90% of the local maximum value. This ensures the exclusion of the erroneous peak which might be caused by nonsmooth behavior in F(y). Generally, the peaks in the crossflow separation region are located far enough apart that the proper peak can be found as described above. However, this logic may break down in the neighborhood of a primary separation and immediately following the secondary separation. Under these conditions the code would choose a value of y_{max} near the top edge of the cavity. Consequently, a cutoff distance is specified in terms of y_{max} from the previous value i.e., $y_{cutoff} = C\ y_{max,previous}$, where C is a constant, chosen equal to 1.5.

The second modification made to the standard Baldwin-Lomax model is the inclusion of multiple wall effects, for computing the value of μ_t near corners. In this case, the eddy viscosity was computed from each wall, and then an effective eddy viscosity was computed by taking an inverse average of the values computed from each wall. For example, on the corner bounded by the cavity floor (flr), forward facing wall (ffw), and the side wall (sw), the eddy viscosity was computed as:

$$\mu_t = \frac{\left(\frac{\mu_t}{y^+}\right)_{flr} + \left(\frac{\mu_t}{y^+}\right)_{ffw} + \left(\frac{\mu_t}{y^+}\right)_{sw}}{\sqrt{(y^+)^{-2}_{flr} + (y^+)^{-2}_{ffw} + (y^+)^{-2}_{sw}}} \tag{2.93}$$

This eddy viscosity is thus influenced greatest by the wall with the lowest y^+ value.

Turbulence in a mixing shear layer or a wake requires elaborate modeling since the measurements indicate that the Reynolds stress approaches a new equilibrium state exponentially due to turbulent history effects [55]. The final modification to the model considers the abrupt change in the eddy viscosity coefficient from the boundary layer to the wake and represents the different length scales in this problem. This modification employs the concept of relaxation eddy viscosity given by Waskiewicz et al. [55]. The eddy viscosity in the wake is computed as:

$$\mu_t = (\mu_t)_e + [(\mu_t)_l - (\mu_t)_e]\,[1 - exp(-x/\beta\delta)] \tag{2.94}$$

where $(\mu_t)_e$ is the calculated maximum eddy viscosity at the upstream lip of the cavity, and $(\mu_t)_l$ is the calculated "outer" eddy viscosity value based on the local vorticity value in the wake. Note that $(\mu_t)_e$ and $(\mu_t)_l$ are at the same spanwise location. The streamwise distance from the corner is denoted by x, and δ is the instantaneous boundary layer thickness at the upstream lip. The parameter β is a relaxation length scale given a value of 10.

2.8 Chapter Summary

The governing equations and their discretization have been presented in Chap. 2. The implicit approximate factorization scheme and the upwind differencing have also been discussed. A discussion on turbulence modeling, initial conditions, and boundary conditions have been included. The baseline solution algorithm presented in this chapter can be used on a single subdomain. However, some modifications to this algorithm are necessary for its implementation on a composite of overlapped grids. A brief description of composite grid generation, the modifications to the solution algorithm, and its extension to overlapped grids are presented in Chap. 3. The multigrid convergence acceleration for a single subdomain and multiple subdomains of grid overlapping techniques are also discussed in Chap. 3.

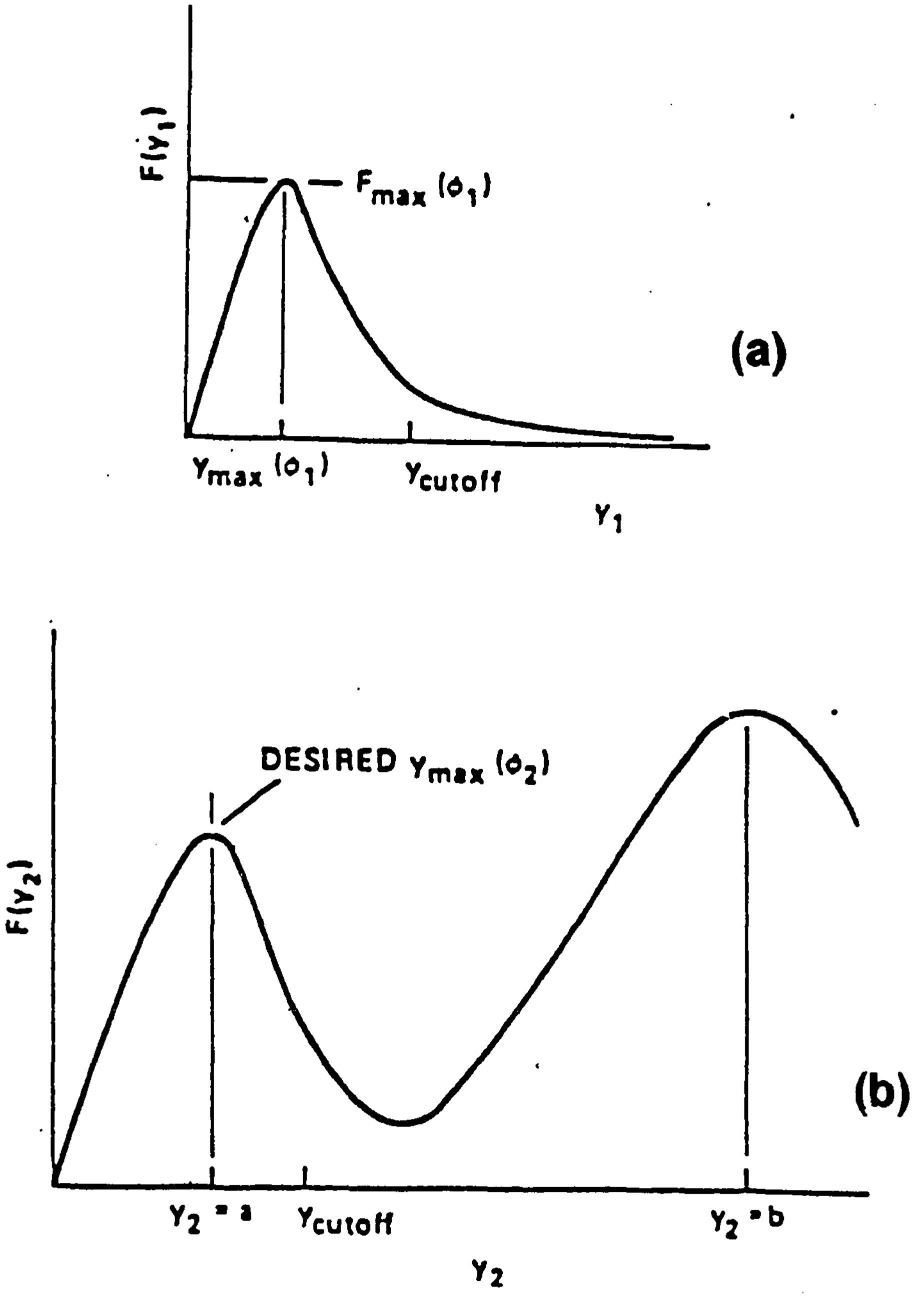


Fig. 2.1 Behavior of F(y).

Chapter 3

SOLUTION ALGORITHM ON OVERLAPPED GRIDS

The baseline algorithm for a single subdomain was presented in Chap. 2. However, the flow domains considered here are discretized with a composite grid. A composite grid, which is a union of the subdomain grids, is constructed to cover the region on which Eq. (2.1) is to be solved. Each subdomain is chosen to ease the grid generation task and create a suitable mesh for the region of that particular subdomain. Various DDT's can be used to decompose the flow domain into simpler subdomains. The present study uses a hybridization of these DDT's and combines their advantages. However, the grid overlapping technique is the primary focus of this study. This is due to the special consideration needed to generate the composite grid using this method and the required modifications to the solution algorithm.

The implementation of the grid overlapping is greatly simplified by setting an order of hierarchy between the subdomains. Once the order is set, the composite grid can be constructed. As a result of overlapping, the embedded or the overlapped grid creates an artificial boundary or hole in the outer grids. The cells (points in finite difference) are then identified and flagged as hole cells (points). These cells are then removed from the solution. The subdomain grids can be embedded completely or simply overlapped. Then the solutions of Eq. (2.1) on each individual grid are matched by interpolation of the conserved variables.

The construction of the overlapped grids for the present study is accomplished by the technique presented in Refs. [56, 57]. This technique is based on the Chimera grid scheme of Steger et al. [16–19]. The Chimera algorithm discretizes the subdomain using finite difference method and solves the flow equations on the composite grid utilizing a central difference scheme. Lessard [57] modified the composite grid generation of this scheme. This modified scheme called "MaGGiE" discretizes the subdomain using a finite volume method (as well as finite difference). It is also capable of constructing the composite grid at coarser grid levels. The resulting composite grid is then well-suited for a finite volume (Sec. 2.3) and multigrid (Sec. 3.3) solver. The construction of composite overlapped grids by Chimera and MaGGiE is detailed in Appendix A and will be discussed here briefly. Note that the composite grid is defined here as the union of subdomain grids overlapped or embedded on one another. A composite grid also includes the information for the necessary intergrid communication, details of which are given in Appendix B.

The primary interests of the present study, however, lies with the solution algorithm presented in Chap. 2 and the required modifications to extend this algorithm to work on overlapped grids. These modifications are two types. The first type modifications are intended to maintain the fully vectorizable characteristic of the implicit scheme. These also include the necessary modifications for using upwind differencing schemes. The second type modifications are focused on the multigrid convergence acceleration scheme for the overlapped grids. Details of the modified solution algorithm and the multigrid scheme for overlapped grids are given in this chapter.

3.1 Composite Grid Construction

The grid overlapping technique allows the subdomain grids to be generated independently. Each subdomain grid which may contain a component of the complex configuration can be generated using any of the well known grid generation methods, such as, the algebraic method or the Poisson's equations method. The independently generated subdomain grids are then translated and rotated to their relative positions. The cell centers are then computed in the coordinate system of the composite grid. The cell center grids are created by averaging the coordinates of the eight cell vortices (Fig.

B.1). The cell vertices are the nodes of the finite difference grid. The details of the grid generation and the construction of the composite grids are given in Appendix A.

An order of hierarchial form between the grids makes the implementation of the grid embedding/overlapping concept more efficient. The hierarchial form between the grids allows the interaction of the appropriate grids, simplifies the development of the data structure required for the interaction, and limits the search to locate points in other grids for the purpose of interpolation. Some examples of hierarchial grid arrangements are illustrated in Fig. A.1. Grids which are on level L of the hierarchy, G_L, are partially or completely embedded in grids of level L-1. At the same time, G_L may overlap other grids at level L and may contain grids of level L+1 which are partially or completely embedded in them. The concept of hierarchy is also detailed in Appendix A.

As subdomains are set on their overlapped position, some cells of one grid may be found to coincide with a solid boundary contained within another grid. Also, a significant number of cells must be interpolated, if every cell common to more than one subdomain grid is to be updated. This becomes computationally expensive and it could have an adverse effect on global accuracy when mesh sizes are not compatible on different subdomain grids. This problem can be avoided if only the boundary of the common regions between each subdomain is updated and the other cells inside these regions are excluded from the calculation. In this process, the subdomains which are overlapped on the others introduce artificial boundaries or holes inside those subdomains (Fig. B.2–B.4). Hence, a search method is used to create and locate these holes. For example, a hole is generated in G_L by the embedded grid G_{L+1} (Fig. B.2). The cells from G_L which lie inside this hole are identified and called HOLE CELLS. A hole cell is flagged for grid G_L by setting an array IFLAG = 0. A cell of G_L which is not in the hole is flagged by setting IFLAG = 1. The details of this search procedure are presented in Appendix B. Note that the search procedures for both finite difference and finite volume grids are similar . For finite volume grids, the cell centers replace the nodes in Figs. B.2–B.4.

The G_L cells which are immediate neighbors of the hole cell are called the FRINGE cells (Fig. B.5), and the intergrid communication of the conserved variables from the G_{L+1} grid is performed with these cells. A fringe cell is also flagged as IFLAG=0. A cell in G_{L+1} with the smallest distance to a fringe cell in G_L is located and called a TARGET cell. Then a search is conducted to find seven cells of G_{L+1} near the target cell. The objective is to form a hexahedron which has these seven cell centers and the target point as the vertices; this hexahedron has to include the fringe point of G_L (Fig. B.6).

The information transfer from the eight cells of the G_{L+1} grid to the fringe cell of G_L is done through trilinear interpolation coupled with characteristic intergrid boundary conditions. This process is called UPDATING. Since trilinear interpolation can only be used on cubes, and since the hexahedron formed around a target cell is generally a warped one due to the curvilinearity of a body-fitted grid, it is necessary to map a warped hexahedron to a unit cube using isoparametric mapping (Fig. B.6). The coordinates of the fringe cell which is being updated, are denoted by α, β, γ, relative to its target cell. Then the interpolated values of the Q vector can be calculated as

$$\begin{aligned} Q = a_1 + a_2\,\alpha + a_3\,\beta + a_4\,\gamma + a_5\,\alpha\,\beta + a_6\,\alpha\,\gamma \\ + a_7\,\beta\,\gamma + a_8\,\alpha\,\beta\,\gamma \end{aligned} \tag{3.1}$$

The values for the eight free variables, a_i, are determined from the known values of Q_i at the eight vertices of the information hexahedron. The details of hole generation and the intergrid communication are given in Appendix B. This procedure is repeated in an opposite manner at the outer boundary of the overlap region, where the information is transferred from the grid G_L to the grid G_{L+1}.

It should be noted that the trilinear interpolation used in this approach for intergrid communication is a non-conservative procedure. However, the effects of lack of conservation can be reduced by taking the following three guidelines: 1. The subdomain grids are to be created in such a manner that when one grid is overlapped within another, the cells of both grids are of comparable sizes within the overlapped region. 2. The flow regions with very high flow gradients should be contained within a subdomain grid as much as possible. 3. Incorporating the weighted variation of the cell volumes into the interpolation can reduce the error introduced by the non-conservative interpolation. The procedure to find the cell-volume weighted variations for a three-dimensional grid is geometrically complicated. Hence, in the current algorithm, the last step has not been implemented yet. However, in producing the results presented in Chap. 4, efforts were made to accommodate the guidelines 1 and 2.

There are other efforts currently being undertaken by researchers to improve the intergrid communication. Some of these approaches are introduced in Appendix C. These approaches include methods for maintaining conservation at the intergrid boundaries and methods for improving the accuracy of the interpolation at those boundaries. For example, a method for matching the solution at the intergrid boundary has been shown to be conservative for one-dimensional overlapped grids [58]. However, this method is rather complicated and irregular even for two-dimensional curvilinear grids.

To improve the accuracy of the interpolation, adding a second set of interpolation points has been suggested in Ref. [59]. For example, the accuracy of the trilinear interpolation of the current algorithm can be improved if two sets of fringe cells are used. However, this increases the risk of illegal communication between the subdomains. An illegal communication occurs when one or more cells, making up the information hexahedrons (for updating process), are fringe cells themselves. This also increases the computer memory and the run time of the flow solver, simply because there are twice as many cells to update. The implementation of two sets of fringe cells is not feasible for the three-dimensional grids considered in this study.

3.2 Modified Solution Algorithm

The implicit scheme described in Chap. 2 is a computationally efficient and fully vectorizable algorithm. However, modifications to the solution algorithm are required in order to utilize the standard block or scalar tridiagonal inversions of Eq. (2.28) on an overlapped grid with holes. The hole, fringe, and outer boundary cells are to be excluded from the computation while maintaining the efficiency obtained by vectorization. Further enhancement to the implicit algorithm is achieved by implementing the multigrid convergence acceleration scheme into the solution algorithm. Modifications to the multigrid scheme are also necessary due to the holes in the overlapped subdomains.

The fringe cells along with the cells located inside the holes or outside the outer boundary which are flagged as IFLAG=0 are treated by the following modification to the implicit algorithm: (a) set the off-diagonal 5×5 blocks in the coefficient matrix to zero, (b) set the diagonal blocks to unity, (c) set the residuals to zero. This results in the computed value of ΔQ for these cells to be zero. Thus, the updated or specified boundary conditions for these cells are automatically preserved. For example, let one of the spatial factors of Eq. (2.28) be written as a system of algebraic equations in block tridiagonal form represented as

$$a_{ij} \times \phi_i = R_i \tag{3.2}$$

where a_{ij} are the 5×5 blocks of the tridiagonal coefficient matrix and ϕ_i are unknown vectors, which represent ΔQ of Eq. (2.28). The right hand side residual is represented by R_i. Then,

$$R_i \leftarrow R_i \times IFLAG_i \tag{3.3}$$

$$a_{ij} \leftarrow a_{ij} \times IFLAG_i, \qquad i \neq j \tag{3.4}$$

$$a_{ij} \leftarrow (a_{ij} \times IFLAG_i) + (1 - IFLAG_i), \quad i = j \tag{3.5}$$

where (←) indicates that its left hand side is to be replaced by its right hand side. It should be noted that all the elements of the block a_{ij}, $i \neq j$, must become zero for the hole cells. However, only the diagonal elements must be set to unity for the diagonal blocks a_{ii}.

In Eq. (2.28), the spatial factors in each direction can be approximated with a diagonal inversion (Sec. 2.5.2). This generates considerable savings in computational time and can be used to advance the solution past the initial numerical transient stage. Therefore, similar modifications used for standard block inversion is also implemented when the approximate diagonal form of Eq. (2.28) is used. In approximate diagonal form, an inversion of the block triadiagonal matrix in a spatial direction is replaced by the inversion of three scalar triadiagonal matrices. Hence, modifications of Eqs. (3.3–3.5) are repeated three times corresponding to each direction. It should be emphasized that a_{ij} are scalar coefficients in this process.

In the upwind differencing discussed in Sec. 2.5, the discretization of the right hand side of Eq. (2.28) uses a five-point stencil in each direction for second-order spatial accuracy. In order to avoid erroneous flux from a cell in a hole when computing a cell neighboring a fringe cell, the Q value of the neighboring hole cell is set equal to the Q value of the fringe cell. This requires a search for such hole cells which border the fringe cells in each of the three directions. Hence, this process reduces the five-point stencil differencing to a three-point stencil differencing for cells in immediate neighborhood of hole. This process also reduces the accuracy of the differencing scheme for these cells. The search for the neighboring hole cells and nullifying the effects of these cells are accomplished by the following procedure. Assuming $IFLAG_i$ represents the flag of the fringe cell, then

$$C_1 = IFLAG_{i-1} + IFLAG_i \tag{3.6}$$

$$C_2 = IFLAG_i + IFLAG_{i+1} \tag{3.7}$$

$$Q_{i-1} = (C_1 \times Q_{i-1}) + (C_2 \times \ Q_i) \tag{3.8}$$

$$Q_{i+1} = (C_1 \times Q_i) + (C_2 \times \ Q_{i+1}) \tag{3.9}$$

where C_1 and C_2 are constant coefficients and their value is either zero or unity. Note that the values of C_1 and C_2 cannot be both zero or unity at the same time. This is because a fringe cell is flagged as IFLAG = 0 and it is always located between a hole cell (IFLAG = 0) and a computed cell (IFLAG = 1). This procedure is repeated in each of the three directions.

The importance of the IFLAG array is now well established. All the cells in the computational domain are flagged with this array and treated accordingly. Hence, the standard fully-vectorizable characteristic remain unaltered.

3.3 Multigrid Scheme for a Single Subdomain

The grid generation with domain decomposition allows one to represent the critical regions of a flowfield with very fine meshes. This helps the optimization of the grid distribution and subsequently the computational time required to obtain the solution on these meshes is reduced. The penalties paid are the overhead time associated with the intergrid information transfer and the duplicate computations in the overlapped regions. In addition, since the logarithm of the spectral radius for a typical scheme generally varies linearly with the mesh size, solving the equations on very fine grids may be impractical. Methods such as the multigrid convergence acceleration can be very successful in accelerating the rate of convergence for steady state solutions. It is also very desirable since it maintains the high level of robustness and accuracy obtained through methods such as upwind differencing. Consequently, it is desirable to offset the additional computing time due to the domain decomposition by the multigrid method. First the multigrid method for a single subdomain will be explained in this section. Then, its application for the overlapped grids will be explained in the next section.

Earlier works by Chima and Johnson [60] and Jameson [61] showed that the multigrid technique can be very useful for two-dimensional inviscid and viscous flows used with explicit and implicit schemes. Anderson et al. [38] later combined the full approximation scheme (FAS) multigrid method with flux vector splitting to obtain efficient solutions to the Euler equations in three dimensions.

Inadequate damping of low-frequency error is responsible for the slow asymptotic convergence rate [62] and these errors can be damped efficiently by the multigrid method. To accelerate the convergence to a steady-state, a sequence of grids are formed for each subdomain by deleting every other grid line on the next finer grid [38]. The rationale for this technique may be explained briefly as follows. The high frequency error component on a given grid level has to be resolved on the same grid level by the solution algorithm explained above. The high frequency error does not appear on the next coarse grid due to increased spacing. However, the remaining low frequency error component appears as a high frequency error on the coarse level grid, which in turn can be reduced by the same solution algorithm that is capable of resolving high frequency errors. This procedure also helps the improvement of the overall robustness. A fixed cycling strategy is used, where a predetermined number of iterations are performed at each grid level. The values of the dependent variables (Q) and residual (R) are passed from a finer grid to a coarser grid through volume-weighted restrictions $\left(I_i^{i+1}\right)$ and $\left(\hat{I}_i^{i+1}\right)$,

$$Q_{i+1} = \left(I_i^{i+1} Q_i\right) = \left[\Sigma V \cdot \Delta Q\right] / \Sigma V \tag{3.10}$$

where V denotes the volume of a cell. Since the full solution is computed and stored on each grid level as opposed to only corrections, this is referred to as the full approximation scheme (FAS). Denoting the discrete analog of the operation in Eq. (2.1) by (L), and the relative truncation error by (E), the following equation is written,

$$L_{i+1}\left(Q_{i+1}\right) = R_{i+1} + E_{i+1} \tag{3.11}$$

where

$$R_{i+1} = \left(\hat{I}_i^{i+1} R_i\right) \tag{3.12}$$

Therefore, from Eqs. (3.10–3.12)

$$E_{i+1} = L_{i+1}\left(I_i^{i+1} Q_i\right) - \hat{I}_i^{i+1}\left[L_i(Q_i)\right] \tag{3.13}$$

Note that the solution on the coarse grid is driven by the fine grid and the relative truncation error between the coarse and fine grids. When the coarsest level is reached, computed corrections to (ΔQ) values at each level are prolonged to the next fine level through trilinear interpolations.

3.4 Multigrid Scheme for the Overlapped Grids

It is apparent that a large portion of the calculations, when multigrid strategy is used, are performed at coarse grid levels. This is advantageous due to the reduction in grid points associated with the coarse levels, which in turn means less operations are required. Hence, it is worth incorporating the multigrid method within the domain decomposition techniques. The multigrid scheme can easily be extended to multiblock grid method and zonal method. However, it requires special consideration when incorporated into the overlapping technique.

The existence of holes in the current domain decomposition method poses major problems for the implementation of a standard multigrid algorithm. Recently, Chesshire and Henshaw [63] developed a technique for the generation of composite overlapped grids and the numerical solution of a two-dimensional, linear, elliptic, partial differential equation on them. This technique allows them to construct a two-dimensional composite grid with any number of subdomains for finite difference computations. It is designed to generate a sequence of coarse grids required for the multigrid method. In the present algorithm, the cells of the coarse level grids of each subdomain can exist everywhere including the holes and the fringe cells. Also, even if the cell itself is not in a hole, cells which are in a hole are conceivably used either at the restrictions or the prolongation stages, or during the coarse level finite volume calculations. This section includes a discussion on the construction of the coarse level composite grid. The modifications to the multigrid scheme for both restriction and prolongation stages are also given here.

The sequence of coarse level grids are generated from the finest level grids of each subdomain as explained in Sec. 3.3. One of the criteria used in creating coarse level composite grids, is to create the holes in such a manner, that during a restriction stage of a multigrid cycle, the restricted functional values are not contaminated by the hole cells on the finer level grids. Secondly, the hole cells of the coarse grids, G_L, are connected to the cells at the coarser levels of other grids, G_{L+1}. This is done to avoid the contaminated information being transferred from within a hole of the coarse grids to non-hole cells of the next finer grids, during the prolongation stage.

The hole cells in the coarse grids are created from holes in the finest level grids of the composite mesh. A search sequence of locating eight finer level cells that make up a coarse level cell is accomplished, such that, if at least one of the eight finer cells is a hole cell, which is designated IFLAGM = 0, then the coarse level cell is designated IFLAGM = 0. If none of the eight finer cells are hole cells, then the coarser level cell is an exterior cell, and it is designated IFLAGM = 1, where M denotes the coarseness level of the grid. The above procedure of defining holes in the coarser subdomains eliminates the restriction errors caused by the holes in the finest level mesh. There are no restriction errors because the restricted value of a coarse cell is determined by weighted values of eight finer cells that make up the coarse cell. The above sequence is repeated for each coarser level of the composite mesh.

Once the hole cells the in coarse subdomain grid, G_L, are located, a search is conducted for interpolation data for these cells, with IFLAGM = 0, from a coarse subdomain grid. G_{L+1}. This search can only fail for those cells of G_L at level M which, for example, coincide with the body around which the grid G_{L+1} is generated. Such a zone is called the ILLEGAL ZONE, and its cells are left with the flag IFLAGM=0, so that they can be effectively excluded form this coarse level calculation. The IFLAGM values of all the other cells, which now have interpolation data, are switched from zero to unity. The next step is locating the outer boundary connection cells of the overlap region where the interpolation is to be performed from G_L to the coarse level of G_{L+1}. Such outer boundary cells of G_{L+1}, for which interpolation data are now available, are flagged as IFLAGM=0. All of the information obtained above for the hole cells, illegal zones, outer boundary cells, and their interpolation data, are written in a data vector form for the multigrid Navier-Stokes solver. The inversions are performed at the coarse levels after the modifications are made analogous to Eqs. (3.3–3.5) with IFLAG_i replaced by IFLAGM_i. The transfer of information between the grids, say G_L and G_{L+1}, therefore, is realized through the updating performed before restrictions, and it is done over much larger physical domains than the overlap regions of the finest level meshes.

There is freedom in choosing the level grid of G_{L+1} from which the interpolation is to be performed to the coarse levels of G_L. The obvious choice is searching for interpolation data between the same level grids. If the cells involved in the interpolation are of comparable sizes at the finest level, as is desirable for accuracy, they are again of comparable sizes at the coarse levels. However, experience shows that the probability of success in searching for the information hexahedrons is less at more coarse levels than it is with the finest level grids. Therefore, an option is built into the algorithm where the information hexahedrons are formed from the finest level grids. This option increases the success in forming the information hexahedrons, but it decreases the accuracy. Also, this option eliminates the possibility of mesh sequencing [38, 64].

Mesh sequencing is a process of accelerating the convergence by solving Eq. (2.1) on the coarse grid levels. This strategy is based on the idea that only enough iterations need to be conducted to smooth the errors below the level of truncation error before switching to the next finer mesh. Mesh sequencing for overlapped grids is only possible when the grids communicate on the same grid level, where Eq. (2.1) is solved at the same coarse level of all the subdomain grids until some level of convergence is reached.

The above procedure details the construction of composite grids at a sequence of coarse levels. At the same time, the procedure is set in such a manner that the solution on coarse level grids are not contaminated by the hole cells of the fine level. Modification is also needed for the prolongation stage and it is performed in the solution algorithm. This modification is necessary due to the hole cells that exist in the illegal zone of the coarse grids. The prolongation stage modification may be summed up as nullifying the weight of the contributions from illegal zone cells. The prolongation is performed from the coarse cell centers, say c1 and c2, to pseudo finer level cell centers between them, say f1 and f2 (f1 is closer to c1 than f2), in one direction as,

$$\begin{aligned} \Delta Q_{f1} &= (A1 \times \Delta Q_{c1}) + (B1 \times \Delta Q_{c2}) \\ \Delta Q_{f2} &= (A2 \times \Delta Q_{c1}) + (B2 \times \Delta Q_{c2}) \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} A1 &= IFLAGM_{c1}\,[1 - (b \times IFLAGM_{c2})] \\ B1 &= IFLAGM_{c2}\,[1 - (a \times IFLAGM_{c1})] \\ A2 &= IFLAGM_{c1}\,[1 - (a \times IFLAGM_{c2})] \\ B2 &= IFLAGM_{c2}\,[1 - (b \times IFLAGM_{c1})] \end{aligned} \tag{3.15}$$

and a, b are the bilinear interpolation constants which are 0.75 and 0.25, respectively. This process is repeated in the second direction, using the pseudo finer level cells of the first direction. Finally, when this process is repeated in the third direction with Eq. (3.14) and using the pseudo finer level cells of the second direction, the ΔQ corrections are recovered for the actual finer level cells. The resultant process is essentially a trilinear interpolation with slight bias around an illegal zone.

In the present study, multigrid acceleration scheme has been utilized for steady cases only. However, Anderson et al. [65] have used the multigrid scheme for unsteady calculations of dynamic meshes. Their numerical experiment indicated that the use of multigrid scheme with two grid levels results in virtually identical solutions as when a single grid is used, but with a substantial reduction in computer time. The use of three grid levels, however, leads to slight discrepancies in the shock region. In this study, the multigrid method is performed on overlapped grids only for a steady flow. The multigrid scheme on overlapped grids for time-accurate calculations have not been examined. This has been left as an area of future investigation.

3.5 Chapter Summary

A brief discussion on the construction of composite grids has been presented in this chapter. Next, the modification of the solution algorithm and the multigrid scheme for grid overlapping technique have been discussed. The following chapter is the discussion on the results obtained by examining the algorithm on both single and composite grids. The time-accurate simulations of the interference flow between multiple and nonsimilar bodies are the main emphasis of Chap. 4.

Chapter 4
RESULTS AND DISCUSSION

The development of a CFD capability to obtain solutions for flows about complex configurations is performed in five subsequent steps. The cylinder-sting-cavity (CSC) assembly is an example of a complex configuration since it consists of multiple and geometrically nonsimilar components. Its components pose considerable challenges themselves. Hence, prior studies of flows about these components as intermediate steps are necessary to gain a better understanding of such flows. These preliminary and preparatory studies, excluding the clean cavity flows are presented as Cases 1, 2 and 3. Detailed studies of flows about rectangular cavities in both two– and three-dimensions have been reported in Refs. [29–33].

Three different two-dimensional profiles (TDP) are considered in Case 1 to examine the effects of shapes on supersonic flows past cylinders. In Case 2, a body of revolution is formed from one of these profiles. Then, the solutions for supersonic flows over this body at high angles of attack are obtained. The grid overlapping technique is tested in Case 3 by considering the flowfield of Case 2 and using a composite grid in place of a single grid. The interference flow problem of a cylinder connected to an offset sting near a cavity is analyzed in Case 4. Since the interference pattern is dramatically different in the event that the cylinder is placed inside a cavity, Case 5 is devoted to this type of flow.

4.1 Case 1: Supersonic Flows Over Two-Dimensional Profiles (TDP)

Supersonic flows about three different TDP's are considered. The schematic diagrams of these profiles are shown in Fig. 4.1. The two-dimensional grids around these profiles are generated by solving the two-dimensional Poisson's equations. The objectives of this case may be summarized as follows: (a) to examine the effects of forebody and base shapes for these type flows; (b) to aid the implementation of the overall computational algorithm described in Chap. 2, and to determine its suitability for the class of flows being considered; (c) to prepare for the simulations of the complex interference flows involving cylinders. These models are the longitudinal profile sections of three different bodies-of-revolution. Only one of the models is chosen for the subsequent three-dimensional investigation. The detail of this investigation is given in Sec. 4.2.

