

Old Dominion University

ODU Digital Commons

Mechanical & Aerospace Engineering Theses & Dissertations

Mechanical & Aerospace Engineering

Summer 1988

Application of Advanced Grid Generation Techniques for Flow Field Computations About Complex Configurations

Monchai Kathong
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/mae_etds



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Kathong, Monchai. "Application of Advanced Grid Generation Techniques for Flow Field Computations About Complex Configurations" (1988). Doctor of Philosophy (PhD), Dissertation, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/tex1-x659
https://digitalcommons.odu.edu/mae_etds/246

This Dissertation is brought to you for free and open access by the Mechanical & Aerospace Engineering at ODU Digital Commons. It has been accepted for inclusion in Mechanical & Aerospace Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

APPLICATION OF ADVANCED GRID GENERATION
TECHNIQUES FOR FLOW FIELD COMPUTATIONS
ABOUT COMPLEX CONFIGURATIONS

by

Monchai Kathong
B.S. January 1980, The Royal Thai Naval Academy
Samut Prakan, Thailand
M.S.M.E. December 1982, University of Wisconsin
Madison, Wisconsin

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 AND MECHANICS

OLD DOMINION UNIVERSITY

August, 1988

Approved by:

Dr. S. N. Tiwari (Director)

Dr. R. E. Smith

Dr. A. S. Roberts

Dr. O. Baysal

Dr. J. M. Dorrepaal

ABSTRACT

APPLICATION OF ADVANCED GRID GENERATION TECHNIQUES FOR FLOW FIELD COMPUTATIONS ABOUT COMPLEX CONFIGURATIONS

Monchai Kathong
Old Dominion University, 1988
Director: Dr. Surendra N. Tiwari

In the computation of flow fields about complex configurations, it is very difficult to construct a boundary-fitted coordinate system. An alternative approach is to use several grids at once, each of which is generated independently. This procedure is called the "multiple grids" or "zonal grids" approach, and its applications are investigated in this study. The method is a conservative approach and provides conservation of fluxes at grid interfaces. The Euler equations are solved numerically on such grids for various configurations. The numerical scheme used is the finite-volume technique with a three-stage Runge-Kutta time integration. The code is vectorized and programmed to run on the CDC VPS-32 computer.

Steady state solutions of the Euler equations are presented and discussed. The solutions include: low speed flow over a sphere, high speed flow over a slender body, supersonic flows over a Butler-Wing at various Mach numbers and angles of attack, supersonic flow through a duct, and supersonic internal/external flow interaction for an aircraft configuration at various angles of attack. The results demonstrate that the multiple grids approach along with the conservative interfacing is capable of computing the flows about the complex configurations where the use of a single grid system is not possible.

DEDICATION

To my parents, Mr. Mongkol and Mrs. Boonrean Kathong.

ACKNOWLEDGMENTS

I would like to thank my academic advisor, Dr. Surendra N. Tiwari, for his help and encouragement. I wish to thank my dissertation committee members, Dr. Robert E. Smith, Dr. A. S. Roberts, Dr. O. Baysal and Dr. J. M. Dorrepaal, for proving me with helpful suggestions. I thank Dr. Robert E. Smith of NASA Langley Research Center for the financial support.

I am indebted to Dr. Lars-Erik Eriksson of Old Dominion University (on leave from the Aeronautical Research Institute of Sweden) for many technical discussions relative to the research and suggested directions to pursue. I wish to thank Dr. C. Wayne Mastin of Mississippi State University for several helpful discussions.

I wish to thank Mrs. Lona M. Howser and the VPS-32 (dayshift) operators for their consultation on the VPS-32 computer. I thank Mr. Micheal R. Wiese of Computer Sciences Corporation for providing me with the reference data. I thank Mrs. Rowena Pierce and Ms. Diane Riddick for typing part of this dissertation. I also recognize Mrs. Merita Aguiard who assisted in the preparation of some figures. I wish to express my deepest appreciation to my parents, Mr. Mongkol and Mrs. Boonrean Kathong, for their moral support. Also, I would like to thank the Royal Thai Navy and the National Aeronautics and Space Administration for their support of my graduate program and the research presented in this study.

TABLE OF CONTENTS

	page
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF SYMBOLS	x
Chapter	
1 INTRODUCTION	1
2 GRID GENERATION	8
2.1 Introduction	8
2.2 Transfinite Interpolation	16
2.3 Mapping Type and Singularities	23
2.4 Multiple Grids	27
2.5 Brief Discussion on Conservative Rezoning Algorithm	36
2.6 Application to The Hyperbolic Equations	39
3 BASIC FORMULATION	41
3.1 Governing Equations	44
3.2 Spatial Finite-Volume Discretization	46
3.3 Artificial-Viscosity Model	53
3.3.1 Nonlinear Artificial Viscosity	54
3.3.2 Linear Artificial Viscosity	55

3.4	Boundary Conditions	55
3.4.1	Solid Walls	56
3.4.2	Coordinate Cuts	59
3.4.3	Inflow/Outflow Boundaries	59
3.4.4	Interface Conditions	62
3.5	Numerical Time Integration	69
3.6	Local Time Step Scaling	69
4	APPLICATION TO THE AERODYNAMIC BODIES	74
4.1	Sphere and Slender Body	76
4.2	Butler Wing Configuration	82
4.3	Rectangular Duct with 10^0 Ramps	99
4.4	Internal/External Flow About a Fighter Aircraft Configuration . . .	104
4.4.1	Grid Generation Procedure	107
4.4.2	Zero Degree Angle of Attack(Without Wake Region)	110
4.4.3	Different Angles of Attack (Without Wake Region)	124
4.4.4	Zero Degree Angle of Attack (With Wake Region)	142
5	CONCLUSIONS	156
	REFERENCES	159
	Appendices	
A	THE IMPLEMENTATION OF THE CONSERVATIVE REZONING ALGORITHM	170
B	THE CALCULATION OF CELL SURFACE AREA AND VOLUME	184
	BIOGRAPHY	191

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2.1 Boundary-fitted coordinate system	10
2.2 Physical versus Computational domain	12
2.3 A computational domain where \vec{f} is known on certain planes	18
2.4 An out-of-surface derivative of \vec{f}	20
2.5 A computational domain where $\frac{\partial \vec{f}}{\partial n}$ are specified on the outer surface	22
2.6 Mapping types for a 3D airfoil.	24
2.7 An O-O mapping for a wing-fuselage configuration.	25
2.8 Singularity lines for the O-O mapping.	26
2.9 The boundary surface is opened to avoid dealing with singularities.	28
2.10 Zoning of multiple-connected region.	30
2.11 Grid patching versus grid overlapping.	31
2.12 Patched surfaces.	38
3.1 A typical volume element.	48
3.2 A solid wall boundary.	58
3.3 Coordinate cut boundaries.	60
3.4 Conditions at the inflow/outflow boundaries.	63
3.5 A typical interface.	65
3.6 A typical patched interface.	66
4.1 Grid system for a sphere.	78

4.2	Grid system for a slender body.	79
4.3	The pressure coefficient on the surface of a sphere.	80
4.4	The pressure coefficient on the surface of a slender body.	81
4.5	Physical model of a Butler-Wing.	83
4.6	Grid system used for a Butler-Wing.	85
4.7	The pressure coefficient on the surface of a Butler- Wing($M = 3.5, \alpha = 0^0$).	86
4.8	Comparison of C_p with the single grid calculation.	87
4.9	C_p on the surface of a Butler-Wing at different cross cut ($M = 3.5, \alpha = 10^0$).	89
4.10	Pressure distribution for a Butler-Wing ($M = 3.5, \alpha = 10^0$).	93
4.11	C_p on the surface of a Butler- wing($M = 2.5, \alpha = 0^0$)	94
4.12	C_p on the surface of a Butler-Wing at different cross cut ($M = 2.5, \alpha = 10^0$)	95
4.13	Grid system for flow through a duct.	100
4.14	C_p contours inside a duct.	102
4.15	Pressure distribution inside a duct.	103
4.16	A fighter aircraft configuration.	105
4.17	Two alternative topologies for the exterior grids.	108
4.18	The surface grid for the fighter aircraft.	111
4.19	Exterior grids for the fighter aircraft.	112
4.20	The enlarge view of a cross cut of grids in the wake region.	113
4.21	Grid lines at the symmetry plane of the aircraft.	114
4.22	Grid lines as in Fig. 4.21 extended to the outer boundary.	115
4.23	Computational domains for the aircraft.	116
4.24	C_p contours on the top surface ($\alpha = 0^0$, no wake).	117
4.25	C_p contours on the bottom surface ($\alpha = 0^0$, no wake).	118

4.26 C_p contours on the symmetry plane ($\alpha = 0^0$, no wake).	119
4.27 C_p contours at various cross cuts ($\alpha = 0^0$, no wake).	120
4.28 Spanwise pressure distribution at the inlet intake ($\alpha = 0^0$, no wake).	122
4.29 Pressure distribution inside the inlet ($\alpha = 0^0$, no wake).	123
4.30 C_p contours on the top surface ($\alpha = 3.79^0$, no wake).	125
4.31 C_p contours on the bottom surface ($\alpha = 3.79^0$, no wake).	126
4.32 C_p contours on the symmetry plane ($\alpha = 3.79^0$, no wake).	127
4.33 C_p contours at various cross cuts ($\alpha = 3.79^0$, no wake).	128
4.34 Spanwise pressure distribution at the inlet intake ($\alpha = 3.79^0$, no wake).	129
4.35 Pressure distribution inside the inlet ($\alpha = 3.79^0$, no wake).	130
4.36 C_p contours on the top surface ($\alpha = 7^0$, no wake).	132
4.37 C_p contours on the bottom surface ($\alpha = 7^0$, no wake).	133
4.38 C_p contours on the symmetry plane ($\alpha = 7^0$, no wake).	134
4.39 C_p contours at various cross cut ($\alpha = 7^0$, no wake).	135
4.40 Spanwise pressure distribution at the inlet intake ($\alpha = 7^0$, no wake.)	136
4.41 C_p contours on the top surface ($\alpha = 10^0$, no wake).	137
4.42 C_p contours on the bottom surface ($\alpha = 10^0$, no wake).	138
4.43 Spanwise pressure distribution at the inlet intake ($\alpha = 10^0$, no wake).	139
4.44 C_p contours on the symmetry plane ($\alpha = 10^0$, no wake).	140
4.45 C_p contours at various cross cuts ($\alpha = 10^0$, no wake).	141
4.46 Pressure distribution along top wall of the inlet.	143
4.47 Pressure distribution along center of the inlet.	144
4.48 Pressure distribution along bottom walls of the inlet.	145
4.49 Pressure distribution from freestream to exit plane of the inlet for various angles of attack.	146
4.50 C_p contours on the top surface ($\alpha = 0^0$, with wake).	147
4.51 C_p contours on the bottom surface ($\alpha = 0^0$, with wake).	148

4.52	Spanwise pressure distribution near the end of the fuselage ($\alpha = 0^0$, with wake).	149
4.53	C_p contours on the symmetry plane ($\alpha = 0^0$, with wake).	151
4.54	C_p contours at various cross cuts ($\alpha = 0^0$, with wake).	152
4.55	Pressure distribution from the inlet intake to end of the wake region.	154
4.56	Pressure distribution from freestream to end of the wake region.	155
A.1	A patched planar interface.	171
A.2	The first method for searching.	175
A.3	A quadrilateral is divided into two triangles.	176
A.4	Mapping of a triangle onto a unit triangle.	177
A.5	The shortest distance between a point P and a grid point (i,j).	179
A.6	Three vectors emanating from a point (i,j).	180
A.7	Conditions that determine in which grid cell the point P lies.	181
B.1	An arbitrary hexahedral.	185
B.2	A surface of a hexahedral.	186
B.3	A hexahedral is composed of five tetrahedra.	188
B.4	An arbitrary tetrahedral.	189

LIST OF SYMBOLS

A_N	Area of new grid cell
A_{NO}	Overlapped area
A_O	Area of old grid cell
A_P	Area of a polygon
E	Total energy
\vec{F}	Vector of fluxes in x direction
\vec{G}	Vector of fluxes in y direction
\vec{H}	Vector of fluxes in z direction
J	Jacobian of coordinate transformation
M	Mach number
N_P	Number of sides of a particular polygon
Q_N	A quantity in the new grid cell
Q_O	A quantity in the old grid cell
$\bar{a}_k(u, w)$	Known function on surfaces of constant u
$\bar{b}_k(u, w)$	Known function on surfaces of constant v
$\bar{c}_k(u, v)$	Known function on surfaces of constant w
c_p	Specific gas at constant pressure
c_v	Specific gas at constant volume
e	Specific internal energy
$\vec{f}(u, v, w)$	A vector value function
h	Fluxes per unit area = $\frac{H}{A}$
p	pressure
\vec{q}	Vector of state variables
q_O	Volumetric density of Q_O
t	time
u, v, w	x, y, z components of velocity
VOL_{ijk}	Volume of a volume element
x, y, z	Cartesian coordinate system
$\alpha_k, \beta_k, \gamma_k$	Univariate blending functions
γ	Specific gas ratio = $\frac{c_p}{c_v}$
δ_{kl}	Kronecker Delta
λ	An eigenvalue
μ_I, μ_J, μ_K	average operators
ξ, η, ζ	Curvilinear coordinate system
ρ	Density
Ω	An arbitrary region

Chapter 1

INTRODUCTION

Basically, there are three approaches or methods which can be used to solve a problem in fluid mechanics and heat transfer. These methods are (1) Experimental, (2) Theoretical, and (3) Numerical. The theoretical method is often referred to as an analytical approach while the terms numerical and computational are used interchangeably.

In the experimental approach, a model is constructed and tested in a testing facility such as a wind tunnel. The flow variables, such as wall pressure and temperature can then be measured. In most cases, experiments are performed on a small-scale model since full-scale tests are prohibitively expensive and often impossible. These small-scale tests do not always simulate all the features of the full-scale tests. General rules for extrapolating the resulting information to full-scale are often unavailable. Also, there are serious difficulties of measurement in many situations, and the measuring instruments are not free from errors. Furthermore, the problem of producing required freestream conditions in the test section of the facility can be quite troublesome and time consuming. Since the facility, for example a wind tunnel, requires large amounts of energy for its operation, its operating costs are quite high. The experimental approach produces the most realistic answers for many flow problems; however, the costs are becoming greater everyday.

In the theoretical approach, assumptions are made in order to simplify the problem. A closed form solution is generally sought. The main advantage of

this approach is that general information which is usually in formula form can be obtained. Its disadvantage is that the problem is restricted to simple geometry and physics.

In the numerical approach, a limited number of assumptions are made and a high-speed digital computer is used to solve the resulting governing fluid dynamics equations. The major advantage of this approach is that the problem is free of some of the constraints imposed on the experimental or theoretical approach. Thus, the numerical approach has the potential of providing information not available by other means. However, the approach does have some disadvantages. The storage and speed of present available computers pose limitations on the method. Other limitations arise due to the inability to understand and mathematically model certain complex phenomena. In spite of these limitations, the numerical approach is becoming more popular. The developments of supercomputers and the reduction in computational costs have made the approach appealing.

A new methodology for attacking the complex problem in fluid mechanics and heat transfer has been developed and has become known as Computational Fluid Dynamics (CFD). Some of the ideas of this numerical approach are very old. CFD is a science of producing numerical solutions to a system of partial differential equations which describe fluid flow. CFD is done by discrete methods and the purpose is to better understand qualitative and quantitative physical phenomena in the flow which then is often used to improve upon engineering design. CFD brings together a number of different traditional disciplines: fluid mechanics, the mathematical theory of partial differential equations, computational geometry, numerical analysis, and the computer science of programming algorithms and processing data structures. Good surveys on the approach can be found in [1-13]*. Also, the

*The numbers in brackets indicate references.

foundations upon which the whole field is built are now reasonably well covered in text books [14-25].

In the CFD calculation, the continuum problem of the differential equations is projected to some finite-dimensional space for the dependent and independent variables. The resulting discrete equations are, then, solved for the final set of numbers. Thus, the first step in CFD is to discretize the domain of the flow by laying out a network of points situated at a finite number of different locations of the independent variables. This brings about a technique called the "grid generation" procedure. A "grid" is conventionally defined as a set of grid points in a coordinate system. The word grid and mesh are also used interchangeably. Grid generation is an essential procedure in CFD. Accuracies and stabilities of the CFD calculations depend a great deal on the properties of these "grids". Grid points are generally generated by letting some coordinate lines coincident with the boundaries of the domain. The purpose of generating these so-called "boundary-fitted" coordinates is to be able to apply boundary conditions directly when partial differential equations are solved on such grid. It is important that the boundary condition be represented accurately since the region in the intermediate vicinity of solid surfaces is generally dominant in determining the character of the flow. The procedure for generating a boundary-fitted coordinate system can be divided into two categories. These are partial differential equation methods and algebraic methods. In the partial differential equation methods, a set of partial differential equations, subjected to some boundary conditions, are solved to obtain a set of grid points. The partial differential equations may be elliptic, hyperbolic or parabolic. The partial differential equation methods offer the smoothness to the resulting grid points but generally require large amounts of computational time. The algebraic methods are based on mathematical interpolation functions and do not require the solution of differential equations or the use of complex variables. The primary advantages of algebraic

methods are speed and directness. Regardless of how grid points are generated, all CFD calculations are usually done on a rectangular domain with a square grid. This is done by transforming the set of partial differential equations of interest, and the associated boundary conditions, to the curvilinear system. The grid points in the physical domain are, thus, mapped into a set of equally spaced grid points in a rectangular region called the computational domain. With the transformation, the CFD calculations can be performed entirely on the fixed rectangular space regardless of the geometry or motion of the boundaries. Thus, numerical grid generation is the process of establishing an ordered and strategic distribution of grid points in a physical coordinate system corresponding to a uniform distributions of grid points in a rectangular computational coordinate system. Details on the grid generation procedures are described in Chap. 2. Surveys of the methods including textbooks on grid generation procedure can also be found in [26-31].

It is obvious that a grid which maps the entire physical domain onto a “slab” in the computational domain is very desirable. This type of grid, called the single grid or single-block grid, offers a considerable simplicity to the CFD calculation. However, for flow about complex configurations, the generation of a smooth and efficient single grid is very difficult. In some cases, especially those of configurations with several components, it may not be possible to obtain this type of grid at all. An alternative approach to this problem is to use several grids at once, each in a different coordinate system. The entire physical domain is, thus, subdivided into several subdomains. The generation of grids in different subdomain is generally independent from each other. This approach called the “multiple grids” or “zonal grid” approach can be categorized into two groups: grid patching and grid overlapping. For the patched grid approach, the subdomain grids are joined or patched together along common boundaries. The subdomain grids are overlapped rather than joined in the grid overlapping approach. The multiple grids approach is

becoming more common as the complexity of the configuration being considered in CFD is increased. However, the approach introduces new boundaries which are not the physical boundaries. Even though, the solution procedure is done separately in each subdomain, certain boundary conditions are needed at these fictitious boundaries. The difficulty of the approach, thus, lies in the treatment of these boundary conditions. Since these boundaries are either joined or overlapped with other subdomains, some information needs to be transferred between subdomains so that the computation of the entire physical domain is consistent. The interpolation of flow variables between subdomains seems to be the simplest choice. However, this procedure does not result in a computational scheme which is conservative. The conservation of a computational scheme is important when the flow being considered contains discontinuities such as shock waves. A computational scheme is said to be conservative when it maintains the discretized version of the conservation law (conservation of mass, momentum and energy) exactly (except for round-off errors) for any grid size over an arbitrary finite region containing any number of grid points. For the multiple grids calculation the information needs to be transferred conservatively between subdomain grids. It has been suggested that fluxes rather than flow variables be transferred between subdomain grids, so that the resulting computational scheme is conservative. It can be shown that the conservation is easier to enforce in the grid patching approach than in the grid overlapping approach. This study follows the grid patching approach along with the conservative treatment at the interfaces, i.e., places where two or more subdomain grids are joined together. The procedure in doing so is discussed in detail in Sec. 2.5.

The viscous Navier-Stokes equations are the ultimate equations to be solved in most CFD applications. However, the Navier-Stokes flow simulation are presently still in the stage of research. A success in the Navier-Stokes flow calculations relies not only on the numerical methods but also on the turbulent modeling.

Since this study focuses on the concept of multiple grids approach, the inviscid Euler equations are sufficient to be used as the model equations. The Euler equations result from dropping the viscous terms from the Navier-Stokes equations. Thus, the problems of computer storage and computational time including the uncertainties of turbulent modeling that arise in the Navier-Stokes calculation can be eliminated. Solutions of the Euler equations, though inviscid, describe the correct phenomena for many flow problems. The integral form of the Euler equations is applied to this study. The integral form may be important for the correct capturing of discontinuities in the flow. The discussion on the Euler equations is given in Sec. 3.1. The Euler equations are discretized by means of the centered finite-volume method. The finite-volume formulation is obtained by applying the integral form of the Euler equations to each grid cell of a given grid. The finite-volume method is cell-oriented rather than grid points oriented. The main advantage of the method is that it can be applied to the general geometry without the need for a global coordinate transformation and it can tolerate the grid singularities since the flow equations are balanced only within the cells of the grid. The steady state solution is obtained by means of the time-dependent technique. The time derivative terms are reintroduced to the Euler equations and the steady state solutions are reached by explicitly marching the solution procedure in time from the initially guessed solutions. The three-stage Runge-Kutta integration scheme is used to serve this purpose. Since the transient solutions are of no concern, the local time step scaling is applied to accelerate the solutions to the steady-state. The linear and non-linear artificial dissipation terms are also added to the discrete Euler equations. The purpose of adding these terms is to impose an entropy condition which is required to eliminate the non-physical shocks. Furthermore, the addition of the artificial dissipation terms helps eliminating the oscillation of solutions which prevents the solutions from reaching the steady state. Boundary conditions are of four types:

solid wall conditions, inflow/outflow conditions, interface conditions and coordinate cuts. The finite-volume discretization along with numerical time integration and boundary conditions is described in detail in Chap. 3. The concept of local time step scaling and artificial dissipation are also addressed. The application of the approach to the flow over a sphere at a low Mach number and to the flow over a slender body at the supersonic Mach number are discussed in Sec. 4.1. Section 4.2 describes the application of the multiple grids approach to the flow over a Butler-Wing configuration. Solutions for supersonic flow through a rectangular duct with 10° ramps are presented in Sec. 4.3. In Sec. 4.4, the application to the internal/external flow about a fighter aircraft configuration are described; solutions are shown at a supersonic speed and various angles of attack.

Chapter 2

GRID GENERATION

2.1 Introduction

Grid generation is an important procedure in CFD calculation. The word “grid” is generally used as a label for a complete set of grid points. Even though it is felt that grid generation and solution procedure are separate and distinct operations, in practice, these two operations can never be totally independent. This is because the accuracy of solutions depends upon grids on which the partial differential equations are solved. In turn, the logistic structure of the data (such as grid spacing), the location of outer boundaries and the nature of coordinate cuts are influenced by the nature of solutions. Perhaps the greatest problem of grid generations is not how to construct grids, rather, the problem is defining in sufficient detail what qualities and properties in a grid are desirable for a particular numerical method.

The representation of boundaries is best accomplished when the boundary is such that it is coincident with some coordinate line, for then the boundary can be made to pass through the points of a grid constructed on the coordinate lines. Different expressions at, and adjacent to, the boundary may then be applied using only grid points (the intersection of coordinate lines) without the need for any interpolation between points of the grid. The avoidance of interpolation is particularly important for boundaries with strong curvature or slope discontinuities, both of which are common in physical applications. Likewise, interpolation between grid

points not coincident with the boundaries is particularly inaccurate with differential systems that produce large gradients in the vicinity of the boundaries, and the character of the solution may be significantly altered in such cases. In most partial differential equation systems the boundary conditions are the dominant influence on the character of the solution, and the use of grid points not coincident with the boundaries, thus, places the most inaccurate difference representation in precisely the region of greatest sensitivity. The generation of a curvilinear coordinate system with coordinate lines coincident with all boundaries, so-called "boundary-fitted" coordinate system (Fig. 2.1), is thus an essential part of a general numerical solution of a partial differential equation system.

Any partial differential equation system can be solved on the boundary-fitted coordinate system by transforming the set of partial differential equations of interest, and associated boundary conditions, to the curvilinear system. Since the boundary-fitted coordinate system has coordinate lines coincident with the surface contours of all bodies presented, all boundary conditions can be expressed at grid points. Normal derivatives on the bodies can be represented using only finite difference between grid points on coordinate lines, without need of any interpolation. The transformed equations can then be approximated using finite difference expressions and solved numerically in the transformed plane. Thus, regardless of the shape of the physical boundaries, and regardless of the spacing of the finite grid in physical field, all computations can be done on a rectangular field with a square grid with no interpolation required on the boundaries. Moreover, the physical boundaries may even be time dependent without affecting the grid in the transformed region. Another major advantage of using boundary-fitted coordinates is that the computer software generated to approximate the solution of a given set of partial differential equation is completely independent of the physical geometry of the problem. Numerical grid generation is thus the process of establishing an ordered and

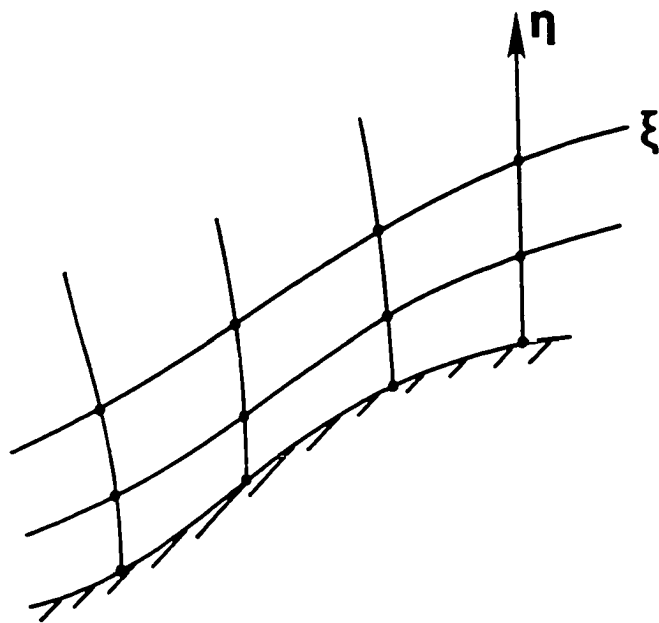
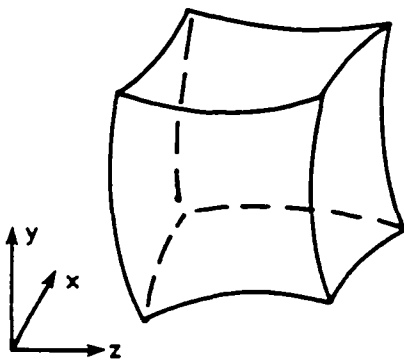


Fig. 2.1 Boundary-fitted coordinate system.

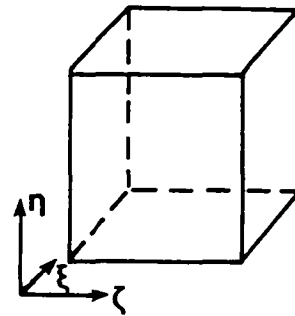
strategic distribution of grid points in a physical coordinate system corresponding to a uniform distribution of grid points in a rectangular computational coordinate system. Some of the basic ideas of the use of boundary-fitted curvilinear coordinate systems in the numerical solution of partial differential equations are discussed in [32]. Examples of physical and computational domains are illustrated in Fig. 2.2.

Two primary categories for arbitrary coordinate generation have been developed. These are algebraic methods and partial differential equation methods. The algebraic procedures include simple normalization of boundary curves, transfinite interpolation from boundary surfaces, the use of intermediate interpolating surfaces, and various other techniques. The partial differential equation may be elliptic, parabolic, or hyperbolic. Included in elliptic systems are both the conformal, and quasiconformal mappings, the former being orthogonal. Orthogonal systems do not have to be conformal, and may be generated from hyperbolic systems as well as from elliptic systems.

Algebraic transformations are attractive in that no numerical solution of a partial differential equation is involved. Thus, the primary advantages of algebraic methods are speed and directness. The major disadvantage of these methods is the lack of smoothness that results when an elliptic partial differential system is used to generate the grid and truncation errors may be significant in regions where the grid is not smooth [33]. For instance, the results of Shang [34] show kinks in the solution corresponding to regions of rapid grid spacing change radiating outward from the boundary. It should be noted that local controls in the multisurface transformation [35] can be used to prevent nonsmooth boundary behavior (e.g., slope discontinuities) from propagating inward. Transfinite interpolation described by Gordon and Hall [36] is a highly generalized algebraic grid generation method. Transfinite interpolation is applied through a series of univariate interpolations where blending functions and the associated parameters (point position and/or derivatives)



(a) Physical domain



(b) Computational domain

Fig. 2.2 Physical versus Computational domain.

determine a grid. For aerodynamic applications, Eriksson [37] and Rizzi and Eriksson [38] have adopted the original transfinite interpolation formulation to use only exterior boundary descriptions and derivatives at certain boundaries. They have also incorporated exponentials into the blending functions to concentrate the grid near an exterior boundary. The multisurface method [31,35] provides formulas for grid definition based on grid descriptions of two boundary surfaces and an arbitrary number of intermediate control surfaces. Choosing interpolants (defined similar to blending functions) and the placement of the control surfaces determines grid shape and spacing. The multisurface method has been used by Eiseman in numerous applications [39,40] but most notably for computing grids about turbine cascades. The two-boundary technique [41-43] is based on the description of two exterior boundaries and the application of either linear or hermite cubic polynomial interpolation to compute the interior grid. For a cubic interpolation, the surface derivatives combined with magnitude coefficients control the orthogonality of the grid at and near the boundaries.

For the partial differential equation methods, a set of partial differential equations must be solved to obtain a coordinate system. The partial differential equations may be elliptic, hyperbolic or parabolic. The methods based on elliptic partial differential equations are more general (since all boundaries can be specified), and more fully developed. Typically, a pair of Laplace equations is solved subject to the Cauchy-Riemann boundary conditions. The earliest successful development was formally reported by Winslow [44], who started with a Laplace system subjected to Dirichlet boundary conditions. Thompson, et al. [45] added periodic boundary condition to produce branch cuts for various topological configurations and suggested that control over the grid could be accomplished by altering the original Laplace system. The alteration is to consider a pair of Poisson equations by including specifications for the right-hand sides. These are called forcing terms

and are general functions of the curvilinear variables. The particular form to be used was established later by Thompson et al. [46]. Mastin and Thompson [47] have been able to show that the two-dimensional system without the forcing term analytically defined nonsingular transformation. Conformal mapping methods can also be included in the elliptic methods. Mehta and Lavan [48] have given a solution about a modified Joukowski airfoil accomplished by generating a coordinate system with a conformal Joukowski Transformation and solving the Navier-Stokes equations on the system. More examples of conformal mapping methods are given by Sampath [49], Wu et al.[50], and Napolitano et al. [51]. When only one physical boundary is specified, hyperbolic partial differential equations may be used to obtain a grid by spatial marching from the given boundary. The remaining boundaries are determined by the solution and are geometrically unimportant in cases such as the external flow about a single object. A fundamental development of the method has been given by Starius [52], and one which is well suited to body concavity has been presented by Stager and Sorenson [53]. The parabolic system can be applied to generate the grid between the two boundaries of a doubly-connected region with each of these boundaries specified [54-56]. The drawbacks of the hyperbolic scheme are: (1) the outer boundary can not be specified, (2) the scheme tends to propagate singularities of the boundary condition into the flow domain, and (3) the solution may become unstable unless an artificial viscosity term is adequately added to the equations. On the other hand, the major drawback of the parabolic scheme is that maintaining orthogonality of grid needs much effort. Nakamura and Suzuki [57] have combined these two schemes into a single scheme that takes advantages of the two but eliminates the drawbacks of each. They have illustrated the used of the scheme by generating grids around an airfoil, an automobile model, and buildings. Both hyperbolic and parabolic methods have the advantage of being generally faster than elliptic methods, but are applicable only to certain configurations.

It has been shown that the partial differential equation approach produces the smoothest grids for general boundary point distributions while the algebraic approach is the fastest procedure. Regardless of which approach is taken, creation of a computational grid requires (1) defining an accurate mathematical description of all solid surfaces in the computational domain and (2) generating an "appropriate" grid around these surfaces according to some criterion, usually with a specified point distribution. Graphic facility is very useful when three dimensions are involved. An important feature is the ability to rotate and translate grid surfaces in real time for inspection.

Generally, the generation of the surface grid is a separate procedure. For simple configurations, such as those with well defined mathematical description, this procedure can be combined with the generation of grid points interior to the domain. For complex configurations, however, it is very difficult to do so. This is because the description of such a configuration is provided usually as a set of data points. It may not be possible to describe them by any mathematical formulation. Also, coordinate points on the configuration are, in general, not coincident with these data points. Moreover, the number of coordinate points are normally different from that of the given data points. These coordinate points must, somehow, be generated, in conjunction with a given data set, in such a way that they provide the correct description of the configuration. One way of doing this is to generate a parametric surface according to a mathematical description and, then, project it on to the surface defined by the given data point [58]. Another method is to use a bicubic spline procedure [59]. The procedure has a patch/plane intersection capability that makes it possible to "slice" any curved surface and find intersections between various independently defined surfaces. In this study, the method described in [59] is used to construct the surface grid of a fighter aircraft configuration (Sec. 4.4). Once a satisfactory surface grid is defined, the transfinite interpolation is used to extend

this grid out into the flowfield. Some details of the transfinite interpolation are given in the next section.

2.2 Transfinite Interpolation

The idea of using interpolation as a means of constructing general curvilinear coordinate systems stems from the fact that in most cases, the coordinates or grid points are known on several or on all of the boundaries of the computational domain and the problem consists of extending this grid into the interior of the domain. Interpolation from the boundaries into the interior of the region can be accomplished by the so-called transfinite interpolation concept (sometimes referred to as the blending function method). Transfinite interpolation is a highly generalized algebraic grid generation method. Transfinite interpolation is applied through a series of univariate interpolations where blending functions and the associated parameters (point position and/or derivatives) determine a grid. The concept was originally developed by Coons [60] and subsequently extended by Gordon [61]. One of the earliest 2D grid generation applications using transfinite interpolation is described by Gordon and Hall [36]. A few examples of 3D applications are provided in the works of Gerhard [62], Anderson and Spradley [63], and Spradley et al. [64]. In these applications, the transfinite interpolation in its simplest form is used, i.e. with no control of the normal derivatives of the grid coordinates at the boundaries. Eriksson [65] and Eriksson and Rizzi [66] have constructed a scheme which allows for the specification of any number of normal derivatives of the grid coordinates on the boundaries. The precise control of the resulting coordinate system or grid has made it possible to generate grids of advanced type that are both smooth and efficient in terms of resolution for a given number of grid points.

Apart from providing a good grid control, the transfinite interpolation concept offers speed and simplicity when implemented on computers. The speed factor

is very important for 3D applications because the generation of a desirable grid for a given geometry is usually a process involving a series of grid generation runs with visual checks and adjustments in between. This fact is not always appreciated when evaluating the cost of grid generation. Naturally, a good graphics software package is an integral part of any 3D grid generation system.

The theory of transfinite interpolation is a very general concept of multivariate interpolation and is outlined here briefly. Let $\vec{f}(u, v, w) = [x(u, v, w), y(u, v, w), z(u, v, w)]$ denote a vector-valued function of three parameters u, v, w defined on the region $u_1 \leq u \leq u_p, v_1 \leq v \leq v_q, w_1 \leq w \leq w_r$. This function is known only on certain planes in the region, Fig. 2.3,

$$\vec{f}(u_k, v, w) = \vec{a}_k(v, w) \quad ; \quad k = 1, 2, \dots, p$$

$$\vec{f}(u, v_k, w) = \vec{b}_k(u, w) \quad ; \quad k = 1, 2, \dots, q$$

$$\vec{f}(u, v, w_k) = \vec{c}_k(u, v) \quad ; \quad k = 1, 2, \dots, r$$

A set of univariate blending functions

$$\alpha_k(u) \quad ; \quad k = 1, 2, \dots, p$$

$$\beta_k(v) \quad ; \quad k = 1, 2, \dots, q$$

$$\gamma_k(w) \quad ; \quad k = 1, 2, \dots, r$$

which satisfy the conditions

$$\alpha_k(u_l) = \delta_{kl} \quad ; \quad \beta_k(v_l) = \delta_{kl} \quad ; \quad \gamma_k(w_l) = \delta_{kl}$$

where

$$\delta_{kl} = 0 \quad ; \quad k \neq l \quad \delta_{kl} = 1 \quad ; \quad k = l$$

is needed to interpolate between these given planes.