Model I is a TDP of a blunt-nose-cylinder which has a forebody defined by a parabola and it is 3 diameters long. The diameter (d) is 3.0 in. and it has the largest length-to-diameter ratio (6.67) of all three models (Fig. 4.1a). The approaching flow is supersonic, turbulent, and at incidence (M = 1.6, $Re_d = 5\times10^5$, $\alpha = 8°$, 12°). The C-type grid about this model has 138×86 cells.

Model II is a TDP of a blunt-nose-cylinder which has a forebody defined by one-half of an ellipse with a relatively large minor axis increasing the bluntness almost to that of a semi-circular cylinder. As shown in Fig. 4.1b, the diameter is 2.4 in. and the length-to-diameter ratio is 5.0. The oncoming flow is supersonic, turbulent and at incidence (M = 1.5, $Re_d = 4\times10^5$, $\alpha = 8°$, 20°). The C-type grid about this model has 120×70 cells.

Model III is a TDP of a cylinder with a diameter of 2.25 in. and length-to-diameter ratio of 6.0 (Fig. 4.1c). Its forebody, other than the tip of its nose, is defined by an ellipse of larger major axis than the Model II. The tip of its nose (0.36–diameter) is pointed. This configuration has boat-tail. The approaching flow is two-dimensional, turbulent with M = 1.6, $Re_d = 3.75\times10^5$, and $\alpha = 8°$. The grid size for this model is 118 ×86.

The computed density contours for Model I at $\alpha = 8°$ and 12° are presented in Fig. 4.2. For Model I at $\alpha = 8°$, the shock stand-off distance and the maximum density ratio across the shock are approximately 0.6d and 1.95, respectively. The separation on the upper surface starts 5 diameters downstream from the nose. An increase in angle of attack from 8° to 12° results in 3% increase in the lift-to-drag ratio. At $\alpha = 12°$, the separation on the upper surface moves further upstream by about one diameter. The lowest density value drops from 10% of the freestream value for $\alpha = 8°$ to 7% for $\alpha = 12°$.

The computed density contours of Model II at $\alpha = 8°$ and 20° and Model III at $\alpha = 8°$ are presented in Fig. 4.3. The flowfield for Model II at $\alpha = 8°$ shows differences chiefly around the forebody from that of Model I. Also, the separation on the upper surface of this model starts further downstream than it does for either Model I or Model III. The effect of higher angle of attack on the shock stand off distance and the growth of the boundary layer for Model II is observed in Fig. 4.3. The stagnation point also moves further down on the nose as angle of incidence is changed from 8° to 20°. The density contour for Model III (Fig. 4.3c) shows that the wake for this model has smaller width than of Model II. The shock stand-off distance for Model III is 0.6d with a maximum density ratio around 2.0.

Although, the general flowfield features may appear to resemble one another, it is shown computationally that the differences in the forebody and the base generate differences in the flow details. The maximum surface pressure pressure (C_p), lift (C_L), and drag (C_D) coefficients for each model at $\alpha = 8°$ are given in Table 4.1.

The lift-to-drag ratio of Model I indicates an improvement of about 100% over Model II and about 20% over Model III. The improvement of Model III over Model II is about 71%. These improvements are due primarily to the larger fineness ratio and the shape of the forebody.

Table 4.1 Pressure and force coefficients of Case 1.

Model Section	Max. C_p	C_L	C_D
I	1.77	0.49	0.21
II	1.63	0.39	0.34
III	1.83	0.45	0.23

4.2 Case 2: Supersonic Flows Over a Body-of-Revolution at High Angles of Attack

Three-dimensional supersonic flows over a blunt-nose-cylinder (BNC) at 32° and 44° angles of attack are considered in Case 2. Model I of Case 1 (Fig. 4.1 a) is chosen for this purpose. The C-O type grid for this case is generated simply by rotating the upper half of the two-dimensional grid of the symmetry plane about the longitudinal axis (Fig. 4.4). The full-body grid (η = 0° to 360°) consists of $81 \times 57 \times 65$ cells in the longitudinal, normal, and circumferential directions, respectively. The choice for a full body grid is based on the fact that flow at the conditions considered for this case has the potential to develop asymmetrically.

The objectives of this case are three fold: (i) to expand the 2–D algorithm used for Case 1 to compute the flow about a body-of-revolution; (ii) to provide valid solutions for supersonic flows over a body-of-revolution at high angles of attack, which compare favorably with experiments; (iii) to prepare for the simulation of the complex interference flows involving cylinders.

The flow conditions for Case 2 are $M = 1.6$, $Re/ft = 2 \times 10^6$, $Re_d = 5 \times 10^5$, $T_t = 585$ °R, where T_t denotes the total temperature. Although the flow Reynolds number over an ogive cylinder at $\alpha = 32°$ and $\alpha = 44°$ may be high enough to give rise to a transitional boundary layer [66], the flow is assumed laminar in this study. It should be noted that the experimental setup of Ref. [67] did not have a transitional strip mounted on the model.

Figure 4.5 shows the agreement between the computed and measured [67] longitudinal C_p distributions on the body at fixed circumferential locations. The computational and experimental values for C_N are 4.11 and 4.19, respectively. The normalized pressure contours at the symmetry plane and crossflow density contours at Station II will be presented later in comparison to those of Case 3. The computed density contour at the symmetry plane for this case is presented in Fig. 4.6. For the case of $\alpha = 32°$, the growth of the boundary layer on the leeside, the nose shock and the shock on the windward side are clearly shown in this figure. For the flow conditions considered here, one would expect that the pressure distributions on the surface to decrease circumferentially in the direction of windside toward the leeside. This would create a suction region on the leeside and in turn generating a lift on this body. However, crossflow separations occur in these flows (Fig. 4.7) and the boundary layer no longer remains attached. The crossflow separation has the positive effect of reducing the drag on the leeside. It also reduces the suction on that side which in turn lowers the lift. These crossflow separations generally cause a lower lift-to-drag ratio. Two axial stations are chosen to analyze the crossflow. Stations I and II (Fig. 4.1a) are located immediately after the forebody-cylinder junction (X/d = 3.17) and upstream of the base (X/d = 6.17), respectively. Comparisons of crossflow density contours at Stations I and II (Fig. 4.7) demonstrate the growth of the leeside vortical region as the distance increases from the nose as well as the cross section of the shock surface on the windward side.

Instantaneous limiting streamlines, which have been shown to follow the skin friction patterns on the body, are shown in Fig. 4.8. The enlarged view of the top side of cylinder are given in three different sections. Crossflow separations are generally indicated by the convergence of limiting streamlines. However, the exact circumferential location of the onset of separation may not be coincident with the convergence. The approximate location of the primary separation, which is designated in Fig. 4.8 as (PS), primary reattachment (PR), secondary separation (SS), and the secondary reattachment (SR), are indicated on Fig. 4.8a and b. The primary separation begins close to nose and rolls into the primary vortex above the surface of the cylinder. The primary vortex induces an adverse pressure gradient which in turn causes the secondary separation. The primary vortex on the right half of the leeside rotates counterclockwise. The rotation of the secondary vortex is clockwise. The limiting streamlines (Fig. 4.8c) indicate that the convergence lines originate close to the nose and move windward as they traverse the length of the forebody. That is, the separation occurs near the nose, as it was also observed by Yates and Chapman [68]. Shown in Fig. (4.9) are the circumferential skin friction distributions on the body at Station II. The existence and the approximate locations of primary and secondary separations locations on the body at Station II are indicated in this figure.

Comparisons of computed and experimental longitudinal surface C_p distributions for $\alpha = 44°$ were shown in Fig. 4.5. The density contours of the symmetry plane for $\alpha = 44°$ (Fig. 4.10a) compare well with the Schlieren photographs (Fig. 4.10b) of this flow (ref. [67]). The crossflow density contours and the velocity vectors at Station II for $\alpha = 44°$ are presented in Fig. 4.11. The approximate locations of primary and secondary separation (PS, SS) and reattachements (PR, SR) are indicated in Fig. 4.10a. The direction of the rotation of primary and secondary vortices for this case is similar to that of $\alpha = 32°$. Tertiary separation (TS) and reattachment (TR) are also observed and indicated in Fig. 4.10a. The tertiary vortex on the right half of the leeside also rotates counterclockwise. The circumferential skin friction distributions at Station II for $\alpha = 44°$ were shown in Fig. 4.9. The computed value of C_N for this case is 6.15 and it is within 4% of its experimental value of 6.38. The computations of both $\alpha = 32°$ and 44° were performed on NASA Langley VPS–32 vector processor using 64 bit arithmetic. Diagonal inversions were performed to initialize the $\alpha = 32°$ case and then switched to the block inversions. The stability bounds for the diagonal inversions appeared to have reached for the case of $\alpha = 44°$, therefore 5x5 block inversions had to be used even for the initialization.

4.3 Case 3: BNC with C-O Grid Overlapped on a Cartesian Grid

The blunt-nose-cylinder of Case 2 is reconsidered in this case. A boundary fitted C-O type grid is wrapped around the BNC and this grid is embedded in a Cartesian global grid. The region with solid circular dots in Fig. 4.12 represents a portion of the overlapped region between the C-O grid and the Cartesian grid. The Cartesian grid and the C-O grid have $81 \times 73 \ \times 73$ and $73 \times 65 \times 45$ cells, respectively. The details of the composite grid is given in Refs. [56, 57]. The computational algorithm used in Case 2 and explained in Chap. 2 is extended, as explained in Chap. 3, to the overlapped grids for this case. The flow conditions for this case are similar to the conditions used for Case 2. However, only incidence angle of 32° is considered. The multigrid convergence acceleration scheme and its extension to work on the overlapped grids, despite holes in the subdomains, are discussed in Chap. 3. To test this extended multigrid scheme and its effect on the computations of the overlapped grids, a two-level composite grid is created. The coarse grid of each subdomain is formed by deleting every other line on the fine grid of that subdomain. The generation of the composite grid and the intergrid communication informations at the coarse level are detailed in Ref. [57].

There are four major reasons for choosing this case. (i) A simple body-fitted grid for a body of revolution, such as a C-O grid, is very different topologically than a Cartesian grid. (ii) There is available computational solution for this case which is obtained on a C-O grid only, i.e., without overlapped composite grids (Case 2). (iii) There is available experimental data [67] for comparison. (iv) This case helps preparing the algorithm for the simulations of the complex interference flows involving cylinders.

The longitudinal pressure coefficient (Cp) distribution on the leeside of BNC at η = 0° is shown in Fig. 4.13. Superimposed on the present computational results are the experimental results of ref. [67], and the computational results of Case 2. The present results obtained on overlapped grids compare favorably with the experimental results. There is only a slight difference between the present results and those of Case 2. In an attempt to show the effect of grid refinement on accuracy, the results obtained on the overlapped coarse grids are also superimposed. As expected, the results improve with the refinement of the grid. Presented in Fig. 4.14 are the normalized pressure contours of the longitudinal symmetry plane. It should be noted here, when plotting in two-dimensions from the three-dimensional data of different subdomain, generally one can not find longitudinal or lateral surfaces of these subdomains which match in location or in curvature. This results in slight postprocessing discrepancies. Also, the postprocessing of the data, especially in curve fitting near intergrid boundaries, is restricted by the capabilities of the plotting programs. In any event, the contour lines cross the intergrid boundaries rather smoothly. The pressure field (Fig. 4.14a) compares very well with the one obtained without grid overlapping (Fig. 4.13b). The crossflow density contours are presented in Fig. 4.15. The leeside vortices are shown to pass freely through the intergrid boundary. Some minor distortions may be attributed to the fact, that the Cartesian section of the crossflow in Fig. 4.15a is a planar one, as opposed to that of the single C-O grid (Fig. 4.15b) which is a curved surface. Their trends and magnitudes, however, agree very well.

To test the effects of refinement and sequencing of overlapped grids, computations without multigrid are performed for the BNC case on the coarse grids, which are the (41×37×37) Cartesian grid and the (37×33×23) C-O grid (Fig. 4.16). Then the solution is prolonged to the fine grids, where the spatial steps are halved and the number of points are twice those of the coarse grids in each direction. The convergence history for the 900 iterations on the fine grid and without multigrid is also presented in Fig. 4.16. Local time stepping is used on both levels of grids with a maximum Courant number of 5. To show the effect of the multigrid computations on overlapped grids, the results of the computations on the coarse grids are restarted to use V-cycles on coarse and fine grids. The solution is obtained after 900 work units. A work unit is defined herein as the ratio of total CPU time used by the multigrid cycles to the CPU time needed for one iteration on the fine grids. It should be noted that the number of iterations indicated for the fine grid computations without multigridding is equivalent to the number of work units. Also, the 900 work units for the multigrid calculations correspond to 750 multigrid cycles for this calculation. The convergence rate of two-level multigrid calculations is approximately 0.98 as opposed to that of the fine grid computations without multigrid, which is 0.99. If more than two levels of grids were used, the reduction of convergence rate would be more dramatic. Of course, this requires the interpolation information to be generated at more than two levels of grids. Since the number of cells in the coarse composite grid is approximately one-eighth of that of the fine composite grid, the convergence rate is better on coarse grids (approximately 0.97). This is the impetus behind performing grid sequencing prior to multigrid computations. However, the solution accuracy, as indicated in Fig. 4.13, is worse on coarse grids. The computer memory requirement increases with the number of grid levels used during multigrid cycles.

The two-level multigrid computations of Case 3 require 200 megabytes (25 megawords) of computer memory as opposed to 168 megabytes (21 megawords) for the fine grid computations without multigrid. Although the diagonal inversions were performed to initialize and advance the solution, block inversions were used to complete the converged solutions. All of these full precision (64–bit words) computations are performed on the CRAY-2 computer of NASA Langely Research Center.

4.4 Case 4: Ogive-Nose-Cylinder with an Offset Sting Near a Cavity

In Case 4, a sharp ogive-nose-cylinder is positioned near a cavity with a length-to-depth ratio of 6.73 (Fig. 4.17a). This CSC configuration is adopted from the experimental investigation reported in [27]. The cavity is 29.363 in. long, 4.363 in. deep, and 5.768 in. wide. The base diameter (d) and the length of the cylinder are 1.2 in. and 24.028 in., respectively (Fig. 4.17b). The cylinder is attached to an L-shaped offset sting which has a 3.2 in. axial section and placed at about 5d above the cavity opening. In the computational model, the diameter of the axial section of the sting is increased from 0.7 in. to 1.2 in. This allowed the connection between the offset sting grid and the cylinder grid without having to refine them any further.

The computational domain of Case 4 is divided into four subdomains. The stretched Cartesian grid of the cavity is a block-structured grid with two subdomains, a lower block inside the cavity and an upper block above the cavity. Conservation across the interface of these two grids is extremely important since the cavity flow is driven by the shear layer. To ensure the conservation, the grid lines are contiguous normal to the interface of these two blocks, where the solutions are matched. A boundary fitted H-O grid is wrapped around the cylinder and the sting is contained in an O-H grid. The cylinder grid is embedded in the upper Cartesian grid. The end surface of the O-H grid containing the sting root is coincident with the cylinder surface. A summary of the composite grid is given in Table 4.2.

As reported in Ref. 28, the flow past a cavity is near symmetric. Therefore, the flow is assumed symmetric with respect to the center plane of the cavity and the cylinder. This allows computations to be performed only for one half of the physical domain, resulting in substantial savings in computer time and memory. This composite grid is shown in Fig. 4.18. Portions of the overlapped regions are marked by the solid circular dots. Computations for this case as well as the initialization of the flows past each component are also performed on a CRAY-2 computer and the required memory is 136 megabytes (17 megawords).

The objectives of this case are three fold: (i) to demonstrate the versatility of the current algorithm since the configuration considered here consists of multiple and geometrically nonsimilar components; (ii) to benchmark the computer code by comparison to available experimental data [27]; (iii) to assess the interference effect of the cylinder on the unsteady cavity flowfield as the cylinder is positioned near the cavity. The inclusion of the sting in the calculation of this case further demonstrates the ability of the present algorithm to handle a complicated configuration, such as a cylinder-sting assembly in which solid surfaces meet. The main reason, however, is to simulate the complete experimental configuration designed for the validation of the computer code developed with this algorithm.

Table 4.2 Component grids of Case 4.

Grid 1: upper Cartesian grid:	103×28×57
Grid 2: lower Cartesian grid:	81×23×43
Grid 3: H-O grid for cylinder:	121×29×33
Grid 4: O-H grid for sting:	21×25×57
Total grid points:	390,219

The flow is turbulent at M=1.65, Re/ft=2×10^6, α=0°, P_t=1092 lbs/ft^2, T_t= 584° R. The ratio of the boundary layer thickness to the depth of the cavity (δ/D) at the front lip is 0.0916.

As discussed in Sec. 2.6, the solutions for the clean cavity flow and the cylinder-sting in freestream are obtained independently. These solutions are then combined and used as the initial solutions for the interference flow problem of the CSC configuration. The diagonal inversions approximation (Sec .2.5.2) are used for the initialization process of cylinder-sting in freestream (no cavity). However, in the clean cavity calculations and the simulations of the CSC configuration standard block inversions are performed for the sake of higher accuracy.

Two characteristic times are defined for two different reasons. The first characteristic time is defined as the maximum of all the times obtained as the ratio of a cavity dimension, l (length or width) to the corresponding component of the freestream velocity, t_{c1} = max(l/u). It is used as a time scale for this unsteady flow. The numerical transients due to the nonphysical initialization of the clean cavity computational domain are assumed to be removed after running the solution algorithm for (4 t_{c1}). The second characteristic time is defined as the ratio of the minimum length of the discretization stencil used by the solution algorithm to the freestream value of the speed of sound, $t_{c2} = \min(5\times\Delta x_m)/a_\infty$. Note that this second-order algorithm uses five-point stencils in each direction. It is used to determine the physical limitation set on the time step size of the solution algorithm which is advanced time accurately. This practice ensures capturing numerically the pressure fluctuations of the cavity flow.

The values of t_{c1} and t_{c2} for the present cavity are 1.59 and 0.073 milliseconds, respectively. The computer time needed to advance the flow for one t_{c1} was about twenty hours. The maximum time step used was 0.26 microseconds. The unit CPU time for this algorithm was 33.4 microseconds per grid point per time step.

The interference flow between the components of the CSC is observed in less than 1 t_{c1} after the advanced initialization. However, the solution is advanced for nearly 2 t_{c1} to match the experimental data. It should be noted that in the experiment, a steady flow measurement technique was used to measure unsteady flows. Measurement of an unsteady flow would require a different instrumentation. In any event, the measured surface pressure data were averaged values of over 100 measurements obtained within a time span of 2 to 3 seconds. An attempt is made here to mimic the experimental process. The surface pressure distribution in this study were averaged over a period of approximately 3 milliseconds. Since the computational time step is 0.26 microseconds, the averaging was done over 11,500 computed values.

Computed values of the time averaged pressure coefficient (C_p) on the cavity surface and the cylinder are compared with the experimental data (Fig. 4.19). The C_p distributions plotted along the horizontal centerline of the floor (designated as F in Fig. 4.19) and the vertical centerline of the rear face (designated RF) indicate that the pressures are near the freestream value on most of the floor, but they increase near and along the rear face. The same trend is observed for the side wall (designated SW). The longitudinal C_p distribution on the inboard ($\eta = 180°$) and the outboard ($\eta = 0°$) surfaces of the cylinder show the sharp increase owing to the nose shock followed by the expansion around the forebody. The computed C_p values compare very well with the experimental data at most locations. It should be noted that these values vary with time for this unsteady flow and the only basis for comparison is through time averaging. Therefore, the initial time and the elapsed time of the averaging process become variables both for the computations and the measurements. The computed values at a few locations appear slightly higher than the data in Fig. 4.19. However, the agreement becomes better at some locations with longer elapsed times of averaging. Since the flow fluctuations are known to be non-periodic at supersonic speeds [29–33], a true characteristic time for averaging is not evident.

The instantaneous streamwise Mach number contours at the plane of symmetry (y = 0.0) are displayed in Fig. 4.20. Evident in this figure is the undulation of the free shear layer which is formed between the high speed external flow and the slower internal flow. For a clean cavity (no interference) with L/D = 6.7, the shear layer is shown to be inherently unstable and continuously pulled into and pushed out of the cavity as a result of the pressure oscillations inside the cavity [30–33]. The shear layer is deflected down when the pressure inside the cavity is below the freestream pressure. At this instant, mass is being pumped into the cavity. This mass is then slowed down by various dissipative processes within the cavity and this in turn increases the cavity pressure to above the freestream value. The shear layer is then deflected out of the cavity by the excess pressure and mass is pumped out of the cavity (Fig. 4.20). However, this cycle is changed when a cylinder is placed near the cavity. The pressure of the flow between the shear layer and the inboard side of the cylinder is increased due to the nose shock of the cylinder and its reflection off of the shear layer. The rise in the pressure value then causes the shear layer to deflect downward deep inside the cavity. Figure 4.21 illustrates this phenomena with a sequence of cross flow total pressure contours over a period of 1.5 ms. The shear layer is shown to be three-dimensional as its shape varies in the lateral and streamwise directions. The formation of the wake flow at the base of the cylinder and the blunt trailing edge of the sting and its interference with the shear layer are also shown in Fig. 4.20.

The vortical nature of the shear layer is depicted via the pressure contours in Fig. 4.22. Also shown in this figure are the expansion at the sharp corner of the front face and the compression at the rear face of the cavity. Also shown is the nose shock on the cylinder and impingement of this shock on the shear layer. The flow over a cylinder at $\alpha = 0^\circ$ without cavity interference is typically a steady flow, i.e., the computed force and moment coefficients of the cylinder remain constant after reaching convergence. The unsteady nature of the cylinder-near-cavity flow is captured computationally and is presented in Fig. 4.23. The oscillation of the C_N due to this unsteadiness is observed over a period of 1.59 ms. The computed mean value of the C_A and C_N of the cylinder over this period are (0.3532) and (0.0056), respectively. The experimentally measured values [27] of C_A, and C_N are (0.3756) and (– 0.0258), respectively.

4.5 Case 5: Ogive-Nose-Cylinder Inside a Cavity

The configuration of Case 5 is similar to that of the previous case with the exception that the offset sting is not included. In this case, the cylinder is placed (1.67d) below the cavity opening. The component grids of Case 4 are put together to form the composite grid for this case without having to change the subdomain grids. In fact, this is one of the advantages of the current algorithm in which the component grids are reusable. The composite grid for Case 5 is shown in Fig. 4.24. The rationale behind Case 4 also applies to this case. The interference patterns are known to be significantly different, however, when the cylinder is placed inside a cavity. The configuration for Case 5 does not contain the sting, in an attempt to gain a better understanding of the flow between the cylinder base and the cavity rear face. Besides, such a sting certainly does not exist when a store is carried internally inside a weapons bay.

The instantaneous total pressure contours of the symmetry plane show the complicated flow structure of Case 5 (Fig. 4.25). The shear layer separates the supersonic external flow and the low speed flow inside the cavity. The nose shock on the cylinder apparent in Case 4 does not occur here due to the low speed flow inside the cavity. The pressure difference between the external and the internal flows just downstream of the forebody junction causes the inward deflection of the shear layer and its impingement on the outboard surface (η = 0°) of the cylinder. The wavy motion of the shear layer causes it to impinge on the cylinder again and then deflect out of the cavity due to higher pressure near the rear face. At this instant, some mass of air is pumped into the cavity from behind the base of the cylinder. This is shown by the instantaneous velocity vectors of the symmetry plane inside the cavity (Fig. 4.26a). For clarity, only every other point in the streamwise direction has been plotted. As the flow streams into the cavity through the opening between the base and the rear face, it is decelerated down by the dissipative processes inside the cavity and it separates near the floor. This clockwise rotating vortex is also illustrated through the instantaneous streamlines of the symmetry plane (Fig. 4.26b). Shown in Figs. 4.26a-b are a relatively weak vortex structure near the front face and a counter clockwise rotating vortex near the floor downstream of the forebody junction. The flow is shown to separate on the outboard side of the cylinder ($\eta = 0°$) just prior to the first impingement and immediately after the second impingement.

The unsteady nature of the flowfield and the interaction of the shear layer with the cylinder is illustrated in Fig. 4.27. Mach number contours of the symmetry plane at five consecutive time levels over a period of 1 ms is shown in Fig. 4.27. The shape of the shear layer and its points of contact with the cylinder change constantly with time. This motion of the shear layer and its interaction with the rear lip of the cavity was shown to generate shedding of vortices in an earlier study [32].

Time averaged pressure coefficients on various surfaces of the cavity and the cylinder are presented in Fig. 4.28. The trends of computed C_p distribution on most surfaces agree very well with the data [27]. The discrepancy in the distribution on rearface (RF) is attributed to the sting, which exists in the experimental model but not in the computational model. The computed C_p at the vertical centerline of rear face (RF) is slightly higher than the experimental data of ref. [27]. This effect is also evident on the surface of the cylinder. The mean value of C_A and C_N for the cylinder over a period of 1.59 ms are (0.0580) and (– 0.0551), respectively. The experimental [27] value for C_A is (0.0283) and for C_N is (– 0.0611). The variation of C_N over this period was illustrated in Fig. 4.23. The difference between a high instantaneous value and a low instantaneous value of C_N, which may be interpreted as the amplitude, is much larger for

Case 5 then that of Case 4. This is somewhat expected since the unsteadiness is much more pronounced inside the cavity and that is where the cylinder is placed for Case 5.

4.6 Chapter Summary

Three preliminary and preparatory studies on two-dimensional profiles and a body-of-revolution have been conducted. These studies have been perfonned to implement the solution algorithm presented in Chaps. 2 and 3 and to prepare for complex interference flows involving cylinders. The time-accurate simulation of a supersonic flow past a CSC configuration has been presented in Chap. 4. Cases 4 and 5 represent two dramatically different interference patterns. The results for Cases 4 and 5 have been presented to depict this complex flow and help understanding the interference effects. The unsteady nature of these flowfields has been simulated and illustrated in this chapter.

Some conclusions and recommendations on the algorithm presented in Chaps. 2 and 3 and the results presented in Chap. 4 are given in Chap. 5.

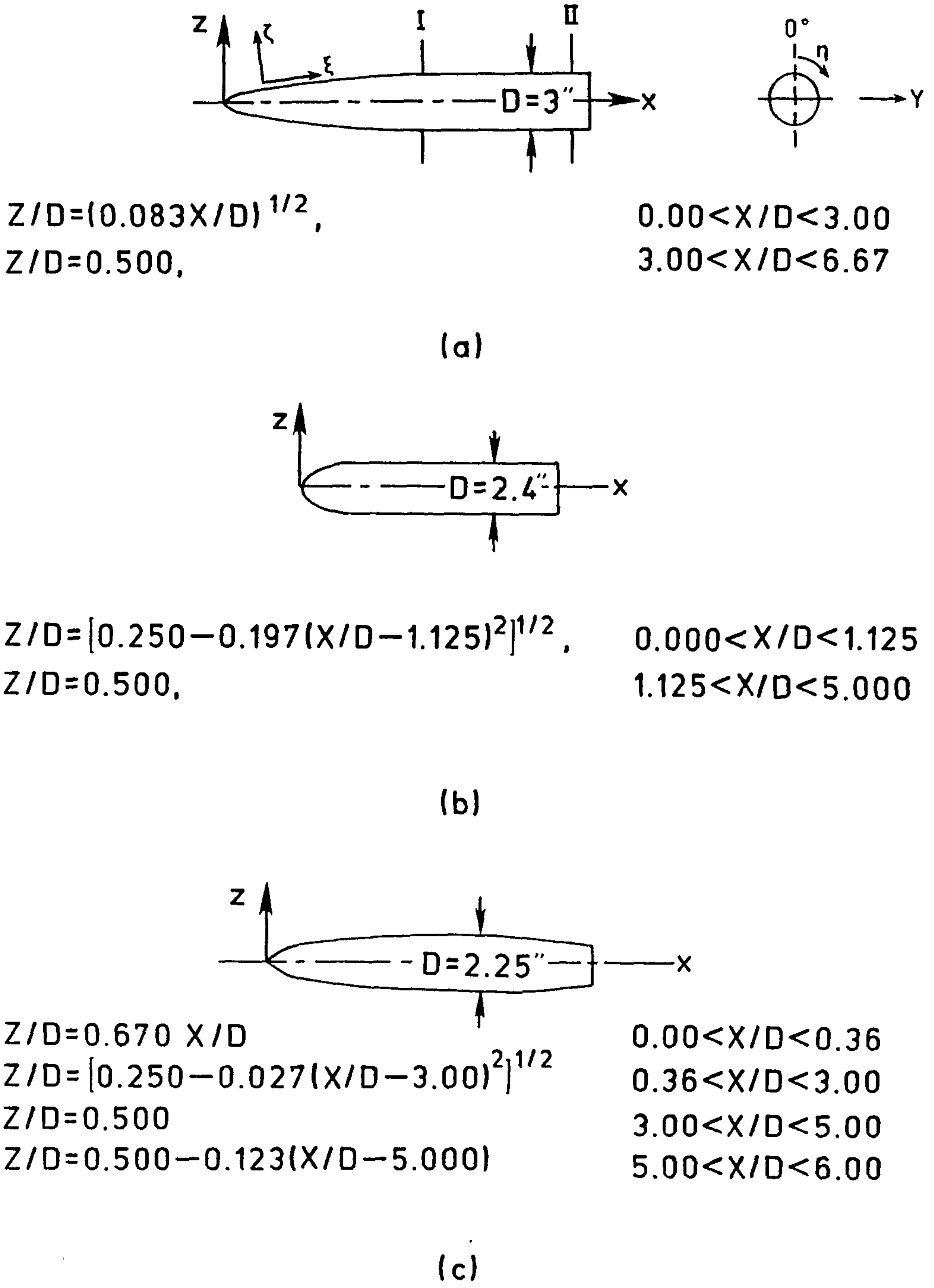


Fig. 4.1 Schematic diagrams and analytical expressions of the two-dimensional profiles (a) Model I (b) Model II, (c) Model III.

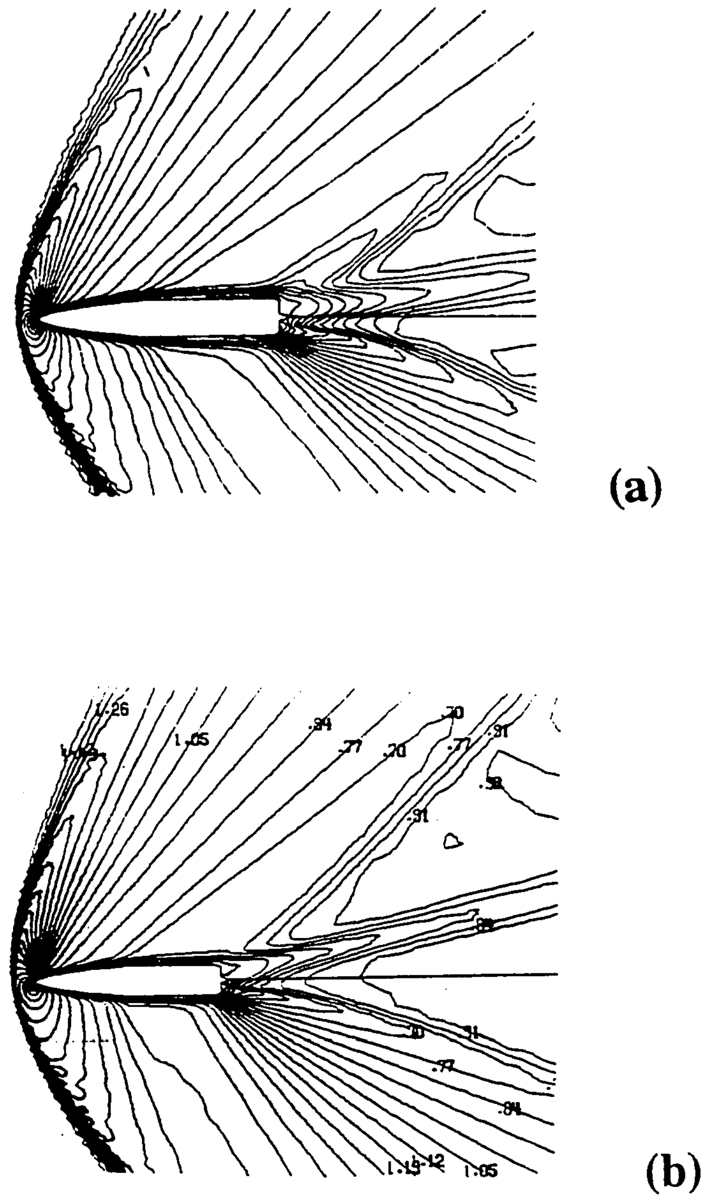


Fig. 4.2 Normalized density contours of Model I at (a) α = 8°, b) α = 12°.

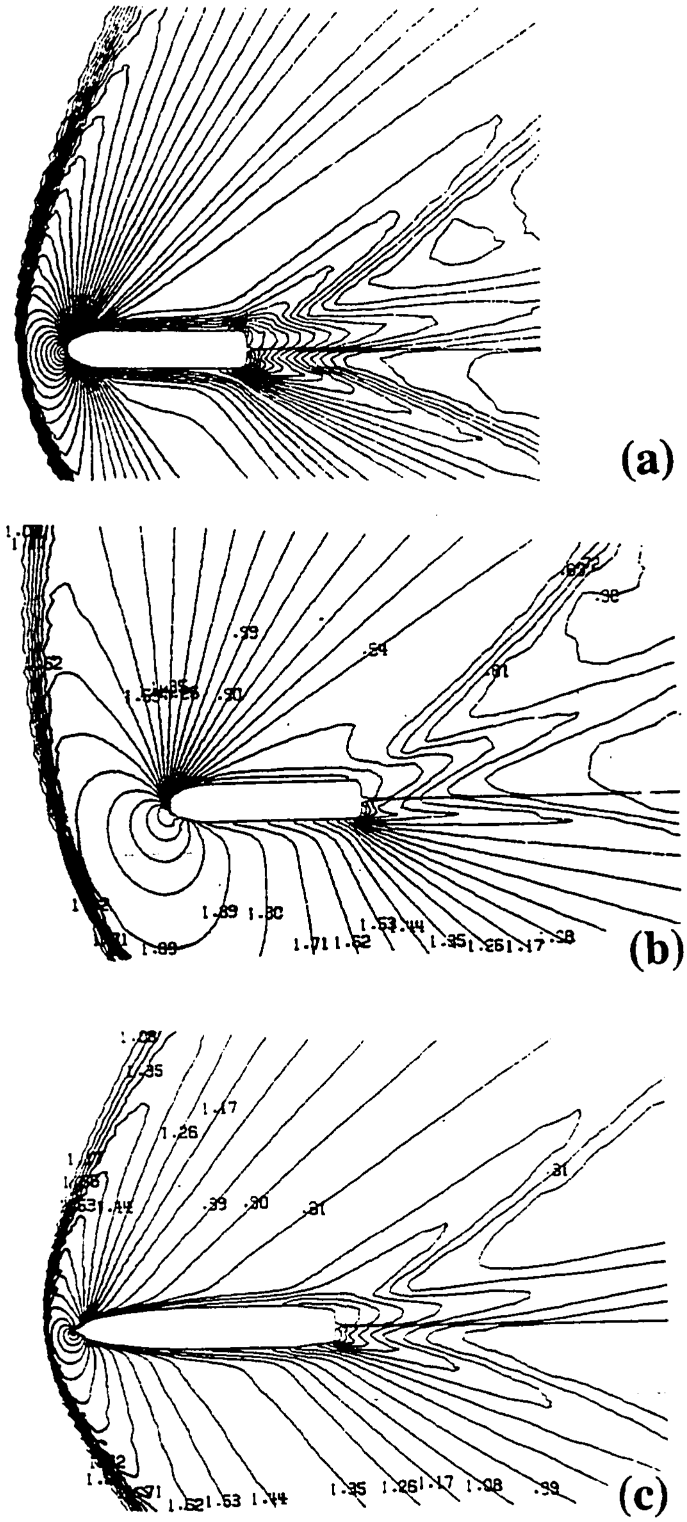


Fig. 4.3 Normalized density contours of (a) Model II at α = 8°, (b) α = 20°, (c) Model III at α = 8°.

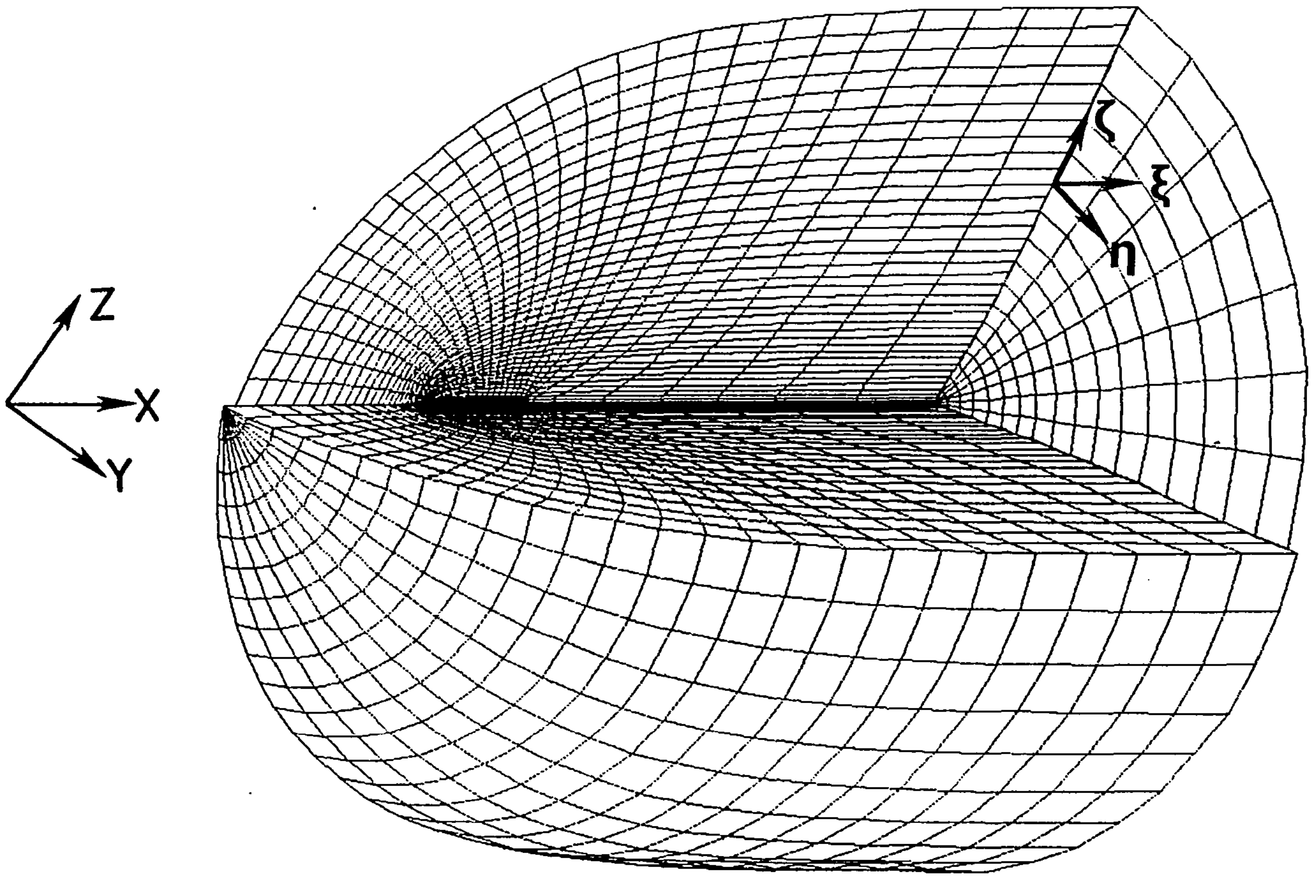


Fig. 4.4 A representative grid section of blunt-nose-cylinder of Case 2.

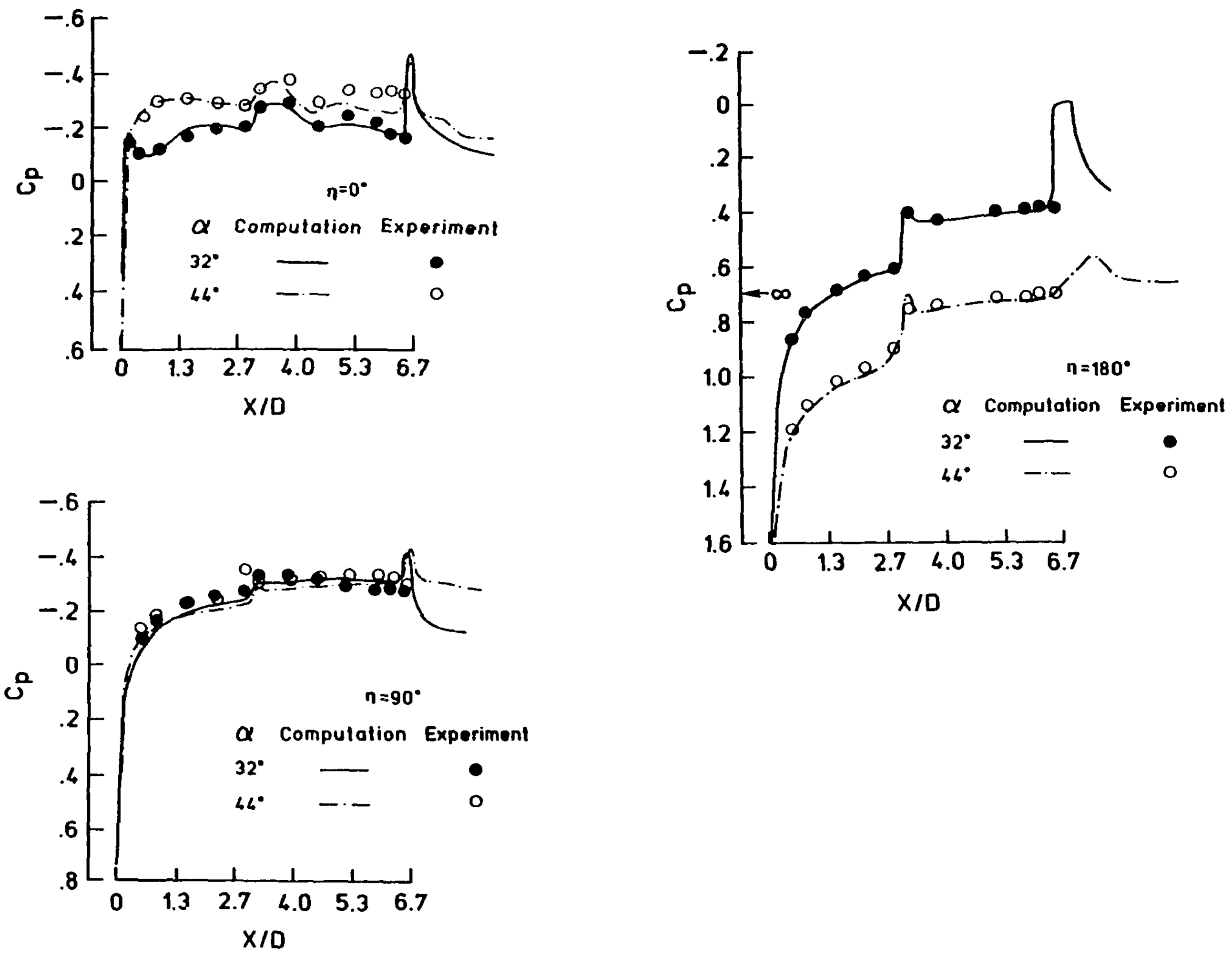


Fig. 4.5 Longitudinal C_p distributions of α = 32° and α = 44° flows in comparison with experimental data of [67].

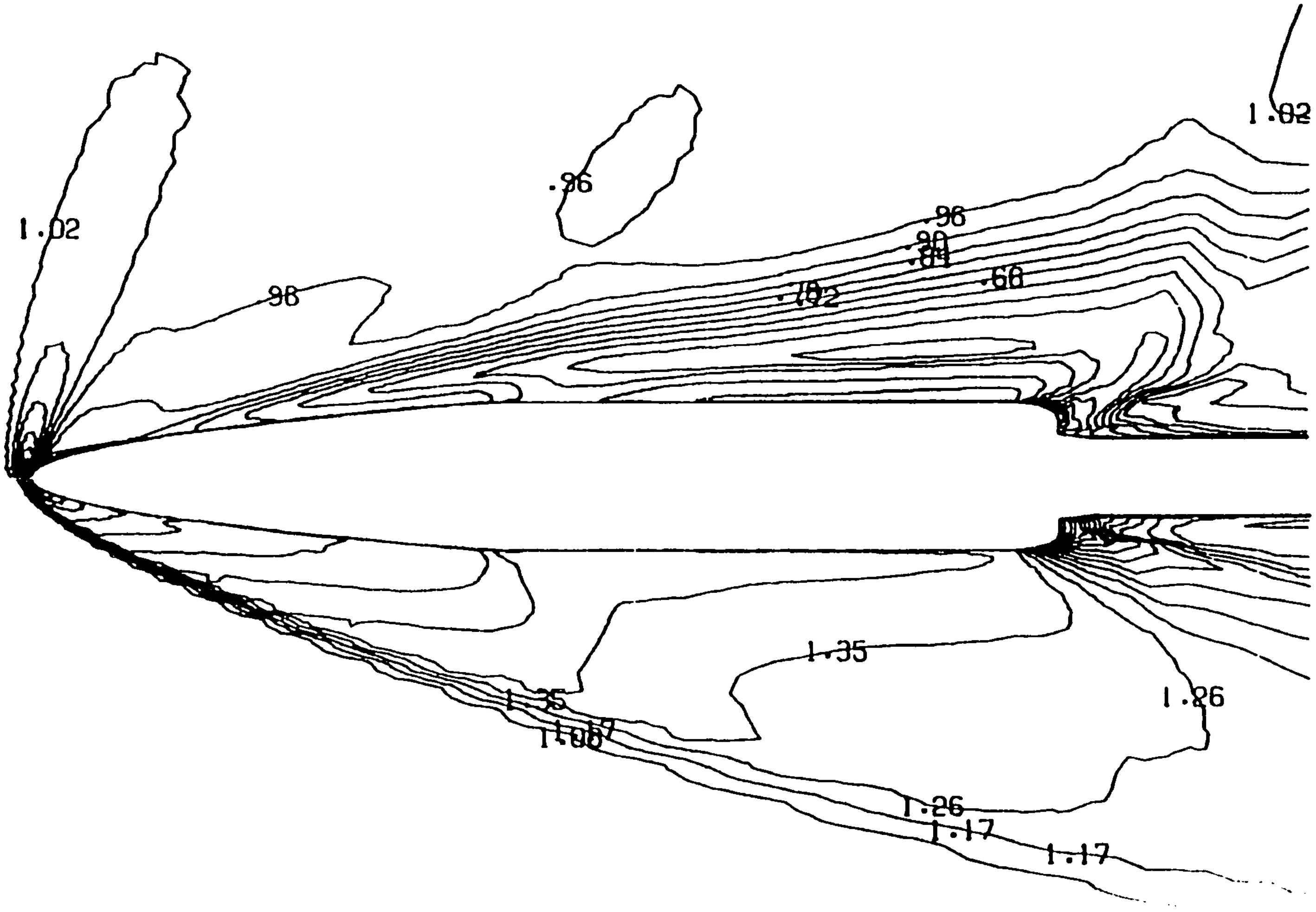


Fig. 4.6 Normalized density contours of the symmetry plane for α = 32° of Case 2.

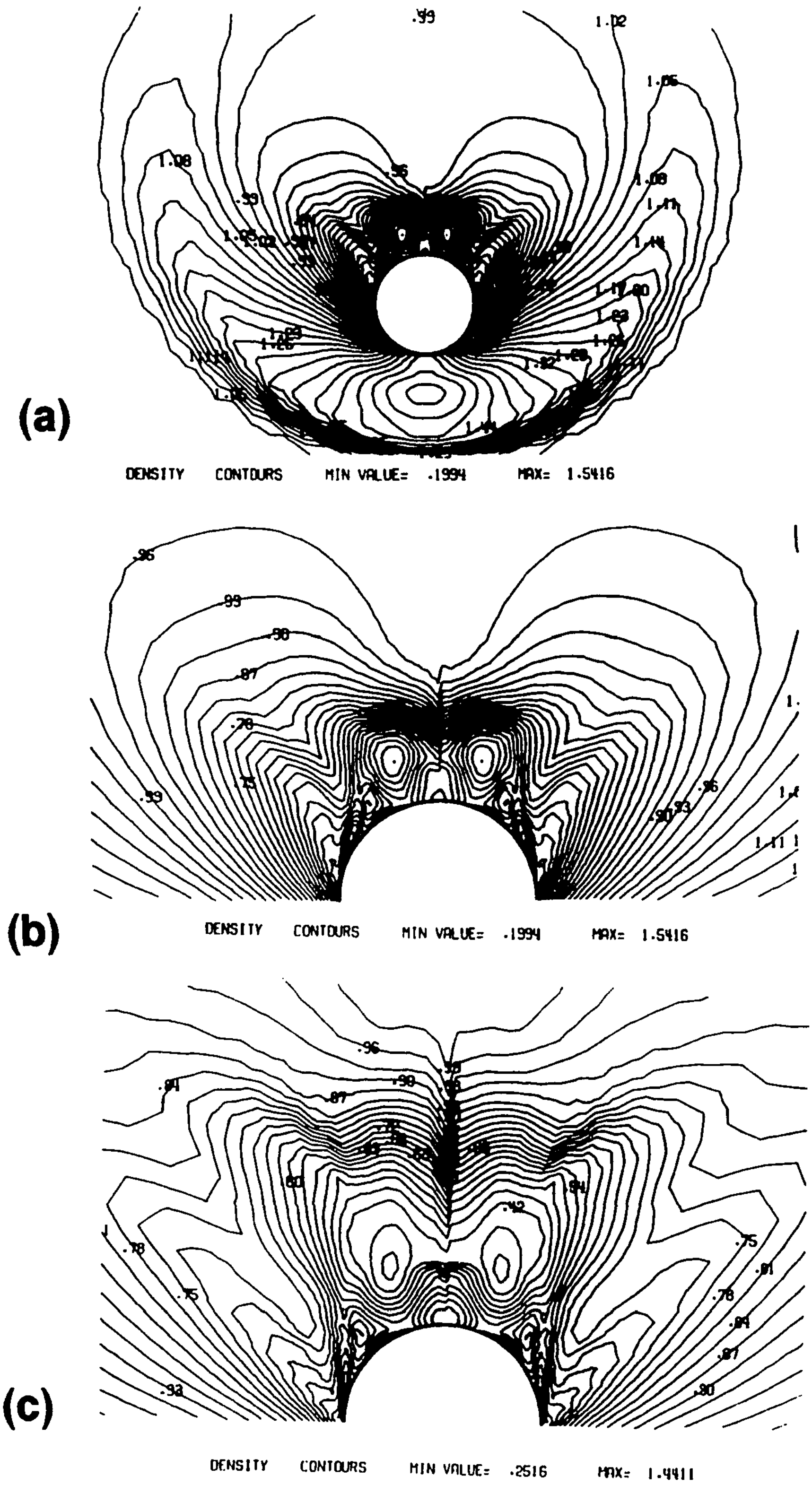


Fig. 4.7 Normalized crossflow density contours for α = 32° (a) Station I; (b) enlarged view of leeside flow at Station I; (c) Station II.

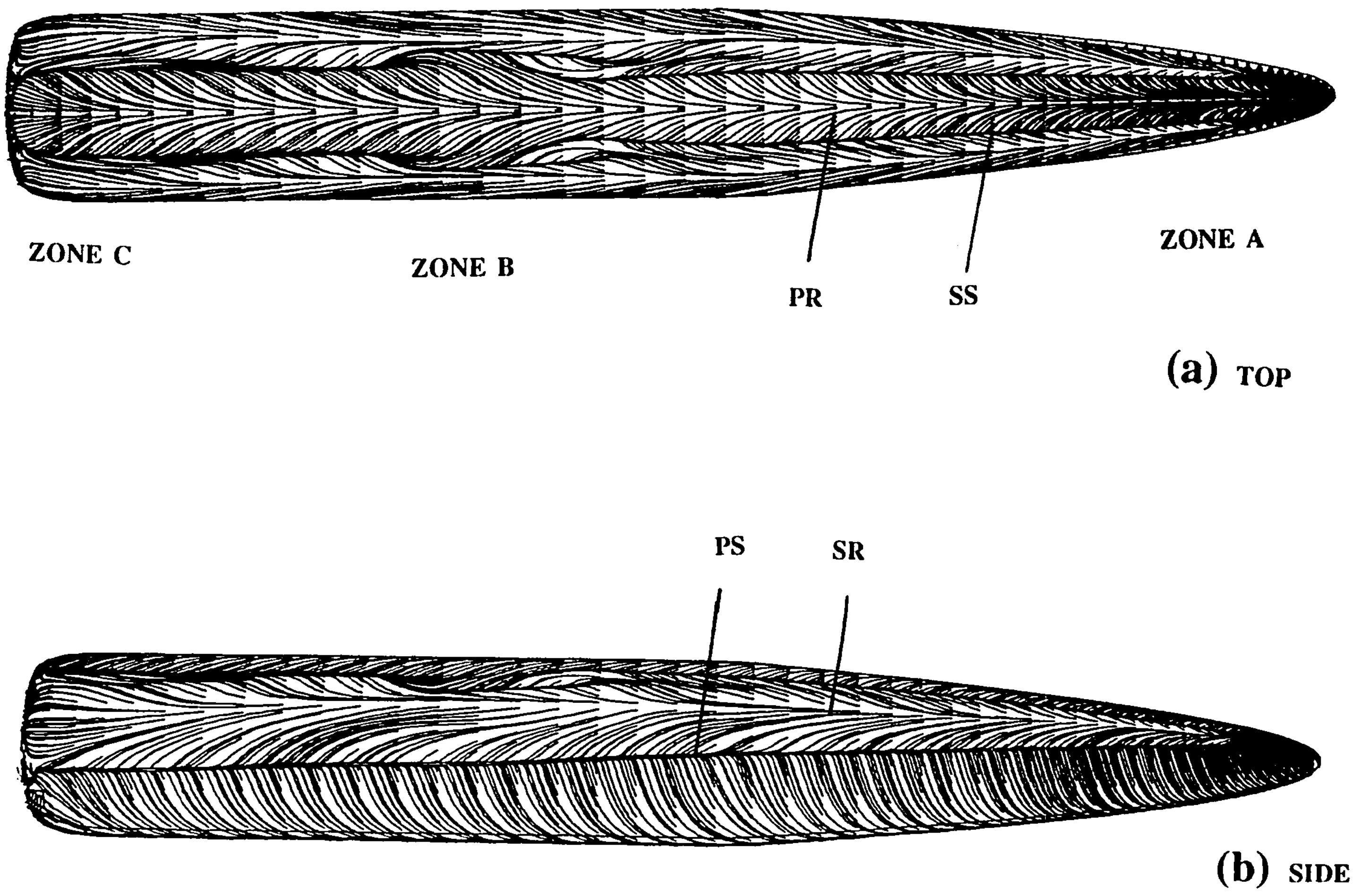
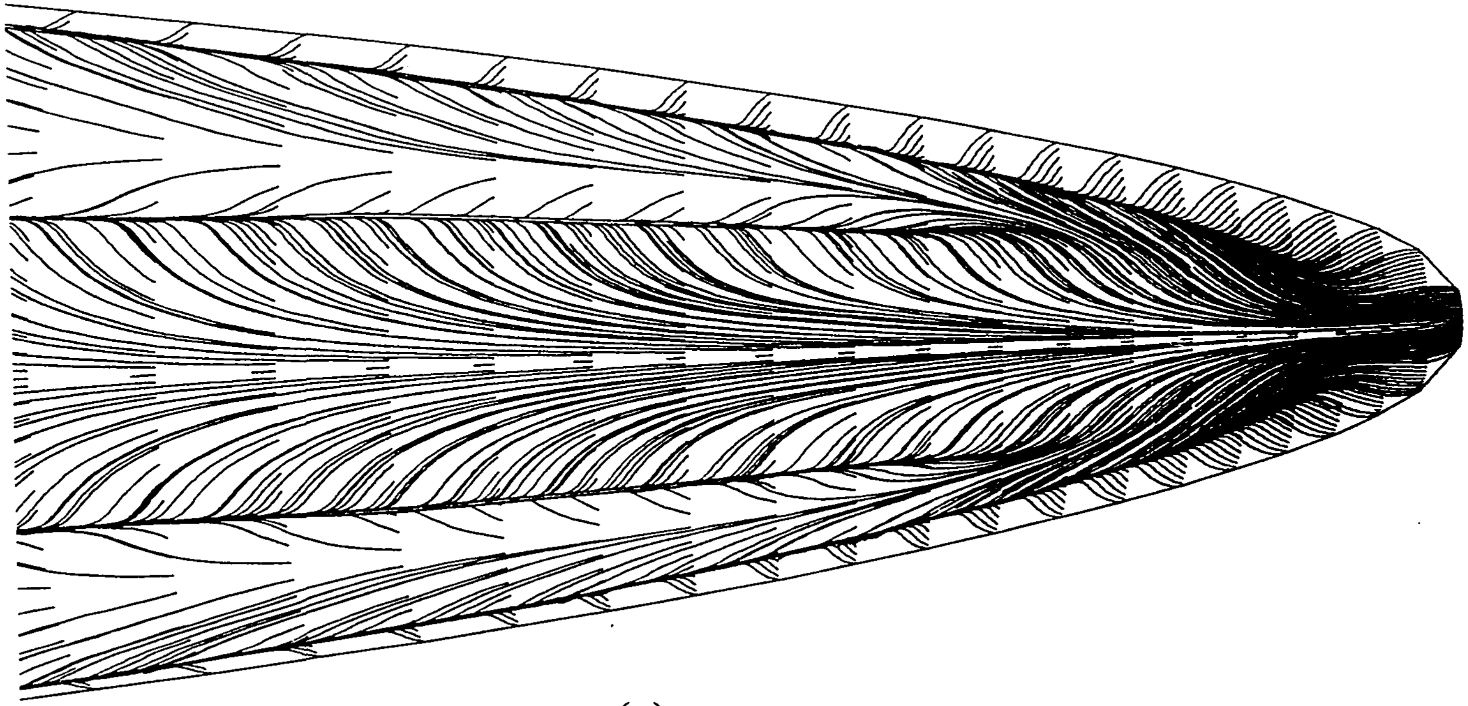
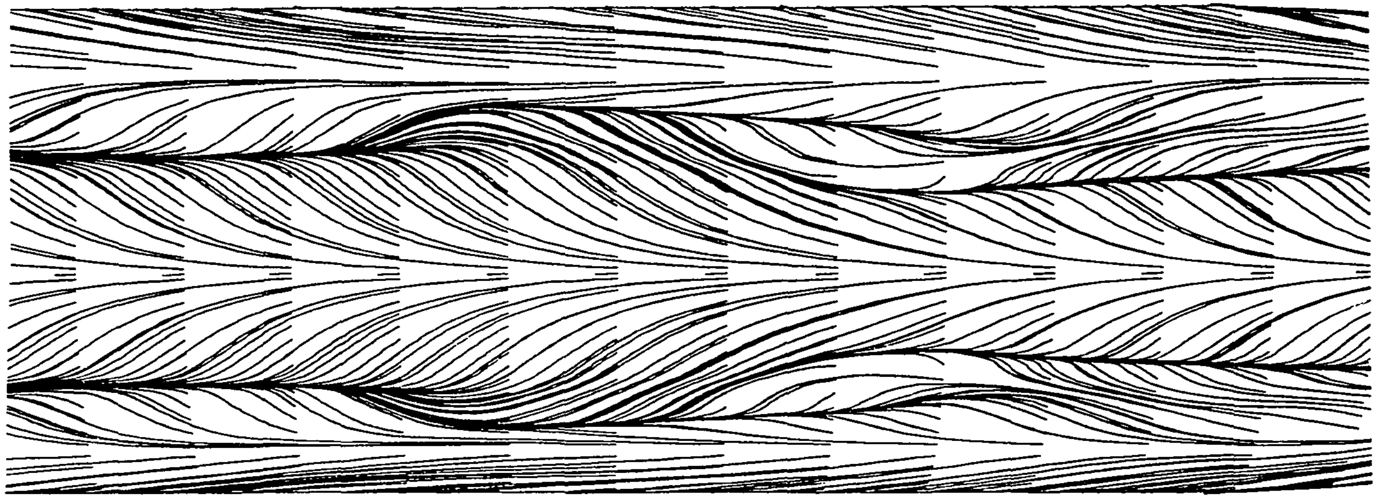


Fig. 4.8 Limiting streamlines of $\alpha = 32°$, (a) top, (b) side, (c) enlarged view of zone A (d) zone B, (e) zone C.



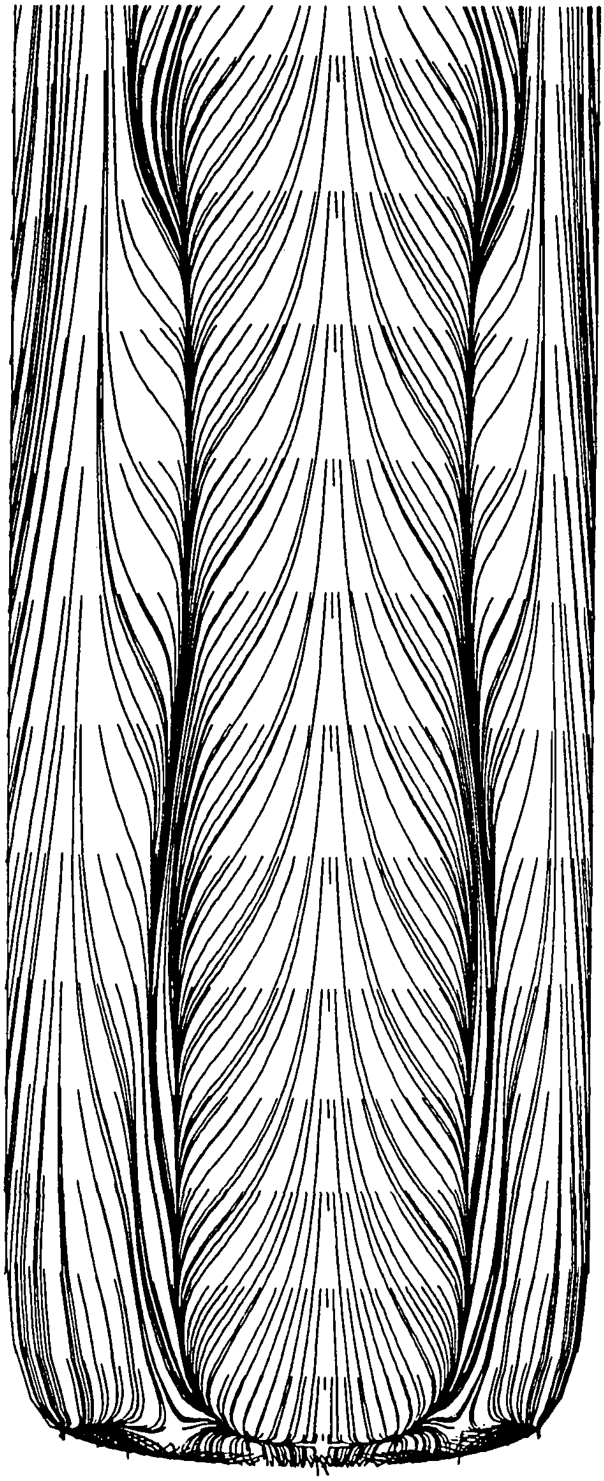
(c) ZONE A

Fig. 8 (Continued).