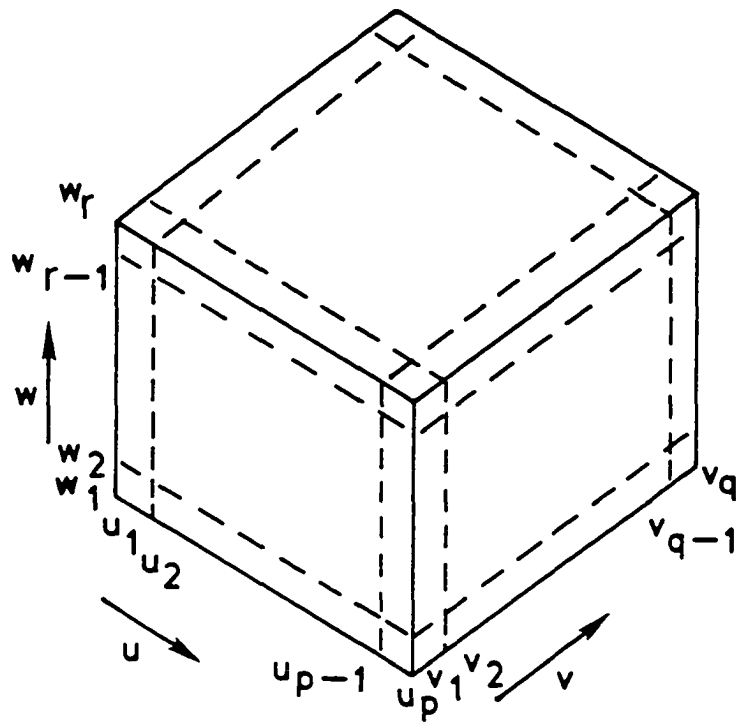


Fig. 2.3 A computational domain where \bar{f} is known on certain planes.

The transfinite interpolation procedure then gives the interpolated function $\vec{f}(u, v, w)$ by the recursive algorithm

$$\begin{aligned} \vec{f}_1(u, v, w) &= \sum_{k=1}^p \alpha_k(u) \vec{a}_k(v, w) \\ \vec{f}_2(u, v, w) &= \vec{f}_1(u, v, w) + \sum_{k=1}^q \beta_k(v) \cdot [\vec{b}_k(u, w) - \vec{f}_1(u, v_k, w)] \\ \vec{f}(u, v, w) &= \vec{f}_2(u, v, w) + \sum_{k=1}^r \gamma_k(w) \cdot [\vec{c}_k(u, v) - \vec{f}_2(u, v, w_k)] \end{aligned} \quad (2.1)$$

The function \vec{f} now defines a transformation from the region $u_1 \leq u \leq u_p, v_1 \leq v \leq v_q, w_1 \leq w \leq w_r$ in u, v, w space to some arbitrarily shaped region in the x, y, z space. It can be verified that if the specification of \vec{f} on the planes $u = u_1, \dots, u_p; v = v_1, \dots, v_q$; and $w = w_1, \dots, w_r$ is continuous at the intersections of these planes, the explicit order of the interpolation directions chosen in the three-step algorithm does not affect the interpolant.

The interpolation procedure just described can give any degree of control if a sufficient number of internal surfaces are specified, but the control is generally poor if no internal surface is defined at all. In order to improve the control while maintaining the minimum input geometry data, a generalized transfinite interpolation procedure \vec{f} which uses derivatives of the function \vec{f} in the out-of-surface direction, in addition to the function itself, can be defined. The effect of specifying out-of-surface derivatives of \vec{f} (Fig. 2.4) is to introduce a direct control of the essential properties of the mapping function in the vicinity of the surface. The specified data are written as

$$\begin{aligned} \frac{\partial^n}{\partial u^n} \vec{f}(u_k, v, w) &= \vec{a}_k^{(n)}(v, w) \quad ; \quad \begin{matrix} k=1,2 \\ n=0,1,2,\dots,p_k \end{matrix} \\ \frac{\partial^n}{\partial v^n} \vec{f}(u, v_k, w) &= \vec{b}_k^{(n)}(u, w) \quad ; \quad \begin{matrix} k=1,2 \\ n=0,1,2,\dots,q_k \end{matrix} \\ \frac{\partial^n}{\partial w^n} \vec{f}(u, v, w_k) &= \vec{c}_k^{(n)}(u, v) \quad ; \quad \begin{matrix} k=1,2 \\ n=0,1,2,\dots,r_k \end{matrix} \end{aligned}$$

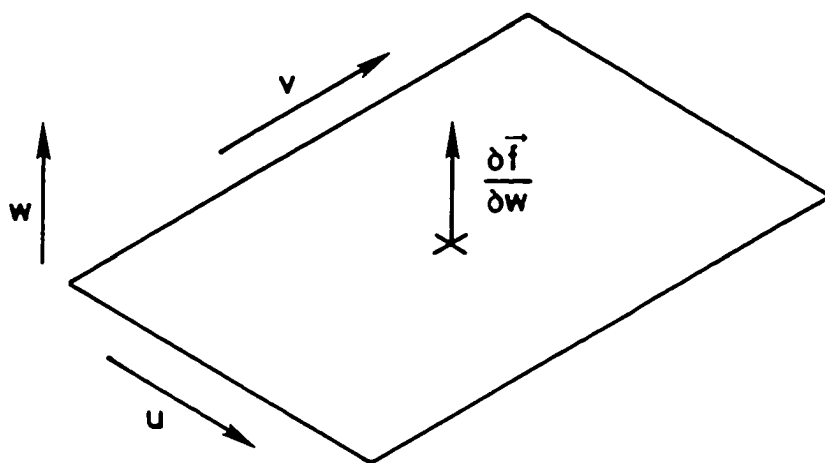


Fig. 2.4 An out-of-surface derivative of \bar{f} .

which is simply the specification of \vec{f} and a finite number of out-of-surface derivatives of \vec{f} on the outer surfaces of the region $u_1 \leq u \leq u_2, v_1 \leq v \leq v_2, w_1 \leq w \leq w_2$ in u, v, w space (Fig. 2.5). To interpolate \vec{f} into the interior of this parametric box, a new set of univariate blending functions is defined as

$$\alpha_k^{(n)}(u) \quad ; \quad k = 1, 2 \quad n = 1, 2, \dots, p_k$$

$$\beta_k^{(n)}(v) \quad ; \quad k = 1, 2 \quad n = 1, 2, \dots, q_k$$

$$\gamma_k^{(n)}(w) \quad ; \quad k = 1, 2 \quad n = 1, 2, \dots, r_k$$

which have to satisfy the conditions

$$\frac{\partial^m}{\partial u^m} \alpha_k^{(n)}(u_i) = \delta_{ki} \delta_{mn}$$

$$\frac{\partial^m}{\partial v^m} \beta_k^{(n)}(v_i) = \delta_{ki} \delta_{mn}$$

$$\frac{\partial^m}{\partial w^m} \gamma_k^{(n)}(w_i) = \delta_{ki} \delta_{mn}$$

The generalized transfinite interpolation algorithm is then written as

$$\begin{aligned} \vec{f}_1(u, v, w) &= \sum_{k=1}^2 \sum_{n=0}^{p_k} \alpha_k^{(n)}(u) \vec{a}_k^{(n)}(v, w) \\ \vec{f}_2(u, v, w) &= \vec{f}_1(u, v, w) + \sum_{k=1}^2 \sum_{n=0}^{q_k} \beta_k^{(n)}(v) \cdot [\vec{b}_k^{(n)}(u, w) - \frac{\partial^n}{\partial v^n} \vec{f}_1(u, v, w)] \\ \vec{f}(u, v, w) &= \vec{f}_2(u, v, w) + \sum_{k=1}^2 \sum_{n=0}^{r_k} \gamma_k^{(n)}(w) \cdot [\vec{c}_k^{(n)}(u, v) - \frac{\partial^n}{\partial w^n} \vec{f}_1(u, v, w)] \quad (2.2) \end{aligned}$$

The function \vec{f} now defines a transformation from the region $u_1 \leq u \leq u_2, v_1 \leq v \leq v_2, w_1 \leq w \leq w_2$ in u, v, w space to some arbitrarily shaped region of x, y, z space. The algorithm given by Eq.(2.2) is also referred to as the ‘‘osculatory’’ transfinite interpolation scheme.

Generally speaking, the method of transfinite interpolation, is a very simple and straight forward concept that offers virtually unlimited possibilities; but for any particular application, it is necessary to supply a certain amount of geometric

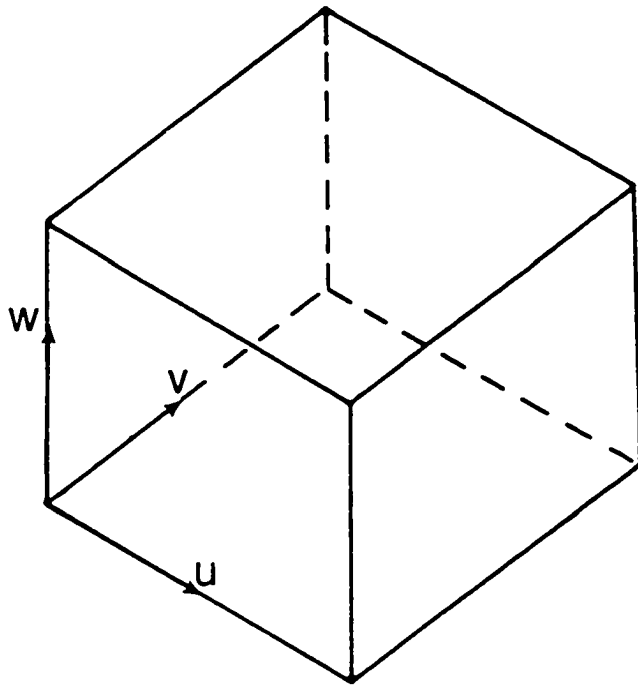


Fig. 2.5 A computational domain where $\frac{\partial \vec{f}}{\partial n}$ are specified on the outer surface.

data to obtain a certain degree of control of the transformation. It is up to the user to balance the requirements of the minimum input geometric data and maximum control.

2.3 Mapping Type and Singularities

It is clear from previous sections that the first stage of a grid generation procedure is the specification of grid coordinate data on the boundaries. Thus, the correspondence between the boundaries in the physical domain and the computational domain has to be made clear. It is, then, necessary to determine the overall structure of the mapping between these domains. For a given geometry, there are generally several possible mapping types with different characteristics in terms of efficiency, coordinate cuts, singularities, etc. For example, there are at least six natural combination of mapping types for the exterior region of a typical airfoil (Fig. 2.6). All of these alternative mapping types give boundary-fitted coordinates but varies markedly in terms of grid efficiency, i.e. the resolution per grid point. It has been shown that the mapping type designated O-O is the most efficient for such a configuration [67]. The notation O-O is to be interpreted as “type O in the chordwise direction, type O in the spanwise direction”, using the 2D notation shown in Fig. 2.6. Figure 2.7 illustrates the O-O mapping type of a wing-fuselage configuration. As shown in the figure, the entire wing is mapped to the bottom of the computational box, the entire outer boundary is mapped to the top and the combined plane of symmetry and fuselage is mapped to one of the side surfaces. The remaining three surfaces of the computational box constitute coordinate cuts, i.e. they correspond to interior surfaces in the physical domain across which the various flow properties are continuous.

Figure 2.8 shows that the O-O mapping type gives rise to two singular lines extending from the two tip corners of the wing to the outer boundary. A

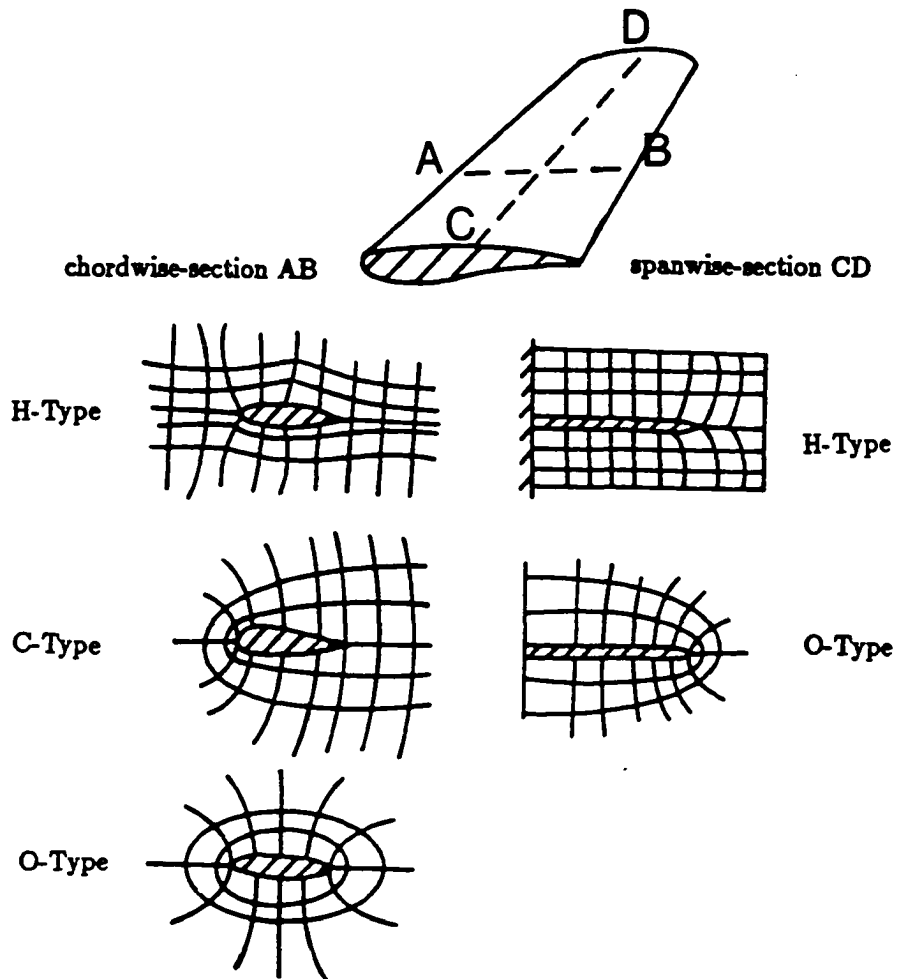


Fig. 2.6 Mapping types for a 3D airfoil.

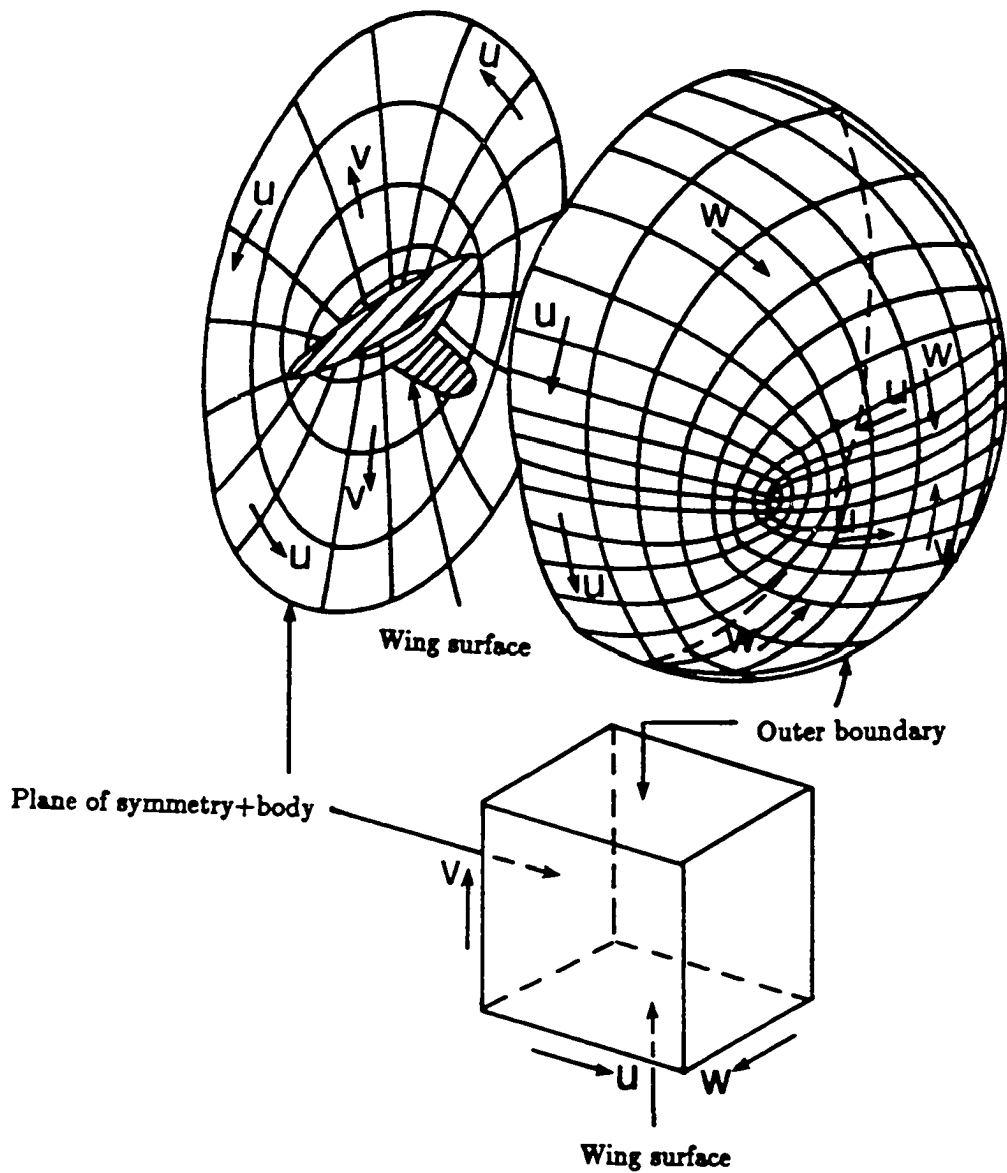


Fig. 2.7 An O-O mapping for a wing-fuselage configuration.

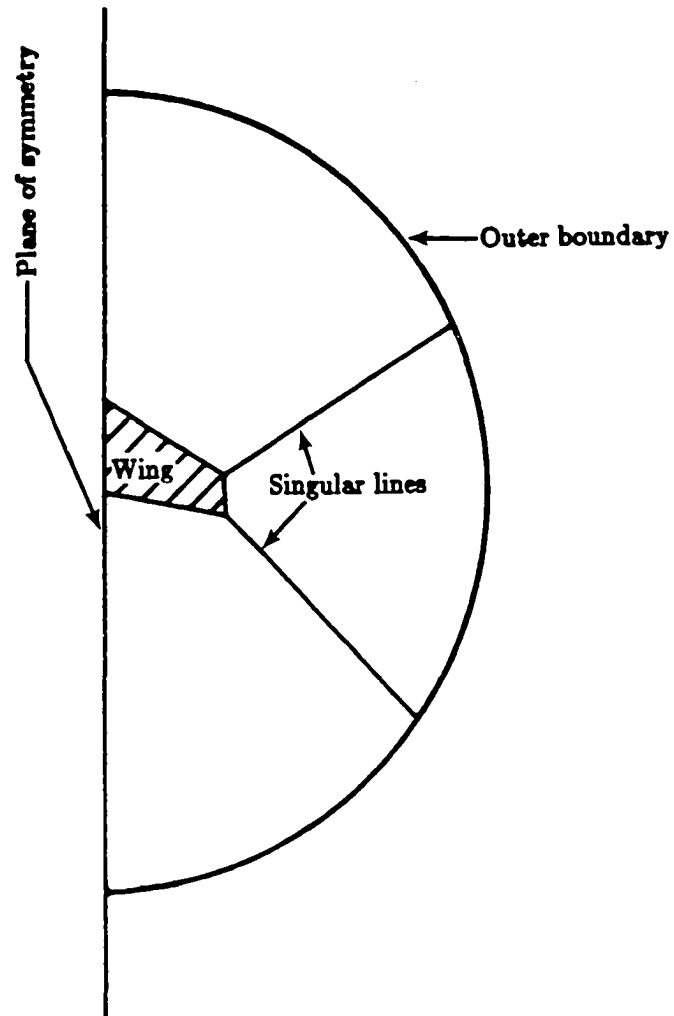


Fig. 2.8 Singularity lines for the O-O mapping.

grid singularity is defined as a place in the physical domain where the Jacobian of transformation is zero or unbounded (depend upon how the Jacobian is defined). The grid singularities are undesirable but very often unavoidable and any practical finite scheme must be able to cope with them. If the physical information near a singularity is not of primary interest, the finite difference solutions can be obtained in this region provided the singular points themselves are excluded. Another practical approach to dealing with singularities is to leave the boundary surfaces open (Fig. 2.9). However, this requires that assumptions be made about the physics that must be included in the solution procedure. This study follows the so-called finite-volume method, which is a conservative cell-oriented method, and can be shown to be stable regardless of the type of singularity involved. The discussion on the finite volume approach is given in Sec. 3.2.

Since singularities always associate with the mapping types, and some types of singularities are more severe than the others, it is important to seek the best type of mapping for a given geometry, both from the viewpoint of efficiency and accuracy. For example, the C-H mapping has been the most popular type for flow computations around wings, even though this mapping has the more severe kind of singular line along the wing tip. Also, it has been shown [67] that this type of mapping is not as efficient as other types of mapping (for example O-O type). The reason that the C-H mapping is popular is because it can be obtained by a simple "stacking" of 2D chordwise grids (C-types) in the spanwise direction, i.e. by a "quasi-3D" method. From the discussion in this section, it may seem that the price to be paid for using such a simple grid generation technique is high.

2.4 Multiple Grids

The discussion so far have been limited to the topic of a single grid, i.e., the grid that maps the physical domain onto a "slab" in the computational

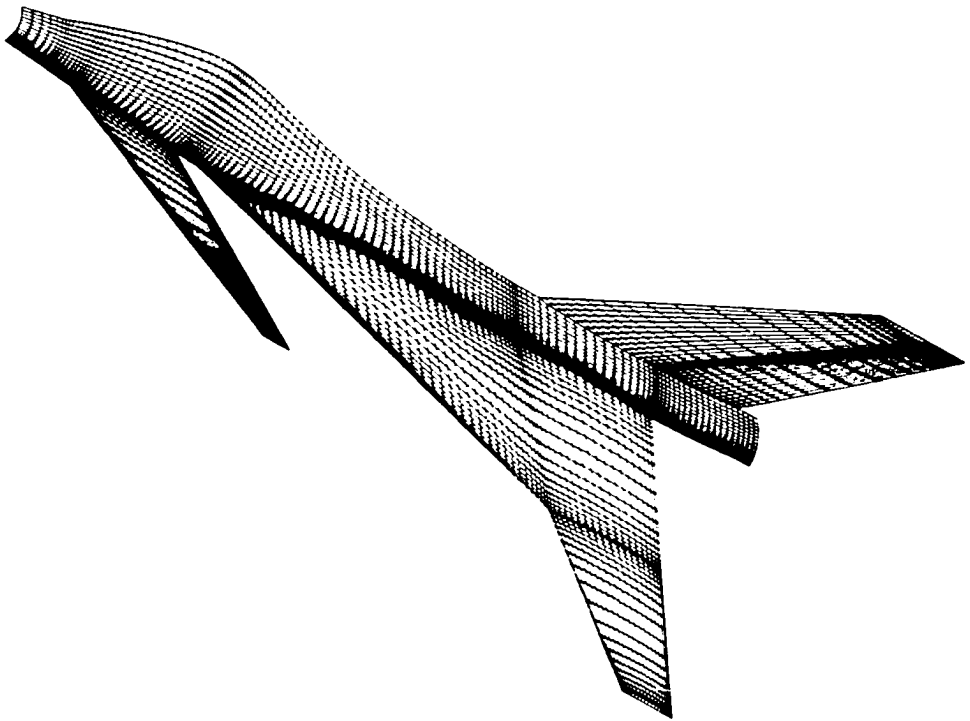


Fig. 2.9 The boundary surface is opened to avoid dealing with singularities.

domain. This type of mapping is very desirable due to its simplicity. However, for very complicated geometries it can be difficult to generate a single grid that is both reasonably smooth and efficient. An example of such a complex region is the exterior of a complete airplane with several lifting surfaces. Each component of an aircraft, in general, requires a grid system that is usually incompatible with the grid systems of the other components. Thus, the generation of a single boundary-fitted grid for the entire configuration is a difficult task, if it is possible at all. In such a global grid, control of grid point distribution, skewness and clustering will be difficult to achieve. For example, a grid which provides sufficient resolution of grid points in a region may result in an excessive number of grid points in other regions. Convergence of the solutions may not be achieved if the number of grid points is excessive. To simplify this problem, it is becoming more common to use several grids at once, each in a different coordinate system [68]. An example of this procedure is illustrated in Fig. 2.10. This approach, called "multiple grids" or "zonal grids" approach (the terms "zone" or "block" is also used interchangeably), falls into two categories: grid patching and grid overlapping (Fig.2.11). The approach subdivides a complicated domain into several subdomains which can accommodate easily generated grids. For the patched grid approach, the global grid is formed by patching together all the individual grids. The computed grid lines in adjacent grids may be made to align at the grid interface with complete continuity [69,70], or with continuous lines slope [71], or discontinuity in slope [72], or perhaps not align each other at all [73]. Rubbert and Lee [72] combine the subdomain grids in such a manner that the resultant global grid is continuous across patch boundaries. However, grid irregularities frequently occur at the corners of the subdomain and at surface perimeter lines. Such irregularities impose constraints upon the choice of the numerical algorithm used for solutions of the flow equations. Lasinski et al. [74] have demonstrated a patched grid technique for solution of the thin layer Navier-Stokes

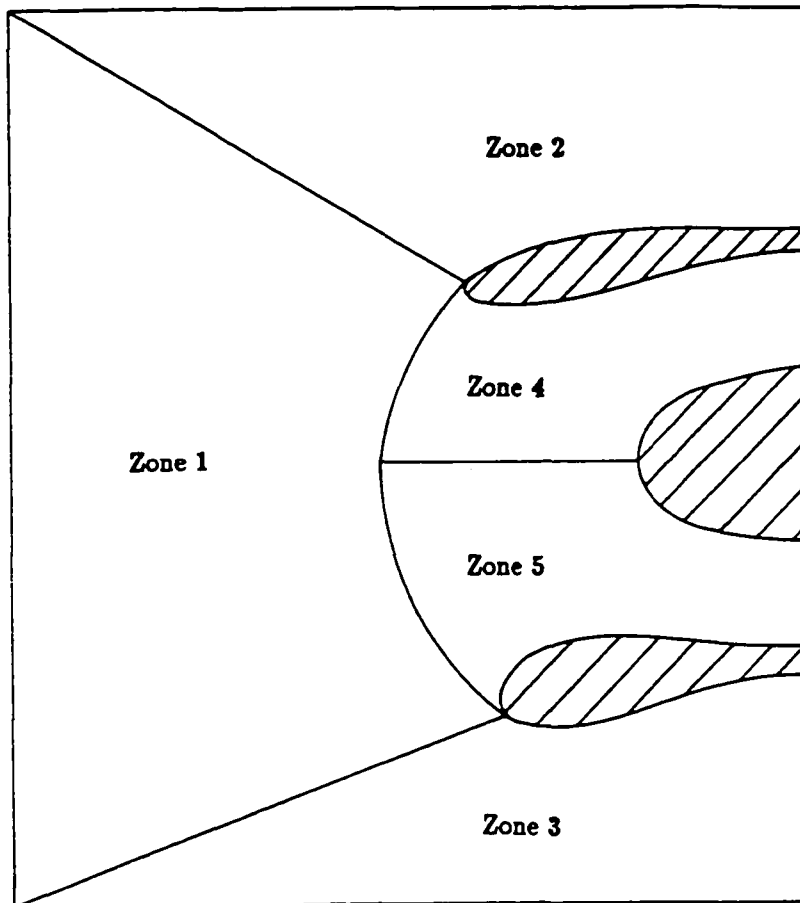


Fig. 2.10 Zoning of multiple-connected region.

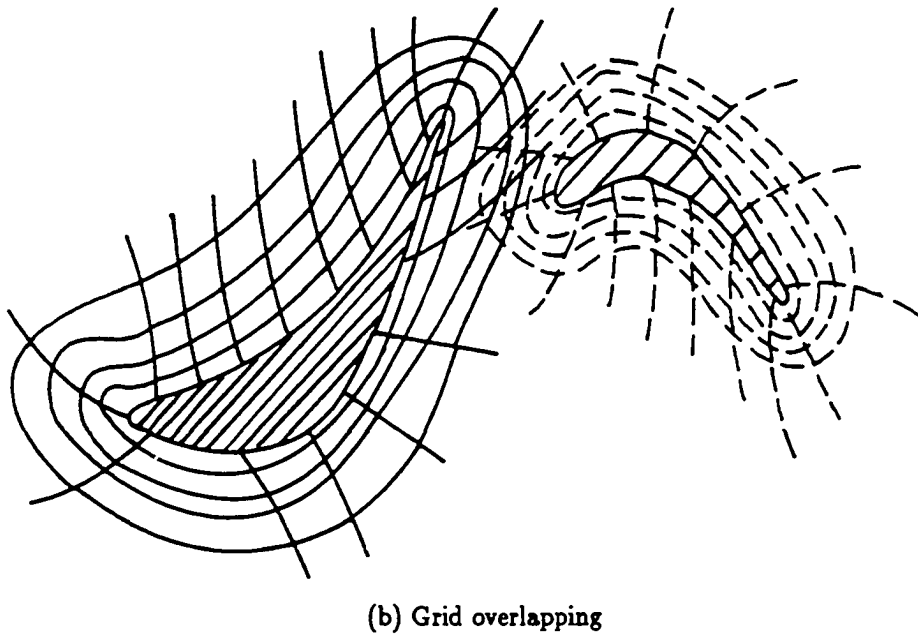
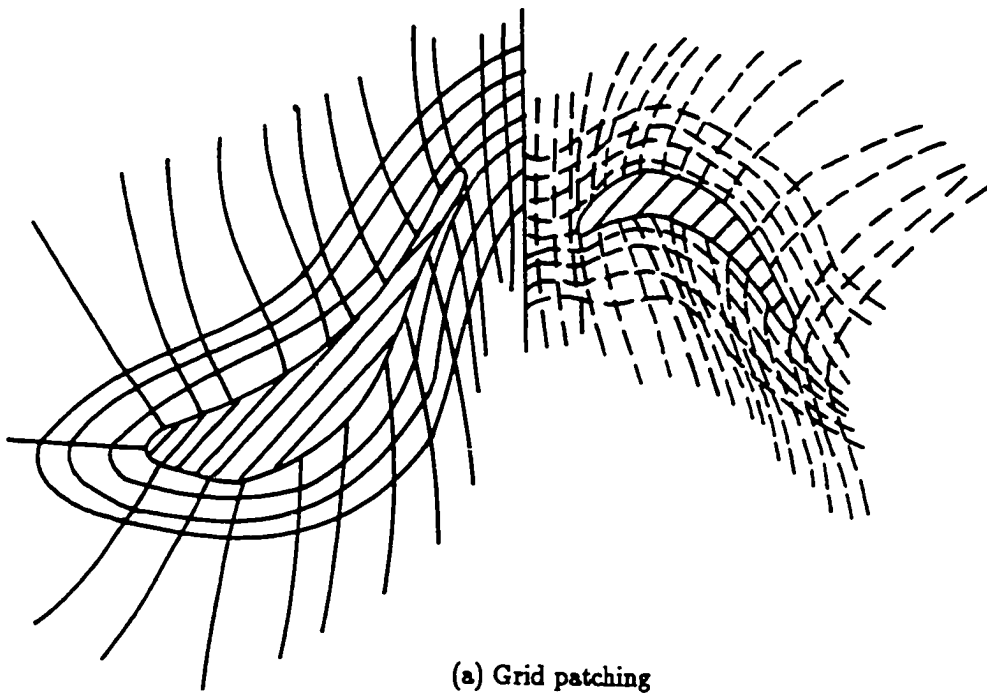


Fig. 2.11 Grid patching versus grid overlapping.

equation. They solve the flow equations on each grid separately. The solutions are coupled by the transfer of boundary data at the coincident boundary points between grids. References 75-78 also illustrate the use of patched grid approach. The grid overlapping approach does not need common boundary between grids, but rather, the various subdomain grids are only required to overlap to provide communication among grids for flow solvers. The development and analysis of solution procedures on grid overlapping approach have been studied by Starius [79,80], Kreiss [81], and Mastin and McConnaughey [82]. The practical application of overlapping grids to the solution of problems in computational fluid dynamics has been demonstrated by Atta [68], Thompson [83], Steger and Buning [84], and Benek et al. [85]. Steger et al. [86,87] have applied the grid overlapping technique to an airfoil/flap in incompressible flow [86] and in subsonic compressible flow [87]. Atta and Vadyak [88] have obtained a potential solution for a wing/nacelle geometry. These studies have demonstrated that the technique can be applied to subsonic flows. However, for the transonic flight regime Benek et al. [87] have found that their single trial solution resulted in an ill-defined shock wave at the grid boundaries and exhibited poor convergence. The studies by Dougherty [89] indicate that for a different grid geometry and algorithm, these problems may not be too severe. Early efforts to predict multiple-component configurations are based on the transonic small disturbance formulation [91-92]. Efforts to predict the flow field about a complete aircraft configuration using a single grid approach have been made by Yu [93]. However, the requirement of exact boundary-fitted grids along certain boundary lines is relaxed. Thus, the exact implementation of boundary conditions is not obtained.

The multiple grids approach has a number of advantages. First, the difficulty in generating three-dimensional grids for different types of complex configurations can be eliminated. Second, the approach allows different types of grid topologies to be implemented in each subdomain in order for grids to be mesh-efficient,

i.e., more grid points near a solid body or shock and less grid points elsewhere. Since it is well known that skewness, rapid volume variation, and large cell aspect ratios degrade the convergence rate of an algorithm, it seems plausible that the enhanced grid point control afforded by the multiple grids approach will also result in an improved algorithm performance. Third, it may also be computationally efficient to solve different equation sets in the various subdomain grids, such as viscous Navier-Stokes equations near the body and inviscid potential equation in the outer field. Chanderjian and Steger [94] have demonstrated this idea by solving the Euler equations in one zone and the dual potential equations in the other for the transonic flow over a lifting airfoil. Finally, computer core memory required by the approach is less than that required if a single grid is used. Thus, the memory limitation on a particular computer can be overcome. This advantage may not be so great in the future, with the development of the supercomputers.

A common difficulty with the multiple grids approach is the construction of a proper scheme for information exchange among the different subdomain grids. The information exchange has to be not only consistent with the governing equations, but should also lead to a stable efficient scheme. These "interface conditions" are required to guarantee the convergence to a weak solution of the governing equations if the algorithm converges. The multiple grids approach results in new boundaries within each subdomain grid, i.e., at the interfaces of various grids. Since these boundaries are not the physical boundaries, it is important to treat grid points on the interfaces with care in order to transfer information from one grid to another accurately. The most obvious procedure is to interpolate the solutions in one grid to provide necessary boundary data for another. Since the classical interpolation formulas were not derived with conservation properties in mind, their use in finite-difference approximation on multiple grids would result in the loss of an exact conservation property. Eberhardt and Banganoff [95] have shown that shock

waves crossing overlapped grid boundaries can become ill defined and convergence is generally degraded when the interpolation procedure is used. They have shown also that the characteristic approach is superior but suggested that the use of conservative properties would be most desirable. For the existence of solutions to certain system of partial differential equations, some conservation laws must be satisfied accurately. The nonlinear nature of the equations of motion permits solutions with discontinuities such as shock and slip surfaces. In order that such discontinuities assume the right strength and physical location, it is imperative that the scheme used for the calculation be conservative [96]. In a multiple grid calculation, it is important that the interfaces are also treated in a conservative manner so that the discontinuities can move freely across the interfaces [97]. The need for conservative grid interfaces is also illustrated in [87].

The question of conservation when switching between two different grids or numerical schemes has been considered by several authors. Warming and Beam [98] have derived transition operators for switching conservatively between McCormack's method and a second order upwind scheme. Hennesius and Pulliam [99] have applied this transition operator approach to derive the so-called zonal interface conditions; this however, resulted in a significant loss of accuracy at the zonal interfaces. Rai [100] has developed conservative zonal interface conditions for multiple grids which share a common grid line, and has provided accurate calculations demonstrating the shock capturing ability of the multiple grids with a discontinuity crossing grids. Cambier et al. [101] have analyzed the zonal-boundary problem for a system of hyperbolic equations and used the compatibility equations to develop a zonal-boundary scheme. Reasonably good results were obtained for transonic channel flow. However, the use of the compatibility equations results in a scheme that is not conservative and, hence, unsuitable for problems in which flow discontinuity move from one grid to another. Rai et al. [102] have presented results obtained

metric discontinuous grids; the integration scheme used is the Osher upwind scheme. Reference 86 provides the results obtained on overlapping grids in conjunction with the stream function approach.

In the patched grid approach, conservation can be easily maintained at the patched interfaces. The extra computing time that is required to implement the zonal boundary condition is less than what is required for overlapping grids. This is because the necessary interpolations, that affect transfer between grids, are performed in a reduced number of spatial dimensions for patched grids. A problem in three dimensions only requires a two-dimensional interpolation procedure. This reduction in the number of dimensions in which the interpolation is performed does not occur for overlapping grids. On the other hand, overlapping grids provide more flexibility in generating grids because there are fewer constraints on the choice of outer boundaries for the different grids. Other disadvantages of grid overlapping approach, beside that of interpolation, are: (i) it is difficult to maintain global conservation and (ii) the accuracy and convergence speed of the calculation seems to depend on the degree of overlapping of the grids and the relative size of each zone, thus introducing a certain amount of undesirable empiricism in the formulation. This study follows the grid patching approach in which the interfaces between subdomain grids are patched as plane interfaces. It can be shown that global conservation can be easily maintained for this type of interface. The study follows the method for transferring a conserved quantity from one generalized grid to another which was first described by Dukowicz [103]. Ramshaw [104] has suggested a procedure for doing so which is similar to the method of Dukowicz, but is simpler and more direct. A computer program following the Ramshaw's procedure has been written and tested with various types of grids and variables. The program has been working well for simple test cases. The objective of this study is to establish

whether or not the technique is feasible for applications to realistic aerodynamic configurations.