(d) ZONE B

Fig. 8 (Continued).



(e) ZONE C

Fig. 8 (Concluded).

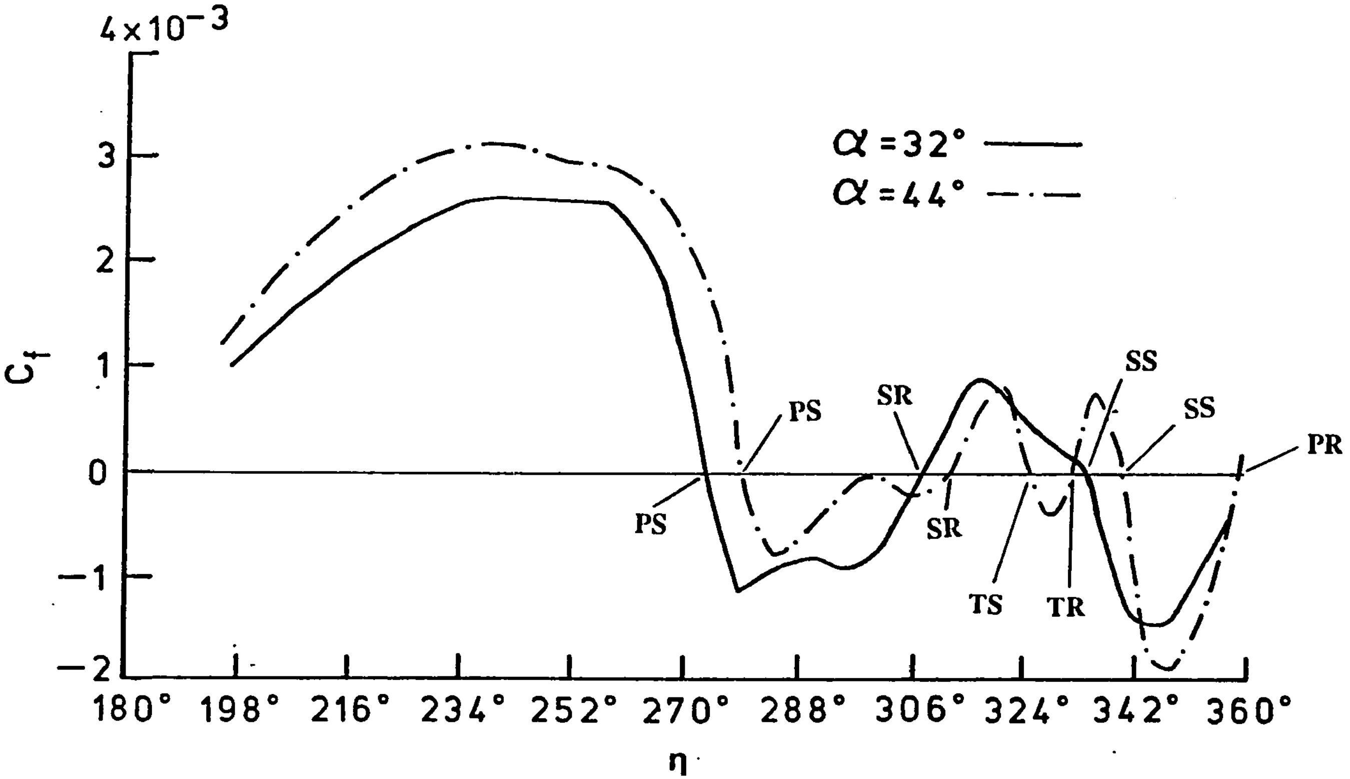


Fig. 4.9 Circumferential skin friction distributions at Station II for $\alpha = 32°$ and $\alpha = 44°$.

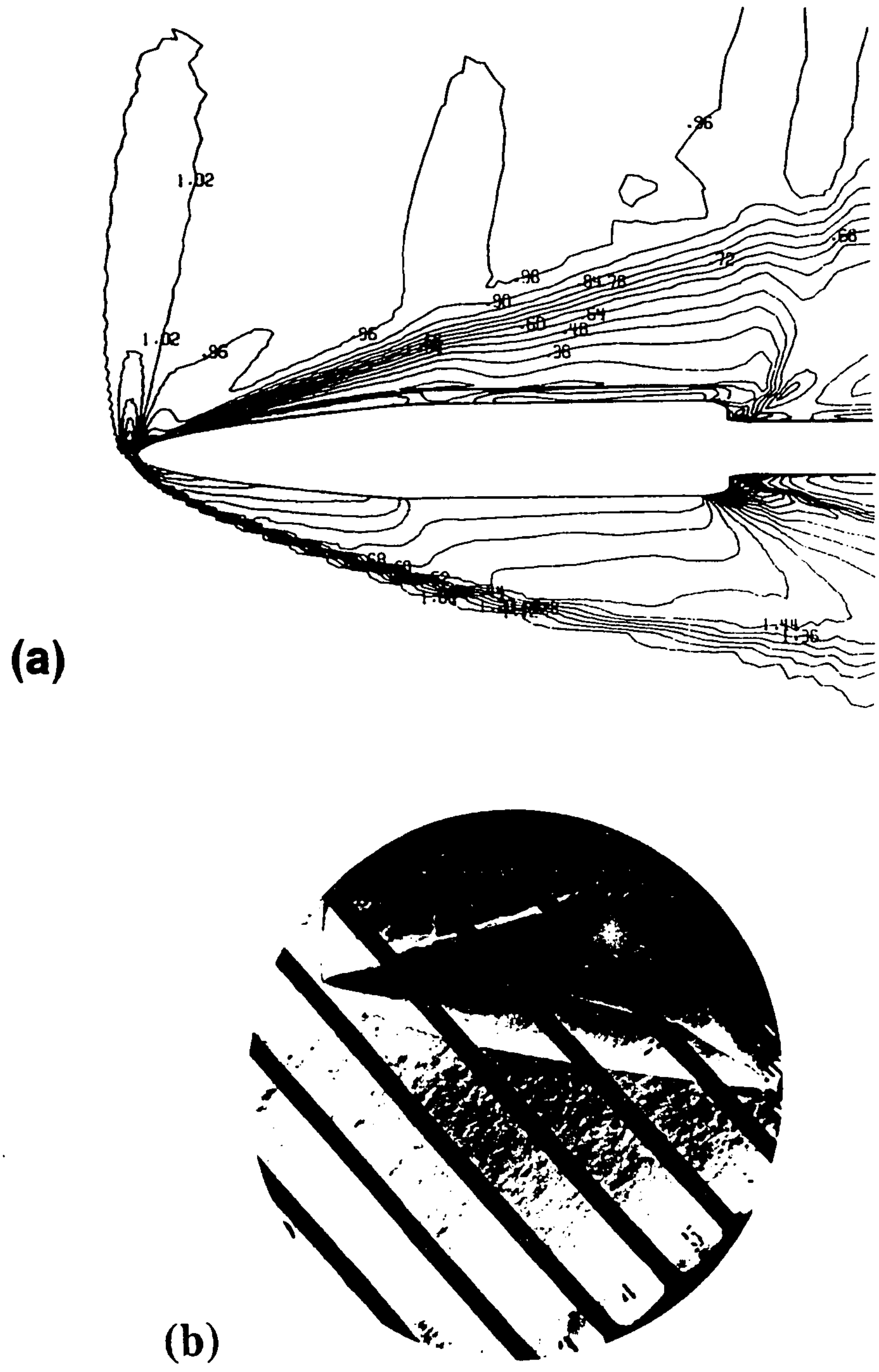
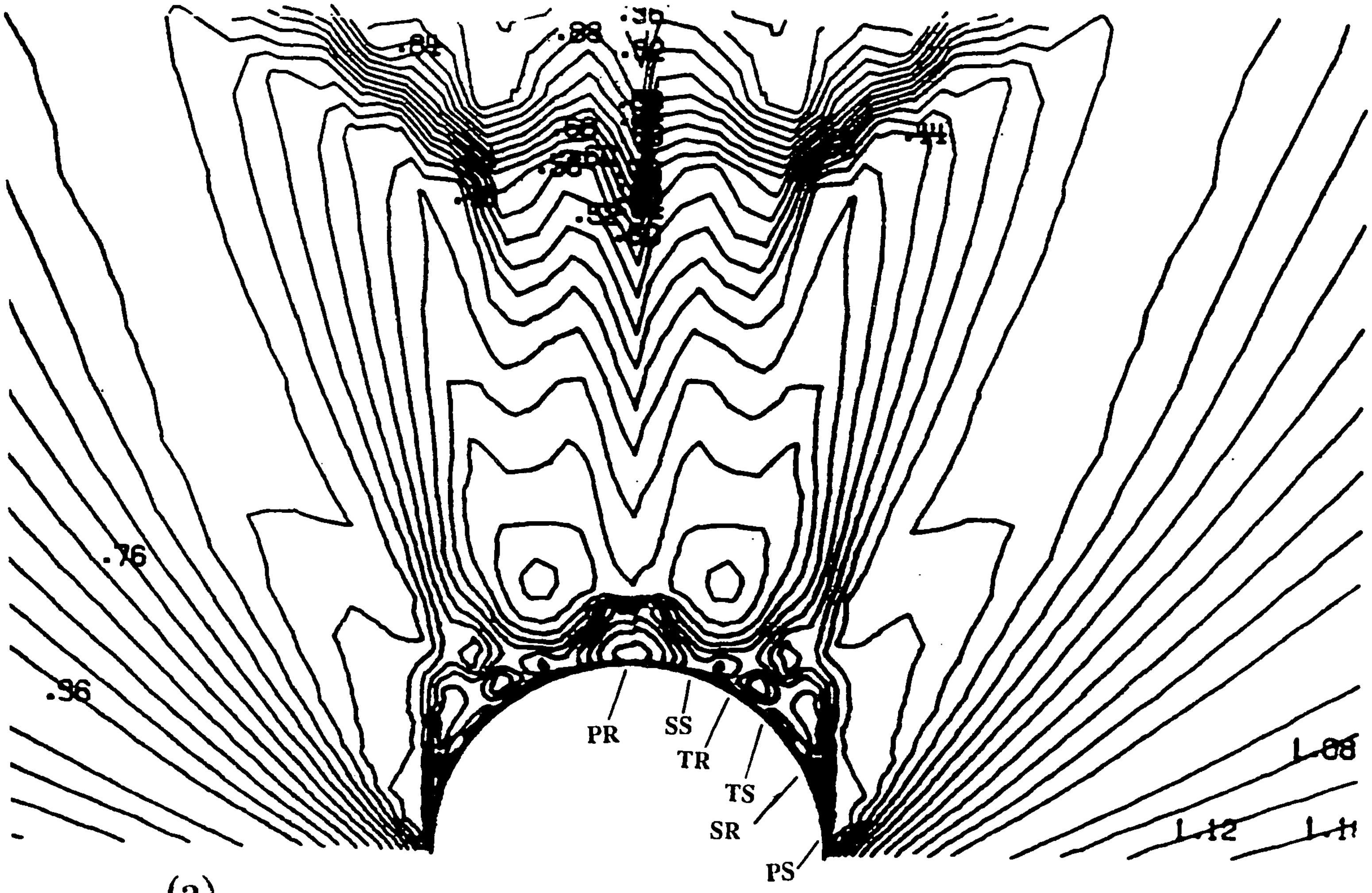


Fig. 4.10 (a) Normalized density contours; and (b) Schlieren photograph [67] of the symmetry plane for $\alpha = 44°$ of Case 2.



(a)

Fig. 4.11 (a) Crossflow density contours, (b) crossflow velocity vectors at Station II for α = 44° of Case 2.

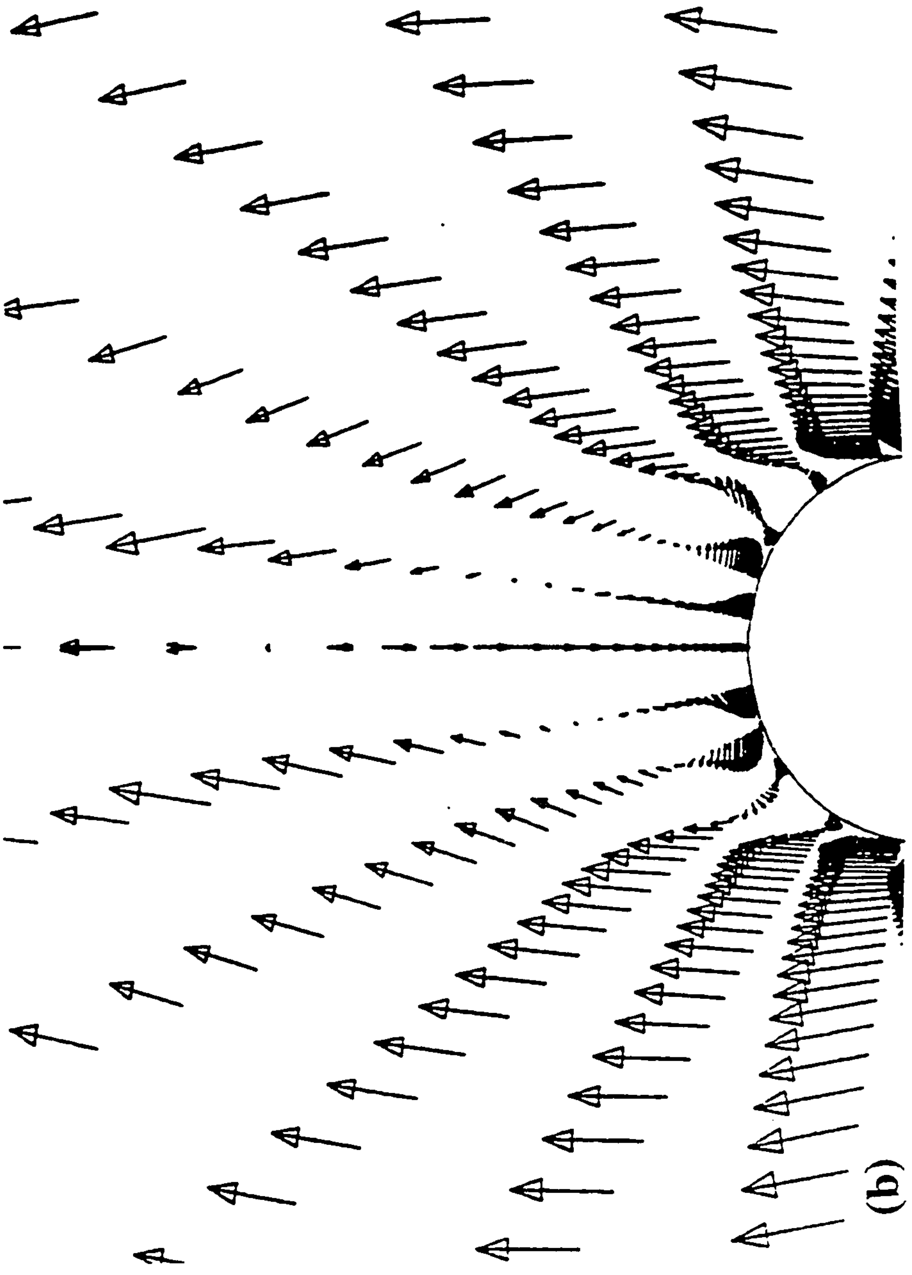


Fig. 4.11 (Concluded).

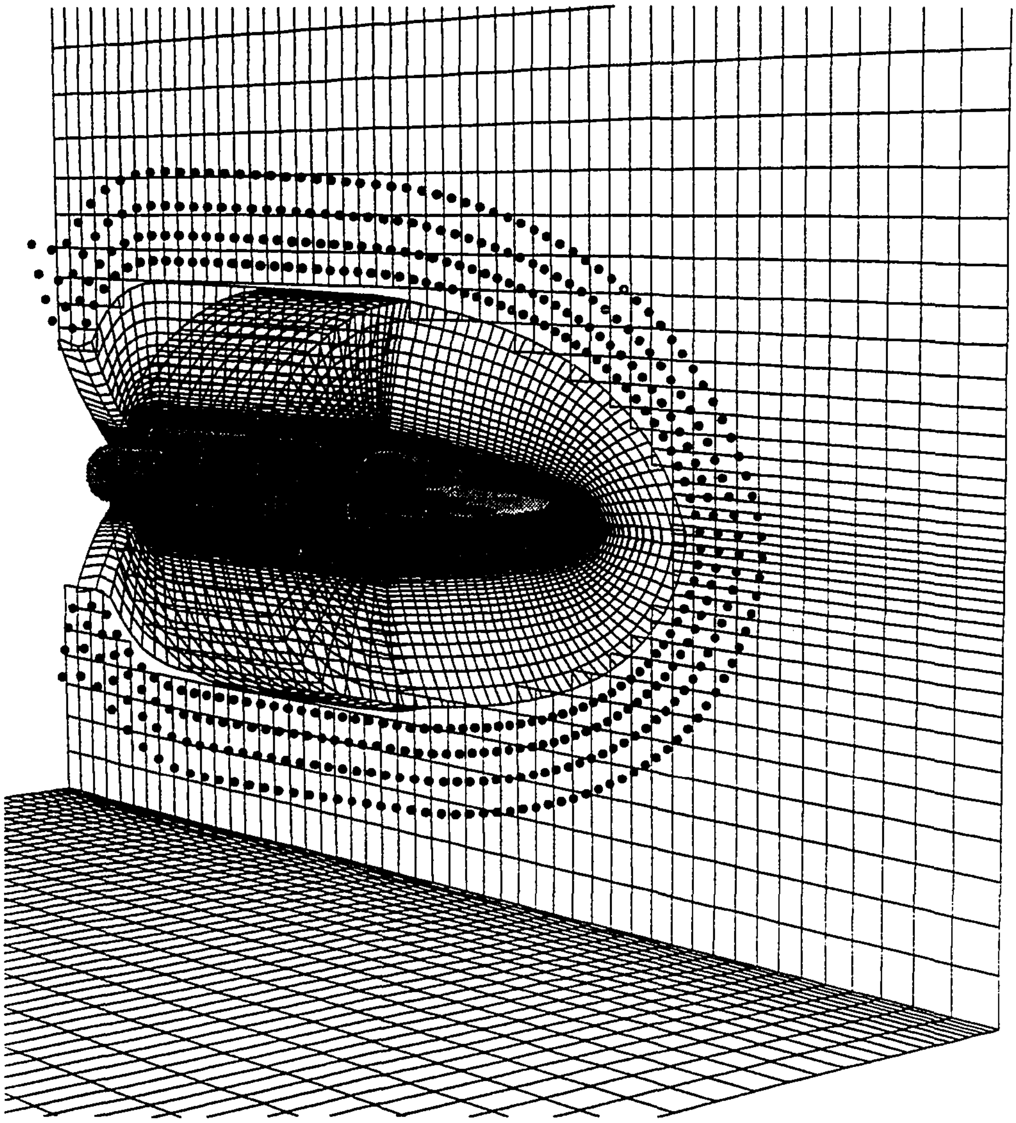


Fig. 4.12 Composite grid the blunt-nose-cylinder of Case 3.

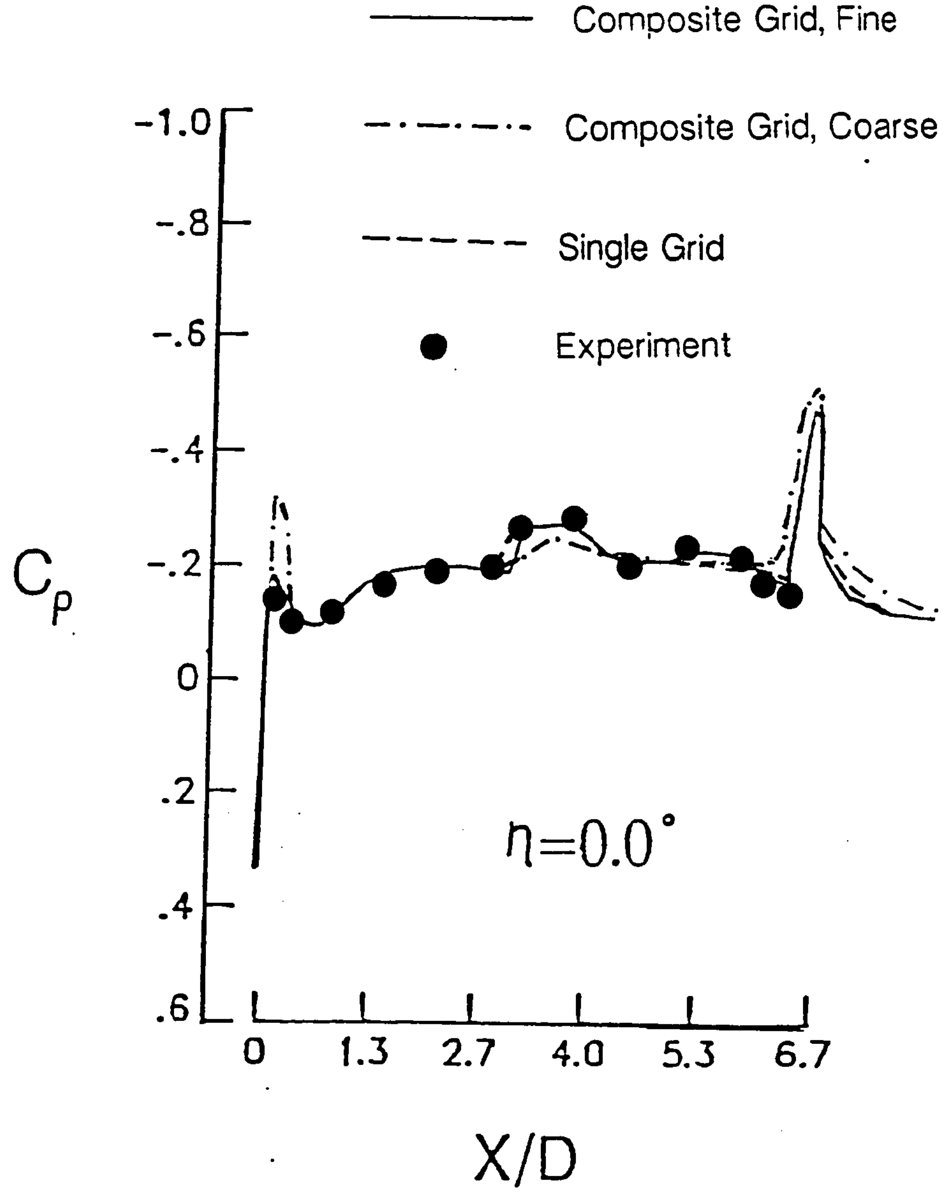


Fig. 4.13 Longitudinal surface pressure coefficient (C_p) distribution on the leeside of BNC with the composite grid of Case 3 in comparison with the single grid of Case 2 and the experimental data of [67].

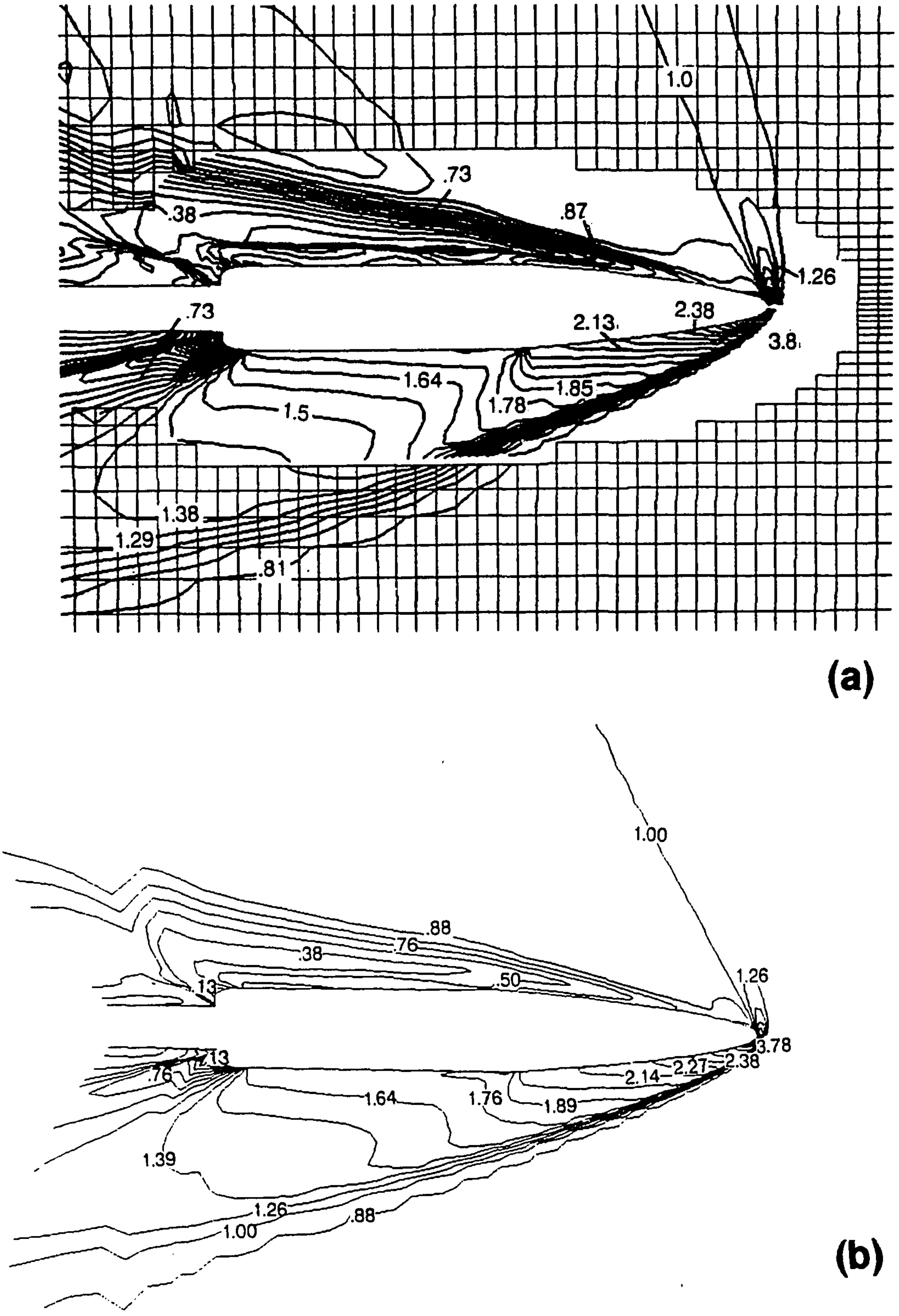


Fig. 4.14 Normalized pressure contours on the symmetry plane of BNC computed on (a) the composite grid of Case 3 (b) the single C-O grid of Case 2.

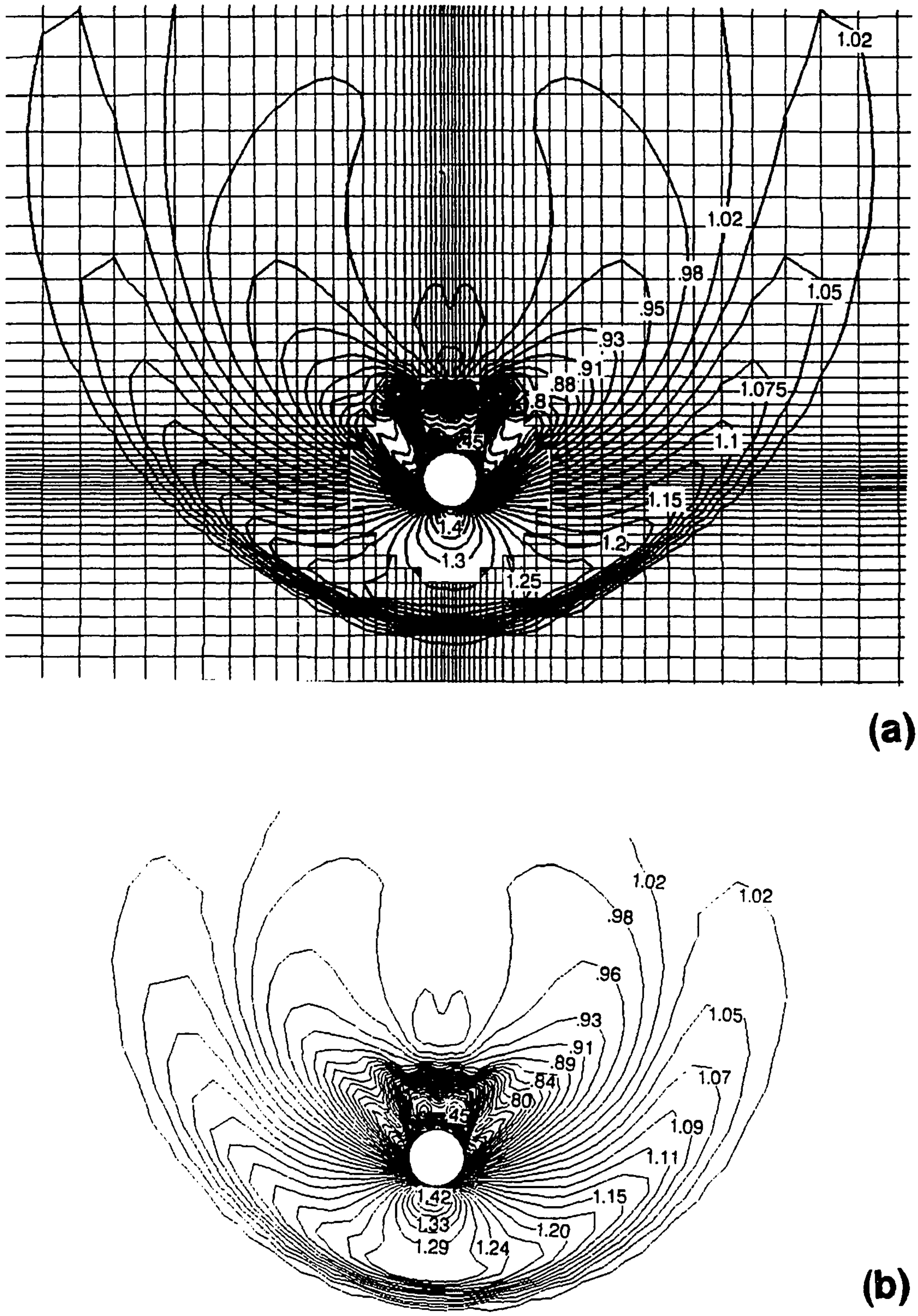


Fig. 4.15 Normalized density contours on the crossflow Plane II of BNC computed on (a) the composite grid of Case 3 (b) the single C-O of Case 2.