The grid generation procedure of multiple grids does not in principle differ from the generation of a single grid. The complete grid is computed by first dividing the entire domain into several subdomain grids and then "filling in" one subdomain grid after the other by transfinite interpolation. Eriksson [105] has obtained good solutions for the inviscid flow around an airplane by applying this concept. There, slope continuity (C^1 continuity) between subdomain grids is obtained by using osculatory interpolation, i.e., by using derivative information as well as grid point locations in the interpolation. The approach used in this study is different from that discussed in [105]. Although the surface must be common between two subdomain grids, there is no restriction on grid slope or density across interfaces. This offers a great flexibility to the generation of each subdomain grid. The details on the treatment of the conditions at the interface are given in Sec. 3.4.4.

2.5 Brief Discussion on Conservative Rezoning Algorithm

A method for transferring a conserved quantity from one generalized mesh to another, when the volumetric density of the quantity is assumed to be uniform within each grid cell of the original mesh, is described briefly. This method was first described in [103]. Reference [104] suggested the procedure in doing so, which is similar in spirit but simpler and more direct. A computer program following this procedure has been written and is working well for example grids and a wide variety of choice of variables. The concept works equally well for any type of grid. However, only the arbitrary quadrilateral grid is demonstrated here. This is because it is the most common type of grids in practice and is convenient to work with since it has the same simple topological and logical structure as a square or rectangular grid.

The basic concept of the algorithm is simple. Consider Fig. 2.12 where two grid surfaces are patched with each other in some fashion. The conserved quantities $Q_{O_{ij}}$ of the original grid surface ($A_{O_{ij}}$ is the area of each surface mesh) is to be transferred to another grid surface in which $A_{N_{ij}}$ is the area of each surface grid. $Q_{N_{ij}}$ is denoted as the transferred quantity in each of these latter surface grid. The quantity $Q_{N_{ij}}$ can be computed by

$$Q_{N_{ij}} = \sum_{n=1}^{N_{NO}} (Q_{O_{kl}}) \frac{A_{NO_{kl}}}{A_{O_{kl}}} = \sum_{n=1}^{N_{NO}} (q_{O_{kl}}) (A_{NO_{kl}}) \quad (2.3)$$

where $A_{NO_{kl}}$ is the portion of the area $A_{N_{ij}}$ which is contained in the area $A_{O_{kl}}$ and N_{NO} is the number of the original surface grids contained in $A_{N_{ij}}$. The quantity $q_{O_{kl}}$ is given by $q_{O_{kl}} = \frac{Q_{O_{kl}}}{A_{O_{kl}}}$, where the volumetric density of $Q_{O_{kl}}$ is assumed to be constant. From Fig. 2.12, it can be seen that the overlapped areas, $A_{NO_{kl}}$ are polygons p whose sides are segments of both the old grid lines and the new grid lines. The number of sides of each type, and total number of sides, will be different for different overlapped areas. Each side of a polygon is common to two overlapped areas, the one on the left (L) and the one on the right [R], and these overlapped areas may be considered to be associated with the side. The objective is to apportion a conserved quantity Q , whose volumetric density q is considered uniform within each cell of the old grid, into the cells of the new grid. The task now, is to find $A_{NO_{kl}}$, the number of original surface grids contained in $A_{N_{ij}}$, and to associate them with the quantities $Q_{N_{ij}}$ and $Q_{O_{kl}}$. The area of the polygon in 2D plane is given by [106]

$$A_p = \frac{1}{2} \sum_{s=1}^{N_p} \epsilon_s^p (x_1^s y_2^s - x_2^s y_1^s) \quad (2.4)$$

where the summation is over all the sides of p , and ϵ_s^p is either +1 or -1 according to p lies to the left or right, respectively, of side s . The endpoint coordinates (x_1^s, y_1^s) and (x_2^s, y_2^s) are considered to be associated with the side s and not with the particular polygon. It would be inefficient and difficult to automate in a computer and naively

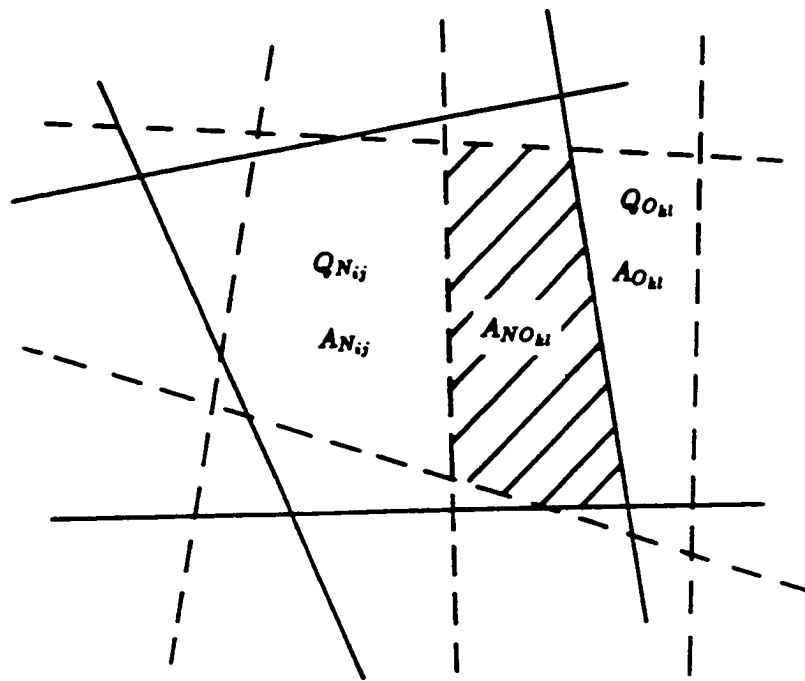


Fig. 2.12 Patched surfaces.

compute the quantities $Q_{N,i}$ directly at a time. This is because the number of sides composing each polygon varies generally. Moreover the parameter N_{NO} for different $Q_{N,i}$ are different. Instead, Ramshaw [104] suggests to evaluate the same contributions by sweeping over the sides or segments s . The side or segment s is any side or segment of the polygon (overlapped area). The coordinate of the two end points of side s are denoted by (x_1^s, y_1^s) and (x_2^s, y_2^s) . This can be done by sweeping over grid lines since the segment s is a part of either old or new grid lines.

If the side s is a segment of the old grid then the quantity Q in the new grid cell containing side s is to be incremented by an amount

$$\Delta_s^O = \frac{1}{2}(q_L - q_R)(x_1^s y_2^s - x_2^s y_1^s) \quad (2.5)$$

If the side s is a segment of the new grid then the contribution to cell on the left is

$$\Delta_s^N = \frac{1}{2}q_O(x_1^s y_2^s - x_2^s y_1^s) \quad (2.6)$$

while the contribution to the cell on its right is just $-\Delta_s^N$ where q_O is the volumetric density of the old grid cell in which side s lies.

Adding Δ_s^O and Δ_s^N for each of new grid cell yields the quantity Q contained in each of the new grid cell. The details on the implementation of the algorithm including the verification of these formulas are given in Appendix A.

2.6 Application to The Hyperbolic Equations

The first step toward the application of the technique to the CFD calculation is to apply the technique to solve some partial differential equations. The hyperbolic equations have been chosen not only because of their simplicity but also because of their hyperbolic nature which is similar to the equations of motion in supersonic flow. The hyperbolic equations can be written in two dimensional space as

$$q_t + a q_x + b q_y = 0 \quad (2.7)$$

and in three dimensional space as

$$q_t + aq_x + bq_y + cq_z = 0 \quad (2.8)$$

where a, b and c have been treated as constances. If the initial conditions are given as $q = f(x, y)$ and $q = f(x, y, z)$, the exact solutions can be found as

$$\begin{aligned} q(x, y, t) &= f(x - at, y - bt) \quad ; \\ q(x, y, z, t) &= f(x - at, y - bt, z - ct) \end{aligned}$$

for the two and three dimensional space, respectively. In two dimensional case, the equation, along with the appropriate boundary conditions, has been solved on a two dimensional grid system which is changed into another grid system at some time. The procedure described in the previous section has been used to transfer the flux (in this case q itself is the “flux”) from one grid to another. The three dimensional equation is more suitable as the model equation of the equations of motion. In this case, the entire domain is divided into two subdomains which are independent from each other. Equation (2.8) along with the appropriate boundary conditions are solved in each subdomain separately. However, some information have to be transferred across the interface between the two subdomains in order that the entire computation is consistent. Again the technique described previously is used to transfer flux (which is aq , bq , or cq depending upon how the interface is oriented) across the interface. It should be mentioned that, in both cases, Eqs. (2.7) and (2.8) are discretized by mean of the centered finite-volume approach. The three-stage Runge-Kutta integration scheme is also used to integrate both equations in time. Results have been compared with the solutions from single grid calculations. Satisfactory results have been obtained. The details of this study can be found in [107].

Chapter 3

BASIC FORMULATION

The ultimate equations to be solved in most CFD studies are the viscous Navier-Stokes equations. However, since solving these equations on modern day computers is still quite time consuming, they are often reduced to a simpler form. Solutions to these simpler equations, namely, stream function formulation [108], full potential equations [109-112], and Euler equations [113-118], have been obtained. The stream function formulation retains the generality contained in the full Euler equations. However, it is limited to two-dimensional or axisymmetric flows, and is made difficult by the fact that the density in the transonic regime is a double-valued function of the unknown stream function. The full potential equation has been used as a standard model and has proved to be a helpful tool in the design of aircraft. As with the stream function, the full potential equation can be solved by efficient relaxation techniques, and requires storage of only a single variable. Furthermore, it permits the solution of three-dimensional as well as two-dimensional flows. The primary disadvantages are the limitation to isentropic and irrotational flows. The isentropic assumption implies that shock waves captured in the transonic regime must be limited in Mach number to a value less than 1.3. The irrotationality conditions requires a uniform incoming flow in two-dimensional situations, and a free vortex condition in three-dimensional flows. The full potential equation will admit the existence of discontinuities in the flow field. However, these discontinuities are isentropic shocks, which do not represent true physical shock waves because they

do not satisfy the Rankine-Hugoniot conditions. These shocks will be approximately of the proper strength and will exist in the proper portion in the flow field if the Mach number of the flow approaching the shock is less than or equal to 1.3.

In this study, the Euler equations are used as the model equations. Methods based on the Euler model are useful tools in CFD since they offer more realism than potential methods and yet are simpler and more economical than methods based on the Navier-Stokes equations. A number of efficient and reliable numerical schemes have been developed for the Euler equations [113-118]. Even though viscous terms are neglected, certain solutions of the Euler equations agree well with the experimental results. Shock waves captured in this model agree with the Rankine-Hugoniot relations regardless of their strength. More importantly, the vortex sheets and vorticity can also be captured as weak and genuine solutions. The applications of numerical methods to solve the Euler equations range from the study of flow field around military aircrafts and missiles where shock waves are strong, to more complex non-uniform shear flows past wings. The details regarding the Euler equations are given in Sec. 3.1.

The solution procedure for the Euler equations used herein is based on a center finite-volume scheme with explicit Runge-Kutta time stepping [119]. This type of scheme was first used by Jameson et al. [120], but the present scheme differs significantly from the original scheme, mainly in the definition of the damping terms and the farfield boundary conditions. It has been extensively tested in both two and three space dimensions, for three different Euler models (the full equations, the constant-stagnation enthalpy model, and the artificial compressibility model for incompressible flow) and for both aerodynamics and turbomachinery applications [121-124]. The finite-volume scheme is described in Sec 3.2.

In most instances the solution to the first order steady state equations is desired. The steady state Euler equations change their character depending upon

the local Mach number. In a totally supersonic flow some very efficient methods exist for their solution. The method of characteristics and a simple marching procedure are two common approaches. In subsonic domain, however, no generally accepted method has yet been devised for solving this system. One approach used for subsonic or transonic flows is to reintroduce the time derivative terms to the equations. The resultant set of equations is everywhere hyperbolic. A steady state solution can be obtained by marching in time from some initial guessed flow field until an asymptotic steady state is achieved. However, the initial conditions give rise to perturbation waves which move through the field as the solution progresses in time. The Euler equations have no inherent dissipation and, therefore, these waves must either be radiated from open boundaries or absorbed by the addition of artificial damping terms. The second and fourth order damping terms are added to the Euler equations. The fourth order terms are global and linear whereas the pressure-controlled second order terms are non-linear and are only activated around shocks. Boundary conditions are mainly of four types: solid wall conditions, interface conditions, inflow/outflow (farfield) conditions and coordinate cuts. Sections 3.3 and 3.4 describe the damping terms and numerical implementation of boundary conditions, respectively, in detail. The explicit three-stage Runge-Kutta integration scheme is also addressed in section 3.5. Generally, to reach a steady state, solution requires a large number of iterations and a long computational time [125]. Since only steady state solutions are desired, and true time accuracy is of no concern, the concept of local time stepping is used to accelerate the convergence to steady state solutions. This concept is introduced in Sec. 3.6.

3.1 Governing Equations

The Euler equations describing three-dimensional, unsteady and compressible flows in conservation form can either be written in the differential form

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = 0 \quad (3.1)$$

where

$$\vec{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}$$

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

or in the integral form

$$\frac{d}{dt} \int_{\Omega} \vec{q} dx dy dz + \oint_{\partial\Omega} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = 0 \quad (3.2)$$

where

Ω = arbitrary finite region.

The perfect gas equation of state is used to define the mean pressure P via the internal energy e :

$$p = (\gamma - 1)\rho e$$

where $\gamma = \frac{c_p}{c_v}$ = specific heat ratio.

An assumption has been made, in writing Eqs. (3.1) and (3.2), that the fluid is not influenced by external body forces. It can be shown that the system of conservation laws given by Eqs. (3.1) or (3.2) is hyperbolic [15]. Thus, Eqs.(3.1) or (3.2) can be integrated in time in order to achieve a steady state solution (if such a solution exists). Equation (3.1) can be obtained by dividing Eq. (3.2) by Ω and then

shrinking it to a point. This leads to the system of the differential conservation laws valid at that point if the partial derivatives are continuous there. The integral approach may be important for the correct capturing of discontinuities in the flow since it formally does not exclude discontinuities from the interior of Ω . This study follows the integral approach in which the difference equation are written directly from the integral system. Therefore, the method is a cell concept rather than a grid-point concept. The discussion on the method is given in Sec. 3.2.

The nonlinear character of the Euler equations generally permits solutions with discontinuities (shocks) where the differential Eq. (3.1) is no longer valid. The equivalence between Eqs. (3.1) and (3.2) is restored by allowing weak solutions to Eq. (3.1). However, both equations can give rise to nonphysical shocks unless an entropy condition is added. A “small” amount of artificial viscosity is added to the inviscid model for this purpose [96]. This artificial viscosity should also mimic the physical viscosity and create a primary vortex for flow past a highly swept wing at an angle of attack. Although secondary vortices brought about by viscous effects, on the leeward side of the wing are not modeled, their effects on the primary vortices are small [126]. The Euler equations admit solutions with distributed vorticity but do not in principle contain any mechanism for generating vorticity. Any vorticity in the solution must be introduced either by boundary conditions or by shocks. Due to the extra entropy condition shocks will lead to an increase of entropy and therefore also generate vorticity according to the Crocco’s theorem [127]. If the boundary conditions at the inflow boundary are such that vorticity is implied, this vorticity will naturally be convected into the domain and eventually be convected out at the outflow boundary. Furthermore, a solid boundary with sharp edge can also generate vorticity since attached flow around such an edge gives rise to shocks and thus also vorticity. In principle, this mechanism would act as an “automatic” Kutta condition [128] for the flow around an airfoil with a sharp trailing edge. However,

some numerical studies reveal that the combination of numerical errors and artificial viscosity will then produce vorticity and thus force the flow to separate at the edge. Section 3.3 discusses the artificial viscosity model in more detail.

For a finite domain it is necessary to construct suitable boundary conditions such that the desired steady state solution is obtained. The theory of absorbing conditions [129] is used in its simplest formulation. By linearizing the equations locally along the boundary and computing the characteristic variables along surface normals, it is possible to give the physically correct boundary information while maintaining good absorption of the transient error waves. The latter property is especially important for internal flows where stationary conditions are usually more difficult to obtain than for external flows. A more detailed description of these absorbing boundary conditions as well as other boundary conditions is given in Sec. 3.4.3.

3.2 Spatial Finite-Volume Discretization

A method to solve the 3D Euler equations has been developed in [130-132]. It is a time-dependent finite-volume approach that uses multistage explicit time integration schemes together with centered space differences. Significant features of this approach are integral conservation form, important for the correct capturing of shock waves and vortex sheets, its amenability to very general geometry without the need for a global coordinate transformation, and its toleration of grid singularities because the flow equations are balanced only within the cells of the grid [133], and not at the nodal point. It has been found that the time-dependent Euler equations permit solutions in which the flows separate from the leading edge of a sharp delta wing at angle of attack, without the implementation of the Kutta condition. In contrast, separated flows are obtained by space marching methods only if the Kutta condition is enforced.

The simplest way to derive the centered finite volume spatial discretization is to apply the integral formulation, Eq.(3.2), of the Euler equations to each grid cell of a given grid (see Fig. 3.1), i.e.

$$\frac{d}{dt} \int_{\Omega_{i,j,k}} \bar{q} dx dy dz + \oint_{\partial\Omega_{i,j,k}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = 0 \quad (3.3)$$

where $\Omega_{i,j,k}$ = volume element (i,j,k).

By using the mean-value theorem, Eq.(3.3) is expressed as

$$VOL_{i,j,k} \frac{d}{dt} \bar{q}_{i,j,k} + \delta_I \bar{F}_{i,j,k} + \delta_J \bar{G}_{i,j,k} + \delta_K \bar{H}_{i,j,k} = 0 \quad (3.4)$$

where the undivided central-difference operators, $\delta_I, \delta_J, \delta_K$, are defined as

$$\begin{aligned} \delta_I \bar{F}_{i,j,k} &= \bar{F}_{i+\frac{1}{2},j,k} - \bar{F}_{i-\frac{1}{2},j,k} \\ \delta_J \bar{G}_{i,j,k} &= \bar{G}_{i,j+\frac{1}{2},k} - \bar{G}_{i,j-\frac{1}{2},k} \\ \delta_K \bar{H}_{i,j,k} &= \bar{H}_{i,j,k+\frac{1}{2}} - \bar{H}_{i,j,k-\frac{1}{2}} \quad , \end{aligned}$$

and,

$$\begin{aligned} \bar{F}_{i+\frac{1}{2},j,k} &= SIX_{i+\frac{1}{2},j,k} \cdot \bar{F}_{i+\frac{1}{2},j,k} + SIY_{i+\frac{1}{2},j,k} \cdot \bar{G}_{i+\frac{1}{2},j,k} \\ &\quad + SIZ_{i+\frac{1}{2},j,k} \cdot \bar{H}_{i+\frac{1}{2},j,k} \end{aligned}$$

$$\begin{aligned} \bar{F}_{i-\frac{1}{2},j,k} &= SIX_{i-\frac{1}{2},j,k} \cdot \bar{F}_{i-\frac{1}{2},j,k} + SIY_{i-\frac{1}{2},j,k} \cdot \bar{G}_{i-\frac{1}{2},j,k} \\ &\quad + SIZ_{i-\frac{1}{2},j,k} \cdot \bar{H}_{i-\frac{1}{2},j,k} \end{aligned}$$

$$\begin{aligned} \bar{G}_{i,j+\frac{1}{2},k} &= SIX_{i,j+\frac{1}{2},k} \cdot \bar{F}_{i,j+\frac{1}{2},k} + SIY_{i,j+\frac{1}{2},k} \cdot \bar{G}_{i,j+\frac{1}{2},k} \\ &\quad + SIZ_{i,j+\frac{1}{2},k} \cdot \bar{H}_{i,j+\frac{1}{2},k} \end{aligned}$$

$$\begin{aligned} \bar{G}_{i,j-\frac{1}{2},k} &= SIX_{i,j-\frac{1}{2},k} \cdot \bar{F}_{i,j-\frac{1}{2},k} + SIY_{i,j-\frac{1}{2},k} \cdot \bar{G}_{i,j-\frac{1}{2},k} \\ &\quad + SIZ_{i,j-\frac{1}{2},k} \cdot \bar{H}_{i,j-\frac{1}{2},k} \end{aligned}$$

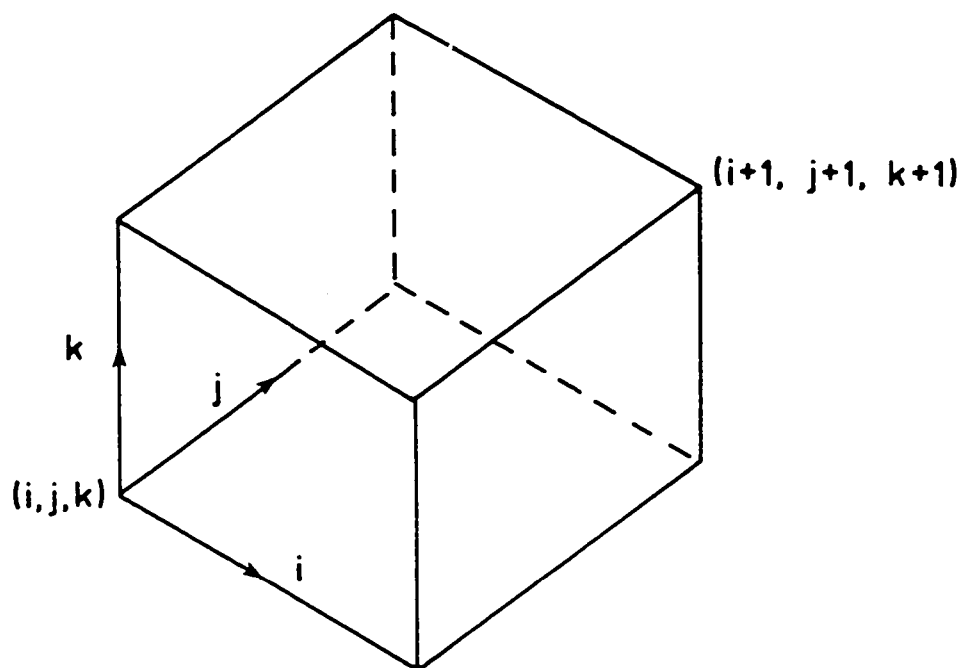


Fig. 3.1 A typical volume element.

$$\begin{aligned}\vec{H}_{i,j,k+\frac{1}{2}} &= SIX_{i,j,k+\frac{1}{2}} \cdot \vec{F}_{i,j,k+\frac{1}{2}} + SIY_{i,j,k+\frac{1}{2}} \cdot \vec{G}_{i,j,k+\frac{1}{2}} \\ &\quad + SIZ_{i,j,k+\frac{1}{2}} \cdot \vec{H}_{i,j,k+\frac{1}{2}}\end{aligned}$$

$$\begin{aligned}\vec{H}_{i,j,k-\frac{1}{2}} &= SIX_{i,j,k-\frac{1}{2}} \cdot \vec{F}_{i,j,k-\frac{1}{2}} + SIY_{i,j,k-\frac{1}{2}} \cdot \vec{G}_{i,j,k-\frac{1}{2}} \\ &\quad + SIZ_{i,j,k-\frac{1}{2}} \cdot \vec{H}_{i,j,k-\frac{1}{2}} \quad .\end{aligned}$$

Since $\vec{q}_{i,j,k}$ is located in the center of the cell but the flux function, $[\vec{F}(\vec{q})]_{i+\frac{1}{2},j,k}$, must be expressed at its surface, some form of local interpolation of the neighboring discrete values must be devised. The simplest, and perhaps most natural, function is

$$\begin{aligned}[\vec{F}(\vec{q})]_{i+\frac{1}{2},j,k} &= \vec{F}(\mu_I \vec{q}_{i+\frac{1}{2},j,k}) \\ [\vec{F}(\vec{q})]_{i,j+\frac{1}{2},k} &= \vec{F}(\mu_J \vec{q}_{i,j+\frac{1}{2},k}) \\ [\vec{F}(\vec{q})]_{i,j,k+\frac{1}{2}} &= \vec{F}(\mu_K \vec{q}_{i,j,k+\frac{1}{2}})\end{aligned}\tag{3.5}$$

Expressions similar to Eq.(3.5) are obtained for \vec{G} and \vec{H} . The average operator, μ , is defined as

$$\begin{aligned}\mu_I \psi_{i+\frac{1}{2},j,k} &= \frac{1}{2}(\psi_{i+1,j,k} + \psi_{i,j,k}) \\ \mu_J \psi_{i,j+\frac{1}{2},k} &= \frac{1}{2}(\psi_{i,j+1,k} + \psi_{i,j,k}) \\ \mu_K \psi_{i,j,k+\frac{1}{2}} &= \frac{1}{2}(\psi_{i,j,k+1} + \psi_{i,j,k}).\end{aligned}\tag{3.6}$$

An alternative is to compute the flux function separately for each of the two neighboring dependent variables and then average the two results, i.e.

$$\begin{aligned}[\vec{F}(\vec{q})]_{i+\frac{1}{2},j,k} &= \mu_I \vec{F}(\vec{q}_{i+\frac{1}{2},j,k}) \\ [\vec{F}(\vec{q})]_{i,j+\frac{1}{2},k} &= \mu_J \vec{F}(\vec{q}_{i,j+\frac{1}{2},k}) \\ [\vec{F}(\vec{q})]_{i,j,k+\frac{1}{2}} &= \mu_K \vec{F}(\vec{q}_{i,j,k+\frac{1}{2}})\end{aligned}\tag{3.7}$$

and similar expressions are obtained for \vec{G} and \vec{H} .

If the flux functions were linear, alternatives represented by Eqs.(3.5) and (3.7) would be equivalent. For a non-linear flux, only the scheme given by Eq.(3.7) provides the correct jump in \vec{q} across the shock. Thus, in this study, each term in Eq.(3.4) is defined as

$$\vec{F}_{i+\frac{1}{2},j,k} = \frac{1}{2}(\vec{F}_{i+1,j,k} + \vec{F}_{i,j,k})$$

$$\vec{F}_{i,j+\frac{1}{2},k} = \frac{1}{2}(\vec{F}_{i,j+1,k} + \vec{F}_{i,j,k})$$

$$\vec{F}_{i,j,k+\frac{1}{2}} = \frac{1}{2}(\vec{F}_{i,j,k+1} + \vec{F}_{i,j,k})$$

Similar expressions are obtained for \vec{G} and \vec{H} . Finally, the other terms are expressed as

$$\begin{aligned} SIX_{i+\frac{1}{2},j,k} &= \frac{1}{2}((y_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\ &\quad \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (z_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}})) \end{aligned}$$

$$\begin{aligned} SIY_{i+\frac{1}{2},j,k} &= \frac{1}{2}((z_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\ &\quad \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (x_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}})) \end{aligned}$$

$$\begin{aligned} SIZ_{i+\frac{1}{2},j,k} &= \frac{1}{2}((x_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\ &\quad \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (y_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}})) \end{aligned}$$

$$SIX_{i,j+\frac{1}{2},k} = \frac{1}{2}((y_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - y_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}})$$

$$\begin{aligned}
& \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\
& - (z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - z_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}})) \\
SIY_{i,j+\frac{1}{2},k} &= \frac{1}{2}((z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - z_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\
& - (x_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}})) \\
SIZ_{i,j+\frac{1}{2},k} &= \frac{1}{2}((x_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}) \\
& - (y_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - y_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}})) \\
SIX_{i,j,k+\frac{1}{2}} &= \frac{1}{2}((y_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& - (z_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}})) \\
SIY_{i,j,k+\frac{1}{2}} &= \frac{1}{2}((z_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& - (x_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (z_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}})) \\
SIZ_{i,j,k+\frac{1}{2}} &= \frac{1}{2}((x_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (y_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& - (y_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - y_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}) \\
& \cdot (x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}})).
\end{aligned}$$

The verification of these formulas including the calculation of a cell volume are given in Appendix B.

From these formulas it is clear that the only quantities needed from the coordinate transformation are the x, y, z - coordinates of the grid points. Equation (3.4) together with (3.7) leads to a spatial-difference operator completely centered in all three coordinate directions, which is second-order-accurate in space if the variation in grid size is reasonably smooth.

The finite-volume discretization bears some similarity to both the conventional finite-difference and finite element discretizations. Its formulation, like the finite-element procedure, begins with the integral equations. Its difference stencil is that of a finite-difference scheme, but it differs in that the cell-averaged quantities instead of point quantities are differenced, and this gives a significant distinction near a grid singularity. In the finite-volume formulation, the flux quantities can be defined and remain finite even in the presence of the grid singularity, since Eq.(3.4) is balanced in the interior of the cell where no coordinates are used. The usual grid-point methods may not yield this feature without special programming considerations. Eriksson [133] has concluded that without any modification the finite-volume technique remains stable in the presence of a grid singularity, but its accuracy decreases to somewhere between first and second order in space. Without the alteration the finite-difference scheme is unstable even if the singularities are straddled. However, if a limiting form of the difference scheme is derived at the singularity point and implemented in the computer code, stability of the finite difference scheme can be restored. An important aspect of the finite-volume approach is that it is well suited for the conservative rezoning approach used in this study. This is true because fluxes are obtained as the average values at the center of the cell faces, i.e., no interpolation from grid points is needed. So, the order of the interpolation scheme does not play a role in the interface treatment.

3.3 Artificial-Viscosity Model

The central difference schemes to solve the Euler equations are inherently dispersive and not dissipative. Even for linear problems, central-difference schemes admit as a solution so-called sawtooth waves. The non-linear nature of the Euler equations gives rise to an aliasing phenomenon whereby these short waves interact with each other, vanish, and reappear as distorted long waves. In nonlinear transport there is a mechanism by which energy migrates from long wavelength motion to progressively shorter and shorter scales until it is removed from the flow by molecular viscosity. The Euler equations possess no such viscosity so, in the discrete representation, this energy would migrate to the smallest scale resolvable on the grid and then returns to large-scale motion via aliasing, which is non-physical and would make a steady state unattainable [134]. In general, these defects could be dealt by digital filtering techniques. However, further deficiencies arise. The nonlinear conservation equations admit non-unique weak solutions when shocks are to be captured. An entropy condition has to be supplied in order to obtain the physically correct weak solution [135]. A standard way to invoke an entropy condition is to model the true physical process inside a shock by the addition of a small dissipation term to the convective differences. This so-called artificial viscosity mimics the real physical viscosity not only by involving an entropy condition but also by removing the short-wave motion out of the flow.

A number of studies has been conducted on construction of such artificial viscosity models, but they vary in detail from method to method. The construction of the models is arbitrary except for the classification according to its order of magnitude in terms of grid spacing. In this study, the dissipation is introduced at the same time as the transport process. Its magnitude lies in or below the range of the truncation error of the discrete approximation. The total difference operator $\vec{F}(\vec{q})$ therefore consists of (1) the convective part $\vec{F}_C(\vec{q})$ that results from

discretizing the Euler equations in space by the centered finite-volume scheme, and (ii) the dissipative part $\vec{F}_D(\vec{q})$. Thus, Eq.(3.4), can be written as

$$\frac{d}{dt}\vec{q}_{ijk} = \vec{F}_C(\vec{q}_{ijk}) + \vec{F}_D(\vec{q}_{ijk}) = \vec{F}(\vec{q}_{ijk}) \quad (3.8)$$

The total discrete dissipative operator $\vec{F}_D(\vec{q}_{ijk})$ includes its own artificial boundary conditions, and comprises both linear and nonlinear terms according to $\vec{F}_D(\vec{q}_{ijk}) = f(C_{ijk}) + D\vec{q}_{ijk}$, where D is a constant matrix. The nonlinear expression $f(C_{ijk})$ is designed to provide dissipation at discontinuities, whereas the linear one is formulated to suppress spurious solutions (sawtooth waves) and to control the migration of energy from large to subgrid scales.

3.3.1 Nonlinear Artificial Viscosity

The nonlinear artificial viscosity in the interior of the domain is expressed by

$$f_{ijk} = \chi\delta_I[S_I(\vec{q}_{ijk})\delta_I] + \delta_J[S_J(\vec{q}_{ijk})\delta_J] + \delta_K[S_K(\vec{q}_{ijk})\delta_K]\vec{q}_{ijk} \quad (3.9)$$

where χ is a constant and S_I, S_J and S_K are coefficients that depend on the solution field through the pressure according to

$$S_I = |\delta_I^2 LP_{ijk}|, S_J = |\delta_J^2 LP_{ijk}|, S_K = |\delta_K^2 LP_{ijk}|,$$

where δ_I^2, δ_J^2 and δ_K^2 are central difference operators,

$$\delta_I^2\psi_{ijk} = \psi_{i+1,j,k} - 2\psi_{i,j,k} + \psi_{i-1,j,k}$$

$$\delta_J^2\psi_{ijk} = \psi_{i,j+1,k} - 2\psi_{i,j,k} + \psi_{i,j-1,k}$$

$$\delta_K^2\psi_{ijk} = \psi_{i,j,k+1} - 2\psi_{i,j,k} + \psi_{i,j,k-1}$$

and $LP_{ijk} = \log(p_{ijk})$. These coefficients are normalized by their maximum value so that their magnitudes lie between 0 and 1. Their purpose is to sense non-smooth flow and increase the filtering of large gradients so that in effect an entropy condition is enforced. At the boundaries, the coefficients S_I, S_J , and S_K are set to zero.

3.3.2 Linear Artificial Viscosity

At all interior cells, the fourth-order difference operator is used and the linear artificial viscosity is expressed as

$$D\bar{q}_{ijk} = -\gamma(\delta_I^4 + \delta_J^4 + \delta_K^4)\bar{q}_{ijk} \quad (3.10)$$

where

$$\delta_I^4\psi_{ijk} = \psi_{i-2,j,k} - 4\psi_{i-1,j,k} + 6\psi_{i,j,k} - 4\psi_{i+1,j,k} + \psi_{i+2,j,k}$$

$$\delta_J^4\psi_{ijk} = \psi_{i,j-2,k} - 4\psi_{i,j-1,k} + 6\psi_{i,j,k} - 4\psi_{i,j+1,k} + \psi_{i,j+2,k}$$

$$\delta_K^4\psi_{ijk} = \psi_{i,j,k-2} - 4\psi_{i,j,k-1} + 6\psi_{i,j,k} - 4\psi_{i,j,k+1} + \psi_{i,j,k+2}$$

and γ is a constant. The linear extrapolation is used at the boundary cells. For example, if $i=1$ denotes grid cells adjacent to a boundary, the linear extrapolation gives

$$\delta_I^4\psi_{1jk} = \psi_{1,j,k} - 2\psi_{2,j,k} + \psi_{3,j,k}$$

$$\delta_I^4\psi_{2jk} = -2\psi_{1,j,k} + 5\psi_{2,j,k} - 4\psi_{3,j,k} + \psi_{4,j,k} \quad .$$

Similar expression can be written for the other boundaries.

Special consideration is given at the interfaces since they are not physical boundaries. The discussion on this topic is postponed until Sec. 3.4.4.

3.4 Boundary Conditions

For the computation of many fluid dynamic problems more difficulties are encountered in satisfying the boundary conditions than in balancing the differential equations at the interior points of the flow field. This is because on the boundary not all of the flow variables are specified by the boundary conditions, and there remain more unknowns than equations. While transformation to a boundary-fitted coordinate system does reduce to one the number of unspecified boundary variables necessary for differencing the interior field, namely the pressure [136], still a

method is needed to couple these unknown values of pressure to those in the interior in a manner consistent with the boundary conditions. Improper treatment of the boundary conditions can lead to serious errors and perhaps instability. In order to treat the flow exterior to a domain an artificial outer boundary must be introduced to produce a bounded domain. This is an artificial boundary in the sense that the actual flow in the physical domain is open, whereas, the computational space must, for practical reasons, be closed. The numerical conditions, therefore, should allow phenomenon generated in the computational domain to pass through the boundary without undergoing significant distortion and without influencing the interior solution. Thus, the maximum amount of transient energy can escape from the field so that the time-dependent solution converges to the steady state. Engquist and Majda [129] have presented a mathematical theory for the practical application of local absorbing boundary conditions at artificial boundaries. Their “First Approximation” is adapted in this study.

Four distinct types of boundary conditions: conditions at solid walls, periodic conditions across coordinate cuts, flow into or out of the artificial boundaries, and conditions at the interfaces, are discussed below.

3.4.1 Solid Walls

At a solid wall the mass flux is zero but the surface pressure contributes to the momentum flux. For this case, Eq.(3.3) is written as

$$\int_{\text{wall}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = \int_{\text{wall}} \vec{S} ds \quad (3.11)$$

where

$$\vec{S} = \begin{bmatrix} 0 \\ \hat{n}_x \cdot p \\ \hat{n}_y \cdot p \\ \hat{n}_z \cdot p \\ 0 \end{bmatrix}$$

Equation (3.11) is used to derive the contribution from those cell walls which coincide with a solid wall. For example, if $j = \frac{1}{2}$ is denoted for these grid cell walls (Fig. 3.2), then Eq. (3.11) is approximated by

$$\int_{\text{wall}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = \begin{bmatrix} 0 \\ SJX_{i,\frac{1}{2},k} \cdot p_{i,\frac{1}{2},k} \\ SJY_{i,\frac{1}{2},k} \cdot p_{i,\frac{1}{2},k} \\ SJZ_{i,\frac{1}{2},k} \cdot p_{i,\frac{1}{2},k} \\ 0 \end{bmatrix} \quad (3.12)$$

where

$$\begin{aligned} SJX_{i,\frac{1}{2},k} &= \frac{1}{2} \left((y_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - y_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \right. \\ &\quad \cdot (z_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (z_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - z_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \left. \cdot (y_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \right) \end{aligned}$$

$$\begin{aligned} SJY_{i,\frac{1}{2},k} &= \frac{1}{2} \left((z_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - z_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \right. \\ &\quad \cdot (x_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (x_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \left. \cdot (z_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - z_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \right) \end{aligned}$$

$$\begin{aligned} SJZ_{i,\frac{1}{2},k} &= \frac{1}{2} \left((x_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \right. \\ &\quad \cdot (y_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - y_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \\ &\quad - (y_{i+\frac{1}{2},\frac{1}{2},k-\frac{1}{2}} - y_{i-\frac{1}{2},\frac{1}{2},k+\frac{1}{2}}) \\ &\quad \left. \cdot (x_{i+\frac{1}{2},\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},\frac{1}{2},k-\frac{1}{2}}) \right) , \end{aligned}$$

and $p_{i,\frac{1}{2},k}$ is approximated by a linear extrapolation from the center of the interior cells, i.e.,

$$p_{i,\frac{1}{2},k} = \frac{3}{2} p_{i,1,k} - \frac{1}{2} p_{i,2,k} .$$

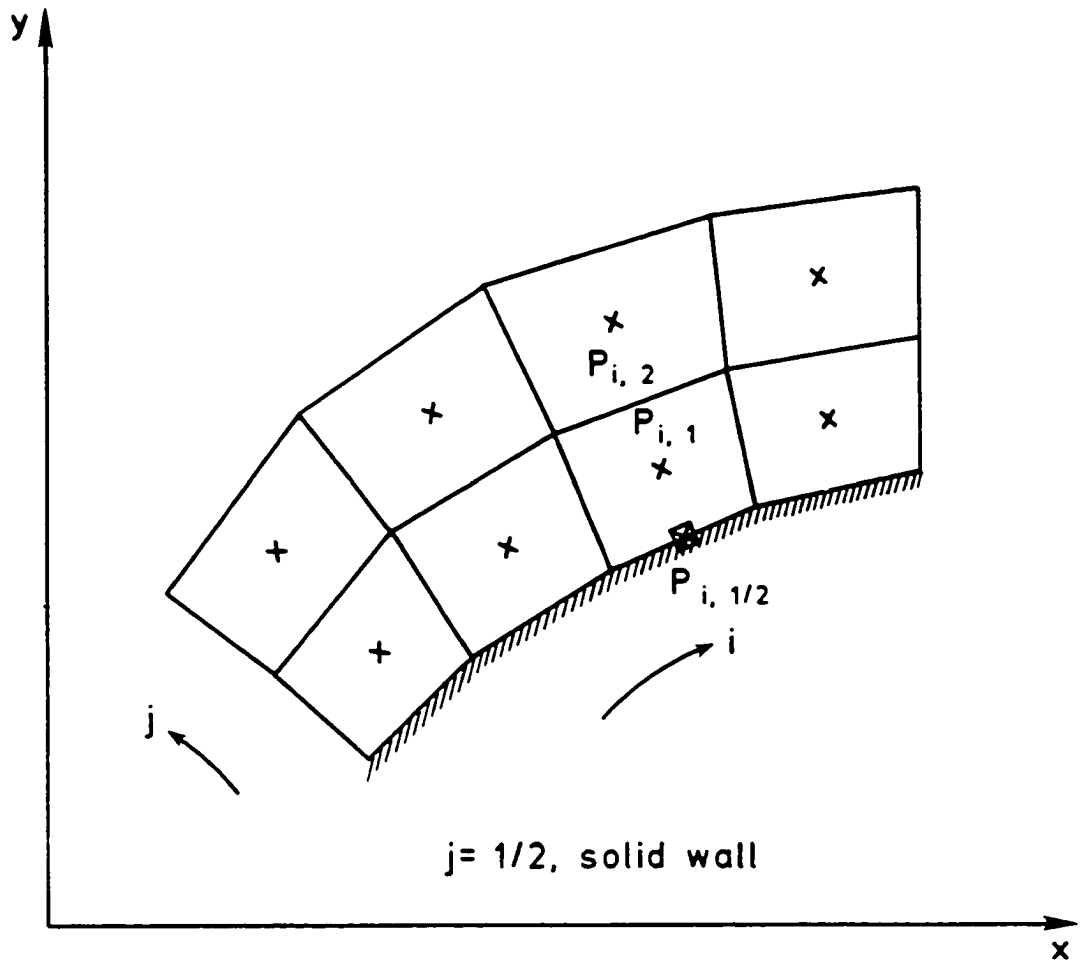


Fig. 3.2 A solid wall boundary.

3.4.2 Coordinate Cuts

At a coordinate cut the physical space folds on to itself and the condition on the flow at the computational boundary is periodicity. This boundary, in fact, does not exist as a physical boundary. It is a boundary only in a practical programming sense. Thus, it does not influence the solutions in the interior. An example of this type of boundary can be seen in Fig. 3.3.

3.4.3 Inflow/Outflow Boundaries

As mentioned previously, these boundaries are artificial boundaries for practical reasons. The theory of absorbing boundary conditions [129] is applied to convert the transient energy out of the flow field so that the steady state solutions can be reached. This is done by linearizing the governing equation locally and computing the characteristic variables in the normal direction. Those characteristic variables which are advected into the domain are then fixed to the desired values whereas those which are advected out of the domain are linearly extrapolated from the interior to the boundary. The resulting complete set of characteristic variables is then transformed back to the primitive variables and used to compute the desired fluxes. The concept of Engquist and Majda's 'First approximation' is described below.

A corresponding one-dimensional system of linear hyperbolic equation can be written in the characteristic form as

$$\frac{\partial \vec{q}}{\partial t} + A \frac{\partial \vec{q}}{\partial x} = 0 \quad ; \quad A = \frac{\partial \vec{F}}{\partial \vec{q}} \quad (3.13)$$

where \vec{q} represent the characteristic variables and the Jacobian matrix A can be

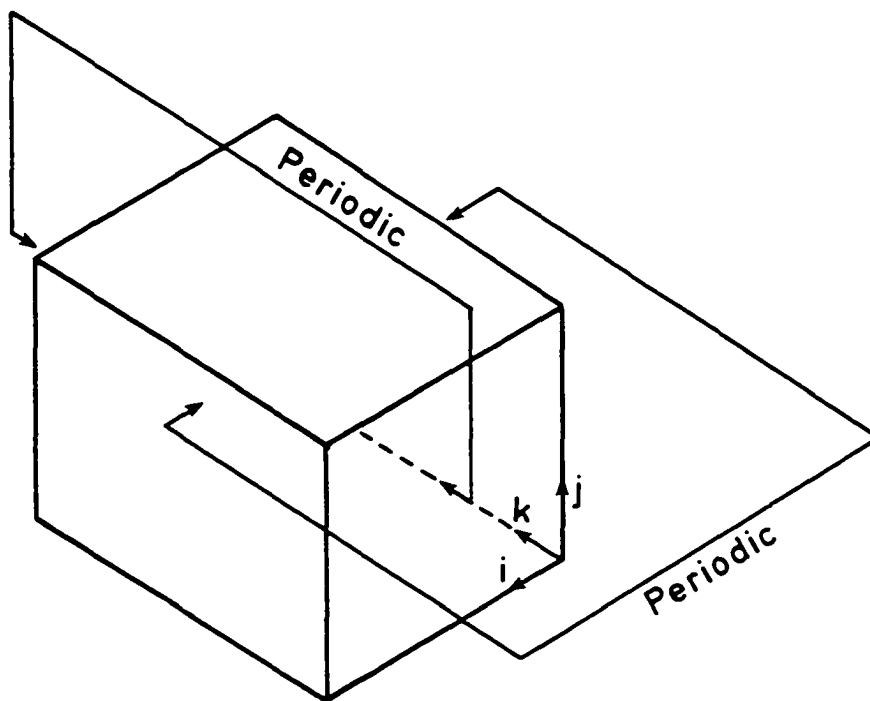


Fig. 3.3 Coordinate cut boundaries.

written as

$$A = \begin{bmatrix} 0 & \alpha & \beta & \varepsilon \\ \alpha c^2 - uU + \kappa\alpha V^2 & \alpha(1 - \kappa)u + U & \beta u - \kappa\alpha v & \varepsilon u - \kappa\alpha w \\ \beta c^2 - vU + \kappa\beta V^2 & \alpha v - \kappa\beta u & \beta(1 - \kappa)v + U & \varepsilon v - \kappa\beta w \\ \varepsilon c^2 - wU + \kappa\varepsilon V^2 & \alpha w - \kappa\varepsilon u & \beta w - \kappa\varepsilon v & \varepsilon(1 - \kappa)w + U \end{bmatrix}$$

where

$$\alpha = \vec{s} \cdot \hat{e}_x \quad ; \quad \beta = \vec{s} \cdot \hat{e}_y \quad ; \quad \varepsilon = \vec{s} \cdot \hat{e}_z$$

$$U = \alpha u + \beta v + \varepsilon w \quad ; \quad V^2 = \vec{V} \cdot \vec{V} \quad ; \quad \kappa = \frac{\gamma - 1}{\gamma} \quad ,$$

c is the local speed of sound and \vec{s} is the surface area of the cell face which coincide with the artificial boundaries.

The eigenvalue λ of A can be found by solving $\det(A - \lambda I) = 0$, as

$$\lambda_1 = U, \quad \lambda_2 = U, \quad \lambda_3 = U - a_+, \quad \lambda_4 = U - a_-,$$

where

$$a_{\pm} = \frac{1}{2}\kappa U \pm \left[\frac{1}{4}\kappa^2 U^2 + c^2(\alpha^2 + \beta^2 + \varepsilon^2) \right]^{\frac{1}{2}} .$$

The left and right eigenvalues associated with these four eigenvalues make up the row and columns of the transformation matrices T^{-1} and T respectively which diagonalize Eq.(3.13) as

$$\frac{\partial \vec{\phi}}{\partial t} + \Lambda \frac{\partial \vec{\phi}}{\partial x} = 0 \quad (3.14)$$

where

$$\vec{\phi} = T^{-1} \vec{q} \quad , \quad \Lambda = T^{-1} A T = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix}$$

After the intermediate variables,

$$\tilde{U} = \beta u + \alpha v, \quad \tilde{V} = -\varepsilon v + \beta w, \quad \tilde{W} = \varepsilon u - \alpha w,$$

$$\xi = \alpha^2 + \beta^2 + \varepsilon^2, \quad Q_{\pm} = \kappa U - a_{\pm}, \quad R_{\pm} = \varepsilon Q_{\pm} - \zeta \kappa w, \quad P_{\pm} = \kappa w a_{\pm} + \varepsilon c^2$$

have been defined for the sake of simplification, it can be found that

$$T = \begin{bmatrix} \kappa\tilde{U} & 0 & R_+ & R_- \\ \kappa vU - \beta(\kappa V^2 + c^2) & \tilde{V} & uR_+ + \alpha P_+ & uR_- + \alpha P_- \\ -\kappa uU + \alpha(\kappa V^2 + c^2) & \tilde{W} & vR_+ + \beta P_+ & vR_- + \beta P_- \\ 0 & \tilde{U} & wR_+ - \kappa(\alpha u + \beta v)a_+ & wR_- - \kappa(\alpha u + \beta v)a_- \\ & & -(\alpha^2 + \beta^2)c^2 & -(\alpha^2 + \beta^2)c^2 \end{bmatrix}$$

and

$$T^{-1} = \begin{bmatrix} \frac{\xi V^2 - U^2}{d_1} & \frac{-\epsilon\tilde{W} + \beta\tilde{U}}{d_1} & \frac{\epsilon\tilde{V} - \alpha\tilde{U}}{d_1} & \frac{\alpha\tilde{W} - \beta\tilde{V}}{d_1} \\ [\kappa w(U^2 - \xi V^2) & [\kappa w(-\beta\tilde{U} + \epsilon\tilde{W}) & [\kappa w(\alpha\tilde{U} - \epsilon\tilde{V}) & [-\kappa w(\alpha\tilde{W} - \beta\tilde{V}) \\ + c^2(\epsilon U - \xi w)]/d_2 & -\alpha\epsilon c^2]/d_2 & -\beta\epsilon c^2]/d_2 & +(\alpha^2 + \beta^2)c^2]/d_2 \\ \frac{R_+}{d_3} & \frac{\kappa\xi V^2 - (U+a_+)Q_+}{d_3} & \frac{-\kappa\xi u + \alpha Q_+}{d_3} & \frac{-\kappa\xi v + \beta Q_+}{d_3} \\ \frac{R_-}{d_4} & \frac{\kappa\xi V^2 - (U+a_-)Q_-}{d_4} & \frac{-\kappa\xi u + \alpha Q_-}{d_4} & \frac{-\kappa\xi v + \beta Q_-}{d_4} \end{bmatrix}$$

The factor d_1 , d_2 , d_3 and d_4 in the dominators are normalizing coefficients so that $T^{-1}T$ equals the unit matrix. For the one-dimensional case it is well known that the number of conditions to be imposed in a cell at the outer boundary should equal the number of characteristic directions that enter the computational domain. Four typical cases are shown in Fig. 3.4. With subsonic inflow the implementation is to set the three ingoing characteristic variables $\phi^{(1)}$, $\phi^{(2)}$, and $\phi^{(3)}$ to their free-stream values, linearly extrapolate the fourth $\phi^{(4)}$ from the computational field, and then solve for the original unknowns $\vec{q} = T\vec{\phi}$. At outflow it is $\phi^{(3)}$ that is given the values of the undisturbed flow, and $\phi^{(1)}$, $\phi^{(2)}$, and $\phi^{(4)}$ are extrapolated from the computational field.

3.4.4 Interface Conditions

An interface, here, is referred to as a common boundary where two or more subdomain grids are patched together. As mentioned previously, these boundaries are not physical boundaries and may compose of grids of different topologies. Care must be taken in order to treat these boundary conditions which exist because the computation is done on each subdomain grid individually. This study follows a

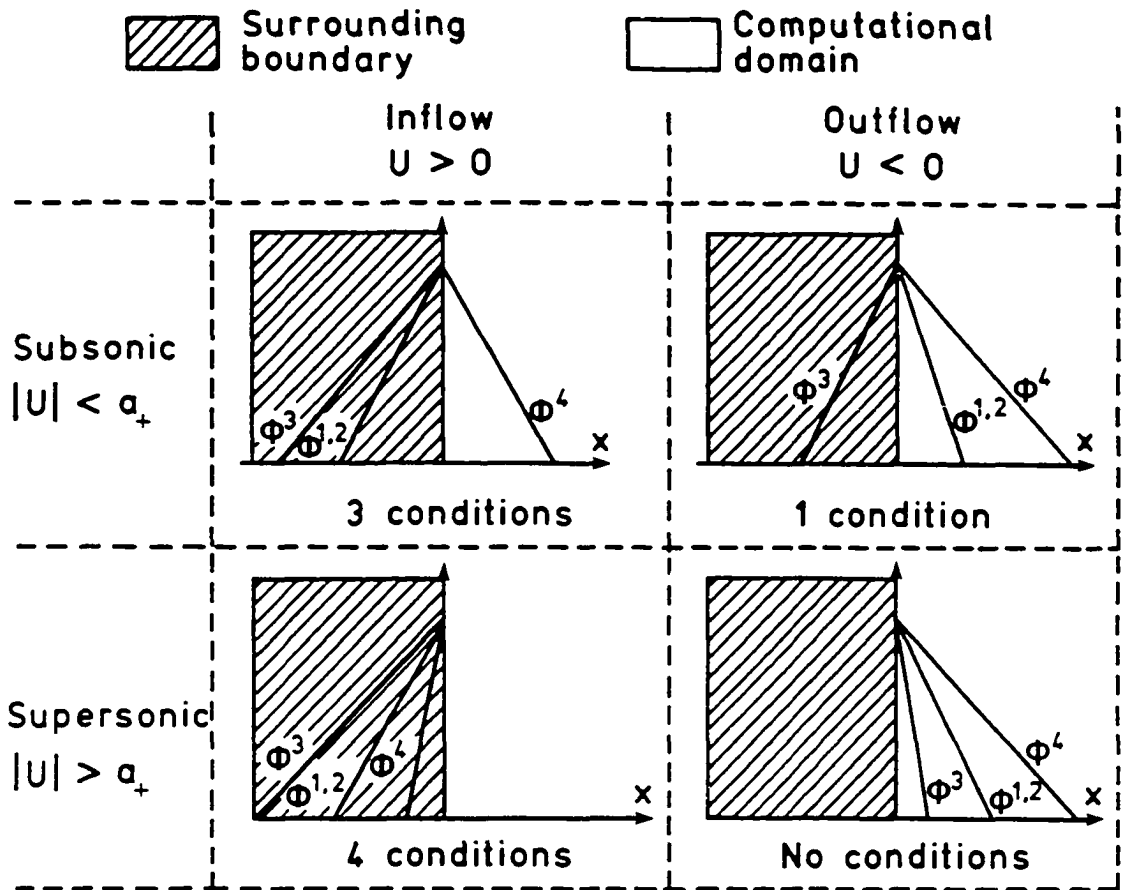


Fig. 3.4 Conditions at the inflow/outflow boundaries.

conservative approach which offers the conservation of fluxes at the interfaces. The conservative treatment at the interfaces is important for the correct capturing of discontinuities crossing them. The discussion on the conservative rezoning algorithm is given previously. An example of its application is described below.

Consider an interface between two subdomain grids as shown in Fig. 3.5. The application of the finite volume approach to the interface cells, denoted as $(i,j,NK-1)$, requires the integrated fluxes, $H_{i,j,NK-\frac{1}{2}}^{(1)} = h_{i,j,NK-\frac{1}{2}}^{(1)} \cdot A_{i,j,NK-\frac{1}{2}}^{(1)}$, where $A_{i,j,NK-\frac{1}{2}}^{(1)}$ are the cell surface areas at the interface. Recall from Sec. 3.2 that fluxes at the cell walls interior to the domain are computed by taking the average of the flux functions evaluated at the cell centers, i.e., $h_{i,j,k+\frac{1}{2}}^{(1)} = \frac{1}{2}(h_{i,j,k}^{(1)} + h_{i,j,k+1}^{(1)})$. Thus, the evaluation of $h_{i,j,NK-\frac{1}{2}}^{(1)}$ requires the flux function $h_{i,j,NK}^{(1)}$ which is located outside the domain (denoted as grid 1 in Fig. 3.5). They can be obtained by several ways. The simplest way is to extrapolate the quantities $h^{(1)}$'s from the interior of the domain. Another way is to interpolate the quantities $h^{(2)}$'s from the interior of the adjacent domain (grid 2 in Fig. 3.5). In this study, however, these interfaces are treated as inflow/outflow boundaries and the theory of Engquist and Majda [129] is applied. Thus, $h_{i,j,NK-\frac{1}{2}}^{(1)}$ are obtained through the combination of the extrapolation and interpolation procedures according to whether the flow is supersonic or subsonic at these interfaces. Here, the incoming characteristic variables are fixed at the interpolated values rather than freestream value as in the case of farfield boundaries.

To proceed the calculation on to the next subdomain grid (grid 2 in Fig. 3.5), the integrated fluxes $H_{i,j,-\frac{1}{2}}^{(2)}$ are required for grid cells $k=1$. The fluxes $H_{i,j,-\frac{1}{2}}^{(2)}$ must be obtained from $H_{i,j,NK-\frac{1}{2}}^{(1)}$ conservatively in order to maintain the global conservation. To see how this can be done, let's consider a general interfaces as shown in Fig. 3.6. Here, solid lines represent grid lines at the interface of the subdomain grid (grid1) with known quantities, i.e., $H_{i,j,NK-\frac{1}{2}}^{(1)}$. The dash lines represent grid lines at the interface of the adjacent subdomain grid (grid 2) with unknown quantities,

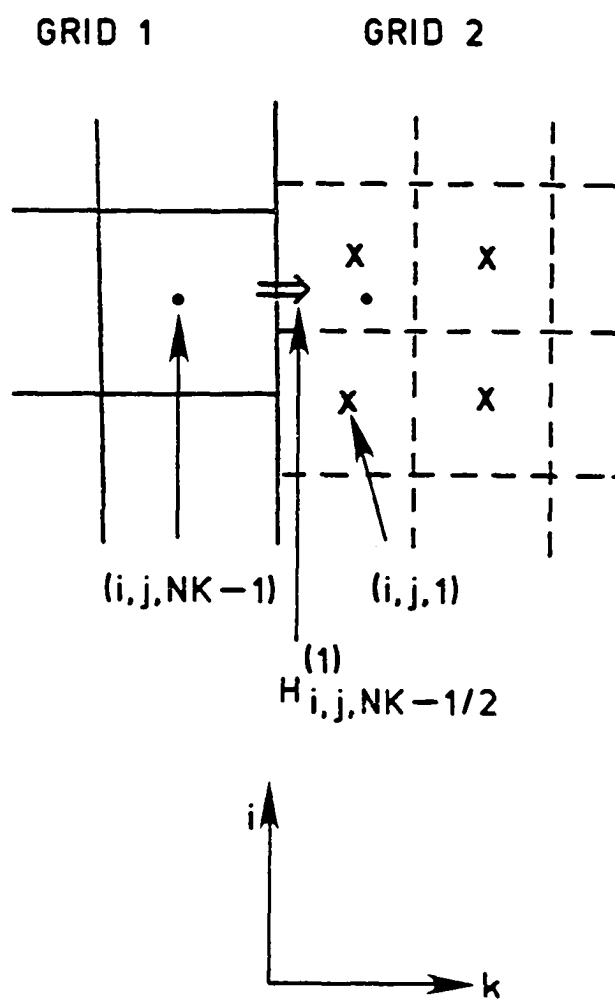


Fig. 3.5 A typical interface.

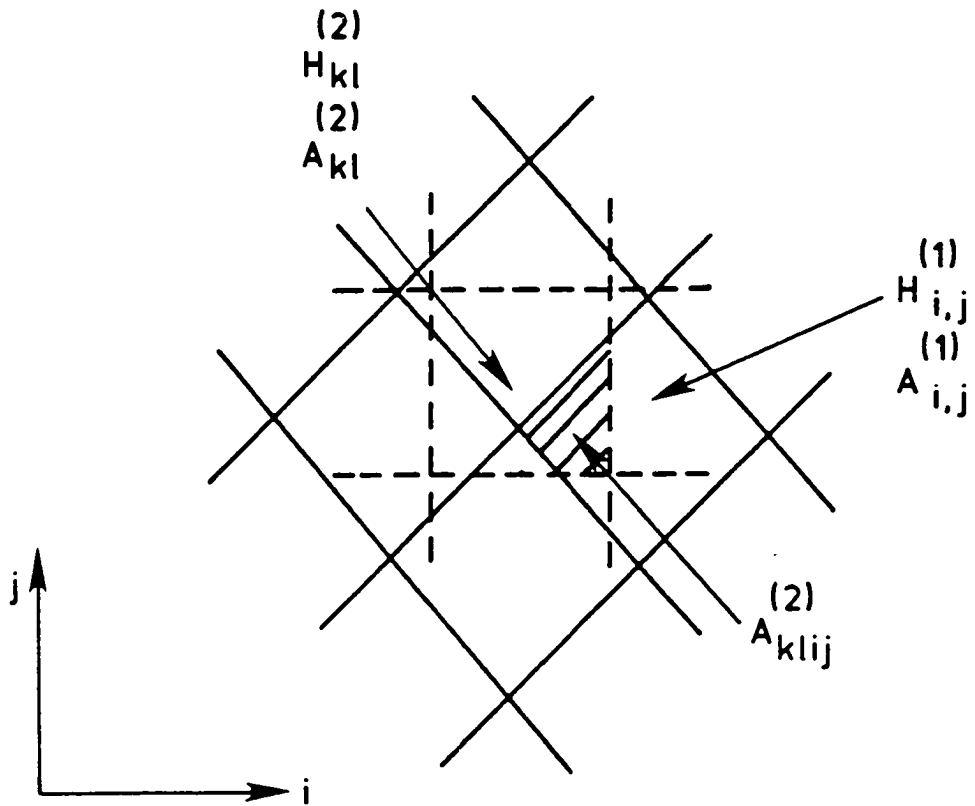


Fig. 3.6 A typical patched interface.

i.e., $H_{i,j,-\frac{1}{2}}^{(2)}$, to be obtained from $H_{i,j,NK-\frac{1}{2}}^{(1)}$ conservatively. In Fig. 3.6 $H_{i,j,NK-\frac{1}{2}}^{(1)}$ is written as $H_{i,j}^{(1)}$ and $H_{i,j,-\frac{1}{2}}^{(2)}$ as $H_{kl}^{(2)}$ to simplify the notation and to avoid the confusion (since the subscript 'i,j' appear in both quantities). The quantities $A_{i,j}^{(1)}$ are the areas of the cell walls associated with $H_{i,j}^{(1)}$ and $A_{kl}^{(2)}$ are the areas associated with $H_{kl}^{(2)}$. The conservative treatment at the interface requires that the integrated fluxes going through an area $A_{kl}^{(2)}$ be the same for both grids, i.e., $H_{kl}^{(1)} = H_{kl}^{(2)}$. From Fig. 3.6, $H_{kl}^{(1)}$ can be evaluated as

$$H_{kl}^{(1)} = \sum_{n=1}^P \frac{\int H_{ij}^{(1)} dA_{klij}^{(2)}}{\int dA_{ij}^{(1)}} \quad (3.15)$$

where $A_{klij}^{(2)}$ is the portion of the area $A_{ij}^{(1)}$ which contain in the area $A_{kl}^{(2)}$, and P is the number of the areas $A_{klij}^{(2)}$ contained in $A_{kl}^{(2)}$. For the finite-volume approach, $H_{ij}^{(1)}$ are constant within the area $A_{ij}^{(1)}$. Thus, Eq.(3.15) can be integrated and written as

$$H_{kl}^{(1)} = \sum_{n=1}^P \frac{H_{ij}^{(1)} A_{klij}^{(2)}}{A_{ij}^{(1)}} = \sum_{n=1}^P h_{ij}^{(1)} \cdot A_{klij}^{(2)} \quad (3.16)$$

Equation (3.18) is in the same form as Eq.(2.3) where $h_{ij}^{(1)}$ are known from the previous interface treatment for grid 1. Thus, the conservative rezoning algorithm can be applied to compute $H_{kl}^{(2)} = H_{kl}^{(1)}$.

Another method which can be used to compute $H_{kl}^{(2)}$ has been given by Hesenius and Rai [137]. The method makes use of the modified Sutherland-Hodgeman clipping algorithm [138] borrowed from the field of computer graphics. They illustrate the use of the method by computing the flow about a wing-canard combination. Recently, Walters et al.[139] have applied the method to obtain the solutions for flow over the hypersonic aircraft and forebodies.

Boundary conditions must also be given for the artificial viscosity terms at the interfaces. Recall that at the regular boundaries, the coefficient S_I, S_J , and S_K in the nonlinear terms are set to zero, and the linear extrapolation is used in the linear terms. The same procedure cannot be used at the interfaces. Several

procedures have been investigated. The procedure described briefly below has been found as the best choice in terms of accuracy, stability, and efficiency. For the nonlinear terms quantities $\bar{q}_{i,j,k}$ and the coefficients S_I, S_J, S_K in Eq. (3.9) which is located outside the current subdomain grid are obtained through the interpolation of the quantities and coefficients from the other grids. The same can be done for the linear terms if only one term in Eq.(3.10) is outside the current subdomain grid. For illustration, recall from Sec. 3.3.2 that the fourth-order difference operator is used to obtain the linear artificial viscosity term. The fourth-order difference operator in I direction is expressed as

$$\delta_I^4 \psi_{i,j,k} = \psi_{i-2,j,k} - 4\psi_{i-1,j,k} + 6\psi_{i,j,k} - 4\psi_{i+1,j,k} + \psi_{i+2,j,k}.$$

If the interface cells are denoted as 'i=1', it can be written

$$\delta_I^4 \psi_{2,j,k} = \psi_{INT1,j,k} - 4\psi_{1,j,k} + 6\psi_{2,j,k} - 4\psi_{3,j,k} + \psi_{4,j,k}$$

where $\psi_{INT1,j,k}$ are the quantities interpolated from those in the adjacent subdomain grids. However, the use of the same procedure, at $i=1$, which results in the formula,

$$\delta_I^4 \psi_{1,j,k} = \psi_{INT2,j,k} - 4\psi_{INT1,j,k} + 6\psi_{1,j,k} - 4\psi_{2,j,k} + \psi_{3,j,k}$$

where $\psi_{INT2,j,k}$ are the other interpolated quantities, may lead to the unstable situation. This is because the use of the interpolation procedure more than once in each grid cells may result in a nonsmooth variation of the artificial viscosity from cell to cell. This, in effect, can amplify the magnitude of the oscillation of solutions instead of damping it, as desired by the purpose of adding these terms. In this study, the terms $\psi_{INT2,j,k}$ are obtained via linear interpolation of $\psi_{INT1,j,k}$ and $\psi_{1,j,k}$. Thus, the above formula is written as

$$\delta_I^4 \psi_{1,j,k} = -2\psi_{INT1,j,k} + 5\psi_{1,j,k} - 4\psi_{2,j,k} + \psi_{3,j,k}.$$

3.5 Numerical Time Integration

The complete semi-discrete scheme, Eq.(3.8), including all boundary conditions defines a unique system of nonlinear ordinary differential equations

$$u_t = F(u) \quad ; \quad u(0) = u_0 \quad (3.17)$$

which must be integrated in time by numerical means. The explicit three-stage Runge-Kutta scheme presented by Gary [140] is used in this study. This scheme is defined by the following algorithm:

$$\begin{aligned} u^*(t_{n+1}) &= u(t_n) + \Delta t F(u(t_n)) \\ u^{**}(t_{n+1}) &= u(t_n) + \frac{1}{2} \Delta t F(u(t_n)) + \frac{1}{2} \Delta t F(u^*(t_{n+1})) \\ u(t_{n+1}) &= u(t_n) + \frac{1}{2} \Delta t F(u(t_n)) + \frac{1}{2} \Delta t F(u^{**}(t_{n+1})) \end{aligned} \quad (3.18)$$

It can be shown that this scheme, when applied to the semi-discretized Euler equations, Eqs.(3.8), is second-order accurate and is stable with a CFL (Courant-Friedrichs-Lewy) number of at most 2. The multistage two level schemes of the Runge-Kutta type have the advantage that they do not require any special starting procedure, in contrast to leap frog and Adam Bashforth methods, for example. The extra stages can be used either (1) to improve accuracy, or (2) to extend the stability region. Another advantage of this scheme is that its properties have been widely investigated, and are readily available in textbooks on ordinary differential equations.

3.6 Local Time Step Scaling

As mentioned previously, to reach a steady state solution by explicit methods, generally, requires a large number of iteration and a long computational time. This is because the time step used in explicit methods is restricted to a maximum value according to the CFL condition [141]. The maximum time step is usually

determined by the smallest grid spacing. On a highly stretched grid, this maximum time step can be extremely small. In applications where only steady state solutions are desired, and true time accuracy is of no concern, it has been found that the use of the “local time step” scaling is highly beneficial to accelerate the convergence to steady state solutions [142]. The simplest way to derive this scaling is by a local Fourier analysis. This concept is outlined below.

To obtain a better understanding of the concept, let's consider the Euler equations in two space dimensions:

$$\vec{q}_t + \vec{F}_x(\vec{q}) + \vec{G}_y(\vec{q}) = 0 \quad (3.19)$$

Assuming that the mapping $x(\xi, \eta), y(\xi, \eta)$ between the physical $x - y$ space and the computational $\xi - \eta$ space is smooth. Equation (3.19) can be transformed into the computational space as

$$(J\vec{q})_t + (y_\eta \vec{F} - x_\eta \vec{G})_\xi + (-y_\xi \vec{F} + x_\xi \vec{G})_\eta \quad (3.20)$$

where $J = \text{Jacobian of transformation} = x_\xi y_\eta - x_\eta y_\xi$. Equation (3.20) is integrated over the region and can be written as

$$\frac{d}{dt} \int \int_{D_{i,j}} \vec{q} J d\xi d\eta + \oint_{\partial D_{i,j}} [(y_\xi \vec{F} - x_\xi \vec{G}) d\xi + (y_\eta \vec{F} - x_\eta \vec{G}) d\eta] = 0 \quad (3.21)$$

The computational space has been discretized according to

$$\xi_i = \xi_0 + i\Delta\xi \quad ; \quad \eta_j = \eta_0 + j\Delta\eta$$

and some notation have been defined as

$$\begin{aligned} x_{i,j} &= x(\xi_i, \eta_j) \quad , \quad y_{i,j} = y(\xi_i, \eta_j) \quad , \\ \vec{q}_{i,j} &= \vec{q}(x_{i,j}, y_{i,j}, t) \quad , \quad \vec{F}_{i,j} = \vec{F}(\vec{q}_{i,j}) \quad , \quad \vec{G}_{i,j} = \vec{G}(\vec{q}_{i,j}) \quad , \\ \delta_\xi \vec{q}_{i,j} &= \vec{q}_{i+\frac{1}{2},j} - \vec{q}_{i-\frac{1}{2},j} \quad , \quad \delta_\eta \vec{q}_{i,j} = \vec{q}_{i,j+\frac{1}{2}} - \vec{q}_{i,j-\frac{1}{2}} \quad , \end{aligned}$$

$$\mu_\xi \bar{q}_{i,j} = \frac{1}{2}(\bar{q}_{i+\frac{1}{2},j} + \bar{q}_{i-\frac{1}{2},j}) \quad , \quad \mu_\eta \bar{q}_{i,j} = \frac{1}{2}(\bar{q}_{i,j+\frac{1}{2}} + \bar{q}_{i,j-\frac{1}{2}}) \quad .$$

Equation (3.21) can be approximated by

$$\begin{aligned} \frac{d}{dt} \bar{q}_{i,j} \int \int_{D_{i,j}} J d\xi d\eta + \delta_\eta [-(\delta_\xi y_{i,j})(\mu_\eta \bar{F}_{i,j}) + (\delta_\xi x_{i,j})(\mu_\eta \bar{G}_{i,j})] \\ + \delta_\xi [(\delta_\eta y_{i,j})(\mu_\xi \bar{F}_{i,j}) - (\delta_\eta x_{i,j})(\mu_\xi \bar{G}_{i,j})] = 0. \end{aligned} \quad (3.22)$$

Equation (3.22) is a semi-discrete centered scheme for the two dimensional Euler equation. To derive the local time step scaling for Eq.(3.22), first the transformation matrices are freezed at grid point (i,j), i.e.

$$x_\xi(\xi, \eta) \approx x_\xi(\xi_i, \eta_j) = \bar{x}_\xi \quad ; \quad x_\eta(\xi, \eta) \approx x_\eta(\xi_i, \eta_j) = \bar{x}_\eta$$

$$y_\xi(\xi, \eta) \approx y_\xi(\xi_i, \eta_j) = \bar{y}_\xi \quad ; \quad y_\eta(\xi, \eta) \approx y_\eta(\xi_i, \eta_j) = \bar{y}_\eta$$

$$J(\xi, \eta) = x_\xi(\xi, \eta)y_\eta(\xi, \eta) - x_\eta(\xi, \eta)y_\xi(\xi, \eta) \approx \bar{x}_\xi \bar{y}_\eta - \bar{x}_\eta \bar{y}_\xi = \bar{J}$$

Thus, Eq.(3.22) becomes

$$\begin{aligned} \bar{J} \Delta \xi \Delta \eta \frac{d}{dt} \bar{q}_{i,j} - \frac{1}{2} \Delta \xi \bar{y}_\xi (\bar{F}_{i,j+1} - \bar{F}_{i,j-1}) \\ + \frac{1}{2} \Delta \xi \bar{x}_\xi (\bar{G}_{i,j+1} - \bar{G}_{i,j-1}) \\ + \frac{1}{2} \Delta \eta \bar{y}_\eta (\bar{F}_{i+1,j} - \bar{F}_{i-1,j}) \\ - \frac{1}{2} \Delta \eta \bar{x}_\eta (\bar{G}_{i+1,j} - \bar{G}_{i-1,j}) = 0. \end{aligned} \quad (3.23)$$

Next, the flux functions $\bar{F}(\bar{q})$, $\bar{G}(\bar{q})$ are linearized around $\bar{q}_{i,j}$ according to

$$\begin{aligned} \bar{F}(\bar{q}) \approx \bar{F}(\bar{q}_{i,j}) + \bar{A} \cdot (\bar{q} - \bar{q}_{i,j}) \quad ; \quad \bar{G} \approx \bar{G}(\bar{q}_{i,j}) + \bar{B} \cdot (\bar{q} - \bar{q}_{i,j}) \\ \bar{A} = \frac{\partial}{\partial \bar{q}} \bar{F}(\bar{q}_{i,j}) \quad ; \quad \bar{B} = \frac{\partial}{\partial \bar{q}} \bar{G}(\bar{q}_{i,j}) \quad , \end{aligned} \quad (3.24)$$

where \bar{A} and \bar{B} are the Jacobian matrices evaluated at $\bar{q}_{i,j}$. With the aid of Eq.(3.24), Eq.(3.23) can be written as

$$\begin{aligned} \frac{d}{dt} \bar{q}_{i,j} + \frac{1}{2 \Delta \eta \bar{J}} (-\bar{y}_\xi \bar{A} + \bar{x}_\xi \bar{B})(\bar{q}_{i,j+1} - \bar{q}_{i,j-1}) \\ + \frac{1}{2 \Delta \xi \bar{J}} (\bar{y}_\eta \bar{A} - \bar{x}_\eta \bar{B})(\bar{q}_{i+1,j} - \bar{q}_{i-1,j}) = 0 \end{aligned} \quad (3.25)$$

which can be treated by the Fourier analysis. Let

$$\bar{q}_{i,j}(t) = \hat{q} e^{st} (e^{i\theta_1})^i (e^{i\theta_2})^j \quad \begin{matrix} -\pi \leq \theta_1 \leq \pi \\ -\pi \leq \theta_2 \leq \pi \end{matrix} \quad (3.26)$$

be the solution of Eq.(3.25). Substitution of Eq.(3.26) into Eq.(3.25) yields the eigenvalue problem

$$[sI + \frac{\hat{i} \sin(\theta_1)}{\Delta \xi \bar{J}} (\tilde{y}_\eta \bar{A} - \tilde{x}_\eta \bar{B}) + \frac{\hat{i} \sin(\theta_2)}{\Delta \eta \bar{J}} (-\tilde{y}_\xi \bar{A} + \tilde{x}_\xi \bar{B})] \hat{q} = 0 \quad (3.27)$$

A nontrivial solution of Eq.(3.27) can be found only if

$$\det \left[\left(\frac{\tilde{y}_\eta \sin(\theta_1)}{\Delta \xi \bar{J}} - \frac{\tilde{y}_\xi \sin(\theta_2)}{\Delta \eta \bar{J}} \right) \bar{A} + \left(\frac{-\tilde{x}_\eta \sin(\theta_1)}{\Delta \xi \bar{J}} + \frac{\tilde{x}_\xi \sin(\theta_2)}{\Delta \eta \bar{J}} \right) \bar{B} - (-\hat{i}sI) \right] = 0 \quad (3.28)$$

Since the original conservation law, Eq. (3.19), is hyperbolic, the matrices \bar{A} and \bar{B} by definition satisfy the condition

$$\det(\alpha \bar{A} + \beta \bar{B} - \lambda I) = 0; \quad \alpha, \beta \text{ real} \Rightarrow \lambda_p(\alpha, \beta)_{p=1}^n \text{ all real} \quad (3.29)$$

Thus, the eigenvalues $s_p(\theta_1, \theta_2)$ of Eq.(3.28) are all purely imaginary and given by

$$s_p(\theta_1, \theta_2) = -\hat{i} \lambda_p(\alpha, \beta) \quad ; \quad p = 1, \dots, n \quad (3.30)$$

$$\alpha = \frac{\tilde{y}_\eta \sin(\theta_1)}{\Delta \xi \bar{J}} - \frac{\tilde{y}_\xi \sin(\theta_2)}{\Delta \eta \bar{J}}$$

$$\beta = \frac{-\tilde{x}_\eta \sin(\theta_1)}{\Delta \xi \bar{J}} + \frac{\tilde{x}_\xi \sin(\theta_2)}{\Delta \eta \bar{J}}$$

For the Euler equations the Jacobian matrices \bar{A} and \bar{B} are known analytically as well as the eigenvalues $\lambda_p(\alpha, \beta)$. Thus, the local spectral radius at the grid point (i,j)

$$\rho_{i,j} = \max_{p, \theta_1, \theta_2} |s_p(\theta_1, \theta_2)|$$

can be estimated by analytic means. The "local time step" scaling of Eq.(3.22) is, then defined by

$$\left(\int \int_{D_{i,j}} J d\xi d\eta \right) \rho_{i,j} \frac{d}{dt} \bar{q}_{i,j} + \delta_\eta [-(\delta_\xi y_{i,j}) (\mu_\eta \bar{F}_{i,j}) + (\delta_\xi x_{i,j}) (\mu_\eta \bar{G}_{i,j})] + \delta_\xi [(\delta_\eta y_{i,j}) (\mu_\xi \bar{F}_{i,j}) - (\delta_\eta x_{i,j}) (\mu_\xi \bar{G}_{i,j})] = 0 \quad (3.31)$$

which evidently scales the problem so that the local spectral radius is equal to 1 everywhere. It can be seen that this type of scaling does not affect a steady solution of the original scheme.