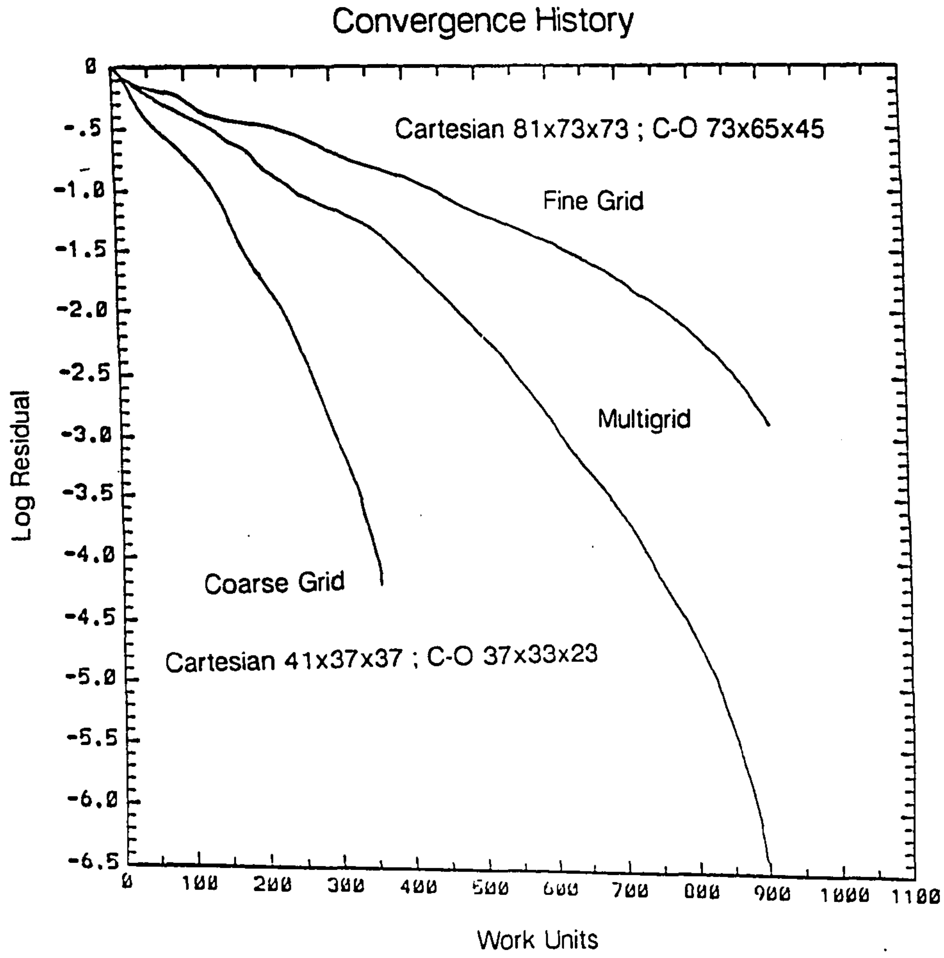


Fig. 4.16 Convergence histories for the computations of Case 3.

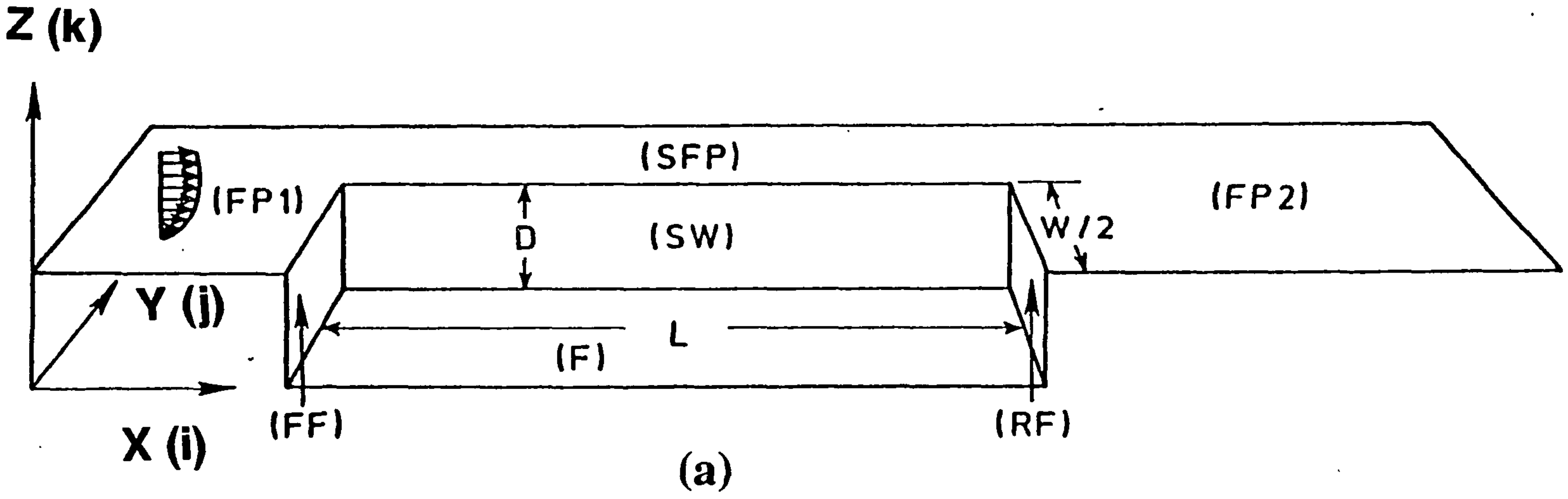
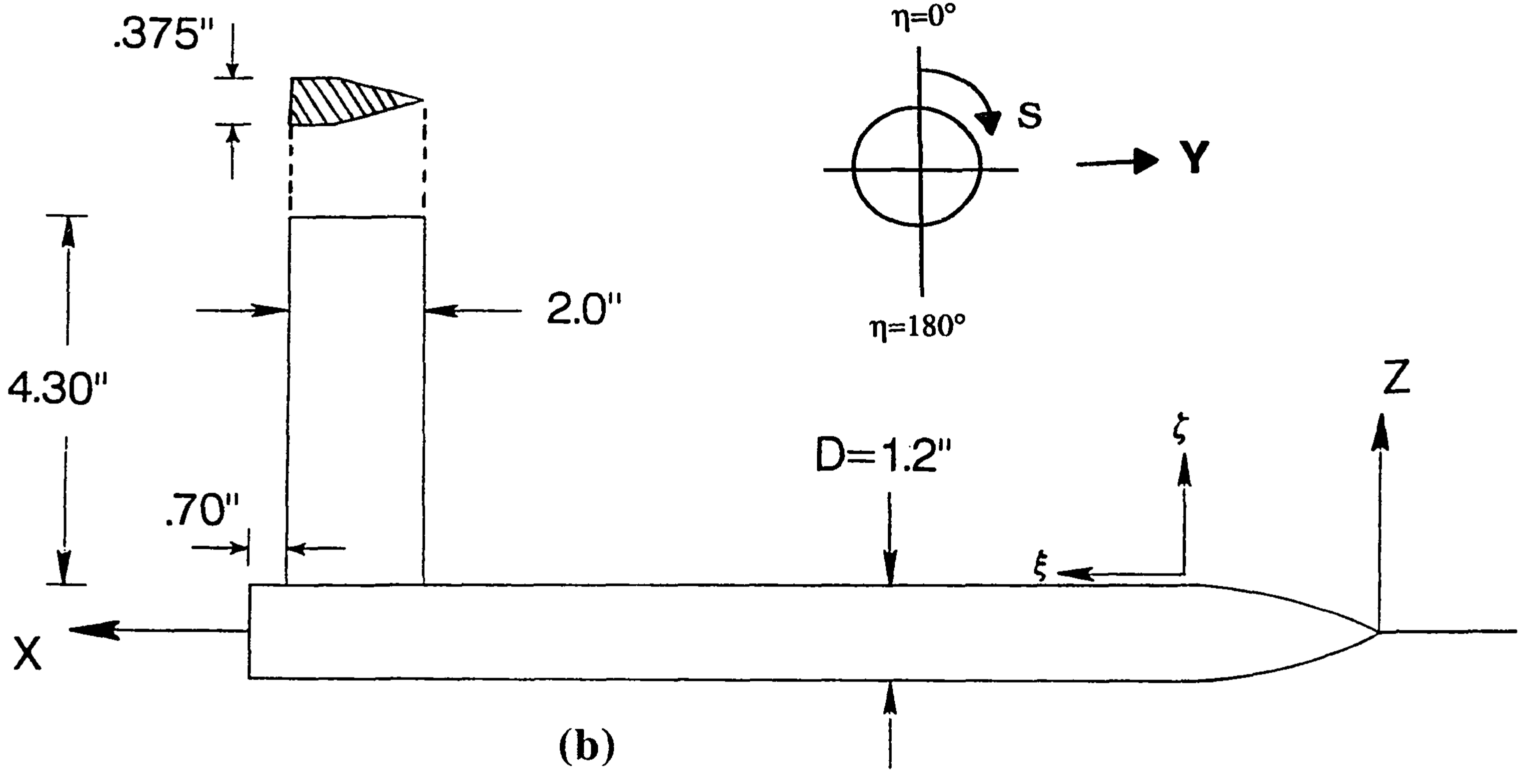


Fig. 4.17 Schematic of (a) rectangular cavity with L/D = 6.73, (b) ogive-nose-cylinder.



Z/D = (9.58**2 - (X/D-3.06)**2)-2.56   0 < X/D < 3.06

Z/D = .500   3.06 < X/D < 22.69

Fig. 4.17 (Concluded).

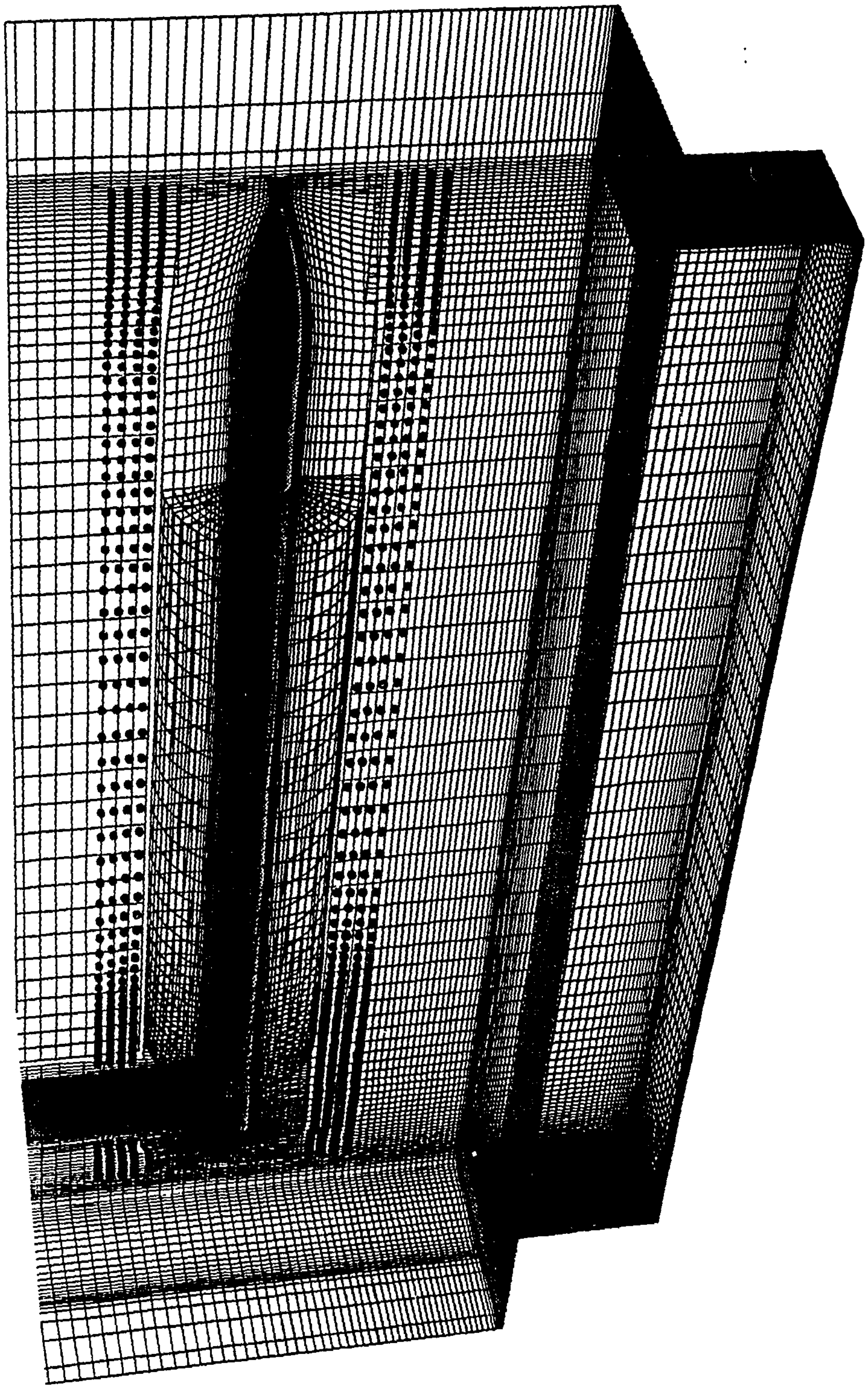
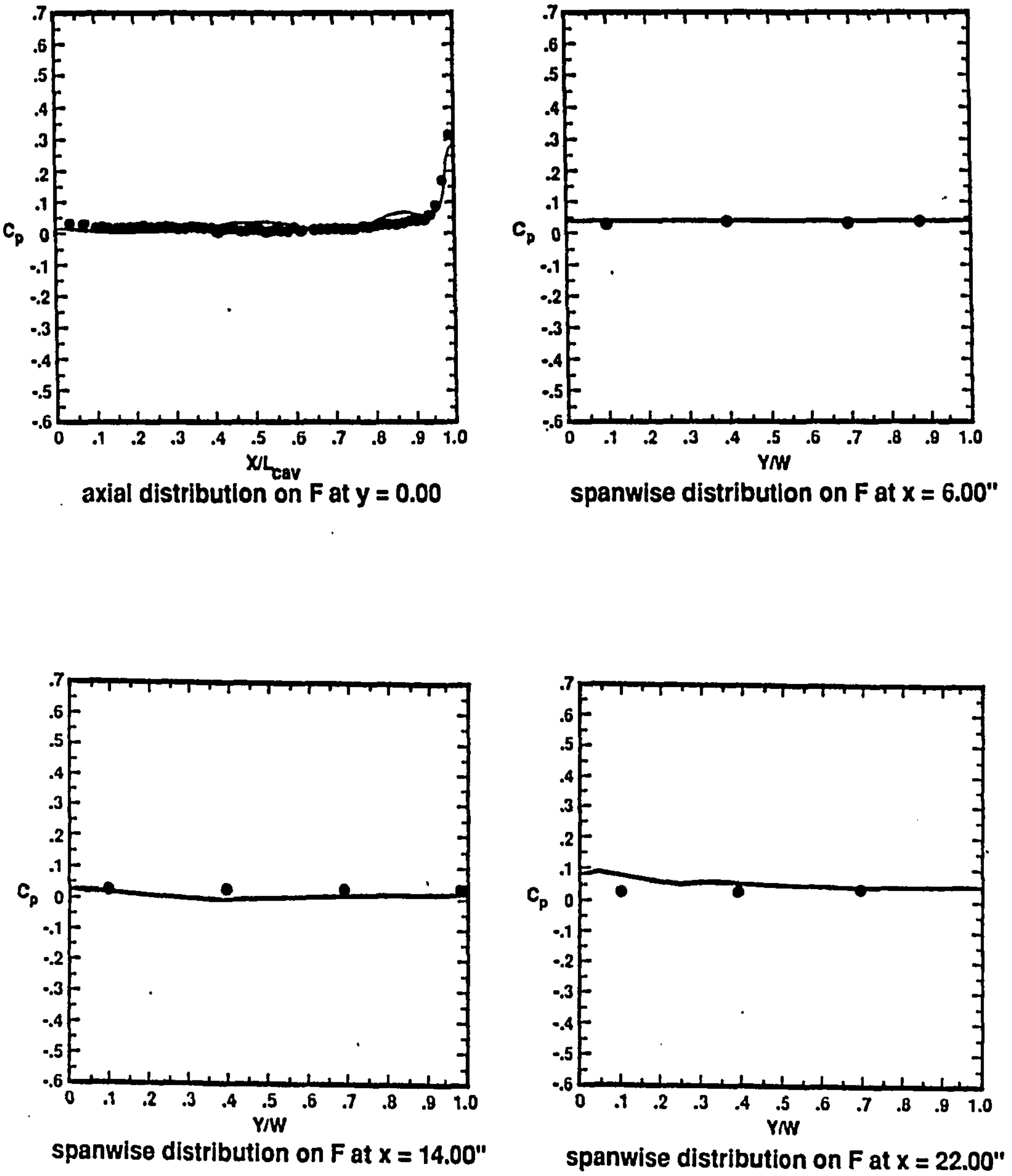


Fig. 4.18 Composite grid of ogive-nose-cylinder near a cavity (Case 4).



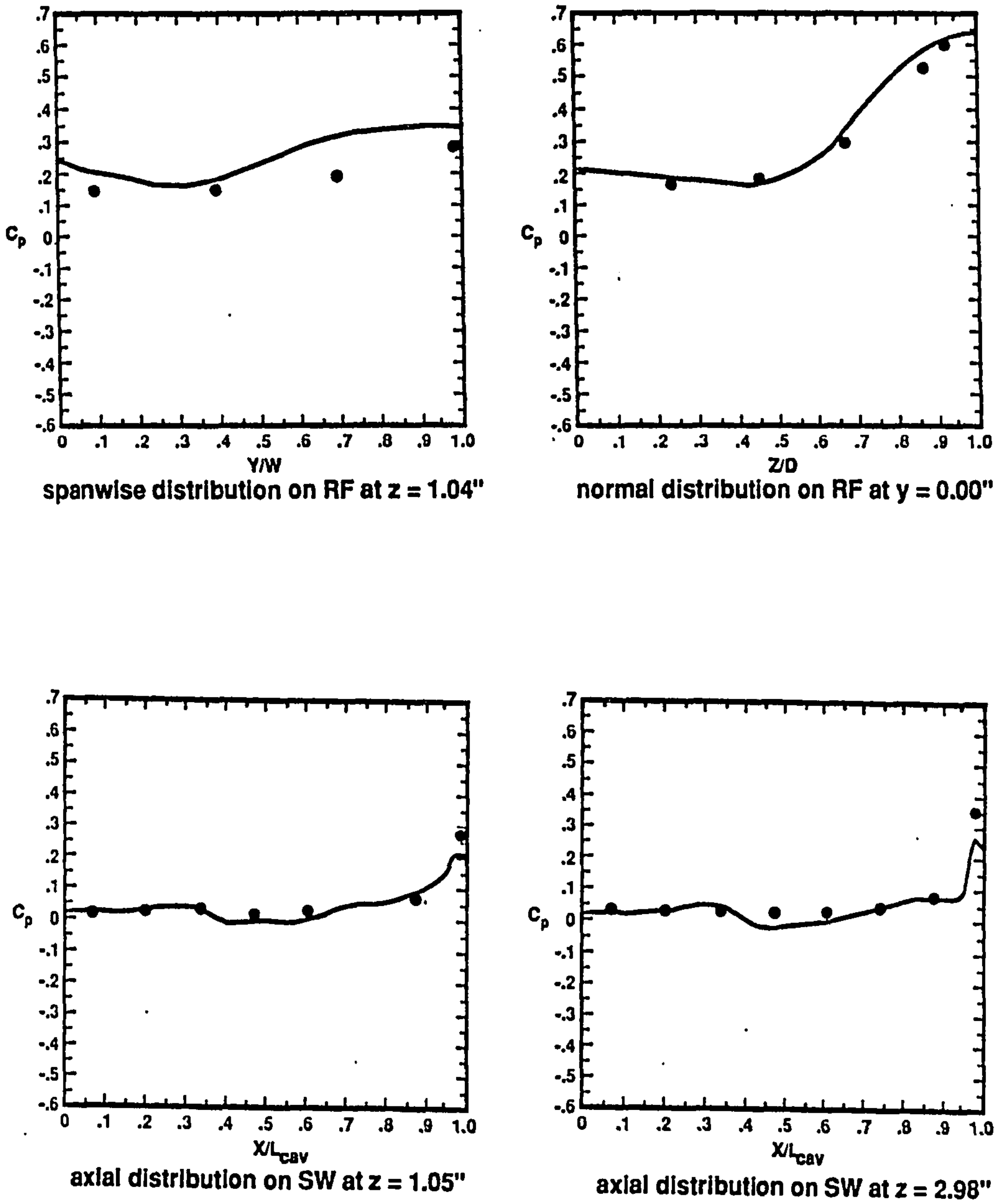
axial distribution on F at y = 0.00

spanwise distribution on F at x = 6.00"

spanwise distribution on F at x = 14.00"

spanwise distribution on F at x = 22.00"

Fig. 4.19 Time averaged surface pressure coefficient distribution for Case 4.



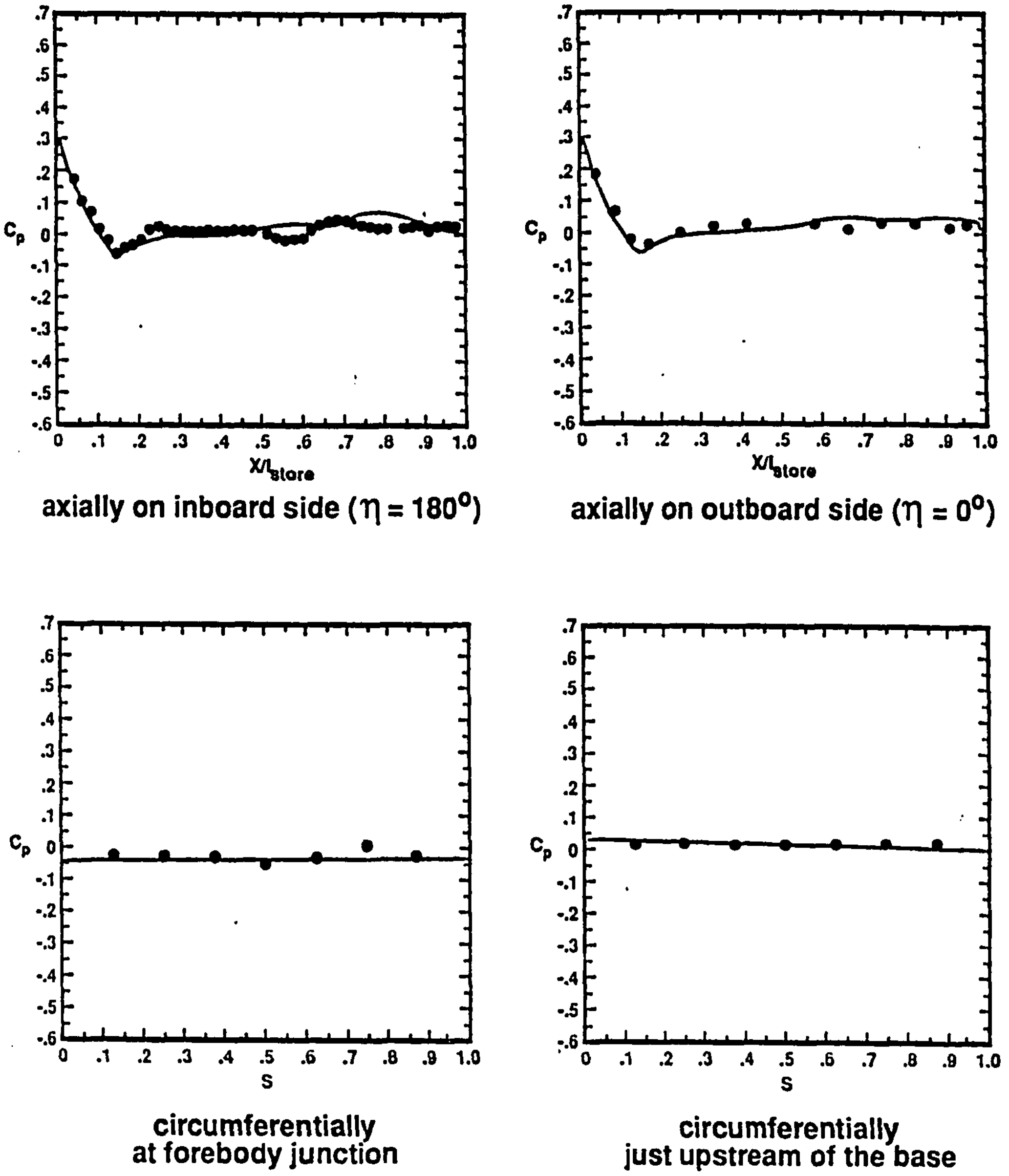
spanwise distribution on RF at z = 1.04"

normal distribution on RF at y = 0.00"

axial distribution on SW at z = 1.05"

axial distribution on SW at z = 2.98"

Fig. 4.19 (Continued).



axially on inboard side (η = 180°)

axially on outboard side (η = 0°)

circumferentially
at forebody junction

circumferentially
just upstream of the base

Fig. 4.19 (Concluded).

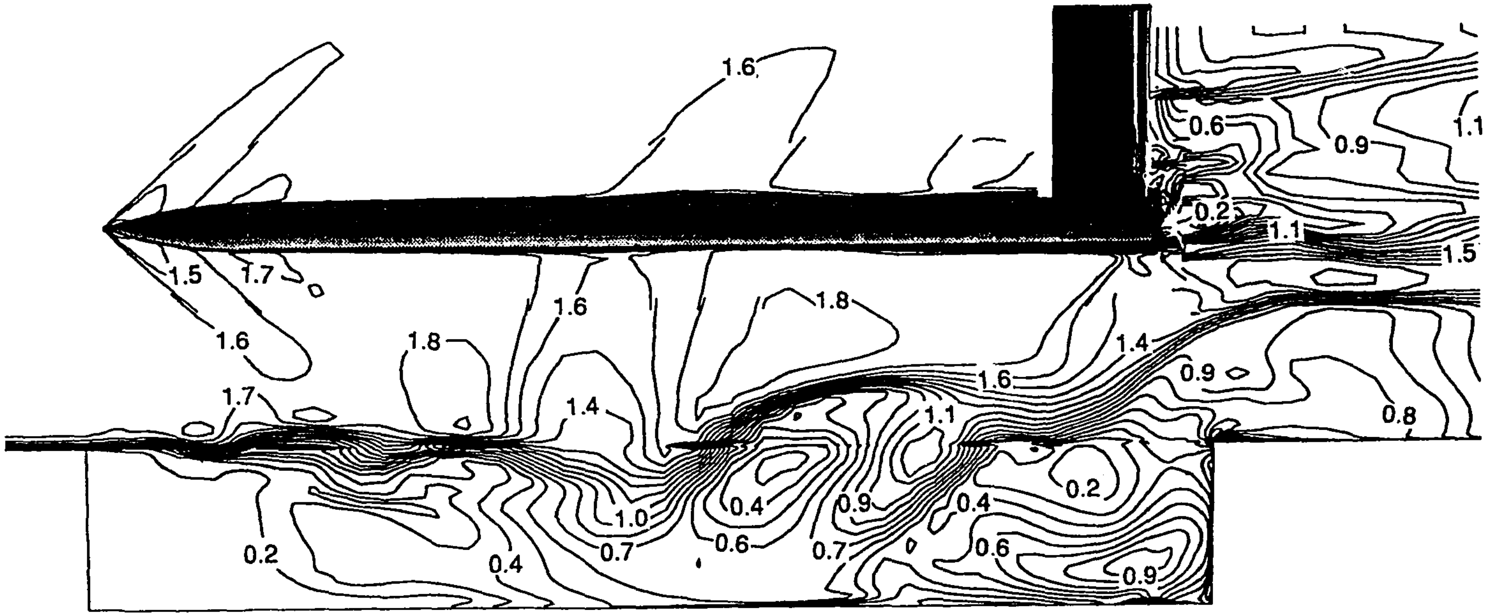


Fig. 4.20 Instantaneous Mach contours on the symmetry plane of Case 4.

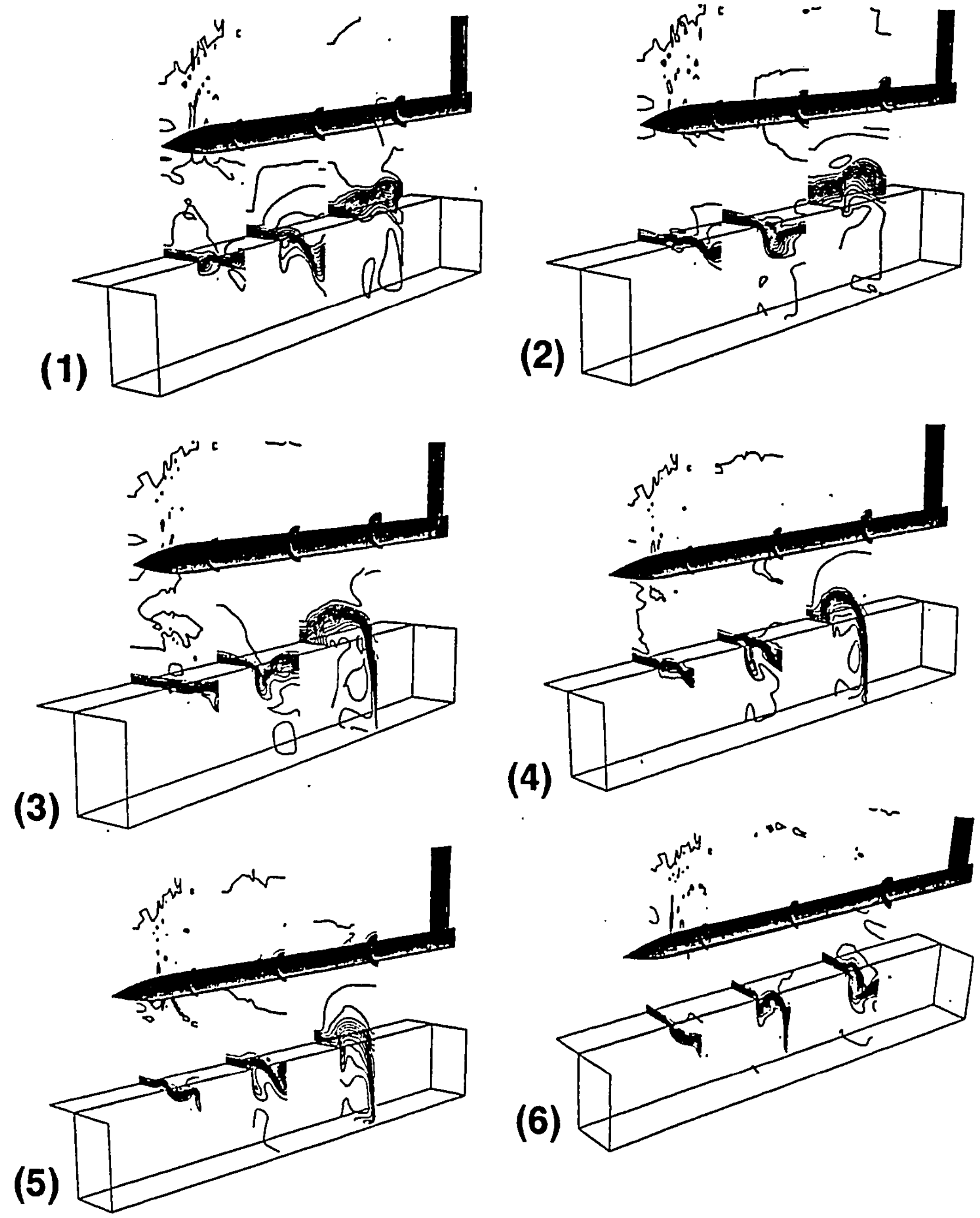


Fig. 4.21 Instantaneous crossflow total pressure contours Case 4 over a period of 1.5 ms.