Chapter 4

APPLICATION TO THE AERODYNAMIC BODIES

The procedure described in Chap. 3 has been applied to obtain steady-state solutions for inviscid flow about various configurations on the multiple grid systems. Basically, the entire domain is subdivided into several subdomains which are patched with each other at the common boundaries (called 'interfaces'). The finite-volume method is used to discretize the Euler equations in each subdomain separately. The discretized Euler equations in each subdomain are integrated in time from the initially guessed solutions (free stream conditions) by the three-stage Runge-Kutta integration scheme. Steady state solutions are reached when the change in solutions between two consecutive time steps is very small (say 1×10^{-9}). The calculation is performed separately in different subdomains. As mentioned previously, some information must be transferred at the interfaces of various subdomains. This is accomplished by applying the interface conditions described in Sec. 3.4.4 at these interfaces. The interface conditions along with the other types of boundary conditions are implemented at every stage. Thus, the flow field of the entire domain is computed for each stage, though the calculation is performed in each subdomain at a time. The configurations are ranked from simple to complex ones. Since the method has not been applied to the aerodynamics calculation, it is a good idea to test the method before applying it to the complex problems. This purpose is fulfilled by considering the flows about simple configurations. This is

because single grids can be constructed about them. Furthermore, the physics of flows about these configurations are relatively simple. Thus, solutions for flow about these configurations have also been obtained using the single grid systems with the procedure described in the previous chapter (finite-volume, etc.). Solutions from single grid calculations are also used as references to compare the solutions from multiple grids. The low speed flow over a sphere serves as the first application of the approach. Since the inviscid flow over a sphere at high speed does not give a realistic phenomena, flow over a slender body is, thus, considered next. Both flow over a sphere and flow over a slender body are discussed in Sec. 4.1. The capability of the conservative rezoning algorithm to handle grids of different topologies is demonstrated in Sec. 4.2 where the multiple grids system is used to compute the flows over a Butler-Wing configuration. Here, grid topologies in different subdomains are completely different. The supersonic flow through a rectangular duct with ten degree ramps considered in Sec. 4.3 demonstrates that strong discontinuities (shocks) can move from one subdomain to another without being distorted. Finally, the usefulness of the approach can be seen in Sec. 4.4 where it is used to obtain solutions for the internal/external flow about a fighter aircraft. In this case, the use of multiple grid system is necessary since it may not be possible to obtain a smooth continuity of grid lines at the interfaces between the interior and exterior grids. It may be disputed that the Euler equations are not suitable in some regions of the flow field considered in this study. Nonetheless, this study demonstrates that the difficulties in constructing the "boundary-fitted" coordinates about complex configurations can be overcome. Indeed, it is the main objective of this study to give such demonstration.

It is demonstrated in [103] that the conservative rezoning algorithm works well for grids with grid cells of comparative sizes. Furthermore, the algorithm yields smaller error when the information is transferred from a finer grid to a coarse grid.

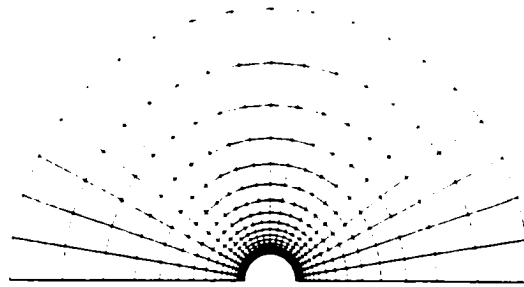
However, this arrangement is not feasible in many applications. It should also be noted that the algorithm can be used only at the planar interfaces. For a non-planar interface, the formula for computing the area of a polygon does not have the associative property as in Eq.(2.4). Thus, it would not be advantage to use it when sweeping through grid lines as done in Sec. 2.5. So far, a practical method for efficiently computing the conservative interface condition at the non-planar interface has not been discovered. Nevertheless, the planar interfaces can be found in many applications. Even with this restriction, the algorithm has been found to be quite useful in CFD applications. As discussed in the following sections, it brings CFD a step closer to reality.

4.1 Sphere and Slender Body

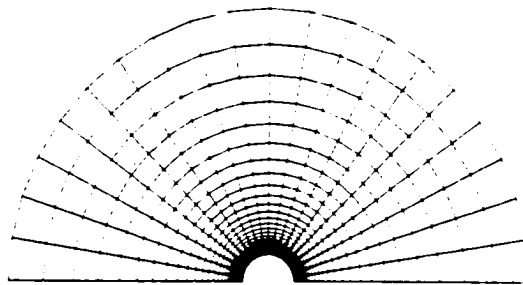
The inviscid flow over a sphere is chosen as the first application of the approach to the aerodynamic configurations. Since the approach has not been applied to a physical aerodynamics problem, the first application should be somewhat straight forward so that only difficulty lies in the treatment of the interface conditions. The simplicity of the sphere, both from grid generation point of view and flowfield calculation, gives a better understanding of the approach. However, the Euler equations model is not suitable for the high speed flow over sphere. This is because such a flow separates somewhere downstream and the Navier-Stokes equations have to be used. In order for the Euler equations to be applicable to high speed flow, and yet simplicity of the configuration is maintained, a slender body is also considered. The free stream Mach number for flow over sphere is 0.2 (nearly incompressible) while flow over the slender body has been investigated at a free stream Mach number of 1.5. Solutions for flows over the slender body have been obtained at zero-degree angle of attack. The entire flowfield is divided into two subdomains at about the center of the configurations. Grids in both subdomains are of

O-O type with different number of grid points, i.e., $21 \times 21 \times 17$ versus $33 \times 33 \times 17$. Figures 4.1 and 4.2 illustrate grids used for the sphere and slender body, respectively. These figures show grid lines at the symmetry plane and at the interfaces. Discontinuity of grid lines at the interface is clearly seen. Singular lines resulting from the O-O mapping type are also shown. As mentioned previously, they do not present a difficulty to the finite-volume method. Direction of the flows is also indicated in the two figures. As seen here, the flows move from the subdomain with a coarser grid to the subdomain with a finer grid. This should yield a larger errors than if they move in the opposite direction. This direction of flows is purposely chosen to demonstrate that the resulting errors do not considerably alter the solutions. As mentioned earlier, the purpose for the investigations of flow over these configurations is to see whether the approach is feasible when applied to the simple CFD calculation. Solutions to these problems can be obtained by using single grids. The use of single grids is probably more efficient. Thus, solutions from single grid calculations are used as references to compare the solutions obtained by multiple grids calculations.

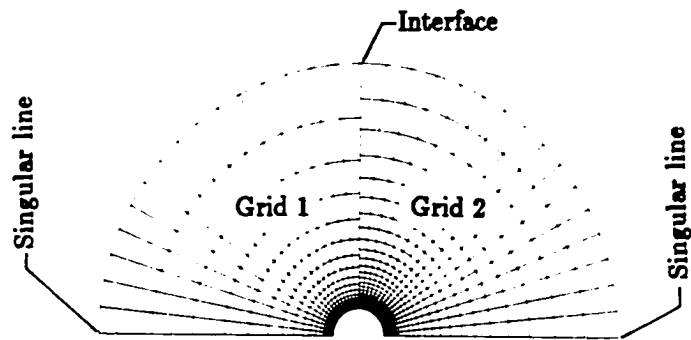
Wall pressure coefficients at the center line are plotted and compared in Figs. 4.3 and 4.4. Good agreement can be clearly seen from the comparisons. Moreover, these figures show all expected features of the flows. The plots of pressure coefficient on the sphere (Fig. 4.3) have the same shape as the corresponding incompressible inviscid (ideal fluid) flow [143]. The high pressure at the stagnation point in Fig. 4.4 indicates that bow shock is formed in front of the slender body and there is a small subsonic region behind the shock. The flow, then, becomes supersonic downstream as seen in the classical flow over a blunt body [144]. The flow is compressed at the rear of the body and shock wave is formed. Even though these comparisons are made for only simple cases, confidence in using the concept of the multiple grids is increased. Indeed, solutions to flow over these configurations



(a) Grid 1



(b) Grid 2



(c) Symmetry plane

Fig. 4.1 Grid system for a sphere.

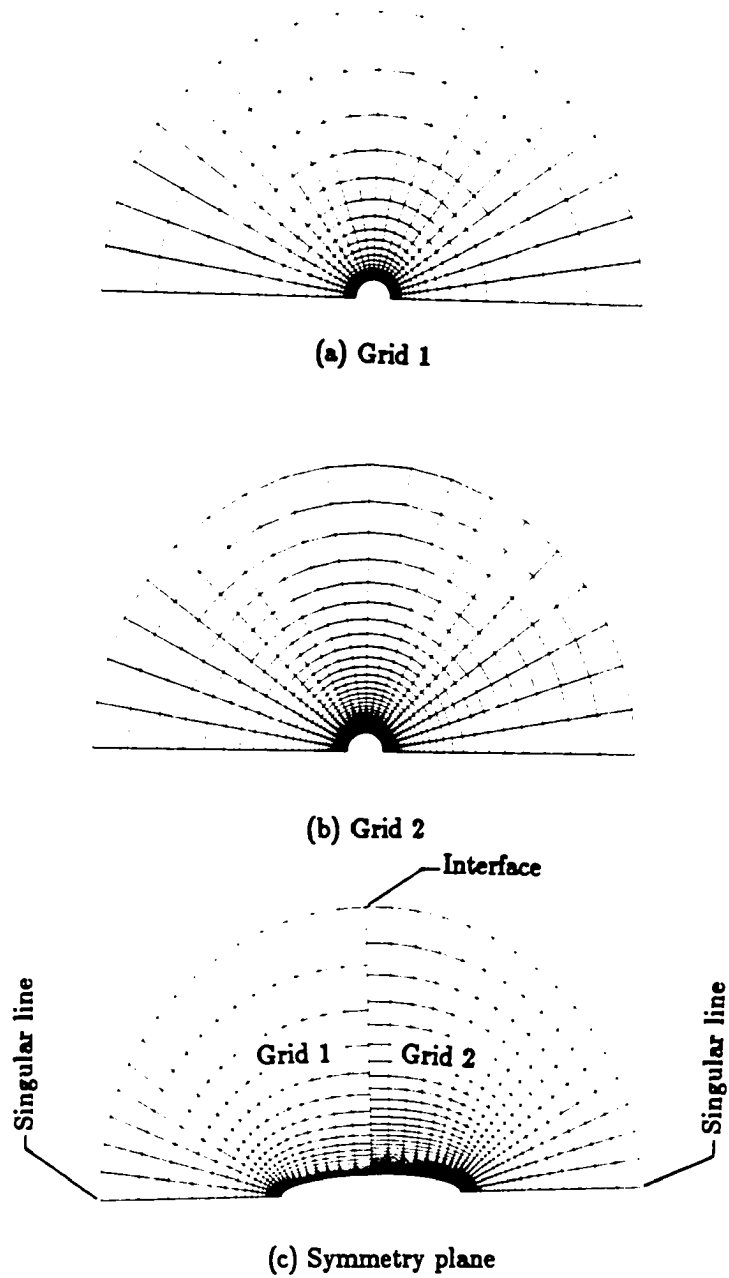


Fig. 4.2 Grid system for a slender body.

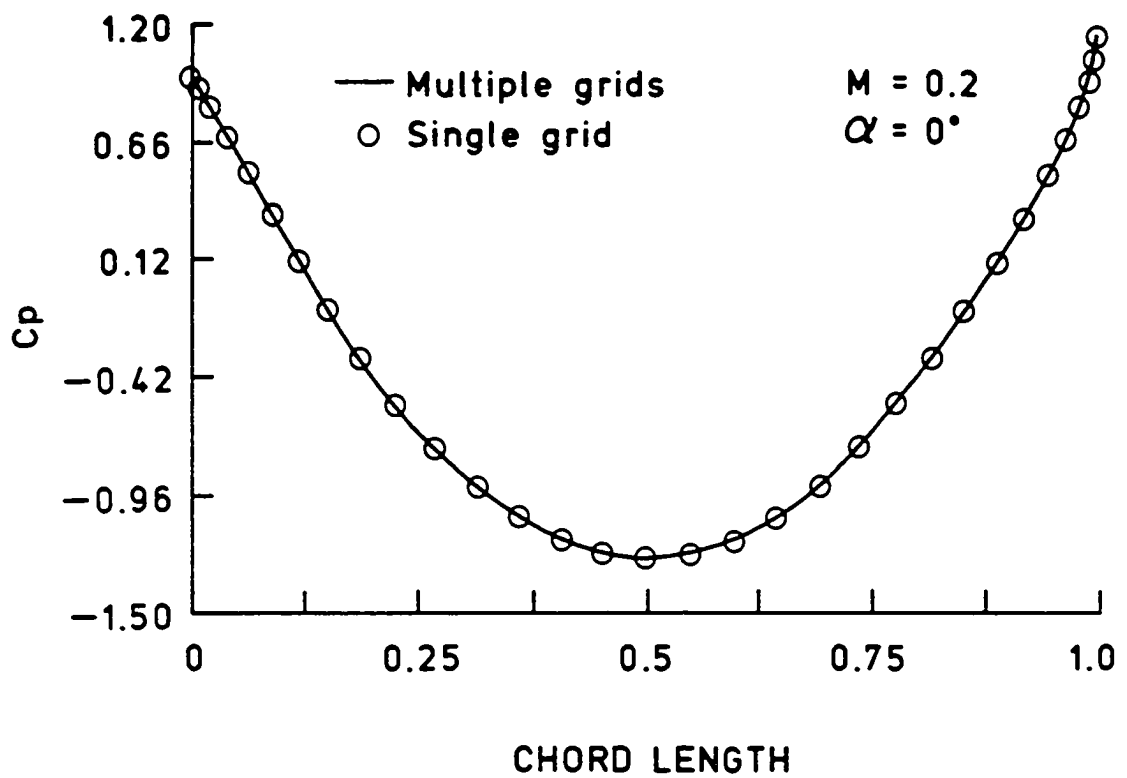


Fig. 4.3 The pressure coefficient on the surface of a sphere.

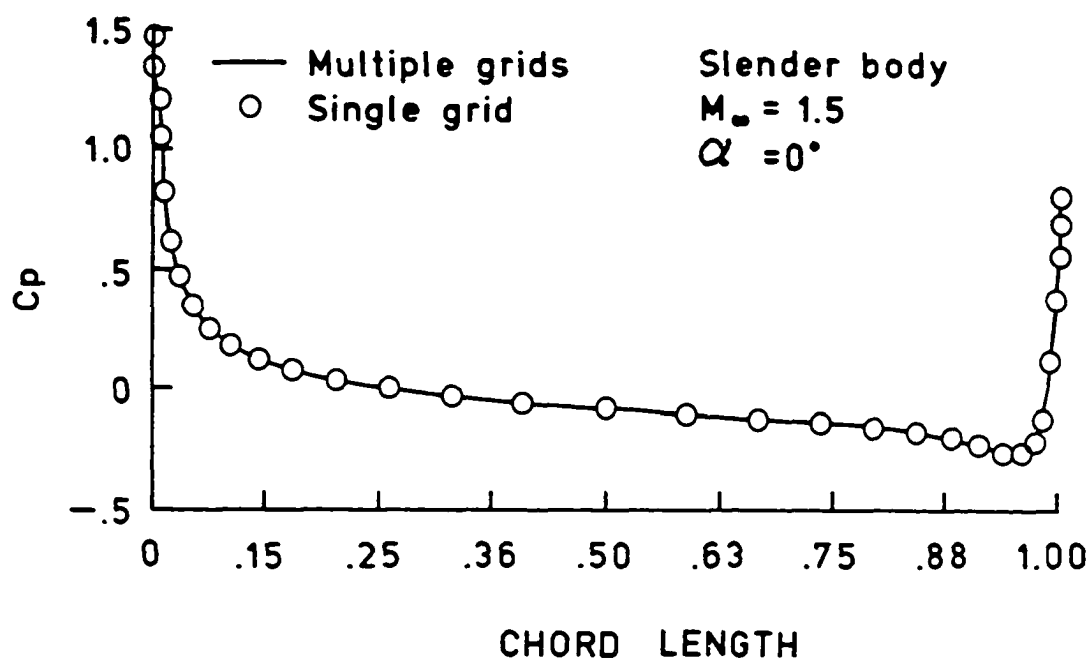


Fig. 4.4 The pressure coefficient on the surface of a slender body.

can be obtained for various flow conditions and number of grid points. The same agreement is expected from these results. However, it is felt that problems with higher level of difficulty should be pursued. Thus, the use of different grid topologies for different subdomain grids is considered next. A Butler-wing described in the next section suits this purpose well.

4.2 Butler Wing Configuration

A Butler wing is a delta wing which was proposed by D.S. Butler [145]. The planform of the body is an isosceles triangle, and the leading edges of the wing lay along the Mach lines of the unperturbed stream. The first 20% of the wing is conical and the last 80% of the wing has elliptical cross sections with increasing eccentricity along the x-axis. At the trailing edge, the elliptic cross section has infinite eccentricity and the last cross section is a straight line. Figure 4.5 shows a physical model of a Butler-Wing. The semi major and minor axes are given by

$$\begin{aligned}
 \text{major - axis} &= \frac{x}{\beta} & ; & \quad 0 \leq x \leq L \\
 \text{minor - axis} &= \frac{x}{\beta} & ; & \quad 0 \leq x \leq 0.2L \\
 &= \frac{x}{\beta} \left[1 - \left(\frac{x - 0.2L}{0.8x} \right)^4 \right] & ; & \quad 0.2L \leq x \leq L
 \end{aligned} \tag{4.1}$$

where $\beta^2 = M_\infty^2 - 1$.

Butler [145] has compared the experimental results for surface pressure with the theoretical results using the slender-body theory approximation to simplify the inviscid equation of motion. Squire [146,147] has obtained experimental results for a Butler-Wing with varying Mach number and angle of attack. Abolhassani et al. [148] have obtained numerical solutions on a Butler-Wing by solving Navier-Stokes equations with the MacCormack time-split method. It should be mentioned that the experimental model is 120 mm. long and is constructed for $M_\infty = 3.5$. That is the semi-apex angle of the planform and of the initial conical nose is $\sin^{-1} \frac{1}{3.5} = 16.602^\circ$.

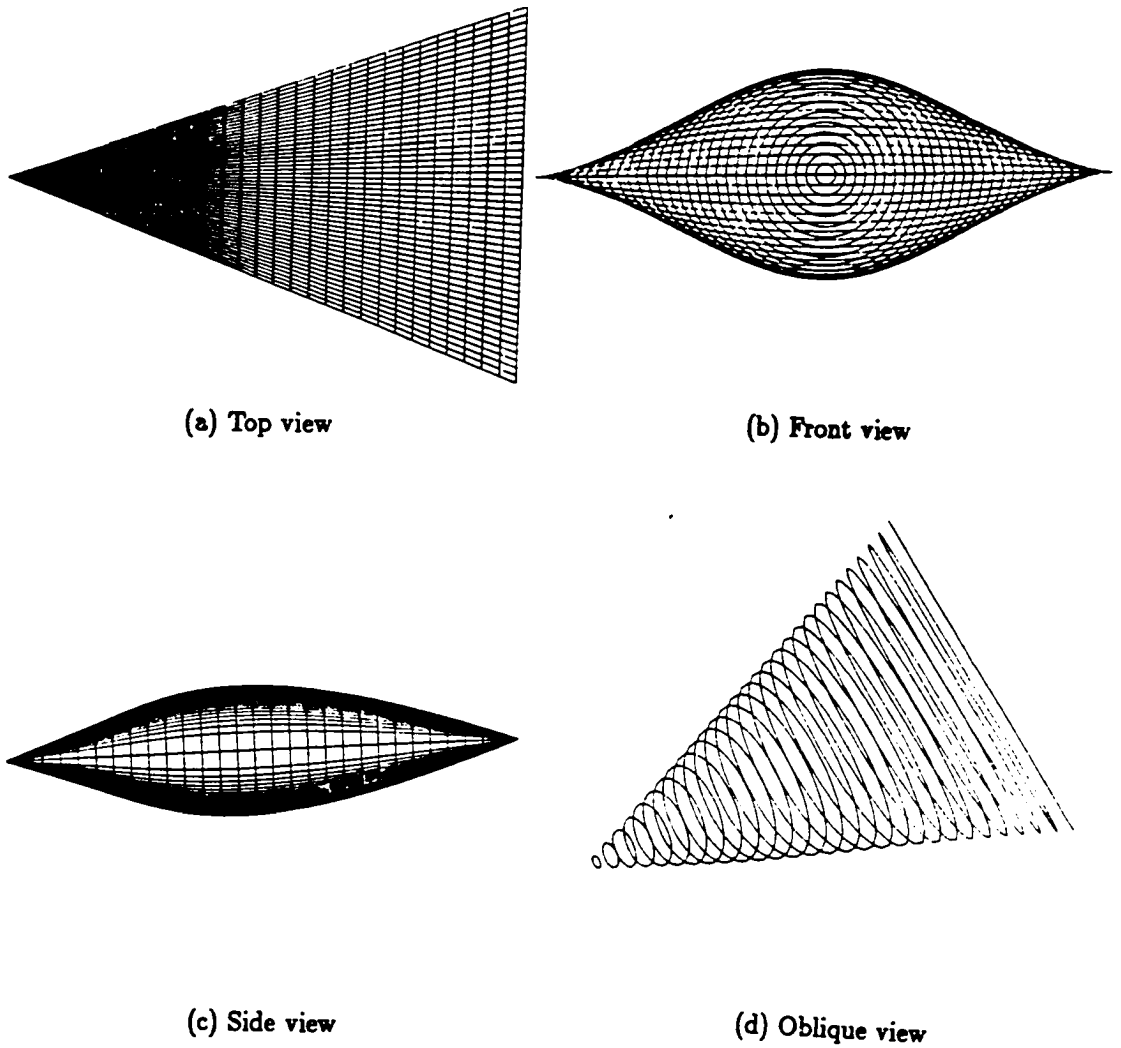
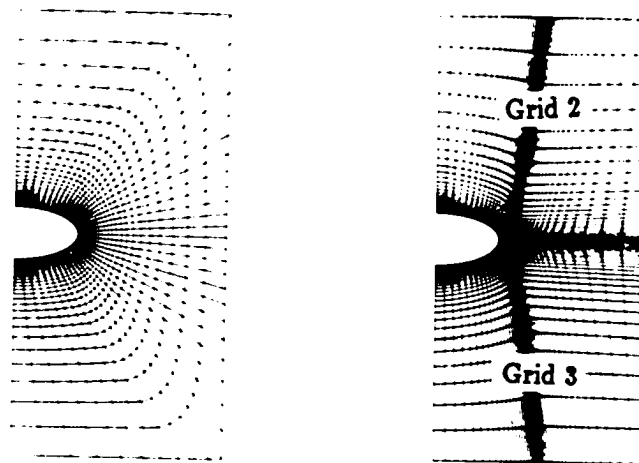


Fig. 4.5 Physical model of a Butler-Wing.

The model is mounted in the tunnel by attaching a sting support of 12.5 mm. diameter to the lower surface.

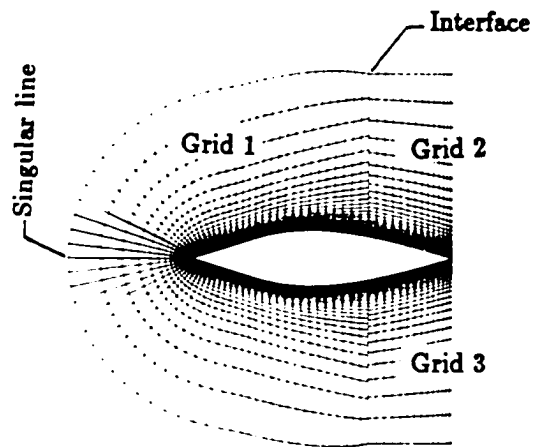
Even though solutions can be obtained using a single grid system, multiple grids solutions on a Butler-Wing suit well for the purpose of this study. This is because a grid of O type is suitable for the front part of the configuration while a H-type grid seems to be a better choice near the trailing edge. The multiple grids system is composed of three subdomain grids with $41 \times 41 \times 35$, $41 \times 41 \times 15$, and $41 \times 41 \times 15$ number of grid points, as shown in Fig. 4.6. The H-type grid at the rear of the configuration is divided into two subdomain grids mainly to avoid programming difficulty. The O-type grid has a singular line emanating from the nose of the configuration as indicated in Fig. 4.6 a. Figures 4.6 b. and 4.6 c. clearly illustrate no continuity or even similarity of grid lines in each subdomain at the interface. The Butler-Wing configuration illustrates the capability of the multiple grids approach to deal with the problem where grid topology is changed from one subdomain grid to another.

The wall pressure coefficients at various flow conditions are plotted in Figs. 4.7-4.11. In all figures, a solid line indicates the wall pressure coefficient obtained from multiple grids calculations. The wall pressure coefficient along the center line are shown in Fig. 4.7 for 3.5 Mach number and zero degree angle of attack. A High pressure at the front of the configuration indicates that an oblique shock wave is generated from the nose of the configuration. As expected, the pressure is constant over the conical section. Further downstream, the flow is expanded as seen from the decrease of pressure with the increase of the eccentricity in the elliptical section. In Fig. 4.7, comparisons are made with results of Refs. 145-148. Discrepancies near the nose region exist because the grid is not fine enough to obtain the correct conical solutions there. Discrepancies at the rear of the wing are believed to occur because of the negligence of viscous effect in the Euler equations. Squire [146]



(a) O-Grid

(b) H-Grid



(c) Symmetry plane

Fig. 4.6 Grid system used for a Butler-Wing.

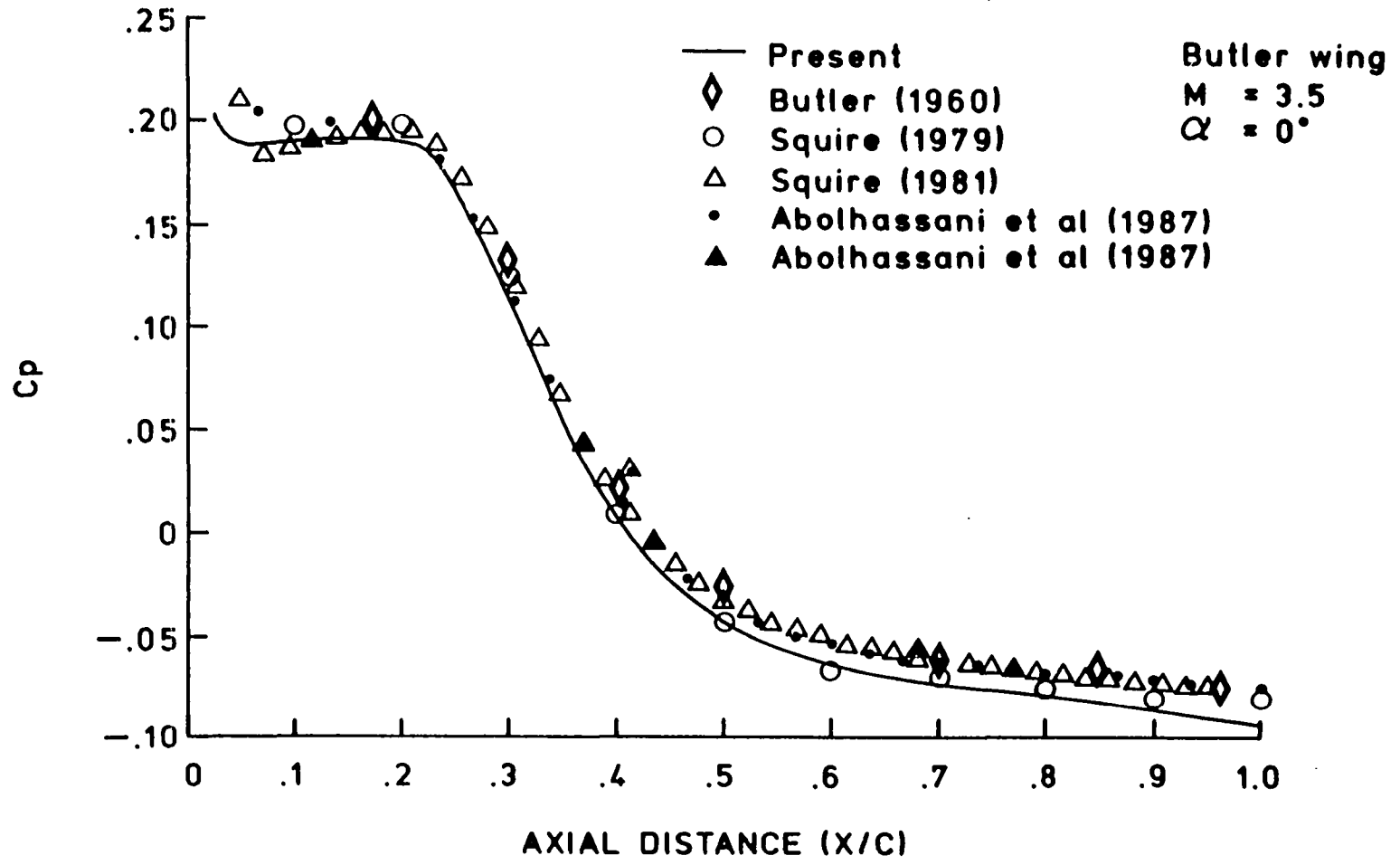


Fig. 4.7 The pressure coefficient on the surface of a Butler- Wing ($M = 3.5, \alpha = 0^\circ$).

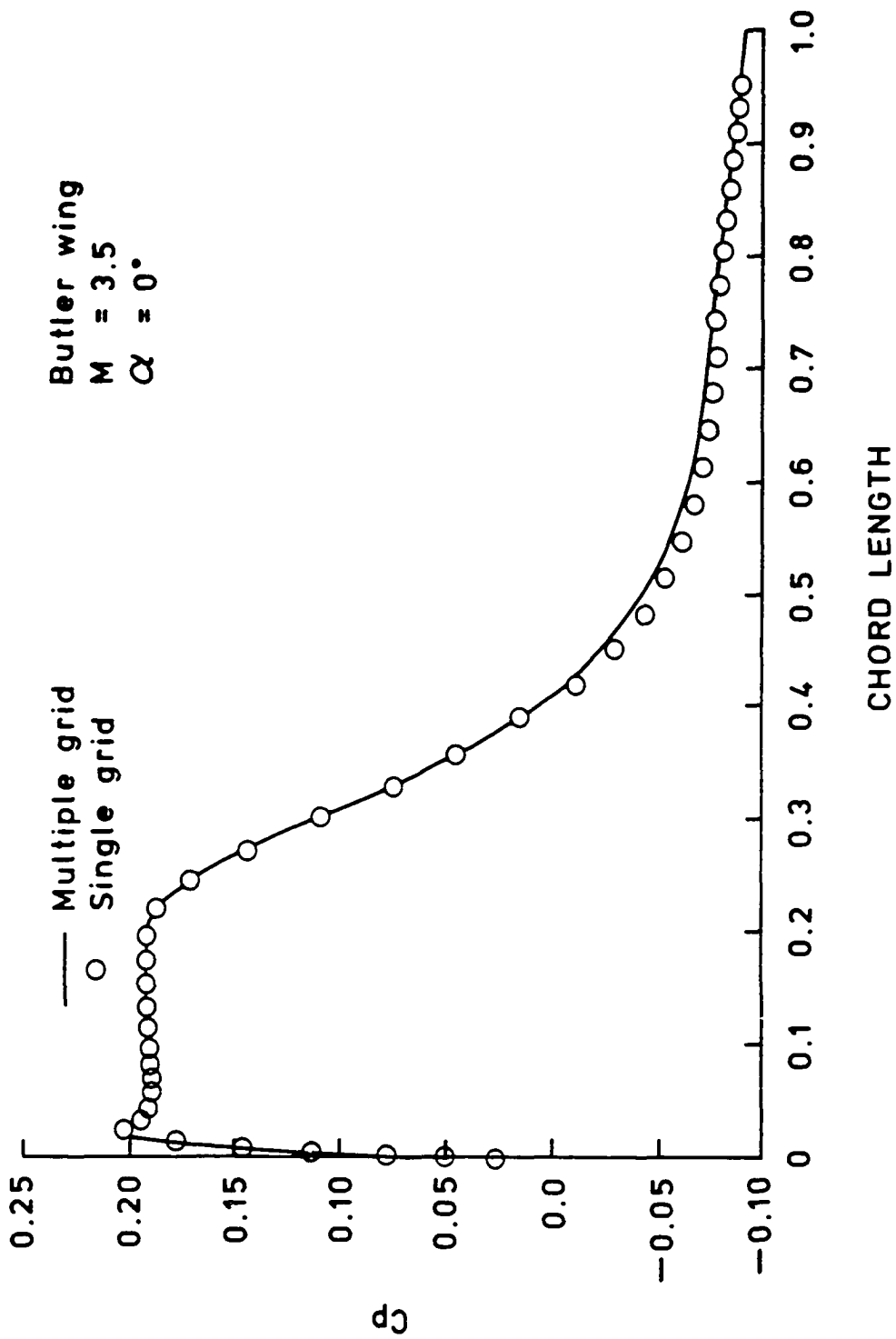
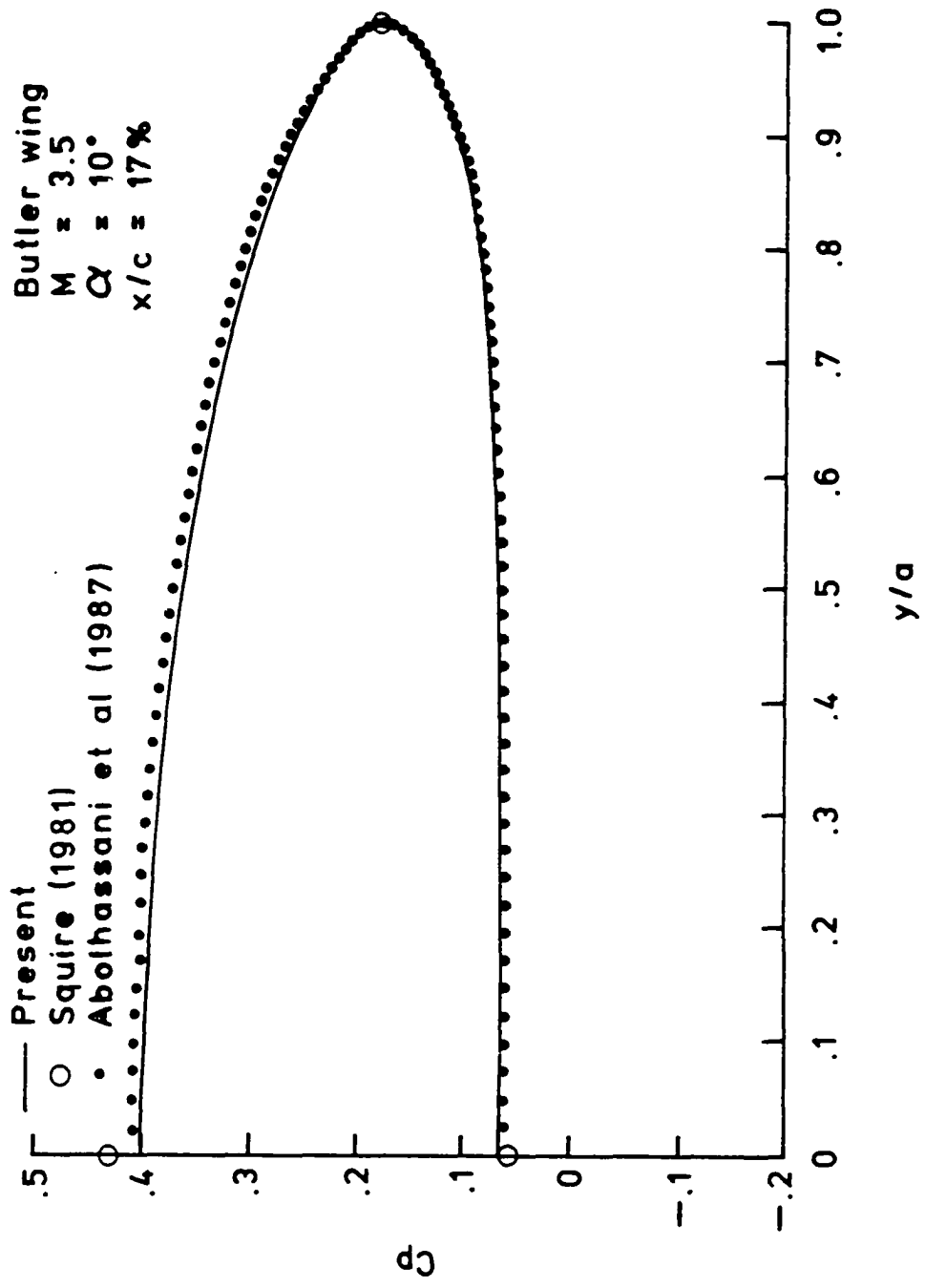


Fig. 4.8 Comparison of C_p with the single grid calculation.

has made similar conclusion about these discrepancies. Good agreement for the comparison between multiple grids solutions and solutions obtained from a single grid calculation is shown in Fig. 4.8. This indicates that the use of multiple grids does not cause the discrepancies in Fig. 4.7.

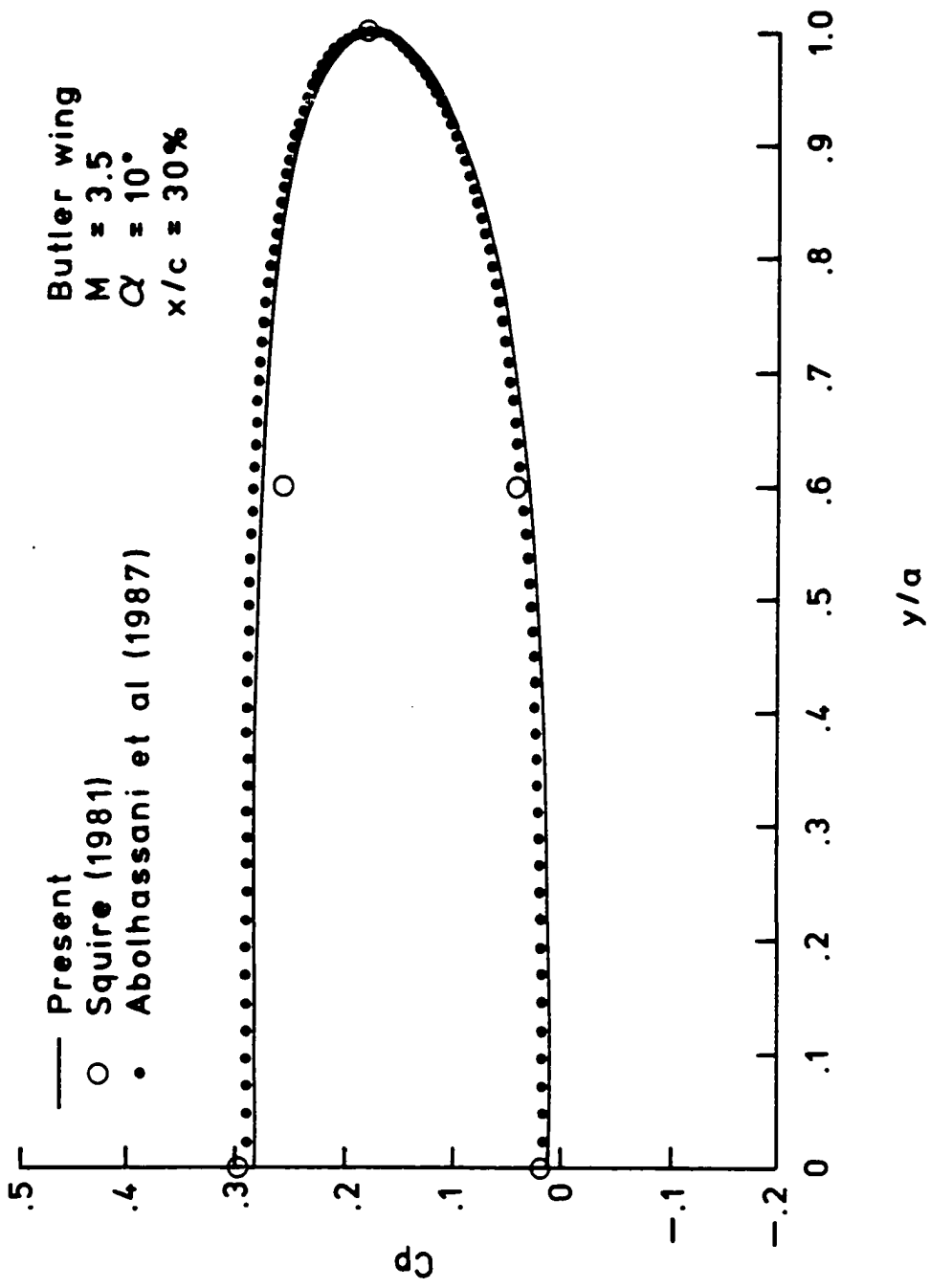
The computed pressure coefficients for ten degrees angle of attack are plotted and compared with the results of Refs. 146 and 148 in Fig. 4.9. At 17%, 30%, 50%, and 70%, chordwise positions, the pressure coefficients are plotted against the conical spanwise coordinate $\frac{y}{x \tan \phi}$. Good agreement can be seen. On the thick sections near the nose, the pressure is highest on the centerline and falls toward the leading edge, whereas near the trailing edge the spanwise distribution is more 'wing like' with the maximum pressure at the leading edge. The changeover is shown by the pressure peaks in the pressure distributions at $\frac{x}{c} = 0.5$ and 0.7 (Fig.4.10). Some discrepancies with the experimental results may be due to the fact that the lower surface of the experimental model is distorted to include a sting support. Results for flow at 2.5 Mach number are shown and compared with results from Ref. 146 in Fig. 4.11 and 4.12. Figure 4.11 shows results at zero degree angle of attack, whereas, results at ten degrees angle of attack are shown in Fig.4.12. The same conclusions as in the case of 3.5 Mach number flow can be made for this case also.

The results obtained, thus far, demonstrate that the use of multiple grids approach is plausible and does not add significant error to the flow model equations even when grid topologies in subdomain grids are completely different. However, more complex problems should be investigated in order to be certain about the capabilities of the approach. The study, thus far, may not yet indicate the usefulness or necessities of the multiple grids approach, since the construction of a single grid system can be made in all cases. In some applications, however, the construction of such a single grid system to cover the entire domain may not be possible at all.

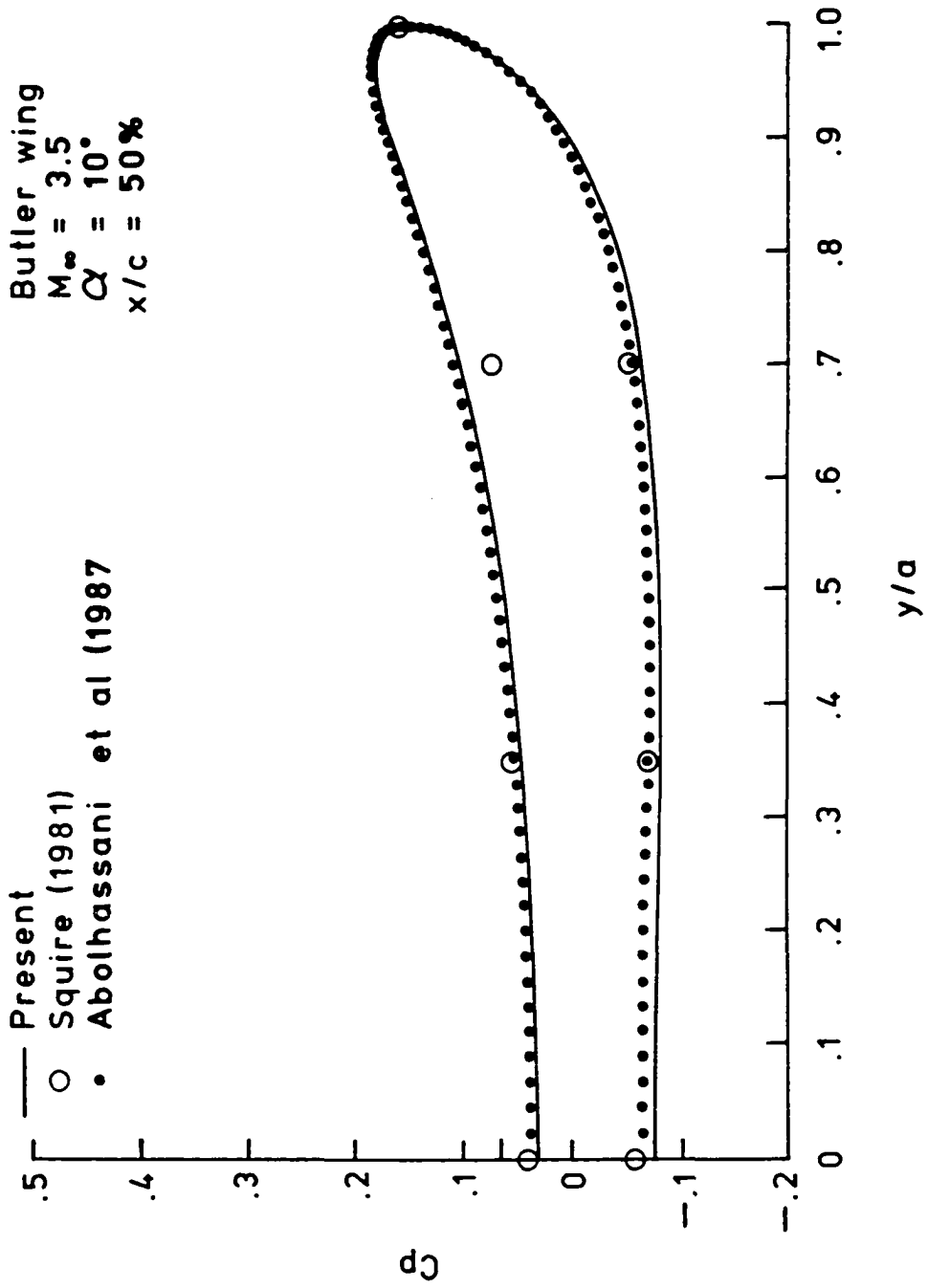


(a) $x/c = 17\%$

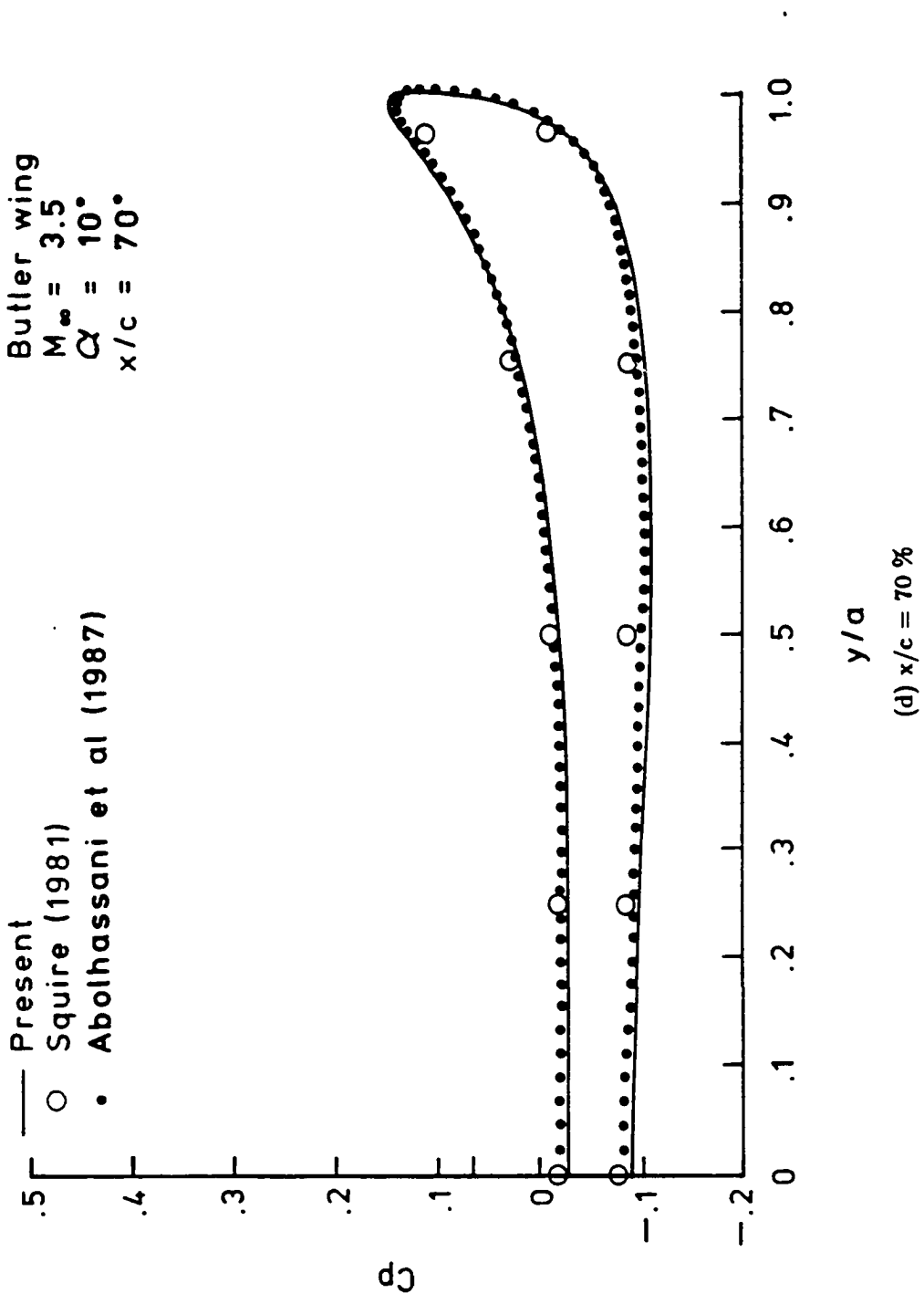
Fig. 4.9 C_p on the surface of a Butler-Wing at different cross cut ($M = 3.5, \alpha = 10^\circ$).



(b) $x/c = 30\%$



(c) $x/c = 50\%$



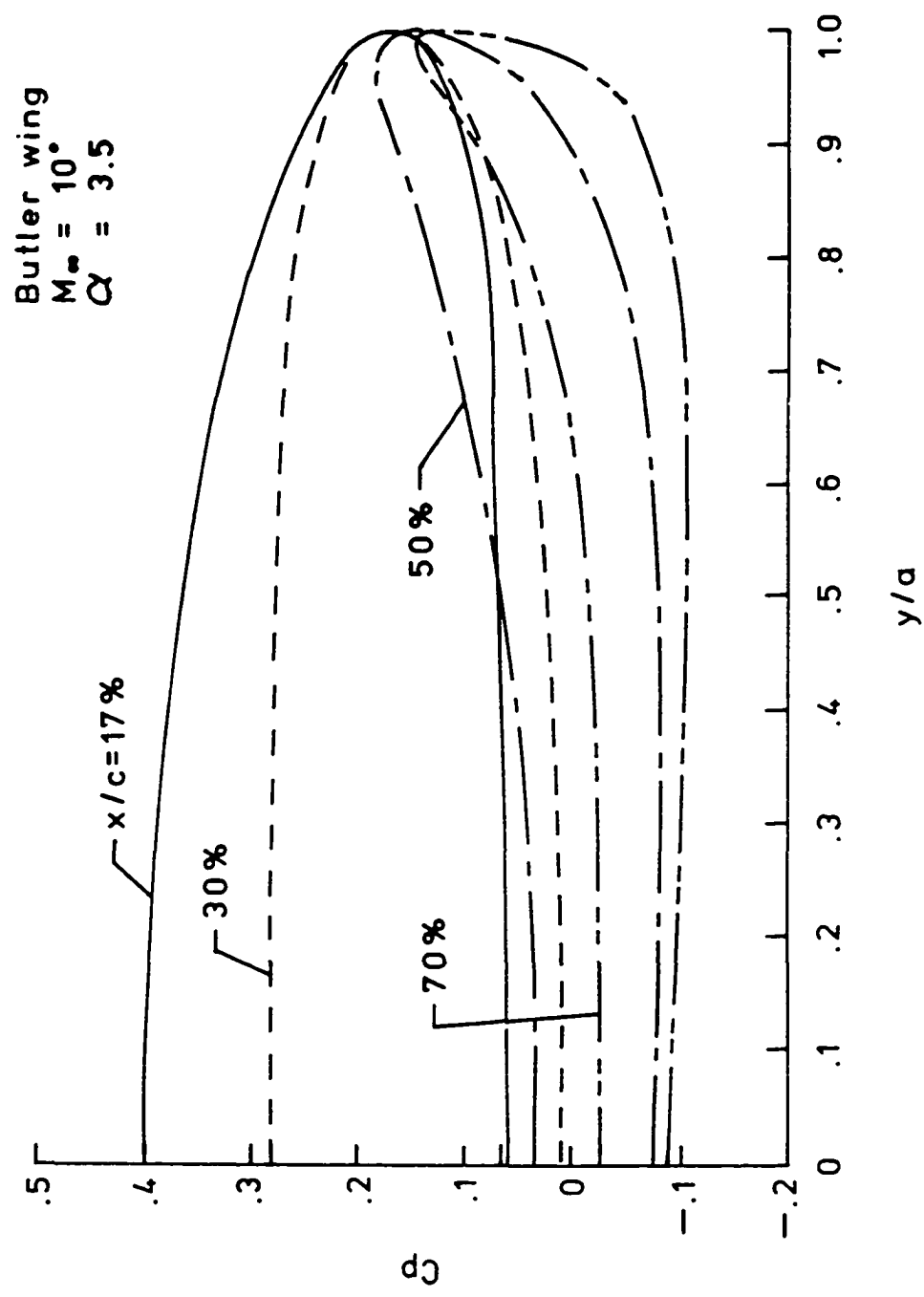


Fig. 4.10 Pressure distribution for a Butler-Wing ($M = 3.5, \alpha = 10^\circ$).

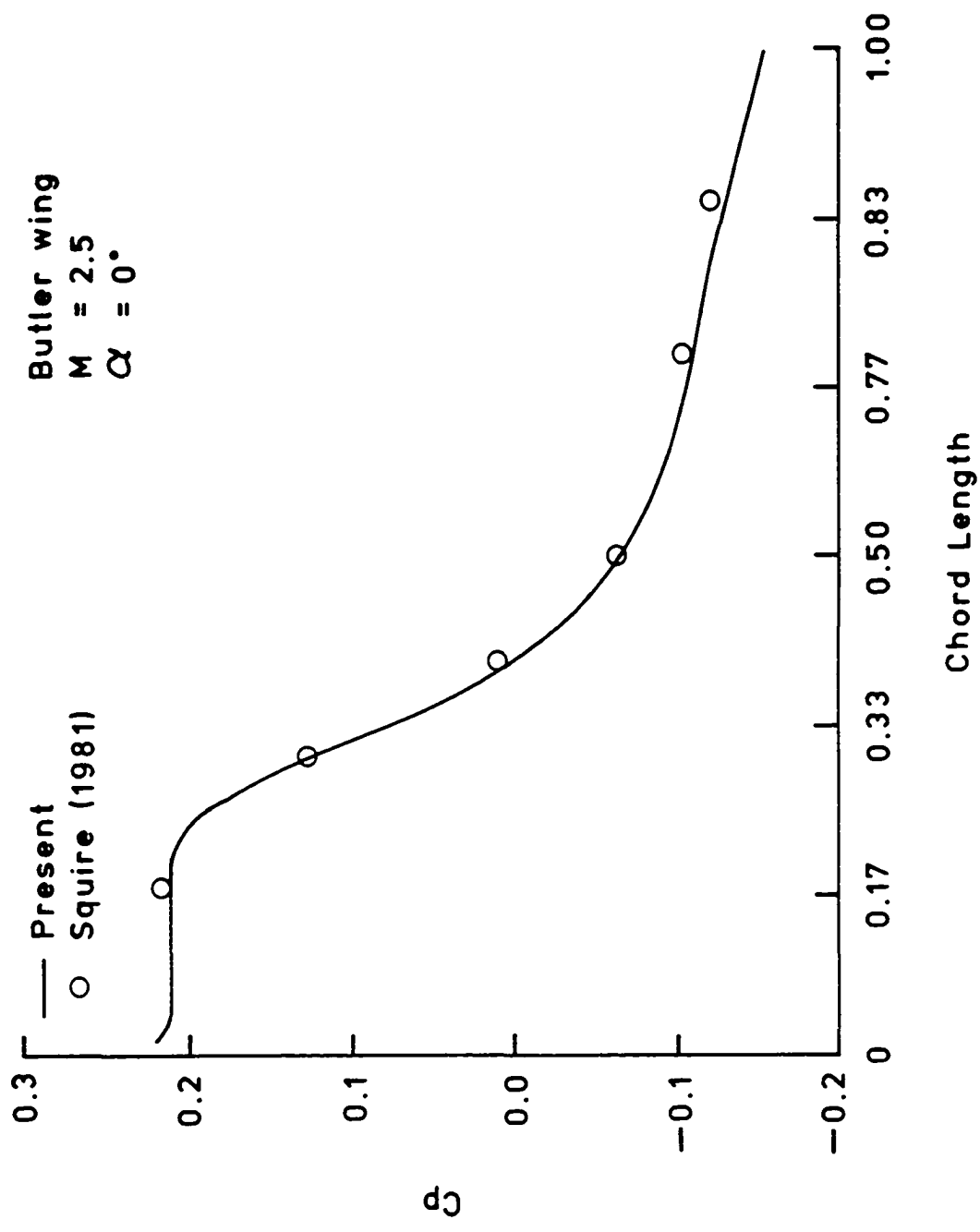


Fig. 4.11 C_p on the surface of a Butler-wing ($M = 2.5, \alpha = 0^\circ$)

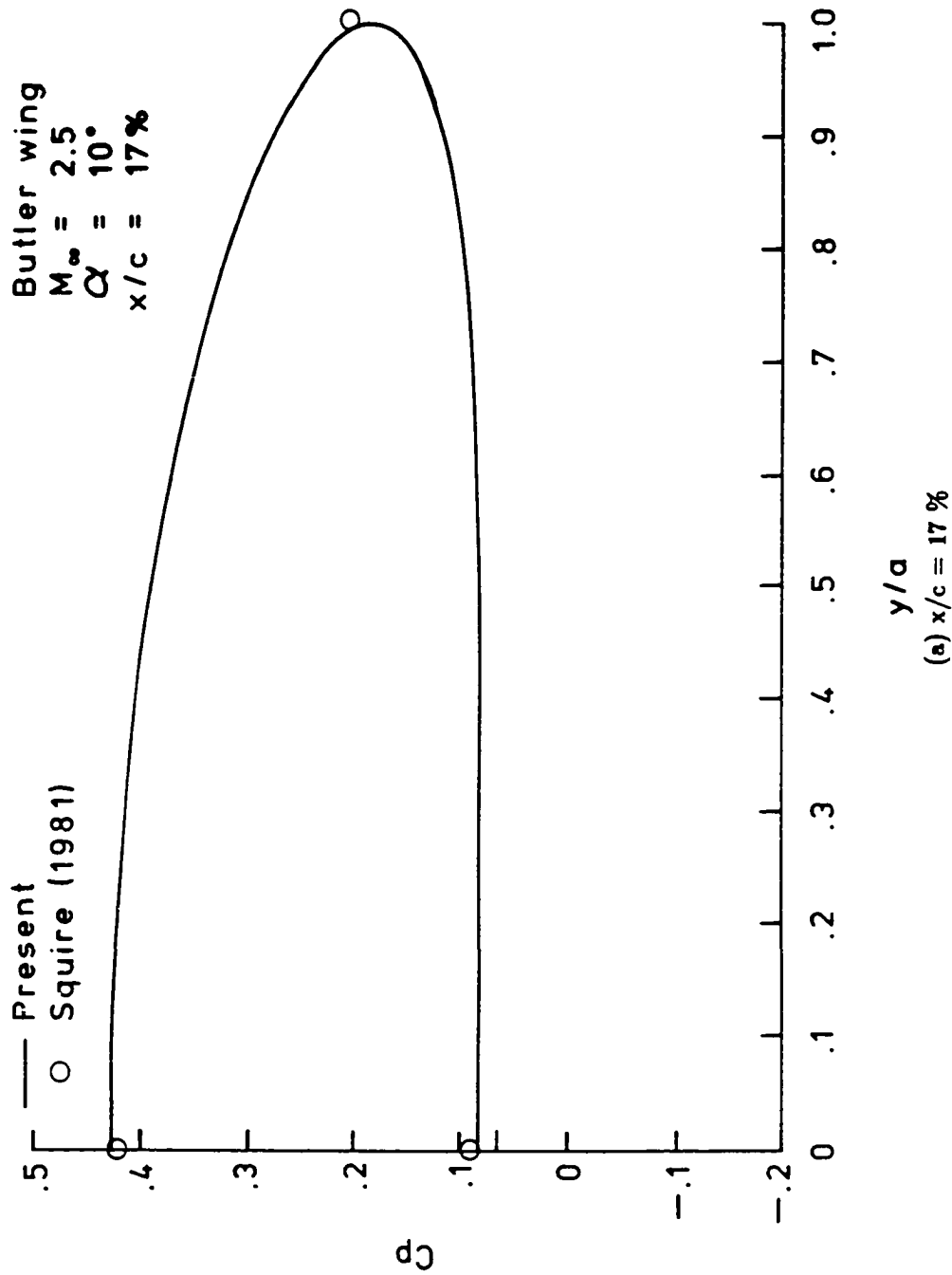
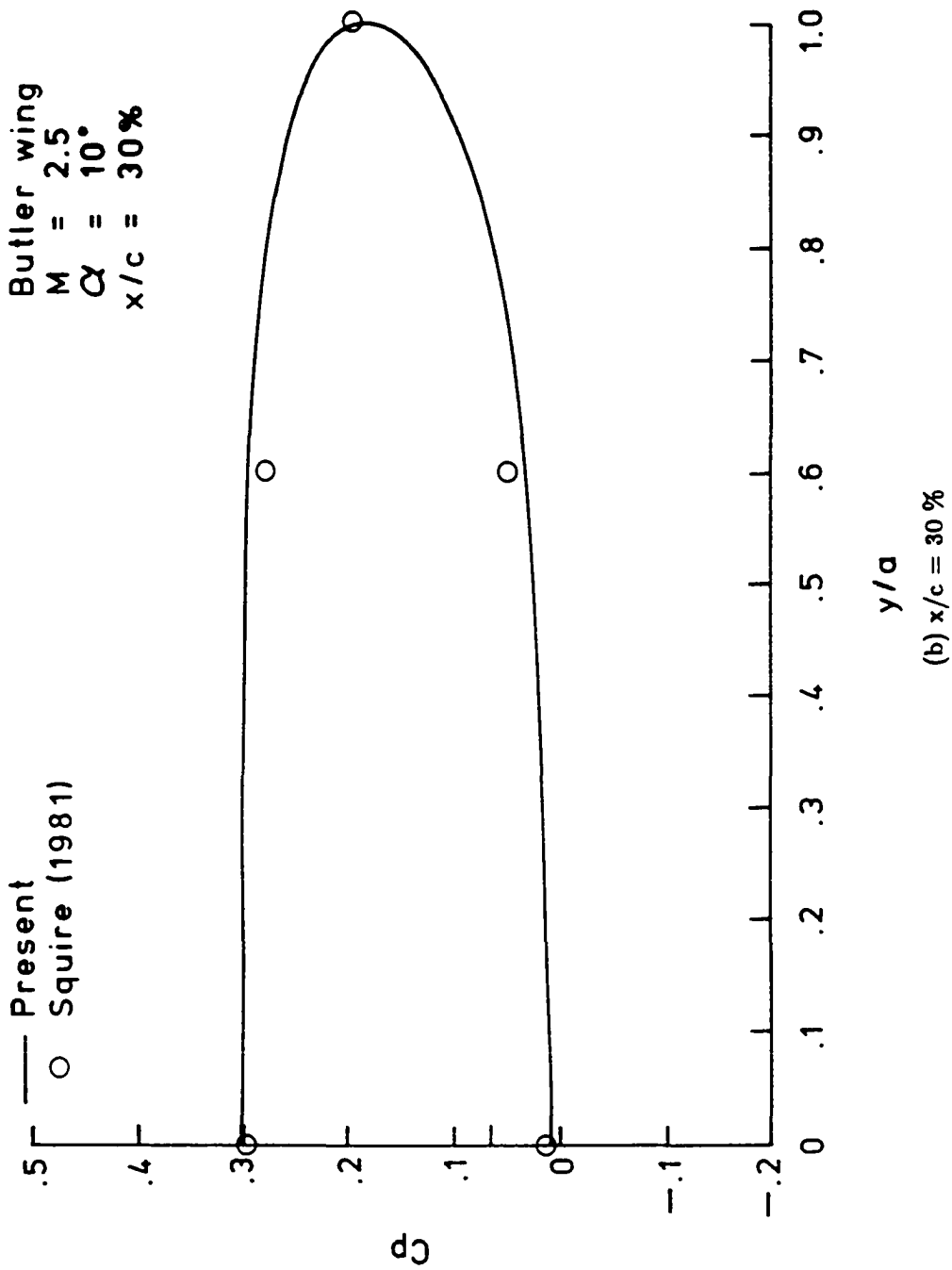
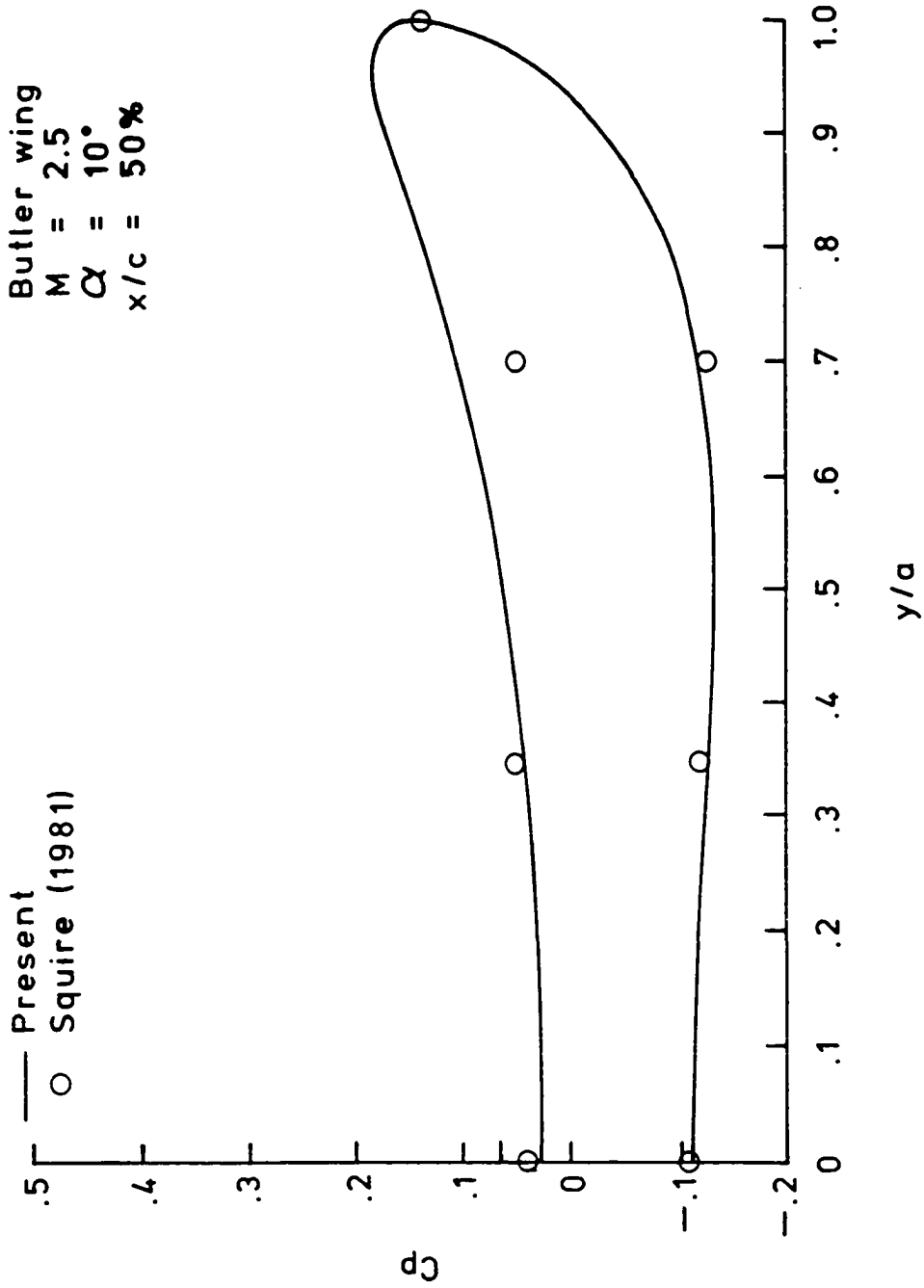
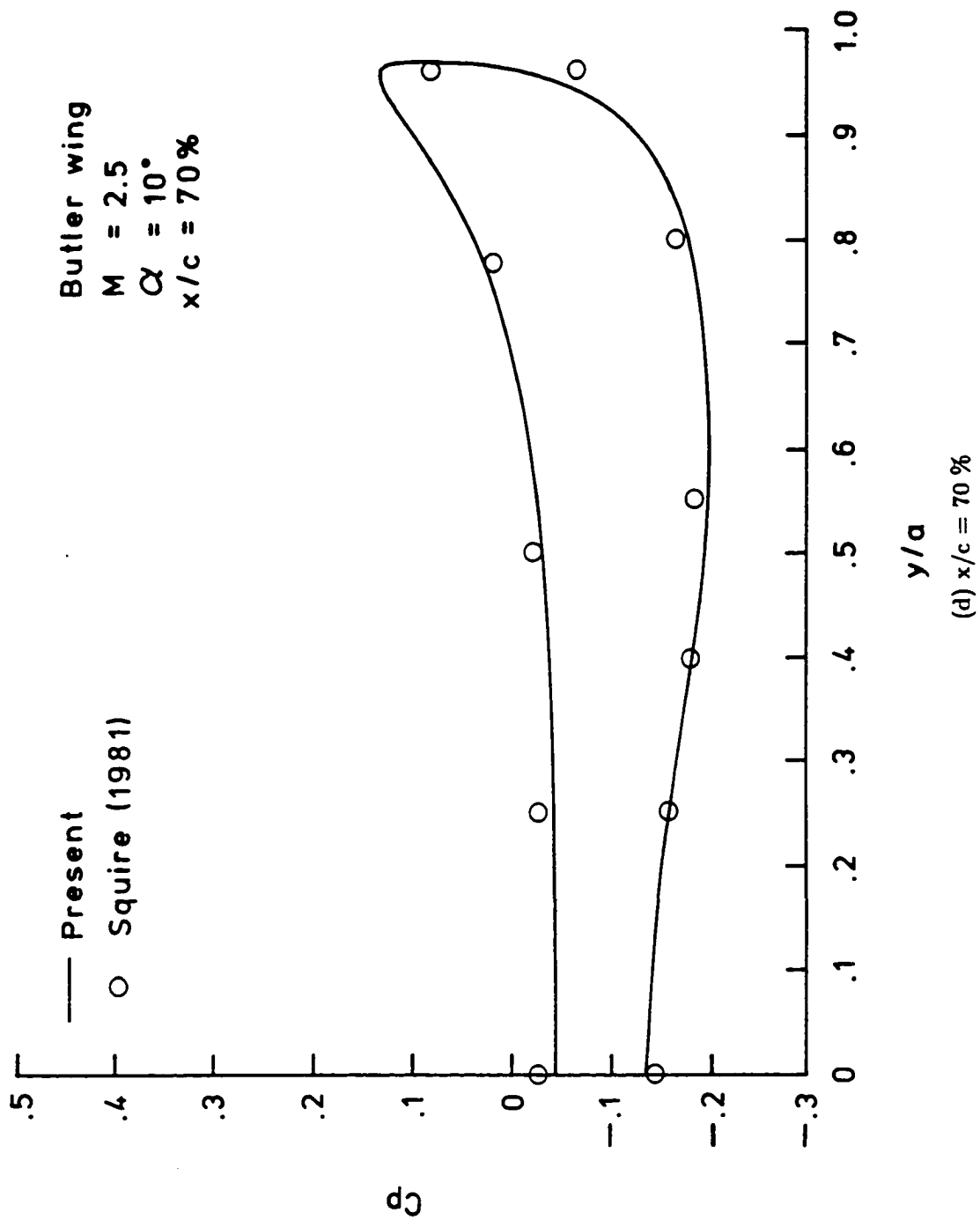


Fig. 4.12 C_p on the surface of a Butler-Wing at different cross cut
 ($M = 2.5, \alpha = 10^\circ$)







4.3 Rectangular Duct with 10^0 Ramps

Thus far, the multiple grids approach has been applied only to external flows over some aerodynamic configurations. Originally, it was not intended to apply it to obtain a solution of flow inside a rectangular duct. This is because a single grid can be constructed easily to model the flow. Moreover, the physics of the flow is relatively simple. However, some consideration regarding internal flow is needed in order to apply the approach to the internal/external flow interaction (Sec. 4.4). Thus, a supersonic flow inside a rectangular duct was chosen in order to verify the modified finite-volume code. This code has been modified from the original code which was written to solve the Euler flows external to some configurations. Two 10^0 ramps have been added to the front of the straight duct to compress the flow and create shock waves, since just a straight duct would not produce any interesting phenomena. Supersonic flows inside this duct have been simulated on a single grid. However, solutions would not converge to steady state if the solid boundaries where the ramps meet the straight portion are not smooth, no matter how smooth the interior grid lines are. Even though the problem could be solved by smoothing out this region or by some other means, it was suspected whether the use of multiple grids would also be an alternative. As a matter of fact, this problem becomes another good case to verify the technique. This is because the physics of the problem is well known. Moreover, shock waves generated by the ramps must go through the interface which is located where the ramps turn into a straight duct. It is seen from this case that the interface can tolerate these shocks without disturbing them.