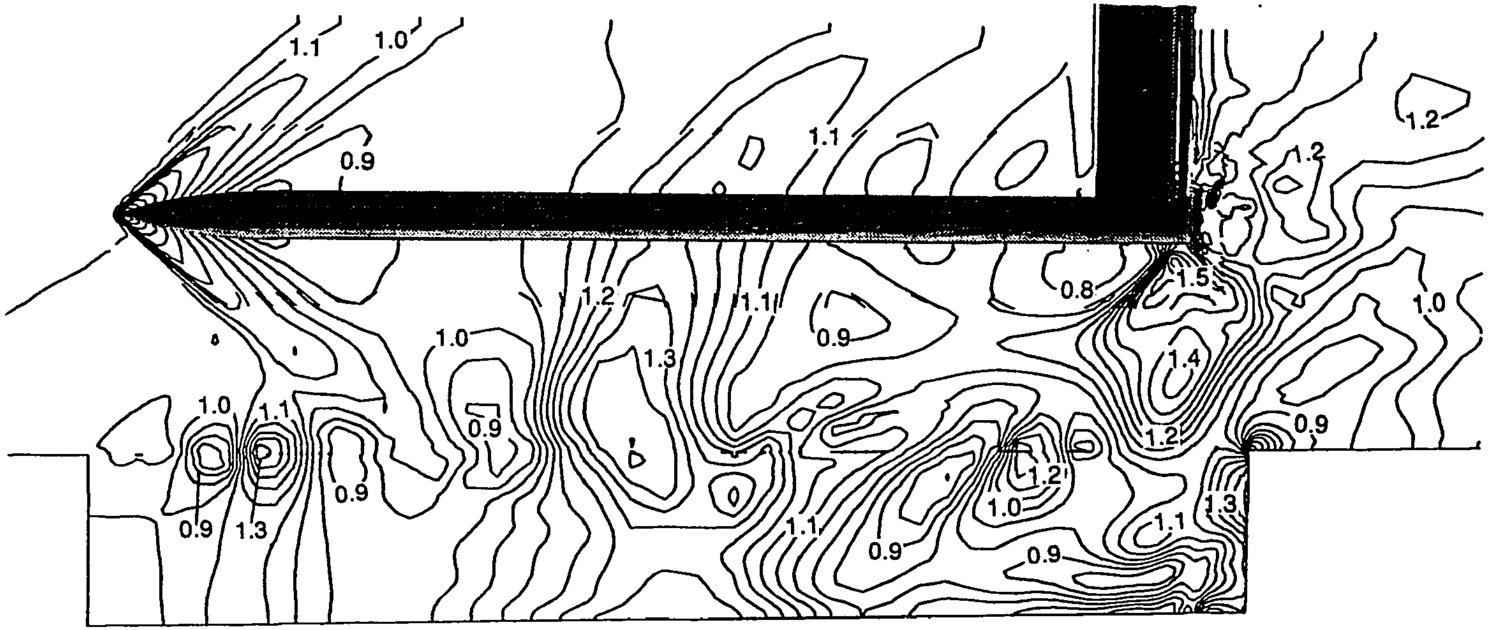


Fig. 4.22 Instantaneous normalized pressure contours on the symmetry plane of Case 4.

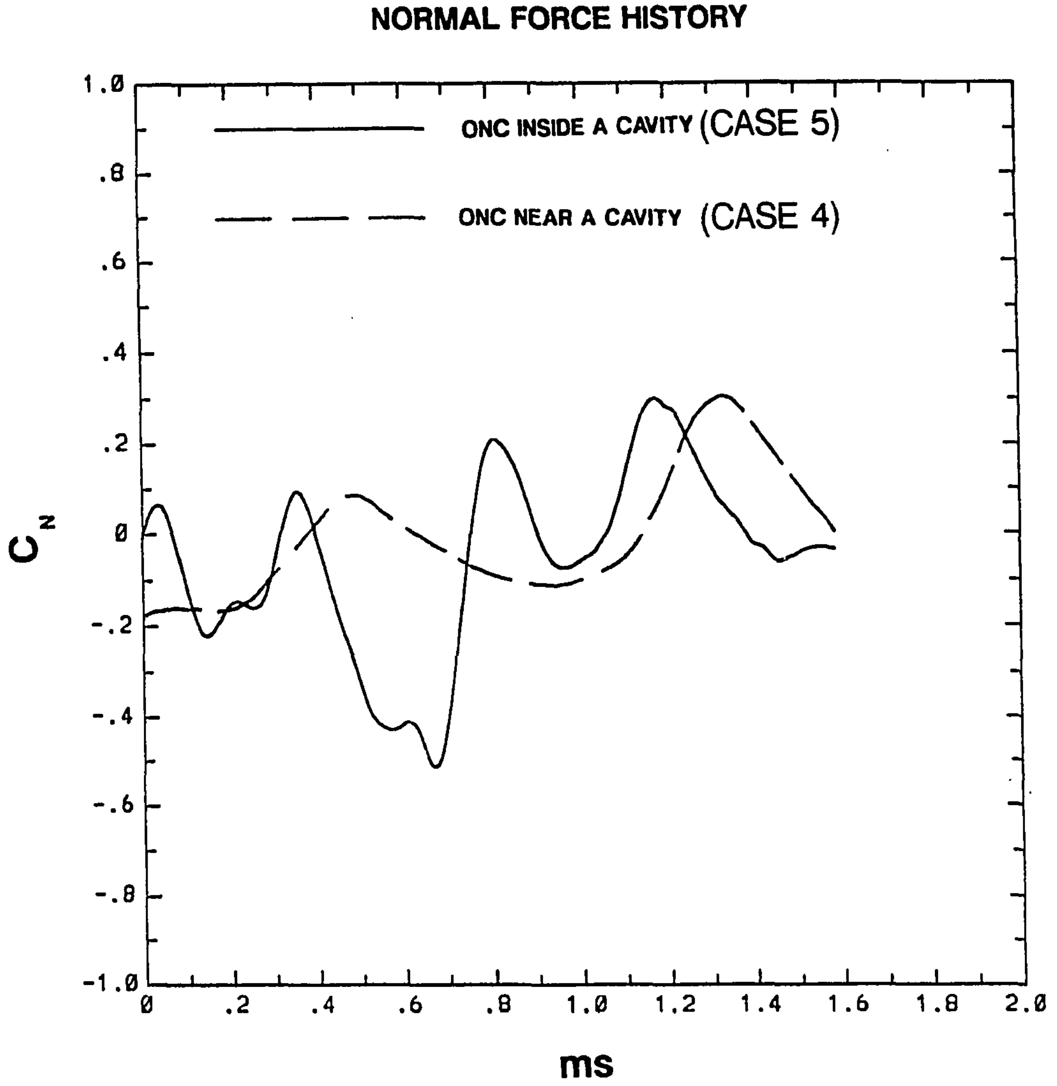


Fig. 4.23 Variation of C_N with time over a period of 1.59 ms for Case 4 and Case 5.

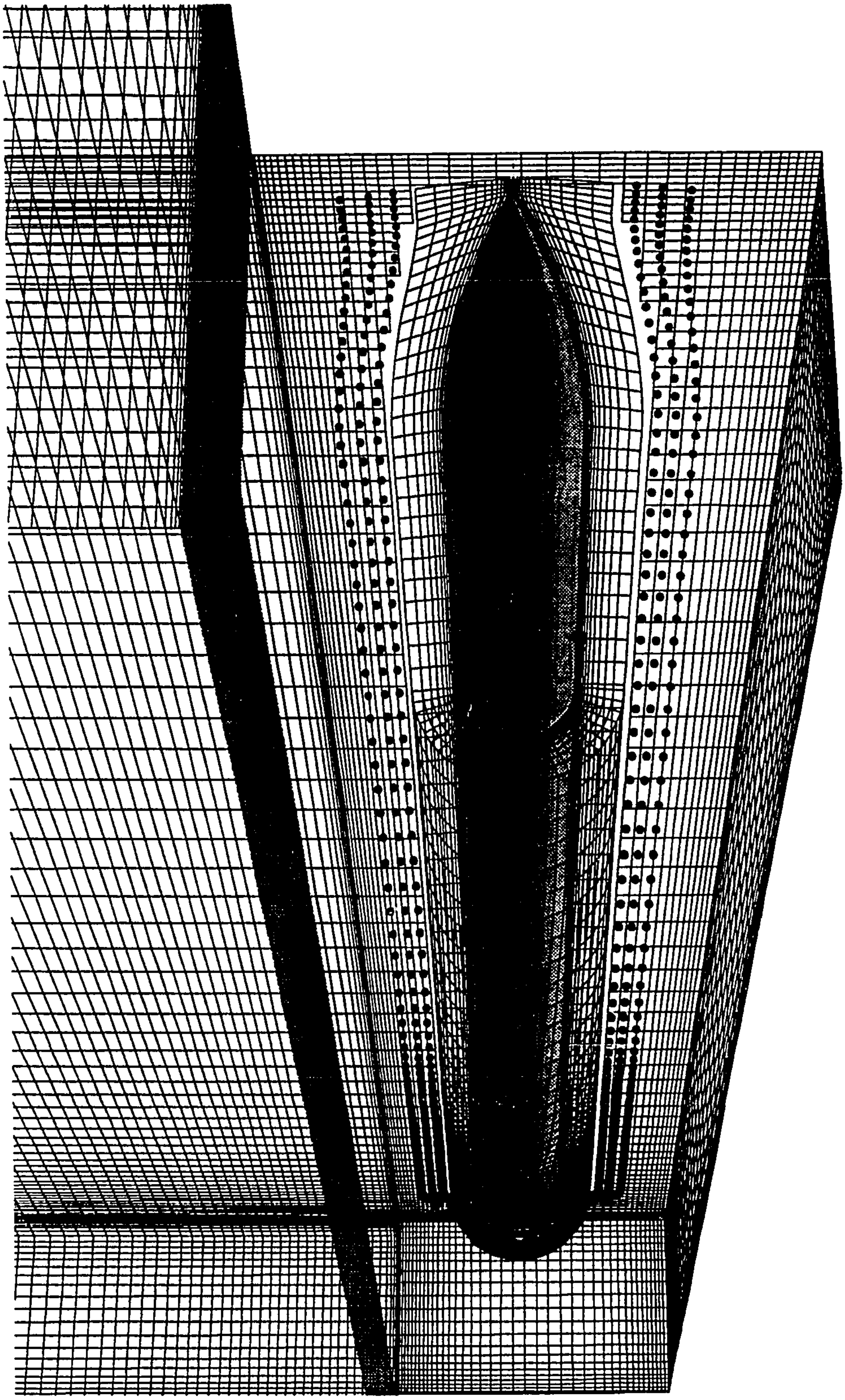


Fig. 4.24 Composite grid of ONC inside a cavity, Case 5.

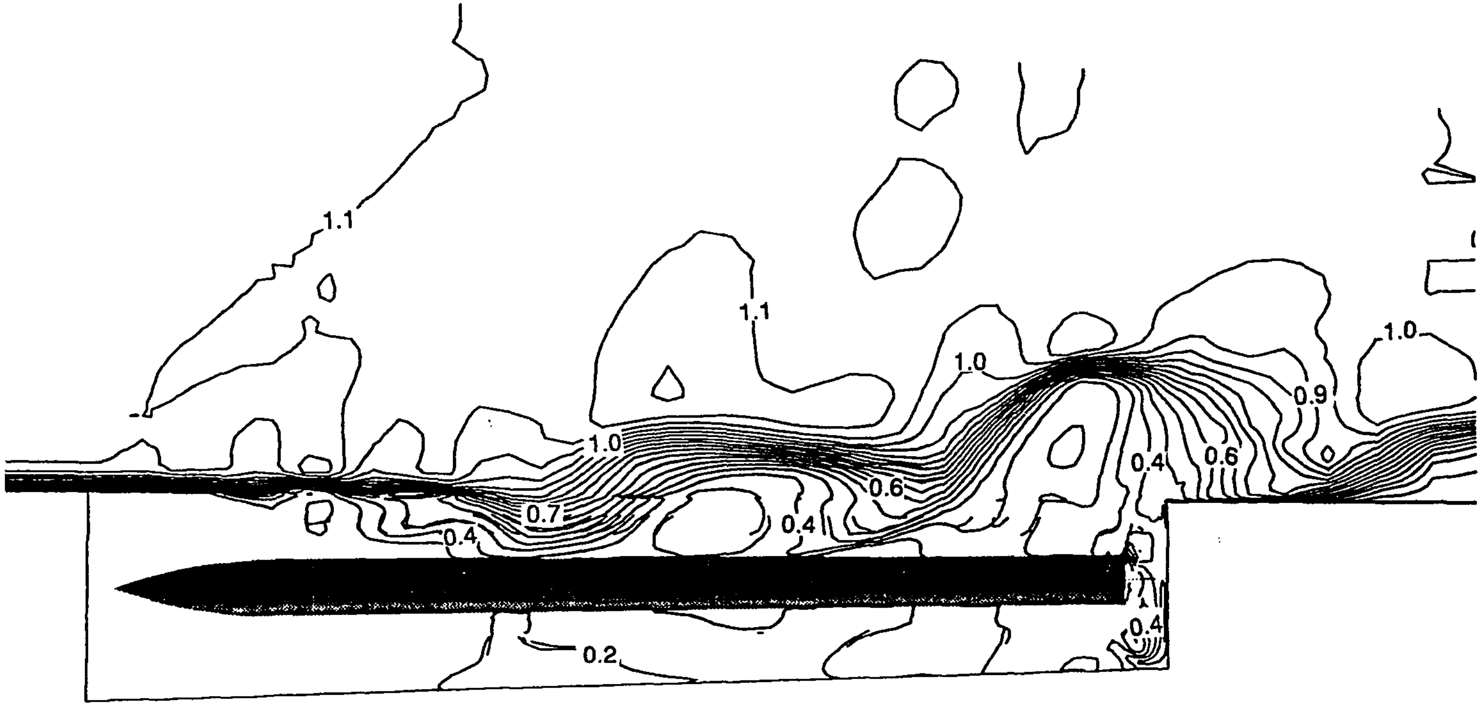


Fig. 4.25 Instantaneous normalized total pressure contours on the symmetry plane of Case 5.

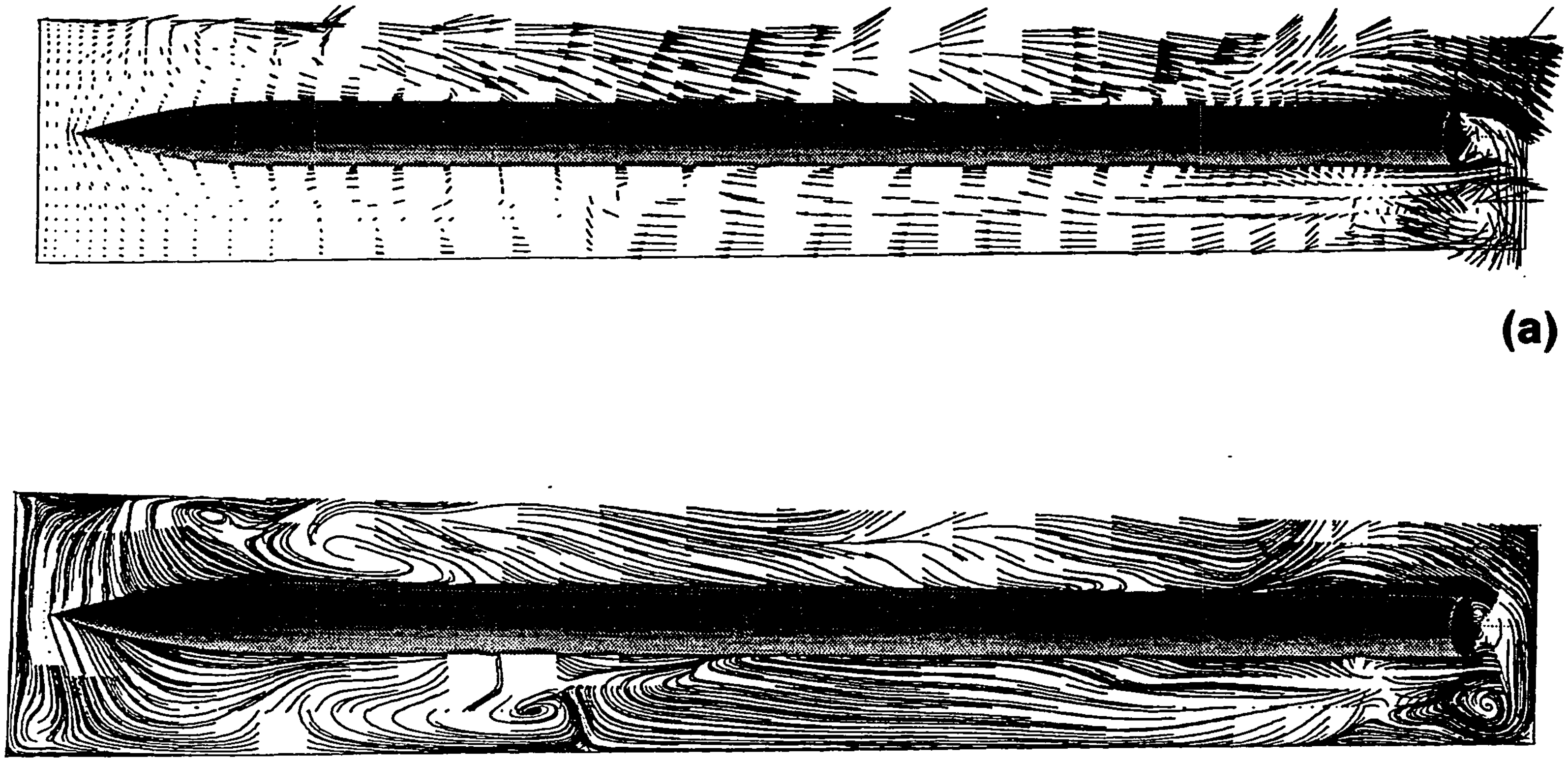


Fig. 4.26 (a) Instantaneous velocity vector (b) Instantaneous streamlines on the symmetry plane of cavity for Case 5.

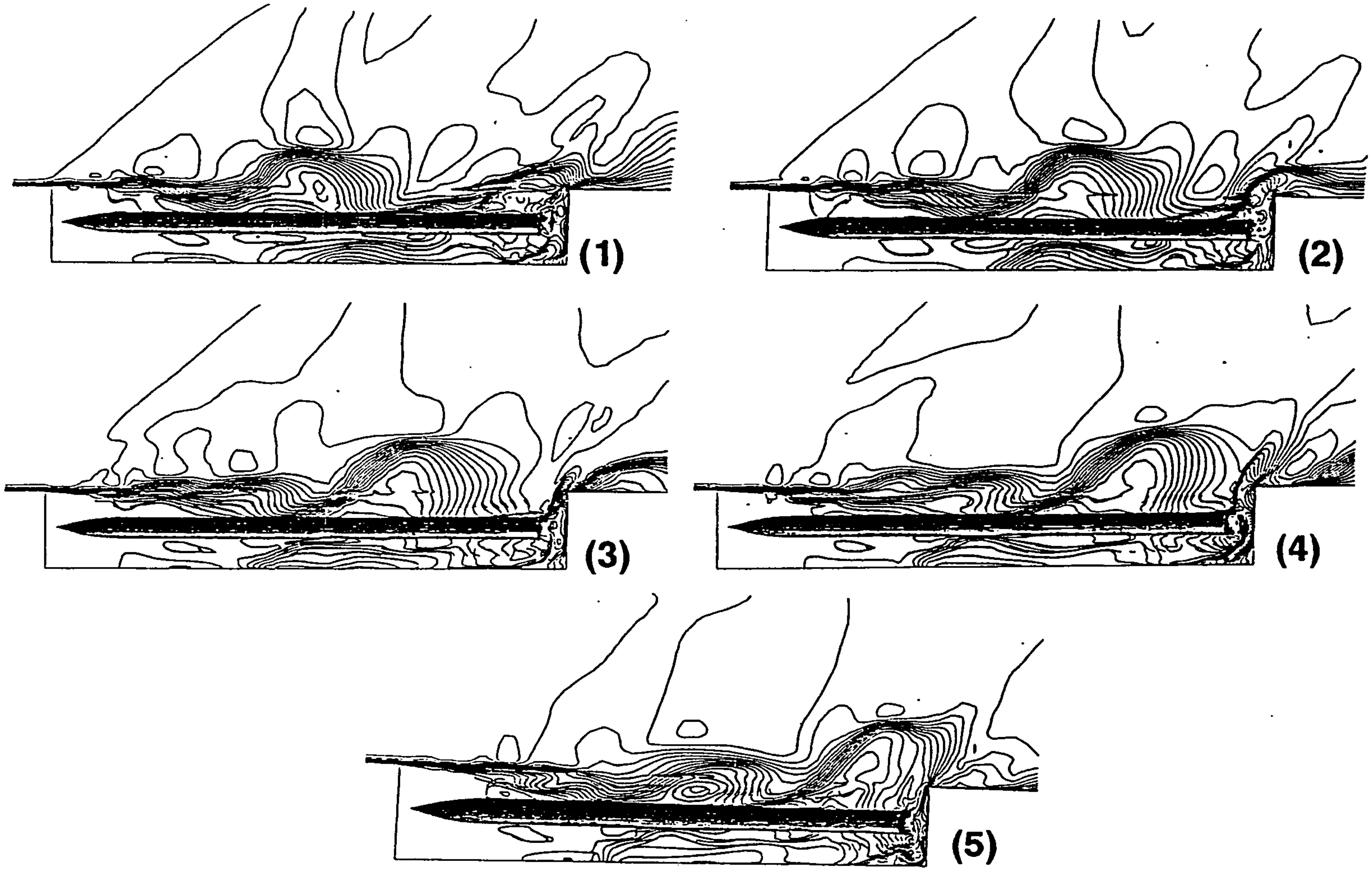
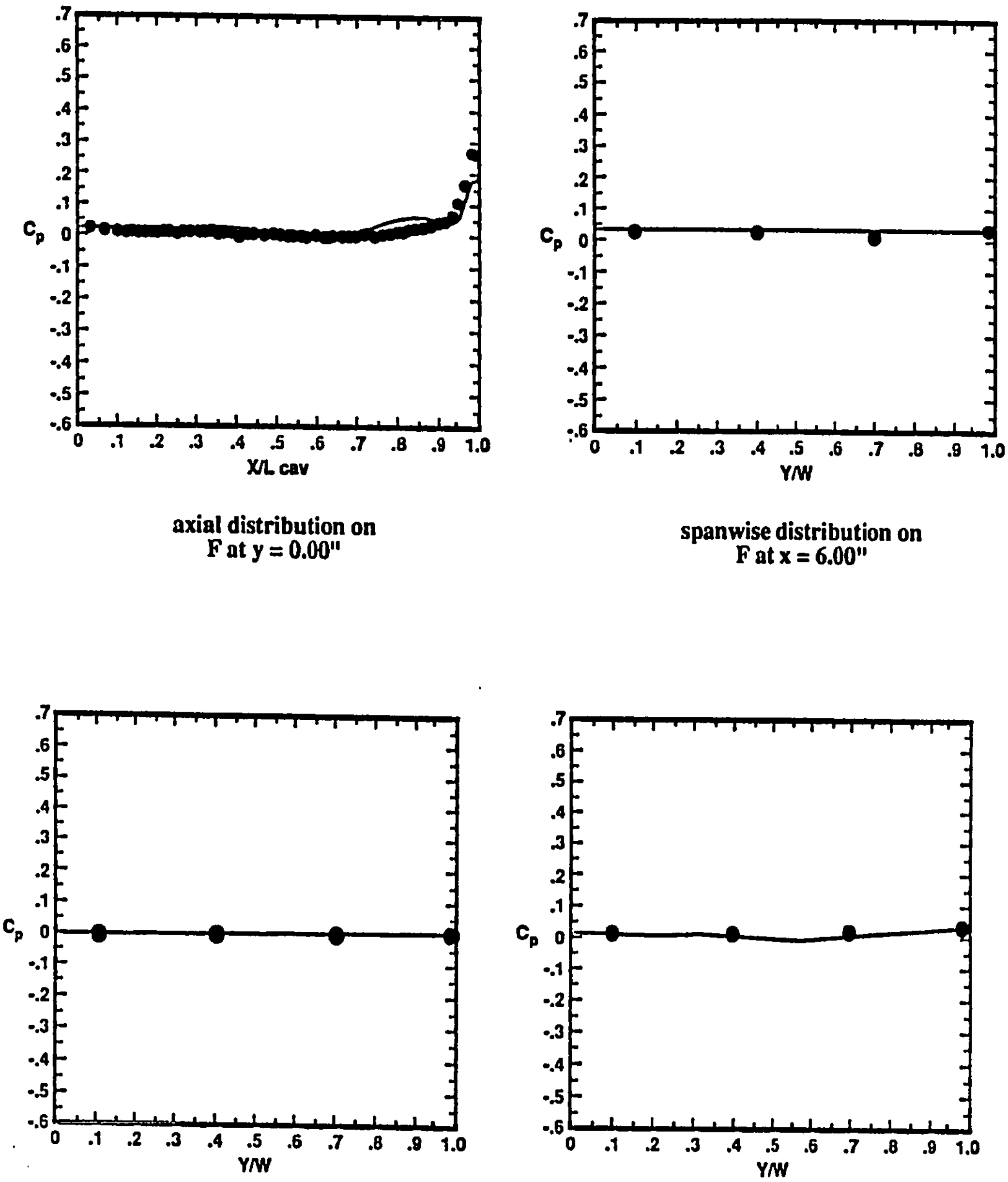


Fig. 4.27 Instantaneous Mach contours on symmetry plane of Case 5.
Frame interval = 1/4 ms.



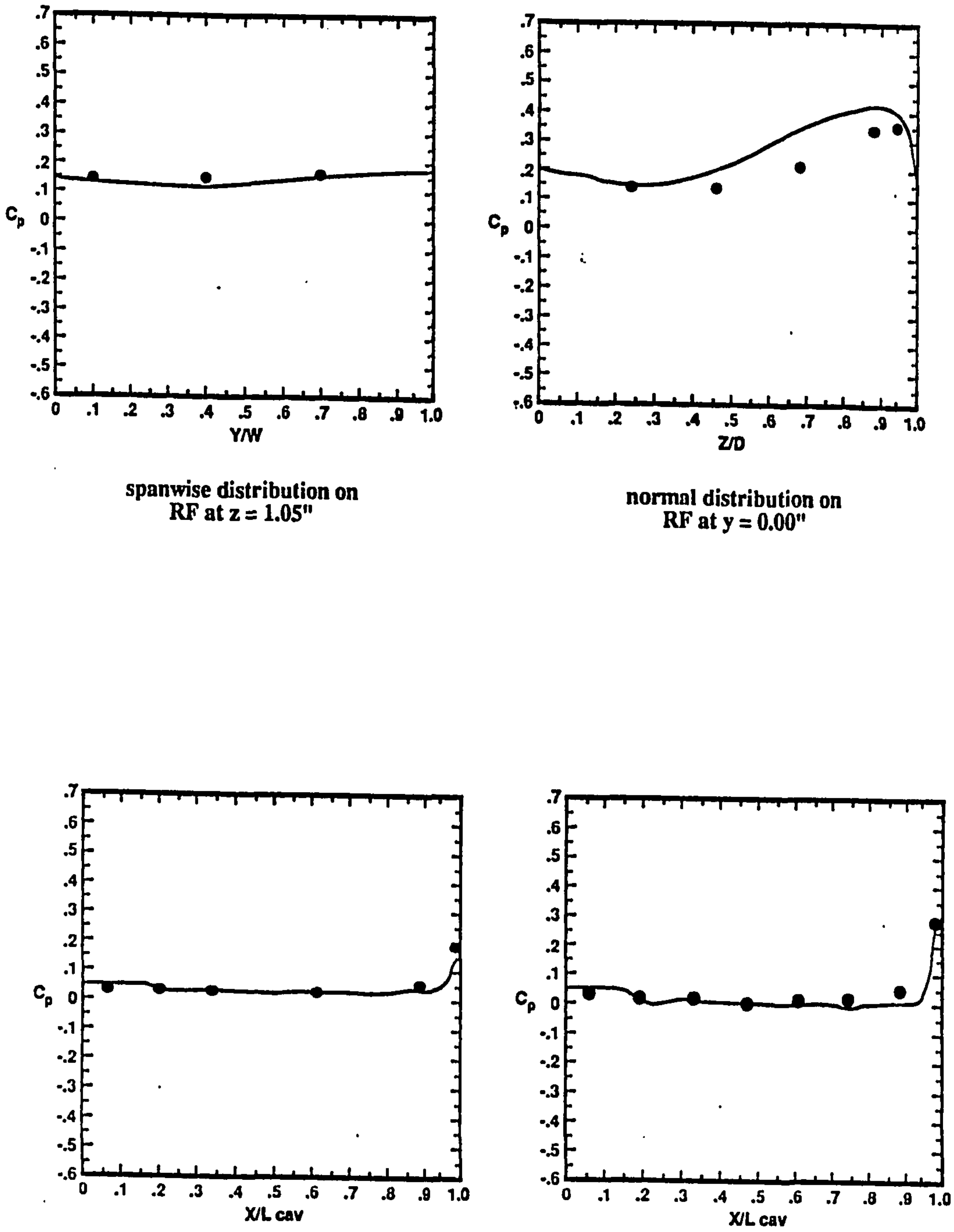
axial distribution on
F at y = 0.00"

spanwise distribution on
F at x = 6.00"

spanwise distribution on
F at x = 14.00"

spanwise distribution on
F at x = 22.00"

Fig. 4.28 Time averaged surface pressure coefficient distribution for Case 5.