Figure 4.13 shows the plane of symmetry and a solid wall of the multiple grids system used in this case. Here, the entire domain is divided into two different subdomains, one covers the ramp portion and another covers the straight portion of the duct with $26 \times 49 \times 11$ and $21 \times 41 \times 41$ number of grid points respectively.

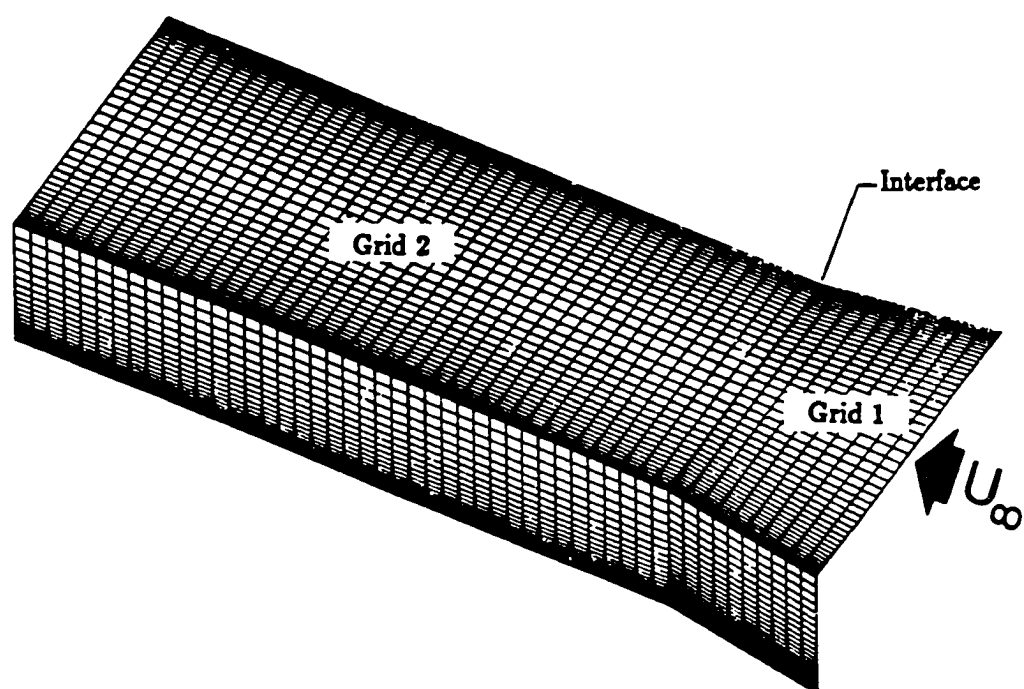


Fig. 4.13 Grid system for flow through a duct.

Both grids are also concentrated in the region near the solid walls. Since grids in each subdomain contains different numbers of grid points, there is no continuity of grid points between them at the patched interface. It should be noted that grid points or even grid slope could easily be made continuous at this interface. Then, the use of the conservative rezoning algorithm would not be necessary and this investigation would be meaningless. Solutions have been obtained for the supersonic flow with various freestream Mach numbers and number of grid points. Since all of the solutions produce a similar phenomena, only the case of freestream Mach number of 2 with the multiple grids system mentioned above is discussed here. Figure 4.14 shows contours of pressure coefficient at the plane of symmetry of the duct. The contours display all expected features of the flow such as shock waves emanating from the ramps, expansion waves generated at the corners where the ramps turn into straight duct. Shock waves meet at the center line about the location of the interface. These shocks reflect off each other and bounce off the walls, then meet again and so on. The pressure coefficients at the symmetry plane are illustrated in Fig. 4.15 along the upper wall and along the center line of the duct. All the features displayed in these figures are identical to those observed on the plots of solutions from single grid calculation. The results from this case indicate that the use of multiple grids along with the conservative rezoning algorithm can accommodate shock waves which pass through the interface without altering the solution. Moreover, it has been demonstrated that the multiple grids approach can avoid problems (convergence to steady state in this case) which may result from the use of just a single grid.

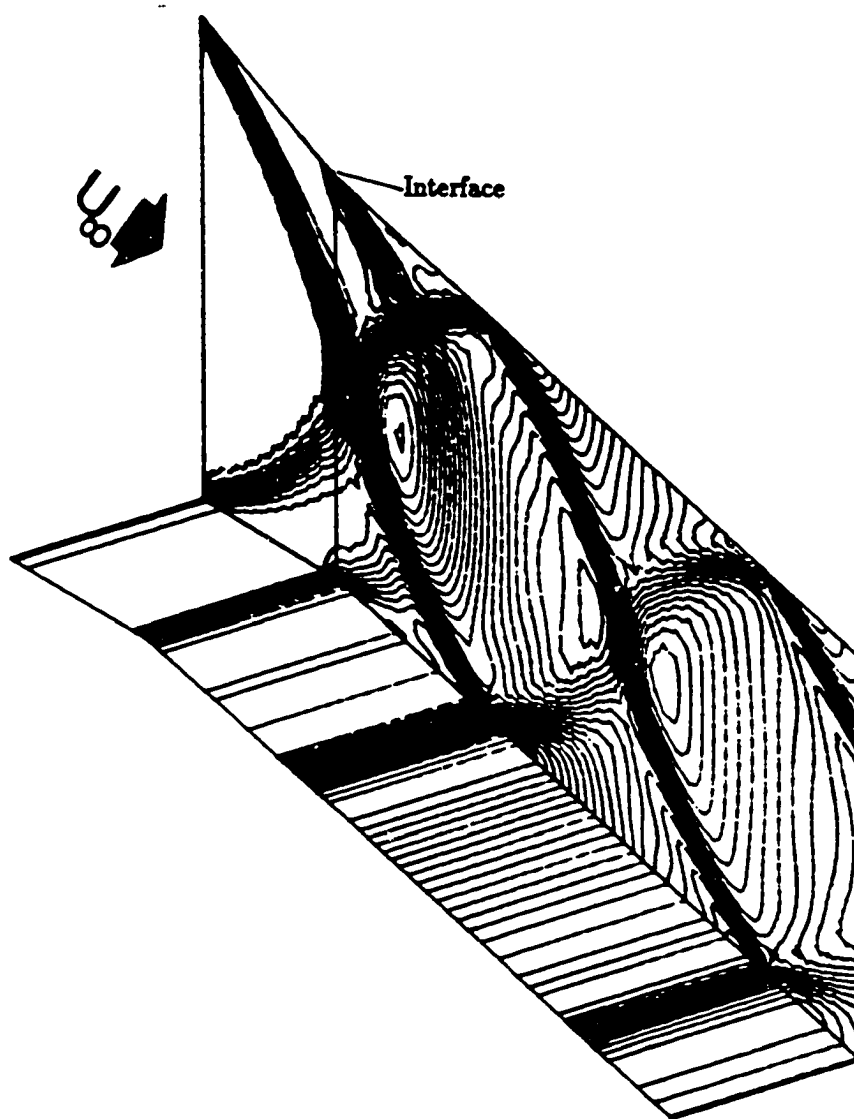


Fig. 4.14 C_p contours inside a duct.

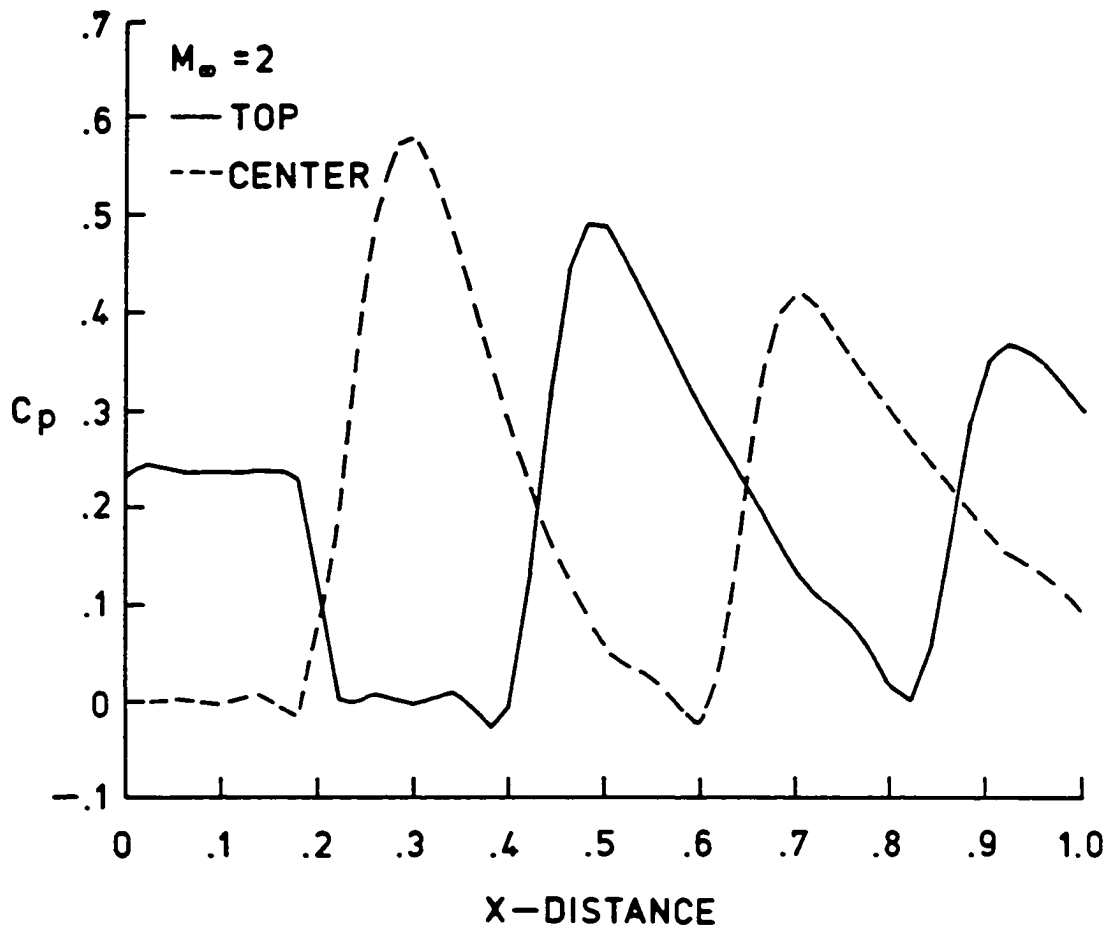


Fig. 4.15 Pressure distribution inside a duct.

4.4 Internal/External Flow About a Fighter Aircraft Configuration

The applications, so far, have been to flows about simple configurations where single grids could be used in each case. The next step is to consider the case in which the use of single grid to cover the entire domain may not be possible. In the recent years, significant improvements in numerical methods and computer technology have made it possible to solve inviscid compressible flow using the Euler equations for moderately complex geometries such as the wing-fuselage [149] or wing-canard [122] configurations. The use of multiple grids approach for such geometries are becoming popular, see for example Refs. 150 and 151. The approach is used in this study to investigate the internal/external flow about a fighter-like aircraft configuration (Fig. 4.16). The external flow over the same configuration has been studied by Eriksson et al. [152]. In Ref.152 the farfield boundary conditions are implemented at the inlet intake to simulate flow into the inlet. This study is an extension of the work available in Ref. 152. In this case, fluids are allowed to flow into the engine inlet and exit at the rear. Here, the interfaces between the interior and exterior grids do not require any continuity regarding grid density or slope (it may be even impossible to enforce any continuity at these interfaces). Flows over fighter aircraft configurations have been studied by several authors. For example, patched grid concepts have been recently used by Karman et al. [153] for a F-16 fighter configuration and by Fritz and Leicher [154] for an EFA-fighter configuration. However, a large number of grid blocks are used in both applications. The exterior grid topology used here is completely different in order to minimize the number of grid blocks and the amount of grid skewness. Some ideas regarding grid



Fig. 4.16 A fighter aircraft configuration.

topology of the exterior grid which have been discussed in [152] are also given here for completeness.

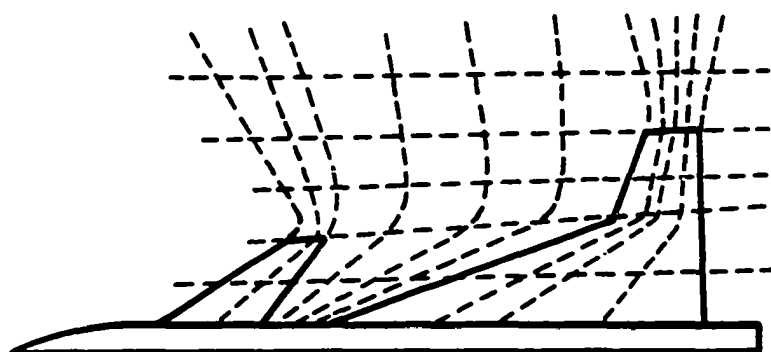
Steady state solutions for supersonic flows have been obtained at a free stream Mach number of 2. The method, however, works equally well for subsonic, transonic, and other supersonic flow conditions. Brief descriptions of steady state solutions for flows at various angles of attack are presented. For flows without the wake region (region behind the aircraft), results are obtained for 0° , 3.79° , 7° , and 10° angles of attack. For the case of flow with the wake region, results are obtained for a 0° incidence. The computed flows are visualized in terms of pressure coefficient contours. Although no experimental results are available, the results shown here demonstrate that the finite-volume scheme has no difficulties in dealing with singular lines. Furthermore, the interfaces between the exterior grids are completely "invisible" to the flow, due to the enforced continuity of all first-order metric quantities. Smooth variations of the pressure contours at the interface between the interior and exterior grids are also shown. The results demonstrate that the method is capable of computing internal and external flows simultaneously even though interfaces between interior and exterior grids do not require any continuities in terms of the grid slope or grid points.

It should be noted that the boundary condition at the inlet intake in [152] are implemented as farfield conditions (zero order interpolation). Solutions in [152] are compared also with the corresponding results of a potential method [155] and of Euler solutions [156]. Solutions of the present study are compared with corresponding results from Ref. 152. The comparisons indicate the influence of flow inside the engine inlet and in the wake region on the flow around the airplane. It should also be pointed out that the exact description of the engine inlet is not modeled since it is not available. Here, the inlet description is modeled by applying the transfinite interpolation procedure to "fill in" the region between the inlet intake and the inlet

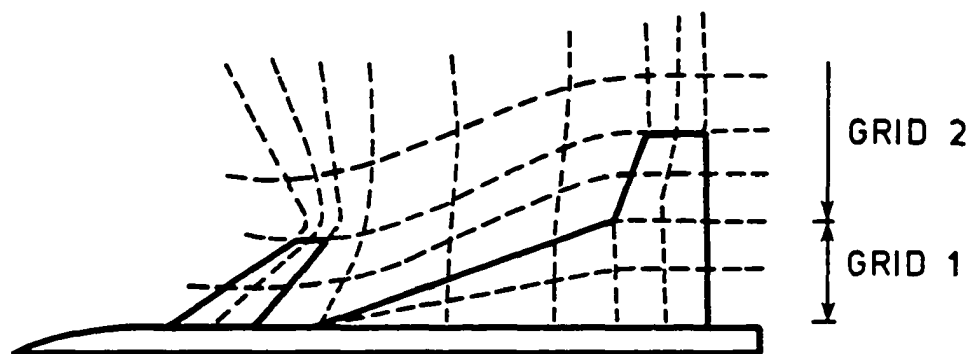
exit plane. It is expected that inlet description can influence the external flow at least in the region near the inlet. The interaction of the boundary layer diverter is not investigated in this study. Different results are presented here in a somewhat logical sequence.

4.4.1 Grid Generation Procedure

The configuration shown in Fig. 4.16 is composed of a fuselage, a canard, a cranked delta wing, a vertical fin and an inlet [157]. The fuselage has an integrated canopy over the cockpit, the engine inlet is separated from the main part of the fuselage by a boundary layer diverter, and is area ruled for supersonic flow. The canards and wings are defined by parabolic arc streamwise sections and the cranked wing is swept 70° followed by a 20° sweep. The intersection of the wing leading edge and the fuselage is near the vertical center of the fuselage and the intersection of the wing trailing edge and the fuselage is near the top of the fuselage. The vertical fin intersects the fuselage in as much the same way as the canard except it is defined vertically rather than in the spanwise direction. The model is 32 inches long. The upper and lower walls of the engine inlet start at about 13 and 16 inches, respectively, from the nose of the airplane. The distance from the center line to the outer wing tip is 9.5 inches. The grid generation about this configuration is not a trivial matter. For the exterior grid, if only patched C^1 continuous grids are considered and grid lines are required to conform to all wing edges, there are two basic alternative topologies (Fig. 4.17). The simplest approach (Alt.1) is to let the selected outgoing grid lines conform to the leading and trailing edges of both canard and wing. In the resulting single-block grid, with the third family of grid lines wrapping around the fuselage, both lifting surfaces can be represented as an interior slit. However, for a cranked delta wing with a highly swept inner region it is clear that this type of grid is not optimal. Not only is the grid highly skewed at



ALT. 1 - SINGLE GRID



ALT. 2 - DUAL GRID

Fig. 4.17 Two alternative topologies for the exterior grids.

the leading edge of the inner part of the wing, but its structure makes it difficult to concentrate grid points in a smooth fashion in the apex region of the wing. A different approach (Alt. 2) is to let the leading edge of the inner wing be represented by a grid line of the other family of lines, as shown in Fig. 4.17. This can be done by introducing an interior grid which only covers the inner wing region and its wake. In the resulting multiple grids, with the third family of grid lines wrapping around the fuselage, both lifting surfaces can again be represented as interior slits, with the inner wing slit in the inner grid and the canard and outer wing slits in the outer grid. It is clear that this topology gives a much less skewed grid and also gives a natural concentration of grid points in the apex region of the wing. This is very important since high resolution is needed in this region to capture vortical flow.

As mentioned previously, this study is an extension of the work available in [152] where the multiple grids have been used to obtain the Euler solutions for the external flow over the configuration in Fig. 4.16 (without wake region). In [152], the continuity of grid slope at the patched interface between subdomain grids is enforced by using the osculatory interpolation (Eq.(2.2)), i.e., by using derivative information as well as grid point location in the interpolation. In the present study, the flow domain is extended to include the wake region. A grid inside the engine inlet is also constructed to simulate the internal flow simultaneously with the external flow. The interior grid is only required to patch with the exterior grids at the inlet intake and at the exit plane. No continuity of grid slope or even grid points is enforced at the patched interfaces. Since the exact description of the inlet is not available, the interior grid is constructed simply by applying transfinite interpolation between these two planes. The extension of the exterior grids of Ref. 152 to include the wake region is achieved by the osculatory transfinite interpolation. However, the entire wake region is not covered by this extension. The void in this region is produced by the absence of the fuselage. It is filled in by the construction of a cylindrical

grid. This grid has a singularity line at the axis of the cylinder since a solid wall is absent there. Slope continuity at the interfaces between this grid and other exterior grids is, again, obtained by the osculatory transfinite interpolation. It is, however, not possible to have any kind of continuity at the interface between this grid and the interior grid since they are of completely different topologies, i.e., O-type versus H-type.

The entire grid about the configuration is, thus, divided into six subdomain grids. The surface grid for the configuration is shown in Fig. 4.18. The bottom and top grids are dual-block grids as described above. Slope continuity at the interfaces between exterior grids is clearly visible in Fig. 4.19. Figure 4.20 shows an enlarged view of a cross-cut of grids in the wake region. Slope continuity at grid interfaces and the point of singularity are illustrated. Grid discontinuities at the interfaces between interior and exterior grids are clearly noticeable in Fig. 4.21 where grid lines are plotted at the symmetry plane. Grid lines extended to the outer boundary at the symmetry plane are shown in Fig. 4.22 (interior grid is not shown). Finally, Fig. 4.23 illustrates the connection of various computational subdomains. Grids are numbered as indicated in the figure. The entire domain is composed of 493,641 grid points. It should be mentioned that the surface grid is constructed by the method described in [59].

4.4.2 Zero Degree Angle of Attack(Without Wake Region)

Steady state solutions for a freestream Mach number of 2 and 0° angle of attack are shown in Figs. 4.24-4.27. These figures show contours of the pressure coefficient on the surface of the airplane, on the symmetry plane, and on various cross-cuts. The contours display all the expected features of the flows with shock emanating from the nose of the fuselage and from the leading edges of the lifting surfaces. The bulging (area ruling) of the fuselage behind the inlet intake gives rise

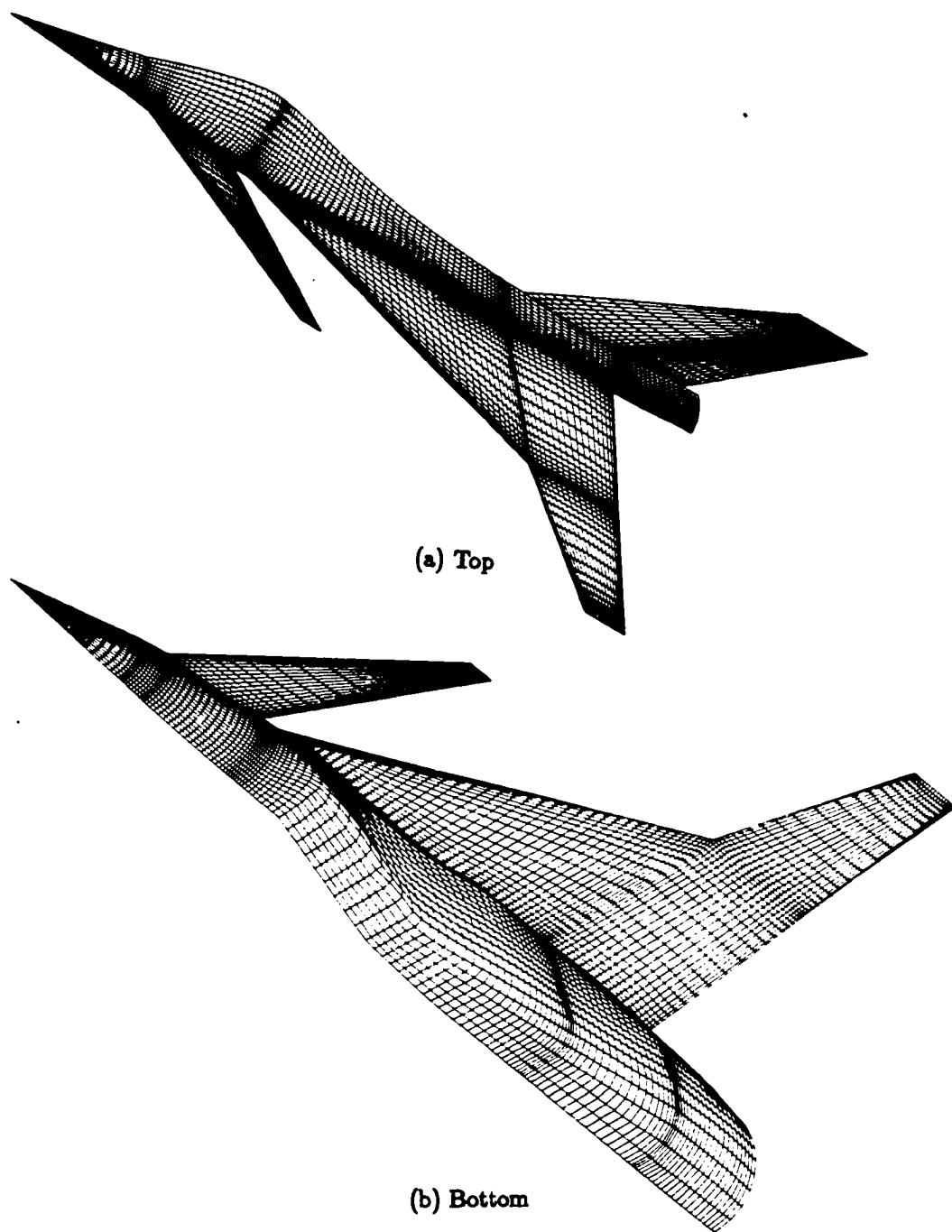


Fig. 4.18 The surface grid for the fighter aircraft.

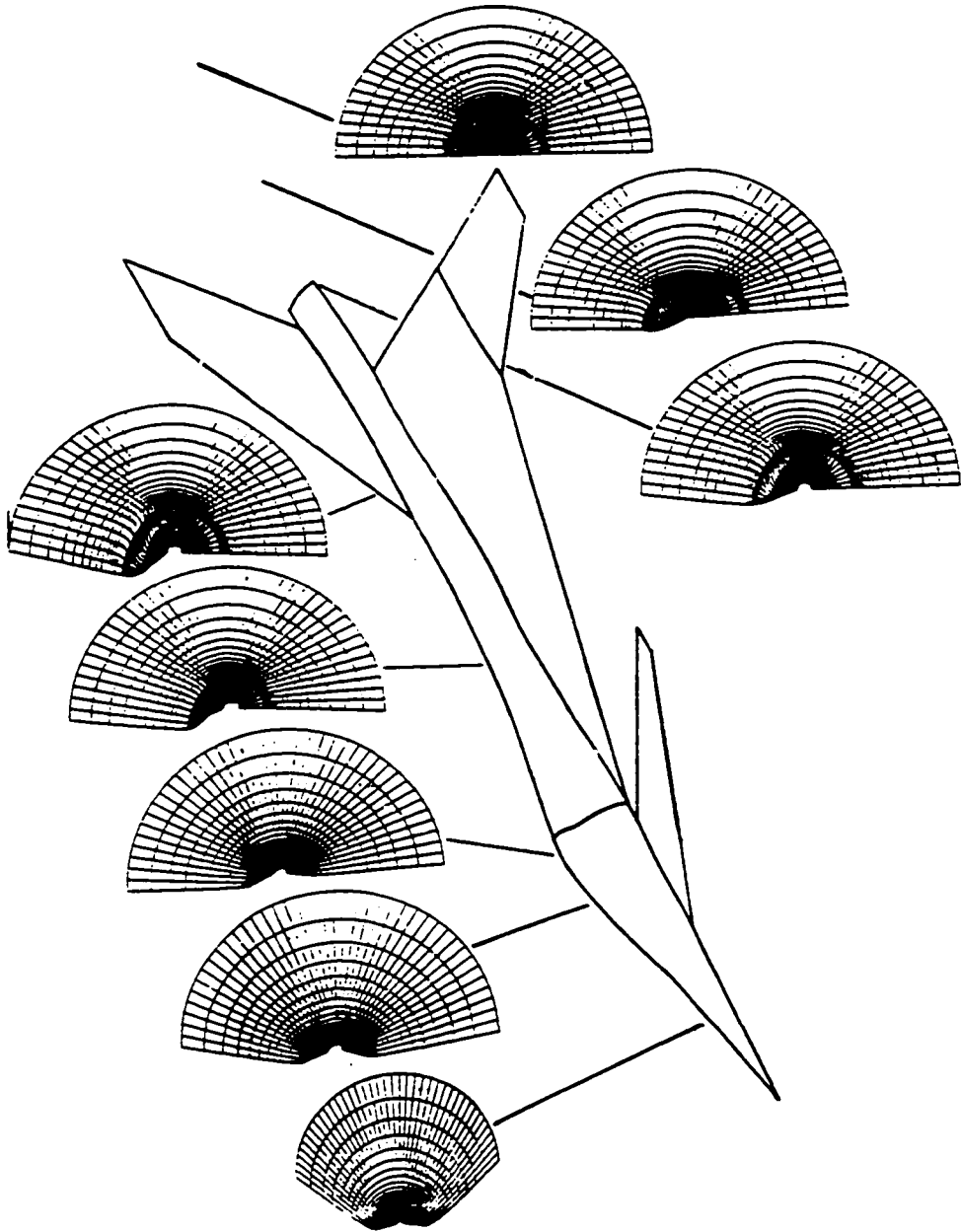


Fig. 4.19 Exterior grids for the fighter aircraft.

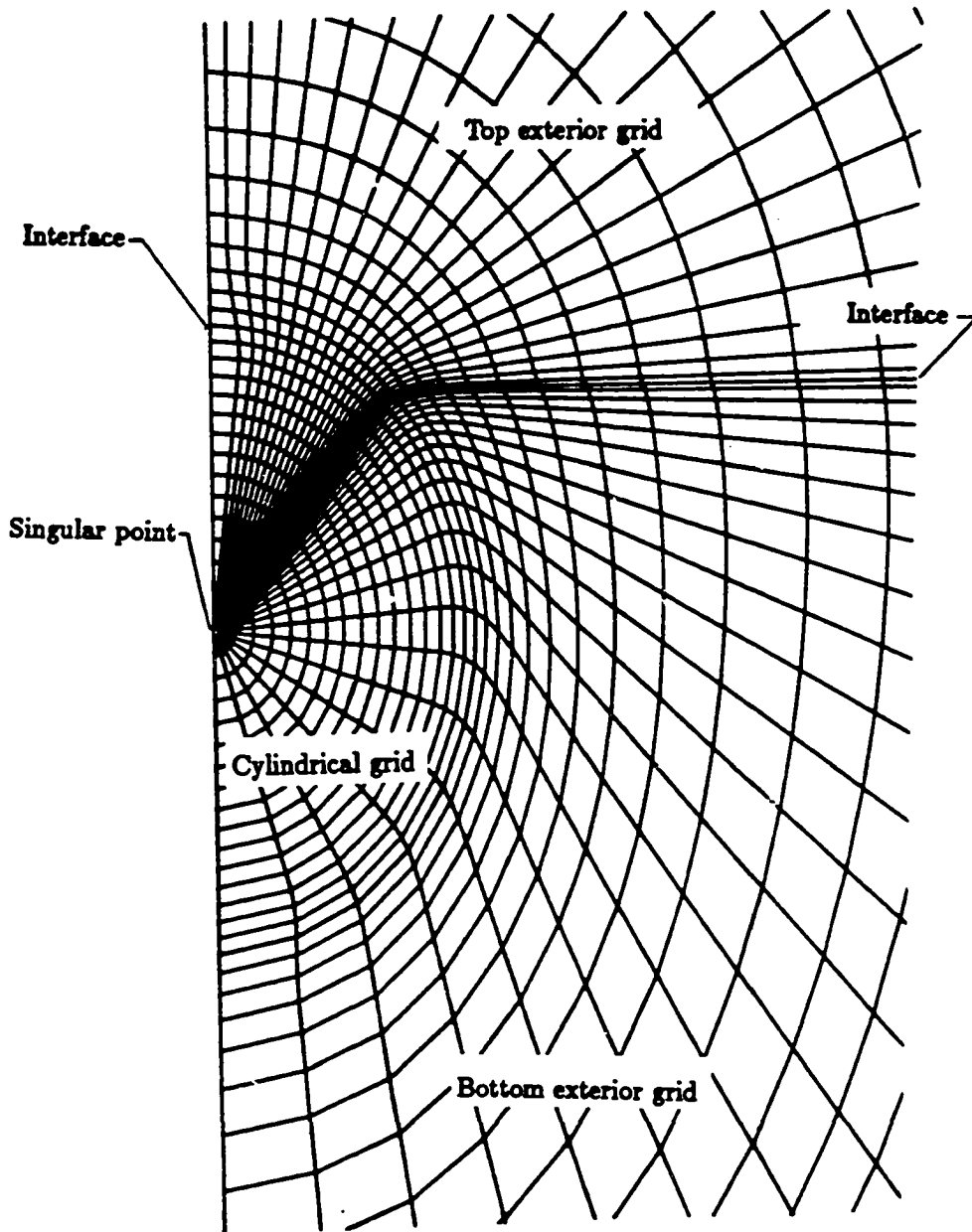


Fig. 4.20 The enlarge view of a cross cut of grids in the wake region.