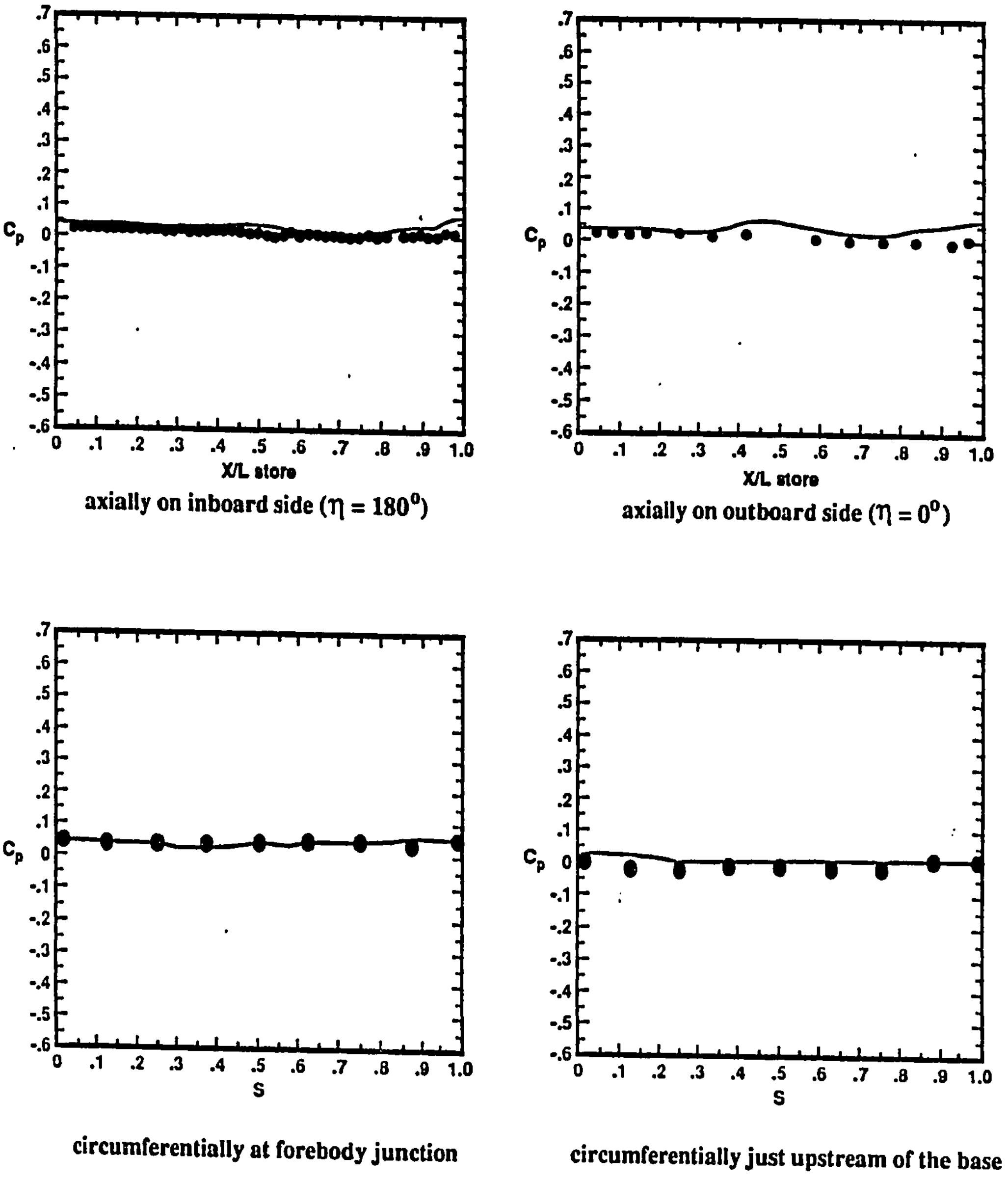
spanwise distribution on
RF at z = 1.05"

normal distribution on
RF at y = 0.00"

axial distribution on
SW at z = 1.05"

axial distribution on
SW at z = 2.98"

Fig. 4.28 (Continued).



axially on inboard side (η = 180°)

axially on outboard side (η = 0°)

circumferentially at forebody junction

circumferentially just upstream of the base

Fig. 4.28 (Concluded).

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

An algorithm is developed to obtain solutions for flows about arbitrary complex configurations with multiple and nonsimilar components. Domain decomposition techniques such as overlapped grids are used to divide the flow domain into simpler subdomains. The time-dependent, Reynolds-averaged, compressible, complete Navier-Stokes equations are solved on each subdomain.

The equations are solved using an implicit, upwind-biased, finite volume method. The implicit scheme eases the restriction on the time-step for highly stretched grids. The upwind method of flux-difference splitting,which is used for the inviscid part of the Navier-Stokes equations, is useful in obtaining crisper resolution of high flow gradients, such as shocks. In addition, the upwind methods generally have the advantage of being naturally dissipative. Hence, the need for separate spatial dissipation terms, to overcome oscillations or instabilities arising in regions of high gradients, is eliminated. The finite volume discretization allows more flexibility with regard to the geometry and it is tolerant of grid singularities. It remains valid in the presence of discontinuities in the flowfield since it begins with the integral equations and satisfies the conservation laws.

The effect of Reynolds stresses is incorporated through the Baldwin-Lomax algebraic turbulence model. Several modifications are done to the model for proper determination of velocity and length scales. This simple algebraic model is used as a compromise because of the large computational time and memory required by the unsteady flows being considered.

Domain decomposition increases the flexibility of grid generation for complex configurations. The grid overlapping technique allows different grid topologies for nonsimilar components of the configuration. In addition, it has the advantage of allowing different solution algorithms for different subdomains. The solution algorithm is modified to remain fully vectorized despite the holes in the overlapped grids. The multigrid convergence acceleration scheme, which is a valuable tool in saving computational time, is modified and extended to work on overlapped grids.

The suitability of the baseline solution algorithm is examined by obtaining the solutions of flows over two-dimensional cylindrical profiles and a body of revolution. This study is also performed to gain a better understanding of this class of flows. Other concurrent investigations have shown the suitability of the current algorithm for complicated two- and three-dimensional cavity flows.

The grid overlapping scheme is then tested by obtaining the solution of a supersonic flow over a blunt-nose-cylinder at high angles of attack. The calculation is performed on a body fitted C-O grid embedded in a Cartesian grid. This solution is compared with the solution of the same flowfield obtained on a body fitted C-O grid with no grid overlapping and the available experimental data. The solution of flowfield obtained with grid overlapping scheme compares very well with the one using no overlapped grids. There is excellent agreement between the computational results and the experimental data. The multigrid algorithm is also used for this case and shows substantial savings in computational time.

The algorithm is then used for the main application, which is to simulate the supersonic flow over an ogive-nose-cylinder near a cavity. The cylinder is attached to an offset L-shaped sting to simulate the complete experimental configuration designed for the validation of the computer code developed with this algorithm. Generally, excellent agreement between the computational results and the experimental data is observed. The unsteady nature of this flowfield is captured computationally.

A similar configuration without the offset sting is considered for further validation of the current algorithm. The component grids of the previous configuration are reused to form the composite grid for this configuration. In this case the cylinder is placed inside the cavity. This case is also crucial to study differences in the interference patterns when the cylinder is not near the cavity but it is placed inside the cavity. The unsteady nature of this flowfield and the interaction of the shear layer with the cylinder are simulated. Computationally obtained pressure coefficient distributions on most surfaces agree very well with the experimental data. However, some of the discrepancies observed may be attributed due to the elimination of the offset sting from the computational configuration. In the case of an internally carried store separating from its bay at supersonic speeds, various types of interference are encountered. The results of Cases 4 and 5 demonstrate, probably, the most important two types of interference.

The present computational algorithm combines the advantages of an efficient, geometrically conservative, minimally and automatically dissipative algorithm with the advantages and flexibility of an embedded and overlapped grid scheme. The applicability, versatility, and the validity of the current algorithm is shown by its applications to complex configurations.

To the best of author's knowledge, the results presented in Cases 4 and 5 represent the first time ever simulations of flows over a cylinder-sting-cavity configuration. These attempts have become possible and successful because of the algorithm presented in this study. However, in order to produce such flow simulations for a sequence of cases representing the separating store from its bay, further reductions in the computational times are necessary. In any event, the results of the present computations should contribute to the much needed database for an internal store carriage and its subsequent separation.

For steady cases, the approximate diagonal inversions (Sec. 2.5.2) and the multigrid scheme (Sec. 3.4) are utilized to accelerate the convergence. However, these strategies are not used for the unsteady flows to maintain a higher order of accuracy. However, other strategies, such as the initialization process considered in this study, should be devised to reduce the computer run time for the time-accurate calculations.

The effects of non-conservation at the overlapped grid interfaces also need to be further investigated. In the present study, efforts are made to reduce this effects. The overlapped grids are constructed as such that the overlapped regions are formed with cells of comparable sizes. Also, the regions of high gradients are either contained within a subdomain and away from grid interfaces, or these regions are decomposed using the conservative multiblock method. The weighted variation of the cell volumes should also be incorporated in the interpolation process to further enhance this algorithm. However, this is a complicated procedure for three-dimensional grids.

Attempts are made in this study to combine the advantages of different DDT's. A hybridization of the overlapped grids and the multiblock techniques are used for this purpose. The overlapped grids are used for configurations where nonsimilar topologies exist. The multiblock method are utilized where conservation across grid interfaces are deemed most essential. However, the flexibility of the current algorithm can be improved by including the zonal method. For example, the need for contiguous grid lines normal to the interface can be eliminated by using zonal grids. A hybridization of all three DDT's can provide a mechanism that further simplifies the selection process of the subdomains.

Future investigations should include rigid body dynamics in the algorithm to study the effects of the induced flow generated by the moving bodies. Furthermore, due to the generality of the present computational method, it can be used for other applications which involve multiple and nonsimilar bodies, such as propulsion integration simulation and studies of ground effects. Other steps should also be taken to capitalize on the advantage of domain decomposition as it relates to computer processing. One of the most natural ways of mapping a CFD problem to a computer with multiple processors is decomposing the domain and assigning each processor the work of a different subdomain.

The only extra programming effort for such concurrent processing is the exchange of intergrid boundary data.

REFERENCES

1. Thompson, J. F., " A Composite Grid Generation Code for General Three-Dimensional Regions," AIAA-87-0275, January 1987.

2. Belk, D. M. and Whitfield, D. L., "Time-Accurate Euler Equations Solution on Dynamic Blocked Grids," AIAA-87-1127-CP, June 1987.

3. Janus, J. M., Advanced 3-D CFD Algorithm for Turbomachinery, Ph.D. Dissertation, Mississippi State University, Mississippi, May 1989.

4. Chow, R. and Marconi, F., "A Navier-Stokes Solutions to Hypersonic Store Separation Flow Fields," AIAA-89-0031, January 1989.

5. Arabshahi, A. and Whitfield, D. L., "A Multiblock Approach to Solving the Three-Dimensional Unsteady Euler Equations About a Wing-Pylon-Store Configuration," AIAA-89-3401-CP, August 1989.

6. Hessenius, K. A. and Pulliam, T. H., "A Zonal Approach to Solution of the Euler Equations," AIAA-82-0969, June 1982.

7. Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA-84-0164, January 1984.

8. Hessenius, K. A. and Rai, M. M., "Three-Dimensional, Conservative, Euler Computations Using Patched Grid Systems and Explicit Methods," AIAA-86-1081, May 1986.

9. Ramshaw, J. D., "Conservative Rezoning Algorithm for Generalized Two-Dimensional Meshes," Journal of Computational Physics, Vol. 59, June 1985, pp. 193-199.

10. Kathong, M., Smith, R. E., and Tiwari, S. N., "A Conservative Approach for Flow Field Calculations on Multiple Grids," AIAA-88–0224, January 1988.

11. Thomas, J. L., Walters, R. W., Reu, T., Ghaffari, F., Weston, R. P., and Luckring, J. M., " A Patched-Grid Algorithm for Complex Configurations Directed Toward the F/A-18 Aircraft," AIAA-89-0121, January 1989.

12. Nakahashi, K. and Obayashi, S., "Viscous Flow Computations Using a Composite Grid," AIAA–87-1128, July 1987.

13. Thompson, J. F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," Applied Numerical Mathematics, Vol. 1, pp. 3–27, January 1985.

14. Atta, E., "Component-Adaptive Grid Interfacing," AIAA-81-0382, January 1981.

15. Atta, E. and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows About Aircraft Configurations," AIAA-82-1017, June 1982.

16. Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," ASME Symposium on Advances in Grid Generation, FED-Vol. 5, Houston, Texas, June 1983.

17. Benek, J. A., Steger, J. L., and Dougherty, F. C., "A Flexible Grid Embedding Technique with Application," AIAA-83-1944, July 1983.

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

19. Benek, J. A., Steger, J. L., Dougherty, F. C., and Buning, P. G., "Chimera: A Grid-Embedding Technique," AEDC-TR-85-64, Arnold Engineering Development Center Report, April 1986.

20. Suhs, N. E., "Computations of Three-Dimensional Cavity Flow at Subsonic and Supersonic Mach Numbers," AIAA-89-0637, January 1987.

21. Dougherty, F. C., Benek, J. A., and Steger, J. L., "On Applications of Chimera Grid Scheme to Store Separation," NASA–TM–88193, October 1985.

22. Dougherty, F. C. and Kuan, J-H, "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme," AIAA–89-0637, January 1989.

23. Meakin, R. L. and Suhs, N. E., " Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion," AIAA-89-1996-CP, June 1989.

24. Kafyeke, F., "Prediction of Wing-Body-Store Aerodynamics Using a Small Perturbation Method and a Grid Embedding Technique," AGARD Symposium on Applications of Computational Fluid Dynamics in Aeronautics, Aix-En-Province, France, April 1986.

25. Löhner, R., " Adaptive Remeshing for Transient Problems with Moving Bodies," AIAA-88-3736-CP, July 1988.

26. Fox, J. H. and Allee, E. G., "Experimental/Computational Study of a Transonic Aircraft with Stores," AIAA-89-1932, June 1989.

27. Stallings, R. L., Jr., Wilcox, F. J., and Forrest, D. K., "Measurements of Pressure, Forces and Moments on a Generic Store Separating From a Box Cavity at Supersonic Speeds," Proposed NASA TP, September 1990.

28. Stallings, R. L. and Wilcox, F. J., "Experimental Cavity Pressure Distributions at Supersonic Speeds," NASA-TP-2683, June 1987.

29. Baysal, O. and Stallings, R. L., Jr., "Computational and Experimental Investigations of Cavity Flowfields," AIAA-87–0114, January 1987.

30. Baysal, O., Srinivasan, S., and Stallings, R. L., Jr., "Unsteady Viscous Calculations of Supersonic Flows Past Deep and Shallow 3–D Cavities," AIAA-88–0101, January 1988.

31. Srinivasan, S., Baysal, O., and Plentovich, E. B., "Navier-Stokes Calculations of Transonic Flows Past Cavities," NASA CR-4210, January 1989. Also in Advances and Applications in Computational Fluid Dynamics (ED. : O. Baysal), ASME-FED, Vol. 66, November 1988, pp. 121–133.

32. Baysal, O. and Yen, G. W., "Implicit and Explicit Computational of Flows Past Cavities with and without Yaw," AIAA-90–0049, January, 1990. Also to appear in Journal of Aircraft.

33. Srinivasan, S., Numerical Simulation of Turbulent Flows Past Three-Dimensional Cavities, Ph.D. Dissertation, Mechanical Engineering and Mechanics Department, Old Dominion University, August 1988.

34. Vinokur, M., "Conservation Equations of Gas-Dynamics in Curvilinear Coordinate Systems," Journal of Computational Physics, Vol. 14, 1974, pp. 105–125.

35. Degani, D. and Schiff, L. B., "Computations of Turbulent Supersonic Flows Around Pointed Bodies Having Crossflow Separation, " Journal of Computational Physics, Vol. 66, September 1986, pp. 173-196.

36. Vinokur, M., "An Analysis of Finite-Difference and Finite Volume Formulation of Conservation Laws," NASA–CR–177416, Ames Research Center, June 1986.

37. Beam, R. and Warming, R. F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form," Journal of Computational Physics, Vol. 22, September 1976, pp. 87-110.

38. Anderson, W. K., Thomas, J. L., and Whitfield, D. L., "Three-Dimensional Multigrid Algorithm for Flux-Split Euler Equations," AIAA-86–0274, January 1986.

39. Mulder, W. A. and Van Leer, B., "Implicit Upwind Methods for the Euler Equations," AIAA-83-1930, January 1983.

40. Hanel, D., Schwane, R., and Seider, G., "On the Accuracy of Upwind Schemes for the Solution of the Navier-Stokes Equations," AIAA-87-1105, January 1987.

41. Moretti, G., "The λ Scheme," Computers and Fluids, Vol. 7, 1979, pp. 191-205.

42. Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "The Split Coefficient Matrix Method for Hyperbolic System of Gas Dynamics Equations," AIAA-80-0268, January 1980.

43. Steger, J. L. and Warming, R. F., "Flux Vector Splitting for the Inviscid Gas Dynamics Equations with Applications to Finite Difference Methods," Journal of Computational Physics, Vol. 40, April 1981, pp. 263-293.

44. Van Leer, B., "Flux Vector Splitting for the Euler Equations," ICASE Report 82-30, September 1982.

45. Van Leer, B., Thomas, J. L., Roe, P. L., and Newsom, R. W., " A Comparison of Numerical Flux Formulas for the Euler and Navier–Stokes Equations," AIAA-87-1104, June 1987.

46. Osher, S. and Solomon, F., "Upwind Difference Schemes for Multidimensional Euler and Navier-Stokes," Computers and Mathematics with Applications, Vol. i2A, 1986, pp. 413-432.

47. Roe, P. L., " Approximate Riemann Solver, Parameters Vectors and Difference Scheme," Journal of Computational Physics, Vol. 43, October 1981, pp. 357-372.

48. Pulliam, T. H. and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," Journal of Computational Physics, Vol. 39, 1981, pp. 347-363.

49. Chakravarthy, S. R. and Osher, S., "Computing with High Resolution Upwind Scheme for Hyperbolic Equations," Lecture given at AMS-SIAM Summer Seminar on Large Scale Computations in Fluid Dynamics, La Jolla, California, June-July 1983.

50. Anderson, W. K., Thomas, J. L., and Van Leer, B., "Comparison of finite volume flux splitting method for the Euler Equations," AIAA Journal, Vol. 24, No. 9, September 1986, pp. 1453–1460.

51. Van Albada, G. D., Van Leer, B., and Roberts, W. W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," Astronomy and Astrophysics, Vol. 108, April 1982, pp. 76–84.

52. Spekreijise, S. P., "Multigrid Solution of the Steady Euler Equations, Ph.D. Dissertation, Centrum Voor Wiskunde en Informatica, Amsterdam, The Netherlands, 1987.

53. Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Flows," AIAA-78-257, June 1978.

54. Cebeci, T., "Calculation of Compressible Turbulent Boundary Layer with Heat and Mass Transfer," AIAA-70-741, June 1970.

55. Waskiewicz, J. D., Shang, J. S., and Hankey, W. L., "Numerical Simulation of Near Wakes Utilizing a Relaxation Turbulence Model," AIAA Journal, Vol. 18, No. 12, December 1980, pp. 1440-1445.

56. Baysal, O., Fouladi, K., and Lessard, V. R., "A Multigrid and Upwind Viscous Flow Solver on 3-D Overlapped and Embedded Grids,"AIAA-89-0464. Also to appear in AIAA Journal. Approx. April 1991.

57. Lessard, V. L., Domain Decomposition for Multigrid, Finite Volume Flow Solvers, Master Thesis, Mechanical Engineering and Mechanics Department, Old Dominion University, December 1989.

58. Berger, M. J., "On Conservation at Grid Interfaces," SIAM Journal of Numerical Analysis, Vol. 24, No. 5, October 1987, pp. 968-983.

59. Chesshire, G. and Henshaw, W. D., " Composite Overlapping Meshes for Solution of Partial Differential Equations," IBM Research Report, RC 14355 (#64300), Mathematics Research Report, January 1989.

60. Chima, R. V. and Johnson, G. M., " Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm," NASA-TM-83376, July 1982.

61. Jameson, A., " Solution of the Euler Equations for Two-Dimensional Transonic Flows by a Multigrid Method," Princeton University, MAE Report No. 1613, June 1983.

62. Stuben, K. and Trottenberg, U. "Multigrid Methods: Fundamental Algorithm, Model Problem Analysis and Applications" in Multigridding Methods", Proceedings of the Conference Held at Koln-Perz, November 1981 (Lectures Notes in Mathematics 960, Springer-Verlag).

63. Henshaw, W. D. and Chesshire, G., "Multigrid on Composite Meshes, "SIAM Journal of Scientific and Statistical Computing, Vol. 8, No. 6, November 1987. pp. 914-923.

64. Baysal, O., Fouladi, K., and Miller D. S., "Computations of Supersonic Flows Over a Body at High Angles of Attacks," AIAA Journal, Vol. 27, No. 4, April 1989, pp. 427-437.

65. Anderson, W. K., Thomas, J. L., and Rumsey, C. L., "Extension and Applications of Flux-Vector Splitting to Unsteady Calculations on Dynamic Meshes," AIAA 87–1152 CP, June 1987.

66. Lamont, P. J., "Pressure Around an Inclined Ogive Cylinder with Laminar, Transitional, or Turbulent Separation," AIAA Journal, Vol. 20, Nov. 1982, pp. 1482–1499.

67. Landrum, E. J. and Babb, C. D., "Wind-Tunnel Force, Pressure and Flow Visualization Data at Mach Number from 1.6 to 4.63 for a Series of Bodies of Revolution at Angles of Attack from –4° to 60°," NASA-TM-3558, October 1977 and NASA-TM-78813, March 1979.

68. Yates, L. A. and Chapman, G. T., "A Numerical Investigation of Crossflow Separation on a Three-Caliber Ogive-Cylinder," AIAA-87-1209, June 1987.

69. Smith, R. E. and Weigel, B. L., "Analytic and Approximate Boundary Fitted Coordinate System for Fluid Flow Simulation," AIAA 80–0192, 1980.

70. Mastin, C. W. and McConnaughey, H. V., "Composite Problems on Composite Grids," AIAA 84–1611, June 1984.

71. Smith, R. E. and Wiess, M. R., "An Interactive Algebraic Grid Generation Technique," NASA-TP-2533, March 1986.

72. Steger, J. L. and Sorenson, R. L., "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations," Journal of Computation Physics, Vol. 33, No. 3, December 1979, pp. 405–410.

73. Sorenson, R. L., "A Computer Program to Generate 2–D Grids About Airfoils and Other Shapes by the Use of Poisson's Equations," NASA-T-M-81198, NASA Ames Research Center, May 1988.

74. Chung, T. J., Finite Element Analysis in Fluid Dynamics, McGraw-Hill, 1978.

75. Ergatoudis, J. G., Irons, B. M. and Zienkiewics, O. C., "Curved Isoparametric Quadrilateral Elements for Finite Element Analysis," International Journal of Solids Structure, Vol. 4, January 1968, pp. 31–42.

APPENDICES

Appendix A
GRID OVERLAPPING METHOD

A.1 Grid Generation

The current grid overlapping method allows the subdomain grids to be generated independently. A subdomain grid topology depends upon neighboring topology only to the extent that they must overlap and that the cell sizes in the overlap region are comparable. The reasons are explained in Sec. 3.1. Two different grid generation methods, the algebraic method and the Poisson's equations method, are used in this study.

The algebraic method is one, in which there is a known explicit functional relationship between the computational and the physical domain [69]. Therefore, the algebraic method are best suited for simple configurations. This technique uses stretching functions to distribute points along simple analytic coordinate curves. It is effective in the area of mesh control at boundaries, but is less effective in the quality of the interior mesh points, particularly for complex domains [70]. An interactive computer program, developed by Smith et al. [71], TBGG, based on a two-dimensional algebraic two boundary grid generation technique is used in creating several subdomain grids. The essence of a two boundary method is to connect a distribution of points between inner and outer boundaries, based on a hermite cubic interpolation procedure.

For more complicated configurations, an elliptic partial differential equation (PDE) approach, developed by Steger and Sorenson [72], is used to generate grids. In particular, a computer program called GRAPE developed by Sorenson [73] is used. The GRAPE program generates two-dimensional grids about airfoils and other shapes by solving the Poisson's equation

$$\xi_{xx} + \xi_{yy} = P \tag{A.1}$$

$$\eta_{xx} + \eta_{yy} = Q \tag{A.2}$$

Control of the spacing between mesh points and control of the angles with which mesh lines intersect the boundaries are incorporated into the right hand side functions P and Q. An iterative procedure is used to solve these equations.

Both codes, GRAPE and TBGG, generate two-dimensional grids. Three dimensional grids are developed by simply stacking the two-dimensional planes in the third-dimension. Further enhancement of cell clustering within high viscous regions are accomplished by a parametric curve fitting procedure. Also, farfield rectangular subdomain grids are created using simple algebraic methods with exponential clustering in viscous regions.

A.2 Overlapping Algorithm

The grid overlapping Chimera algorithm developed by Steger et al. [16–19] is modified by Lessard [57] to serve a multigrid, finite volume (as well as finite difference) upwind solution algorithm. The modified version of the composite grid generator is given the name MaGGiE, short for Multi-Geometry Grid Embedder.

The program MaGGiE creates a three-dimensional composite mesh from individual subdomain grids. It also provide the necessary information for intergrid communications. The subdomain grids create holes in other subdomain grids in which they are embedded or overlapped. The holes that are created in the grids are excluded from the solution. To obtain a logical sequence of grid communications between overlapped grids, a form of grid hierarchy is needed. An order of hierarchial form between the grids allows the interaction of appropriate grids, simplifies the development of the data structure required for this interaction, and limits the search to locate points in other grids for the purpose of interpolation. Grids which are on level L of hierarchy are designated $G_{L,i}$ where "i" is the grid index on level L. In general, grids on a given level L are partially or completely embedded in grids of level L-1. Grids on level L may overlap other grids of level L, and they may contain grids of level L+1 partially or completely embedded in them. Examples of such hierarchial grid arrangements are illustrated in Fig. A.1.

MaGGiE's composite mesh generation consist of: (1) establishing the proper lines of communication among the grids through appropriate data structure; (2) constructing holes within grids; (3) identifying points with holes and illegal zones (solid surfaces); (4) locating points from which outer and hole boundary values can be interpolated; and (5) evaluating interpolation parameters. The MaGGiE code is divided into six stages. The first three stages are used to acquire fines level grid communication data, and the last three stages are used to acquire multigrid level communication data.

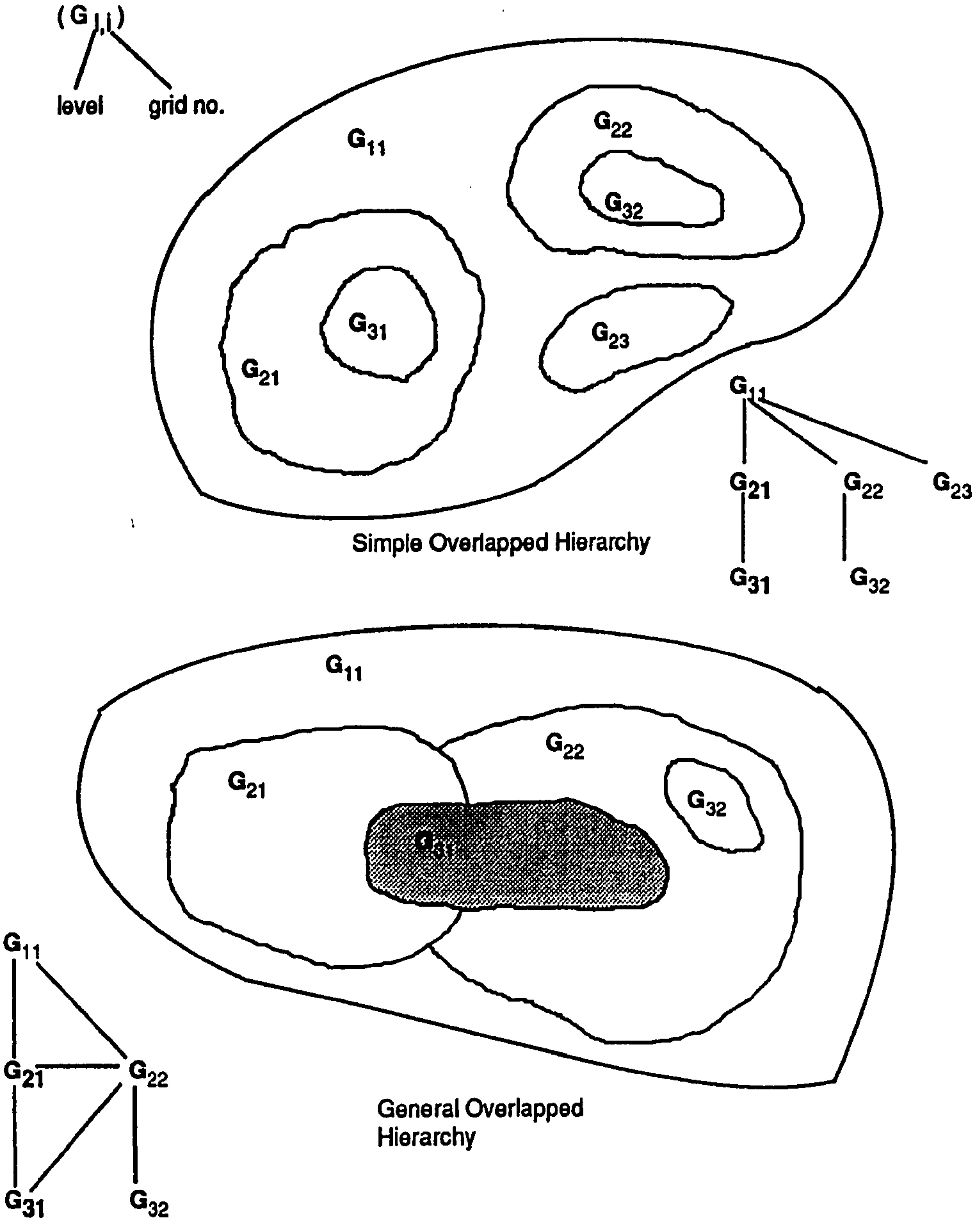


Fig. A.1 Hierarchial ordering for intergrid communications.

Appendix B
HOLE BOUNDARY

The composite grid generation starts with the subdomain grids being translated and rotated to their proper locations relative to a fixed, global origin. If cell center interpolation data between connected grids are needed, the subdomain grids are transformed from cell vertices to cell center points. The transformed grids are created in Stage 1 and are used throughout the six stages. The cell center grids are created by averaging the coordinates of the eight cell vertices (Fig. B.1). For example, the x-coordinate of the cell center is calculated as

$$X^{cell}_{i,j,k} = \{X_{i,j,k} + X_{i+1,j,k} + X_{i+1,j+1,k} + X_{i,j+1,k} + X_{i,j+1,k+1} + X_{i,j,k+1} + X_{i+1,j,k+1} + X_{i+1,j+1,k+1}\}/8 \tag{B.1}$$

Collapsed cell centers and edge points are defined on the last grid planes in the three coordinate directions. This is done to create the same number of cell centers as there are nodes. The collapsed cell centers are calculated by averaging the four vertices of a cell surface. For example, the collapsed cell center on the KMAX plane is calculated as

$$X^{cell}_{i,j,k\,\mathrm{max}} = \{X_{i,j,k\,\mathrm{max}} + X_{i+1,j,k\,\mathrm{max}} + X_{i+1,j+1,k\,\mathrm{max}} + X_{i,j+1,k\,\mathrm{max}}\}/4 \tag{B.2}$$

The edge points are defined on the IMAX, JMAX and KMAX grid corners. For example, the grid edge formed by the intersection of the JMAX and KMAX planes of the grid is calculated as

$$X^{edge}_{i,j\,\mathrm{max},k\,\mathrm{max}} = \{X_{i,j\,\mathrm{max},k\,\mathrm{max}} + X_{i+1,j\,\mathrm{max},k\,\mathrm{max}}\}/2 \tag{B.3}$$

It is important to note that the physical space constructed by the cell centers in less than the space constructed by the nodes. Thus, care is needed in connecting cells located at and near boundaries.

A search method is used to locate the holes created in each subdomain or global grid caused by other overlapped subdomains. The search procedure can be divided into six steps. It should be emphasized here that the search procedure given below is for the nodes of a finite difference grid. However, it is also valid for the cell centers of a finite volume grid. Hence, the same search procedure can be repeated for the finite volume grids if the cell centers are imagined in place of the nodes in Figs. B.2–B.4 (The words cell center and point are used interchangeably herein.).

Step 1: An initial hole boundary is specified as a surface, C, in the overlapped grid $G_{L+1,i}$ (Fig. B.2). The "i" index of $G_{L+1,i}$ will be dropped from now on for convenience.

Step 2: Outward normal vector, N, is constructed at each hole surface cell center using a vector cross product technique. Further details of this technique are given in Ref. [57].

Step 3: A temporary origin, P_o, of the initial hole is located by averaging the hole surface coordinates.

Step 4: A maximum search radius, R_{max}, is defined as the maximum distance from the origin of the hole to a cell on the hole boundary surface (Fig. B.3).

Step 5: The initial search determines whether a cell center (P) from the grid G_L lies within the search radius R_{max}. If the cell center P lies within the search circle then a vector dot product test is used.

Step 6: A vector dot product ($N \cdot R_p$ is the position vector from a hole surface point to a cell center P in G_L (Fig. B.3). If $N \cdot R_p > 0$, the cell center P lies outside the initial hole; otherwise the cell center P lies inside the initial hole and thus is defined as a hole point in grid G_L. A cell is defined as a hole cell, if its cell center lies inside the hole.

Figure B.4 shows a hole and its boundary in grid G_L generated by the overlapped grid G_{L+1}. A hole point is flagged for grid G_L by setting an array IFLAG = 0. A cell of G_L which is not in the hole, is flagged by setting IFLAG = 1. The next task is to locate the G_L cells which are immediate neighbors of the hole cells. These are called fringe cells, and the intergrid communication of conserved variables from G_{L+1} grid is performed on these cells. A fringe cell is also flagged IFLAG = 0. The fringe and hole cell centers in grid G_L are shown in Fig. B.5. A cell in G_{L+1} with the shortest distance to a fringe cell in G_L is located and called a TARGET cell. The TARGET cell is the starting point in the search for the cells are used for interpolation. The number of cells in G_{L+1}, surrounding the fringe cell in G_L, that need to be connected depends upon the order and accuracy of the interpolation procedure.

B.1 Intergrid Communication

A trilinear interpolation procedure is used in the intergrid communication of conserved variables. The significance, accuracy and conservative nature of using linear interpolation is discussed in Appendix C.

Once a target cell of G_{L+1} is located, a search is conducted to locate seven other cells in G_{L+1} near the target cell. The objective is to form a hexahedron which has the seven cell centers and the target cell center as the vertices, such that the hexahedron includes the center of a fringe cell from G_L. The information is transferred from the eight cells, which their centers define the vertices of the interpolation cell of G_{L+1}, to the fringe cell of G_{L+1} using trilinear interpolation. A typical interpolation cell of a body fitted grid is a warped hexahedron. The trilinear interpolation can only be used on cubes. Each interpolation cell containing a fringe cell at which a function value is to be interpolated is mapped to a unit cube using isoparametric mapping. Isoparametric mapping [74, 75] is the process of defining the same function that describes the geometry of the element as the function used to interpolate spatial variations of a variable at location P within the element (Fig. B.6). The isoparametric mapping assumes that the transformation between the natural ξ, η, ζ coordinates and the global X, Y, Z coordinates is unique. The order of the polynomial function used to represent the field variable within an element depends upon the number of nodal variables to evaluate the coefficients of the polynomial. Hence, the interpolation cell has eight nodal variables, and thus leads to the transformation/interpolation equation of the following form.

$$f = a_1 + a_2\xi + a_3\eta + a_4\zeta + a_5\xi\eta + a_6\xi\zeta + a_7\eta\zeta + a_8\xi\eta\zeta \tag{B.4}$$

where a_i, i = 1,...8 are coefficients depending on the values of f_i at the vertices of the unit cube (Fig. B.6). The terms, ξ, η, ζ are coordinates of the center of interpolated cell, P, relative to the center of target cell in the unit cube. The unit cube is mapped so that $0 \leq \xi$, η, $\zeta \leq 1$. For example, $a_1 = f_1$ is obtained when (ξ, η, ζ) = 0, 0, 0. The other coefficients are

$$\begin{aligned}
a_2 &= -f_1 + f_2\\
a_3 &= -f_1 + f_4\\
a_4 &= -f_1 + f_5\\
a_5 &= f_1 - f_2 + f_3 - f_4\\
a_6 &= f_1 - f_2 - f_5 + f_6\\
a_7 &= f_1 - f_4 - f_5 + f_8\\
a_8 &= -f_1 + f_2 - f_3 + f_4 + f_5 - f_6 + f_7 - f_8
\end{aligned} \tag{B.5}$$

Identifying the origin of the cube in the interpolation space relative to the coordinates in the physical space as (0, 0, 0) = $(X, Y, Z)_{i,j,k}$, the f_i values withe vertices become

$$\begin{aligned}
f_1 &= f_{i,j,k} & f_5 &= f_{i,j,k+1}\\
f_2 &= f_{i+1,j,k} & f_6 &= f_{i+1,j,k+1}\\
f_3 &= f_{i+1,j+1,k} & f_7 &= f_{i+1,j+1,k+1}\\
f_4 &= f_{i,j+1,k} & f_8 &= f_{i,j+1,k+1}
\end{aligned} \tag{B.6}$$

Note, the interpolation stencil can be identified by the target cell (i, j, k) because the other seven vertices are an extension of it. This simplifies the storage requirements for the interpolation data, since only the information for the target cell is needed. The last agenda of the interpolation procedure is to determine the values of ξ, η, ζ from the isoparametric mapping. The procedure to determine these values are given in Ref. [57].

In certain cases, the trilinear interpolation procedure fails for particular boundary cells. For those boundary cells that lie within another mesh that cannot obtain interpolation data corresponding to the three coordinate directions with values between 0 and 1, zeroth-order interpolation is used. The zeroth-order interpolation is performed from the TARGET cell, which is at a minimum distance away. There are several possible causes for failure of the trilinear interpolation or isoparametric mapping procedures. Failure can occur if the interpolation cell, that contains the boundary cell is extremely warped, which may cause improper transformation to the cube space, or if the Newton's iterative method of determining ξ, η, ζ from the system of equations fail. A loss of accuracy occurs at these cells is usually less than five percent of the total number of boundary cells, this method is usually acceptable. The inclusion of zeroth-order interpolation procedure in MaGGiE increases the robustness of the grid connection algorithm for subdomains of different topologies.

B.2 Outer Boundary

The procedure described in Stage 1 for fringe cells is repeated in an opposite manner at the outer boundary cells of the overlap region where information is transferred from the grid G_L to the grid G_{L+1}. Again, target cells in G_L and interpolation data are determined for the outer boundary cells in grid G_{L+1}.

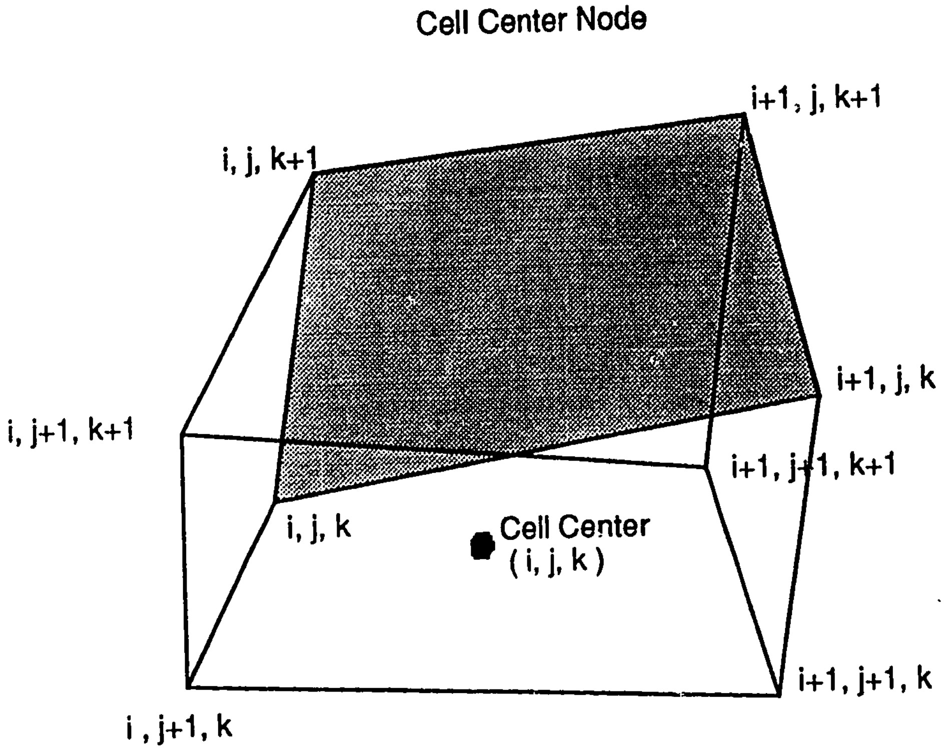


Fig. B.1 Sketch of a cell center node.

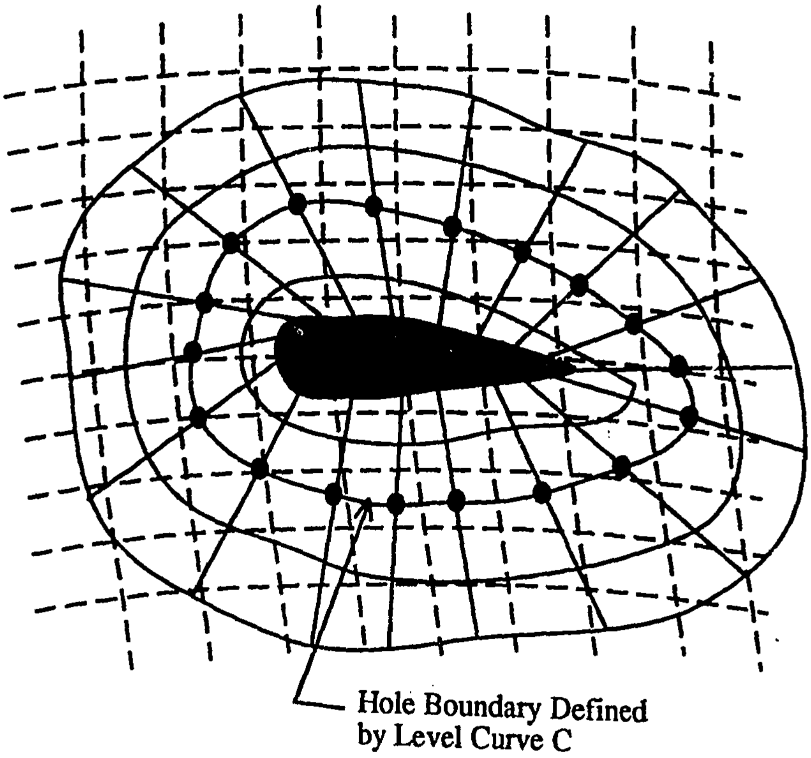


Fig. B.2 Sketch of an initial composite mesh hole boundary surface.

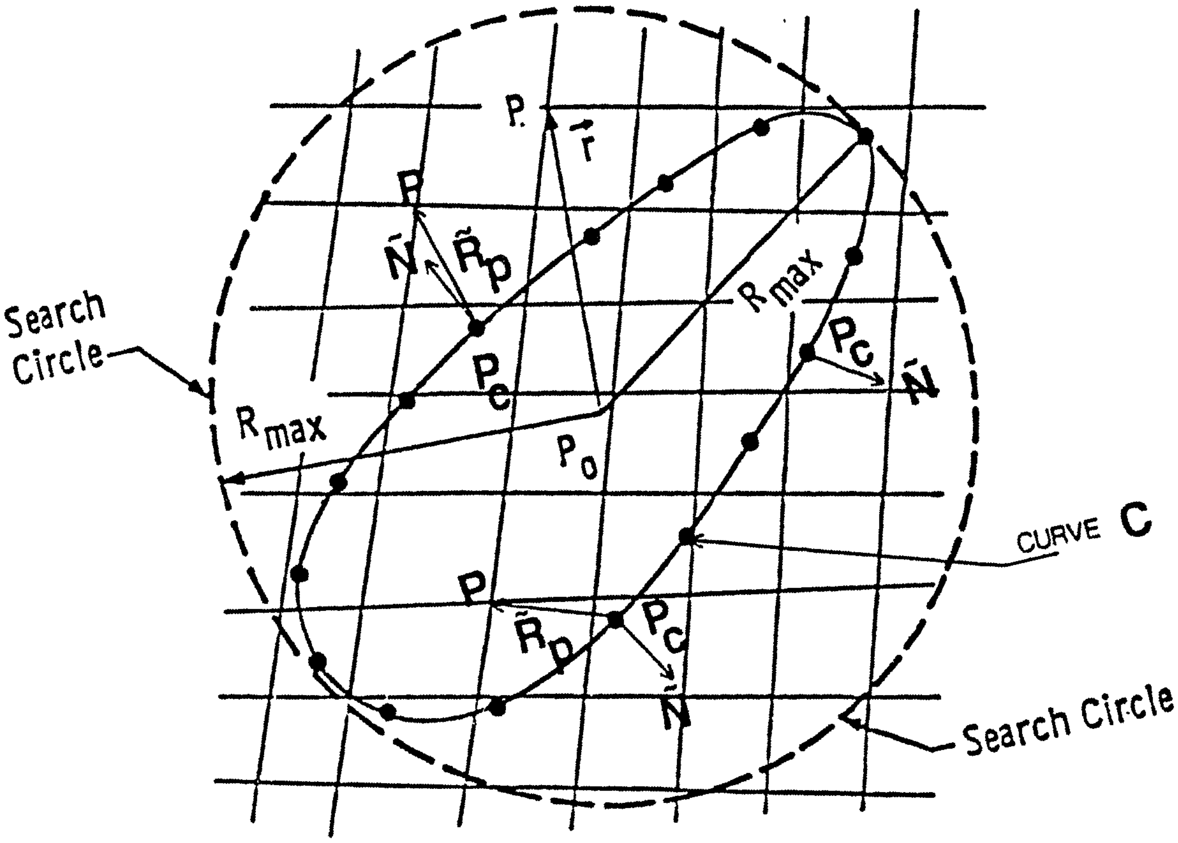


Fig. B.3 Sketch of hole search method.

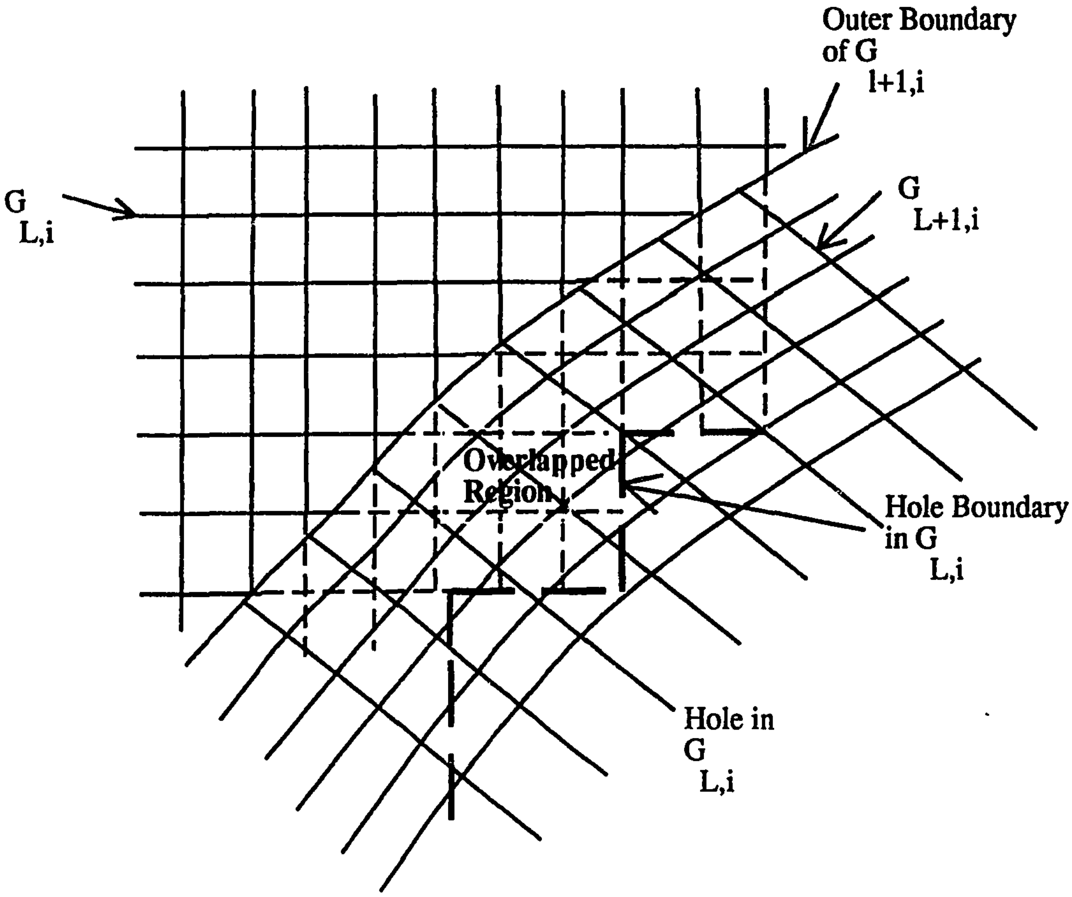


Fig. B.4 Sketch of the hole and outer boundary of a composite mesh.

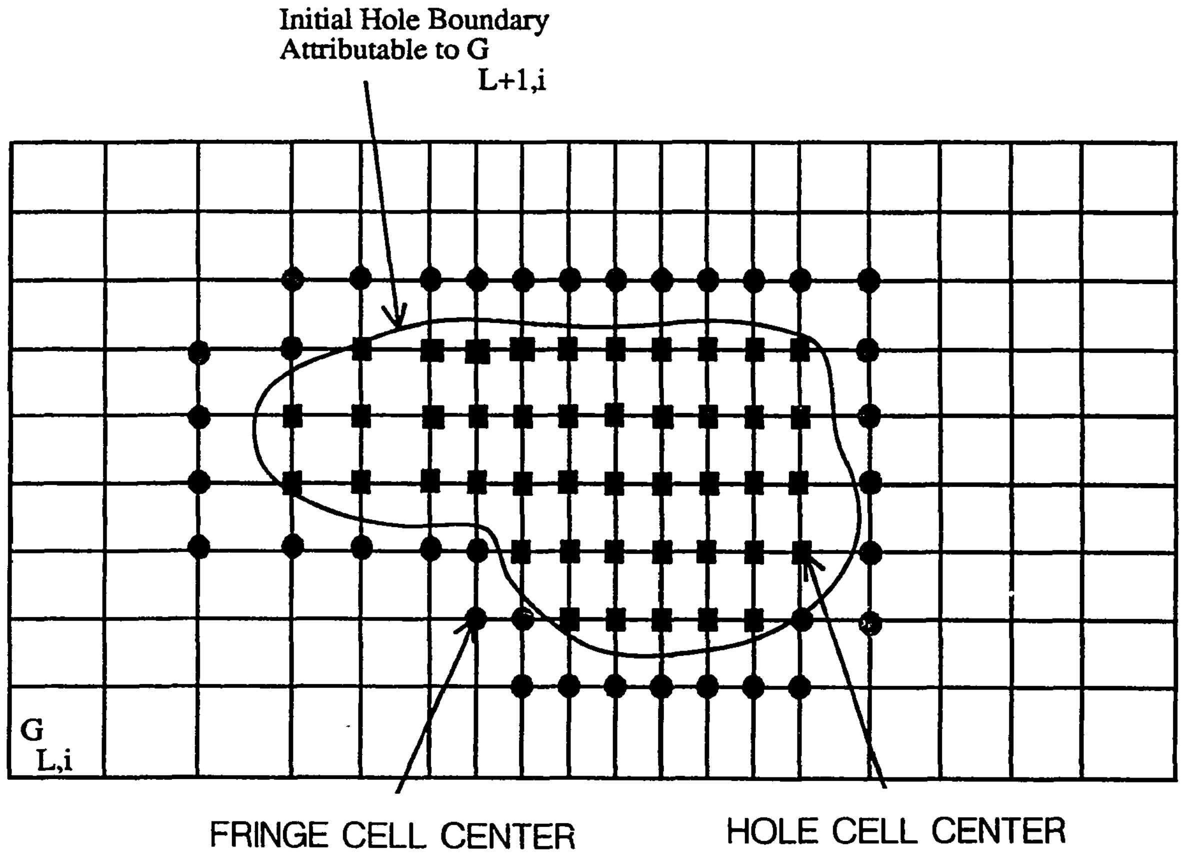


Fig. B.5 Sketch of fringe and hole cell centers in a composite mesh.

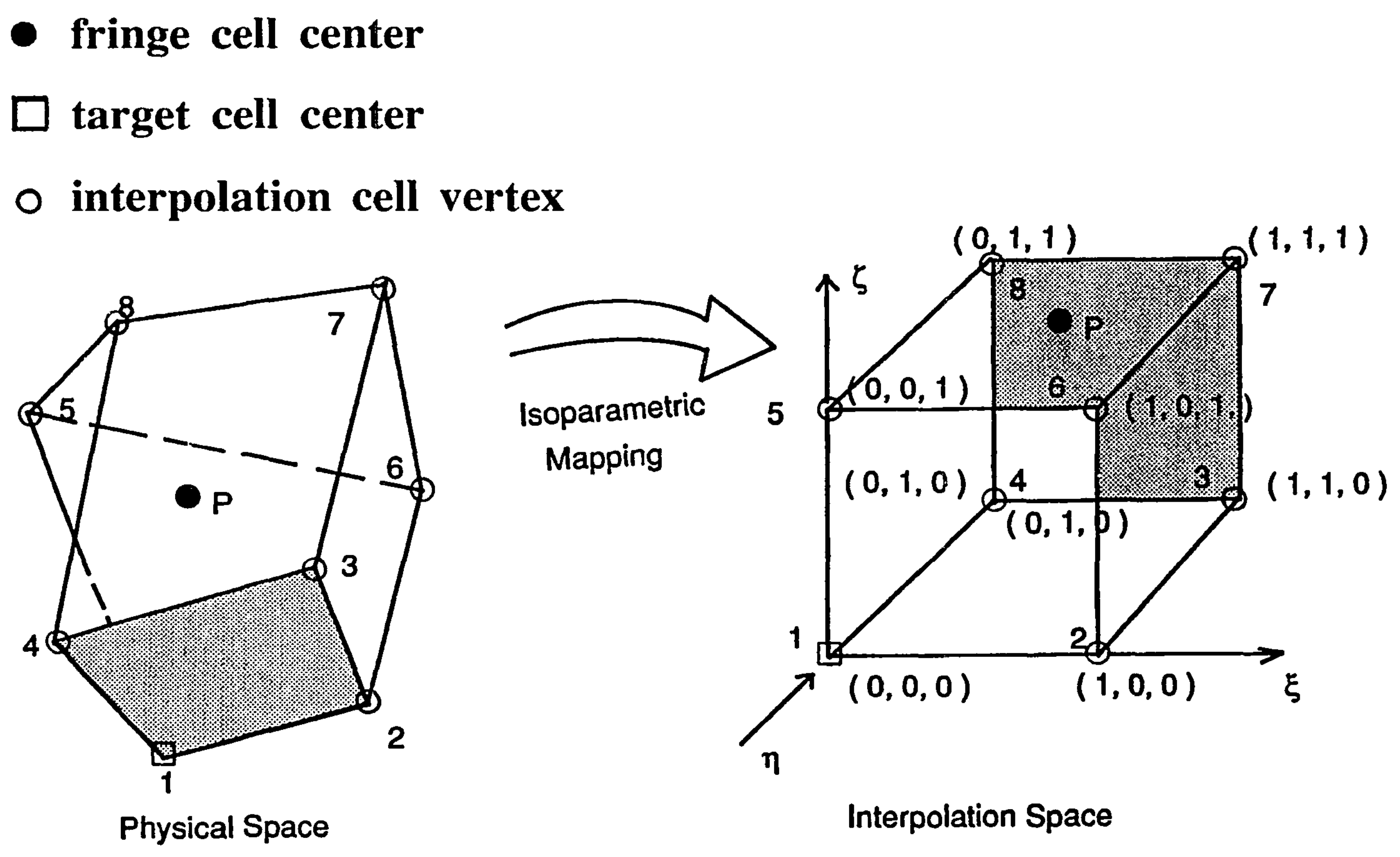


Fig. B.6 Skctch of isoparametric mapping.

Appendix C
ACCURACY

Subdomains generally overlap one another in an arbitrary fashion in the grid overlapping technique. Hence, it is desirable to utilize a conservation procedure for intergrid communication at the interface. This is beneficial, for example, in the case of maintaining shock capturing properties at the intergrid boundaries. The conservative intergrid communication guarantees the correct location of shocks passing through grid boundaries and eliminates the generation of artificial shocks at these boundaries. Some of the approaches currently being considered to maintain conservation at the intergrid boundaries or to improve the accuracy of the interpolation at those boundaries are presented in this appendix. It should be noted that this study uses the trilinear interpolation (a non-conservative approach) for the intergrid communication. The trilinear interpolation has been discussed in Appendix B.

C.1 Inter-Subdomain Conservation

(i) Matching Method: A general algorithm to derive a conservation interface condition for meshes which overlap in an arbitrary fashion is given by Berger [58]. The algorithm gives weak solutions to the differential equation if they converge on one- and two-dimensional overlapped grids. It also gives rise to formulas of arbitrary order of accuracy which requires approximation to the higher order derivative of the solution. However, formulas of higher order of accuracy may be unstable [58]. The remedy may be to use a conservative method of matching. A one-dimensional conservative method of matching is proposed in Ref. [58]. A consistent approximation to the integral of conserved variables determines a matching equation, which gives a weak solution to the partial differential equation being solved, if this approximation converges as the limit of h (spatial step size) goes to zero. If this matching equation is linear, the interpolation of fluxes, as needed for conservation, is equivalent to interpolation for the solution itself. The differencing stencil required for this scheme, based upon the concept of weak solution to the differential equation, is rather irregular even for two-dimensional curvilinear grids.

(ii) Spatial-Flux and Time-Flux Conservation Approach: The spatial-flux approach of Rai [7, 8] for zonal girds maintains global conservation by enforcing spatial-flux conservation along the interface. In ξ–direction, for example, it is expressed as

$$\int\int F^{(1)}\, \partial\eta\, \partial\zeta = \int\int F^{(2)}\, \partial\eta\, \partial\zeta \tag{C.1}$$

where $F^{(1)}$ and $F^{(2)}$ are the spatial flux of mass, momentum, and energy in the ξ–direction for Zone 1 and Zone 2, respectively (Fig. C.1). This scheme requires two points on either side of the interface to compute the flux. The flux in Zone 1 can be constructed from the data in Zone 1 and an interpolation of Zone 2. This scheme is used primarily for grid interfaces where the subdomain grids are patched at the boundaries.

The time-flux conservation presented by Thomas et al. [11] is an alternative to the spatial-flux approach. It does not rely on the boundaries being coincident, hence it may be suitable for overlapped grids. The time-flux of mass, momentum, and energy can be represented by Q, the conserved variables at the cell centers. In the overlap region, the conserved time fluxes can be expressed as

$$\int\int\int Q^{(1)}\, \partial\xi\, \partial\eta\, \partial\zeta = \int\int\int Q^{(2)}\, \partial\xi\, \partial\eta\, \partial\zeta \tag{C.2}$$

where $Q^{(1)}$is the time-flux in the Zone 1 and $Q^{(2)}$ is the time-flux in the Zone 2. Equation (C.2) and an analogous equation for the projection of Zone 1 into Zone 2 satisfy Eq. (C.1) only to within truncation error but it has been found [11] to maintain the conservation properties at the boundaries. If the grid spacing across the interface is assumed equal, Eq. (C.2) is reduced to a two-dimensional equation. On the other hand, if the assumption is not true, then the conservation of time-flux is accomplished by interpolation to the cell centers of one grid assuming a weighted variation of time-flux with the cells of the other grids. The interpolation procedure is conservative when the weighted variation of the time-flux is dependent on the percentage of the volume of the cell of the Zone 1 that overlaps a cell in Zone 2. However, the procedure to find the cell-volume weighted variations for a three-dimensional grid is geometrically complicated and usually cannot be generalized.

C.2 Global (Overall) Accuracy

An important issue in regarding the interpolation is its effect on the overall accuracy of the solution. The continuity and the conservation of the flow variables that can be maintained have been investigated previously, for example in Ref. [59]. An interpolation assumes the continuity of the interpolant. For the overall accuracy to be as good as the discretization formula used in Eq. (2.1), it is shown [59] that the width of the interpolation formula should be (0.25 p r + 1) if the width of the overlapped region is constant. In this formula, p denotes the order of the differential equation being solved, and r denotes the order of accuracy of the spatial discretization. Hence, if the differential equation is of order two (second-order) and spatial discretization is second-order accurate (r = 2), then the width of the interpolation formula is two. That is, two sets of fringe cells are needed for second-order accurate matching of the solutions to the second-order differential equations being solved here. For the present algorithm, the implementation of a second-order accuracy may not be feasible since it requires a second set of fringe cells. The width of the overlapped region should be increased in order to include the second sets of fringe and outer boundary cells; otherwise, the smaller overlapped region increases the risk of illegal communication between the two subdomains. An illegal communication between subdomains occurs when one or more cells, making up the information hexahedron (for updating process), are fringe cells themselves. Also, increasing the width of the interpolation formula increases the computer memory and the run time of the flow solver, because there are twice as many cells to update.

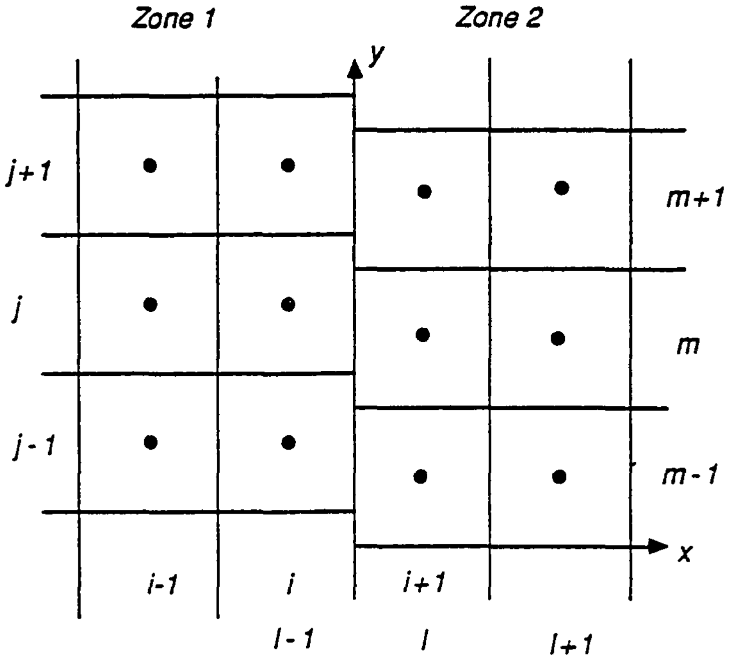


Fig. C.1 A two-dimensional zonal grid.

BIOGRAPHY

The author was born on August 6, 1961 in Tehran, Iran. He came to the United States in 1978 in pursue of higher education. He received his Bachelor's degree in April 1982 from Florida International University, Miami, Florida. Following a year of work experience, he enrolled in the graduate program in Mechanical Engineering at Old Dominion University. He received his Master's degree in May 1986 and then continued in the Doctoral program. The author has been involved in research work pertaining to CFD at NASA Langley Research Center.