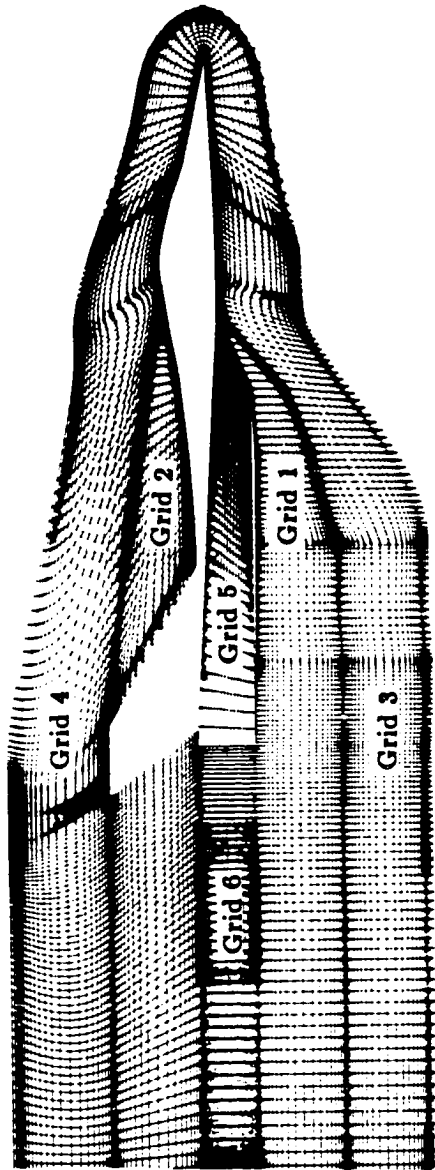


Fig. 4.21 Grid lines at the symmetry plane of the aircraft.

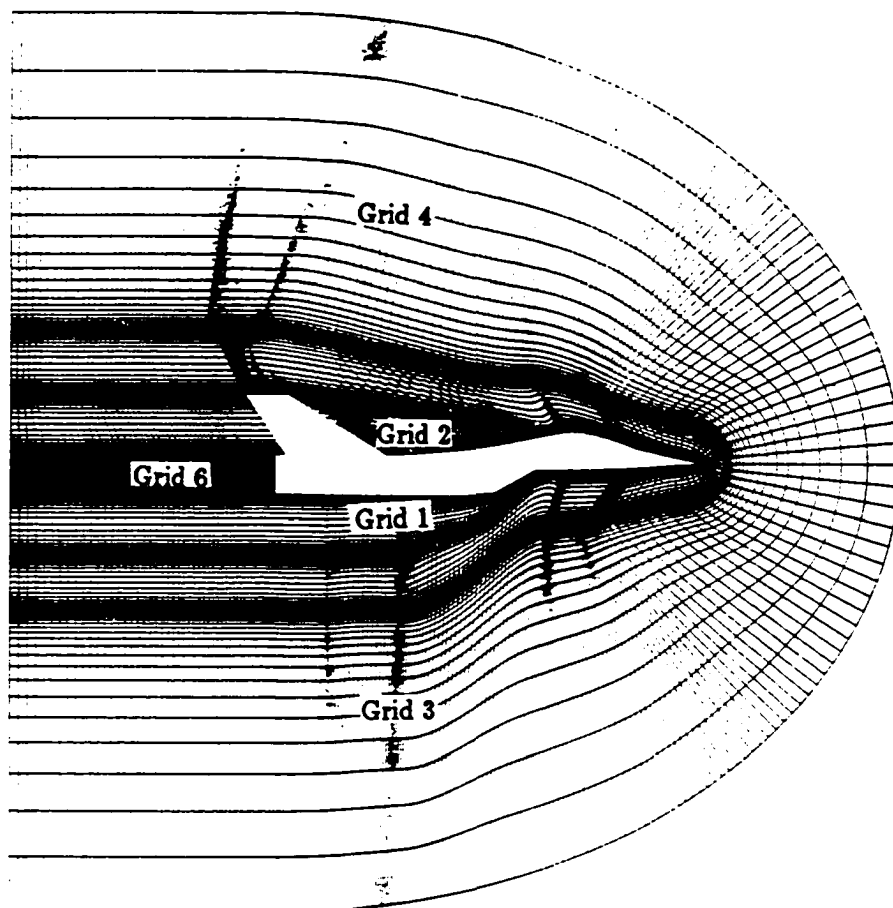


Fig. 4.22 Grid lines as in Fig. 4.21 extended to the outer boundary.

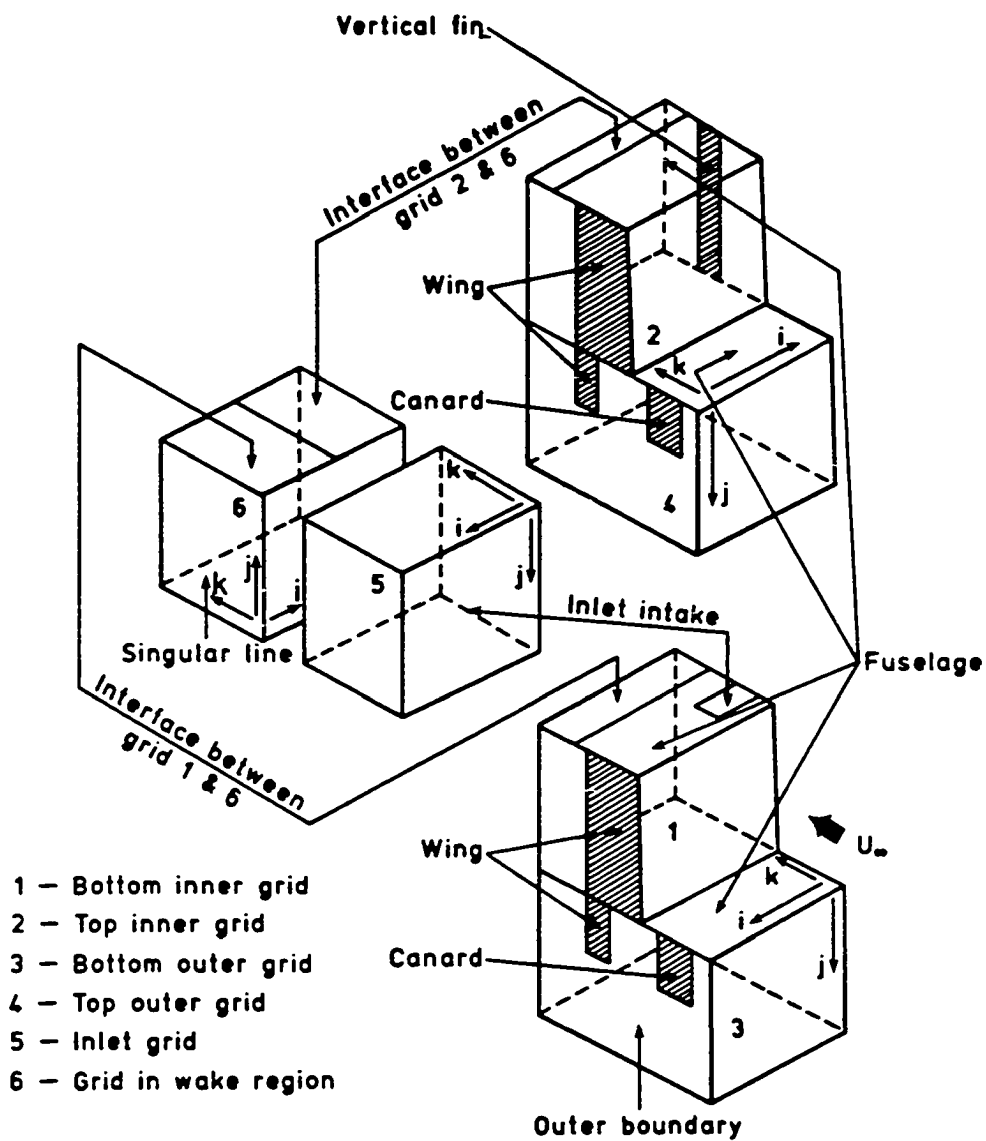


Fig. 4.23 Computational domains for the aircraft.

$$\begin{aligned}M_{\infty} &= 2 \\ \alpha &= 0^{\circ} \text{ (Without Wake)} \\ \Delta C_p &= 0.025\end{aligned}$$

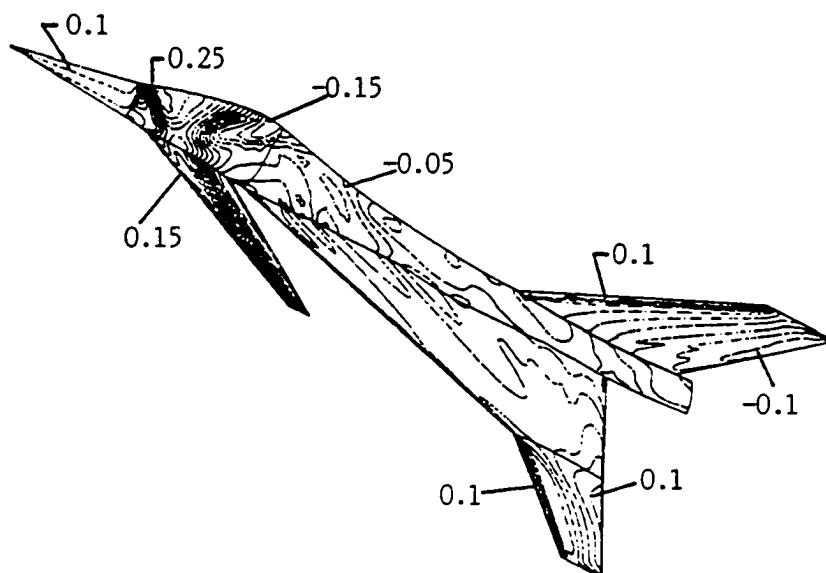


Fig. 4.24 C_p contours on the top surface ($\alpha = 0^{\circ}$, no wake).

$M_\infty = 2$
 $\alpha = 0^\circ$ (Without Wake)
 $\Delta C_p = 0.025$

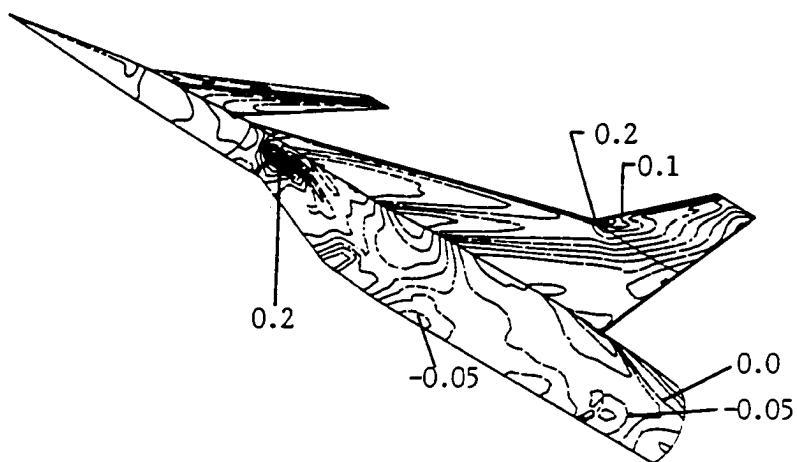


Fig. 4.25 C_p contours on the bottom surface ($\alpha = 0^\circ$, no wake).

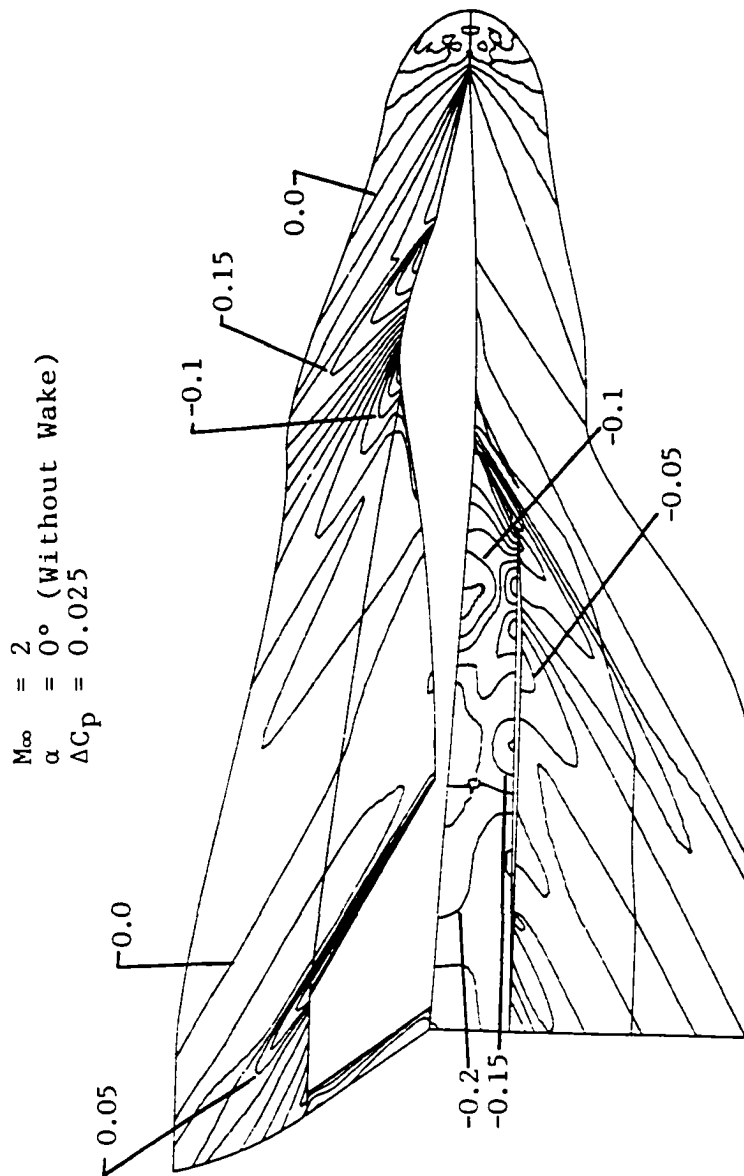


Fig. 4.26 C_p contours on the symmetry plane ($\alpha = 0^\circ$, no wake).

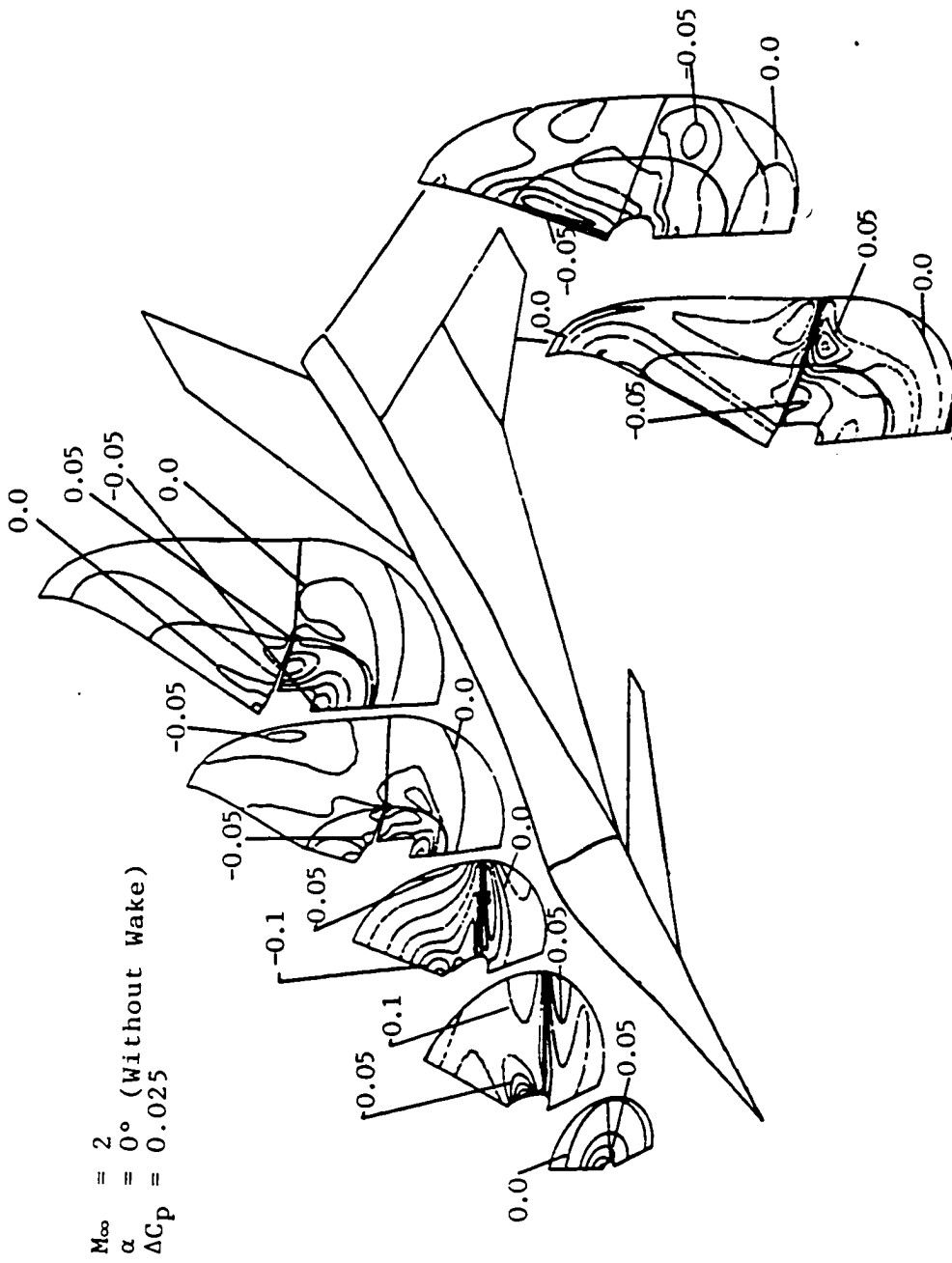


Fig. 4.27 C_p contours at various cross cuts ($\alpha = 0^\circ$, no wake).

to the high pressure in the region on the fuselage and on the bottom of the wing surface just downstream of the inlet intake. It will be seen (Sec. 4.3.3) that this high pressure region moves forward with increasing angle of attack. The flow enters the inlet region in a smooth manner, but there is a considerable increase in pressure because of some chocking at the inlet.

The spanwise pressure distribution at a cross-section which cuts through the inlet intake is shown in Fig. 4.28. The solid line represents the solution from the present study, whereas the dash line represents the solution from the study of Ref. 152. The top curves correspond to the pressure distribution on the bottom surface whereas the bottom curves represent the pressure distribution on the top surface. In the figure the horizontal distance is measured from the symmetry plane out to the wing tip. The distance is given as the real distance (inch) and the normalized distance. The normalized distance is based on the distance from the symmetry plane to the outer wing tip. The large pressure drops on the top curves, i.e., at the distance about 1.2 inches from the symmetry plane, correspond to the point where the wing starts on the bottom surface. The label 'no inlet' in the figure simply means no simulation of flow inside the inlet but the inlet itself is still attached. The comparison in the figure indicates differences between the two solutions at the inlet intake. This is because, in the present case, the flow is allowed to go through the inlet while only farfield condition (zero order extrapolation) is implemented in Ref. [152]. This demonstrates the influence of the inlet to the external flow near the inlet intake. It is expected that the flow in this region change with the description of the inlet.

Variations in the pressure coefficient at the symmetry plane of the inlet are shown in Fig. 4.29 for three different locations (top, center, and bottom). The horizontal axis (x-distance) is given in two different scales, one is the real scale in inches and the other is the normalized scale. The normalized scale is based on

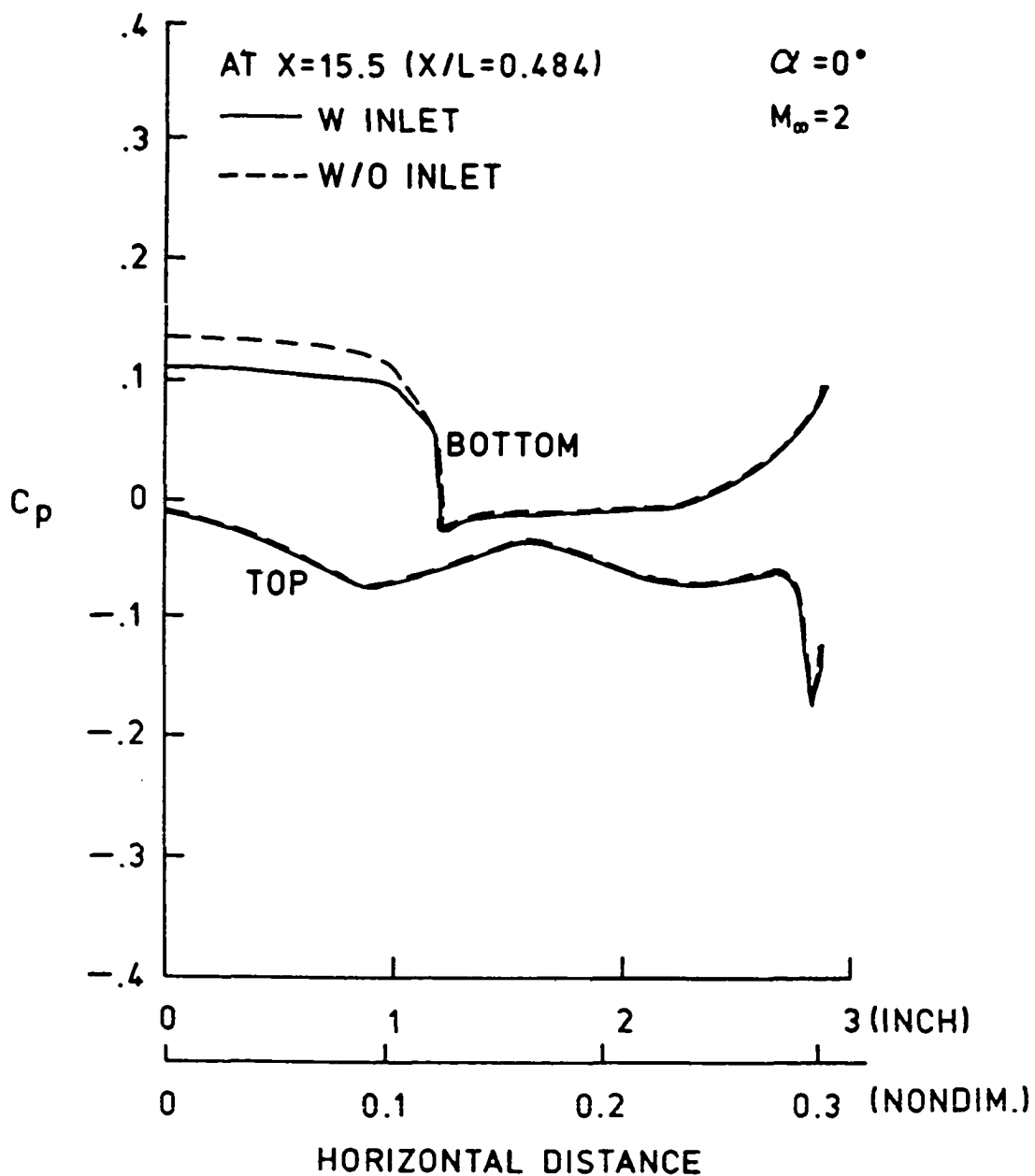


Fig. 4.28 Spanwise pressure distribution at the inlet intake ($\alpha = 0^\circ$, no wake).

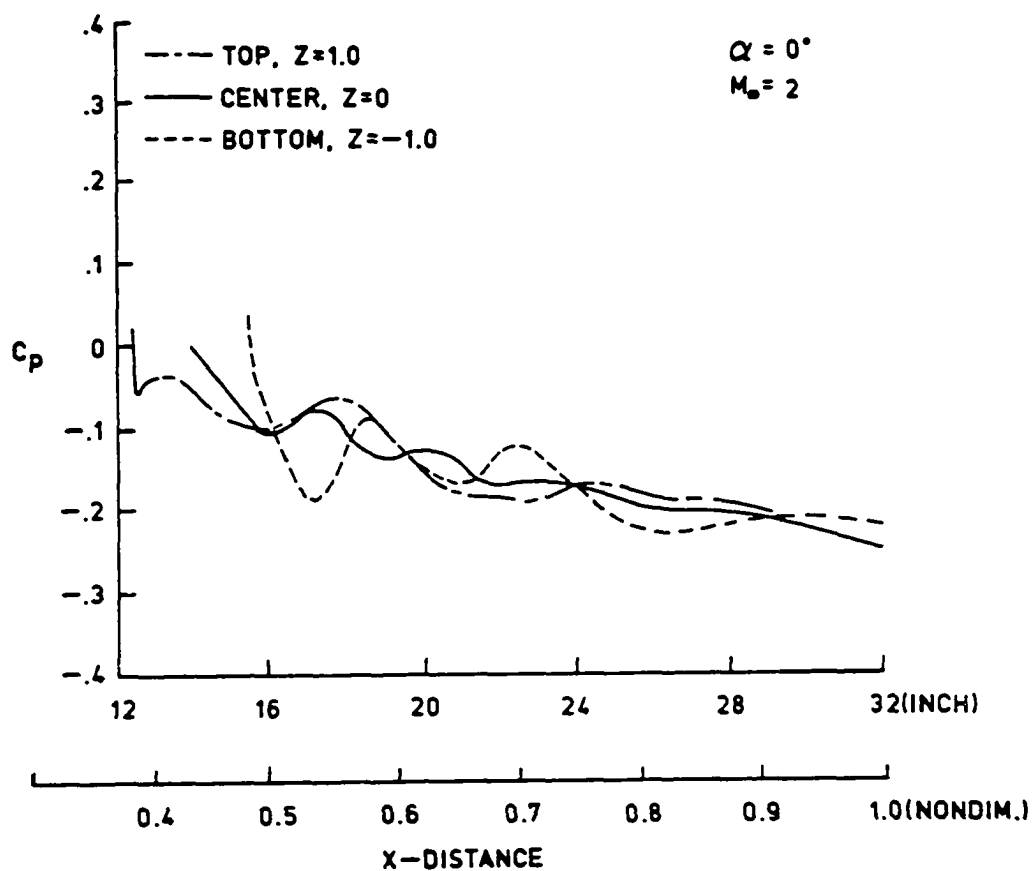


Fig. 4.29 Pressure distribution inside the inlet ($\alpha = 0^\circ$, no wake).

the total length of the model (32 inches from the nose to the rear end of the fuselage). The height of the inlet (z -distance) is normalized based on its height from the center line. The results show that the flow, in general, expands after it goes into the inlet. This is due to the inlet shape created by generation of the interior grid. However, the situation can be entirely different if another inlet shape is generated. The results also indicate that there exist shock waves which reflect from the inlet walls. These shocks lose their strength as the flow moves away from the inlet intake.

4.4.3 Different Angles of Attack (Without Wake Region)

Increasing angles of attack while holding the Mach number at 2 produces expected results. Pressure increases on the bottom surface and decreases on the top surface. The pressure coefficient contours on the top and bottom surfaces of the fuselage, canard, and wing are shown in Figs. 4.30 and 4.31 for flow at 3.79° angle of attack. Figures 4.32 and 4.33 show the corresponding contours of pressure coefficient at the plane of symmetry and at various cross-cuts. Again these figures show all the expected features of the flow such as high pressure in front of the canopy and low pressure behind it, low pressure along the upper leading edge of the inner wing and high pressure under the outer wing. An interesting feature is the continuation of the low pressure region along the top leading edge of the inner wing to the region between the outer and inner wing, downstream of the crank. This indicates that there is some vortex generation and thus entropy also increases in this region. There is also some chocking at the inlet (Fig. 4.32) as in the previous case. The comparison of the spanwise pressure distribution at a cross-section which cuts through the inlet intake (Fig. 4.34) also indicates pressure difference between the two solutions at the inlet intake. Plots of pressure coefficient at the symmetry plane of the inlet along three different locations are shown in Fig. 4.35. This figure

$$\begin{aligned}M_{\infty} &= 2 \\ \alpha &= 3.79^{\circ} \\ \Delta C_p &= 0.025\end{aligned}$$

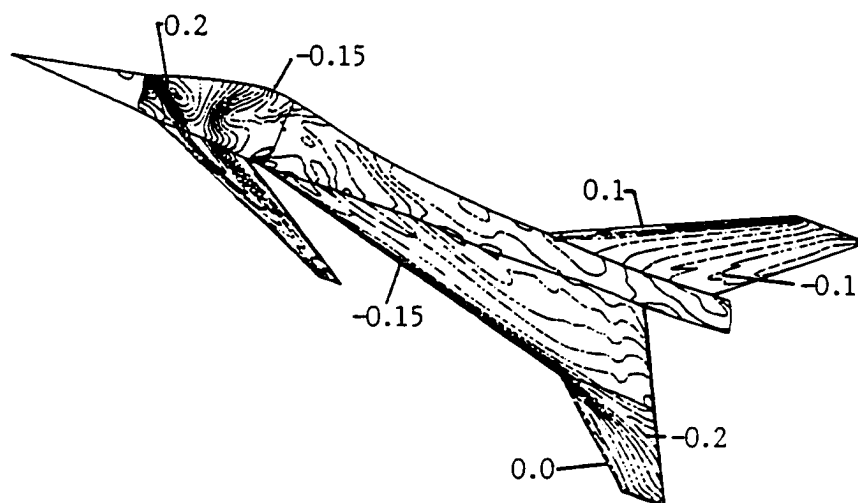


Fig. 4.30 C_p contours on the top surface ($\alpha = 3.79^{\circ}$, no wake).

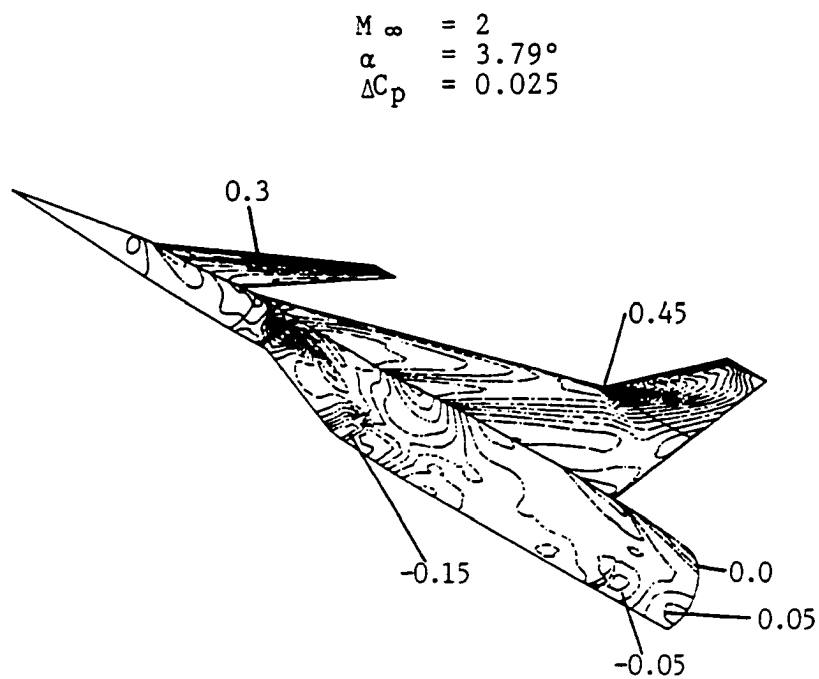


Fig. 4.31 C_p contours on the bottom surface ($\alpha = 3.79^\circ$, no wake).

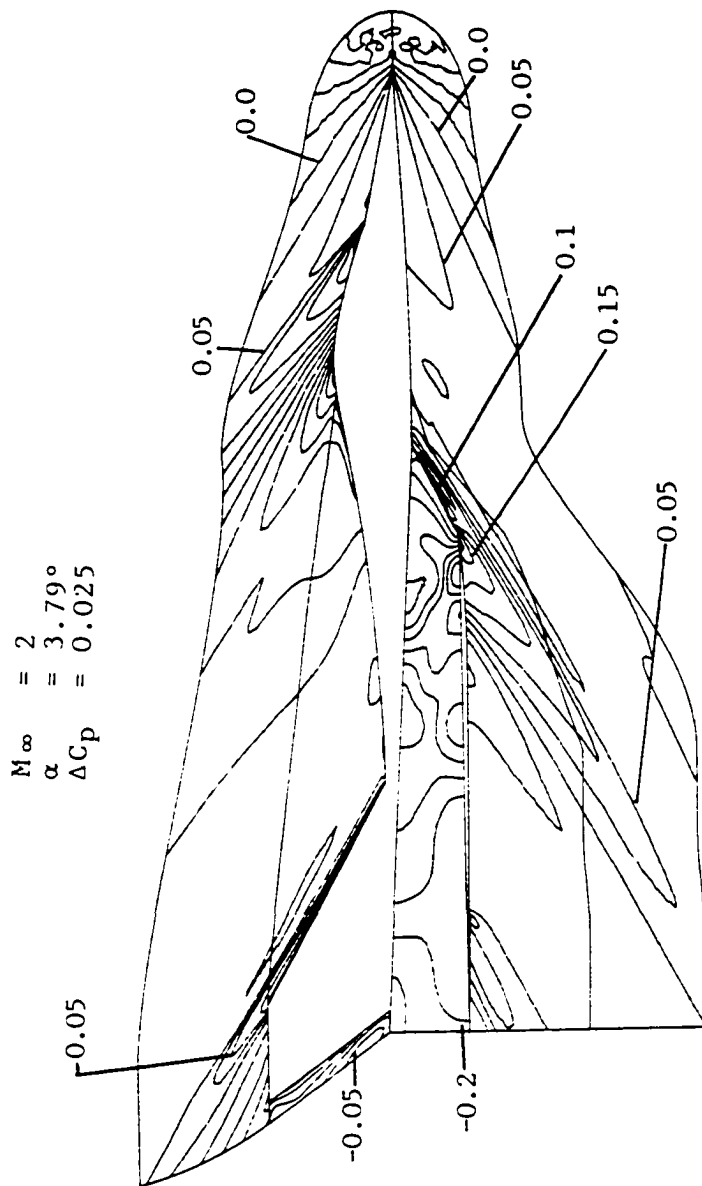


Fig. 4.32 C_p contours on the symmetry plane ($\alpha = 3.79^\circ$, no wake).

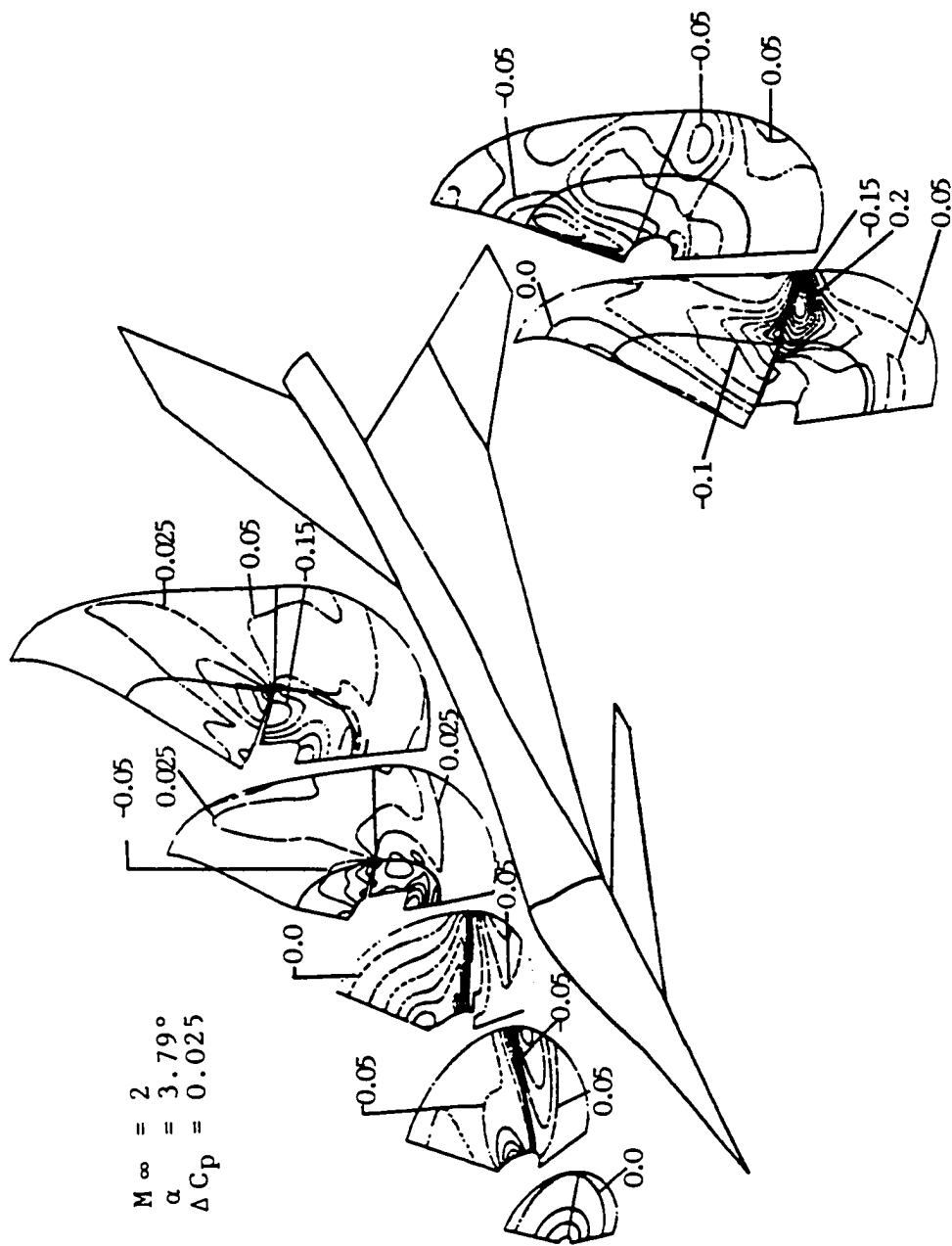


Fig. 4.33 C_p contours at various cross cuts ($\alpha = 3.79^\circ$, no wake).

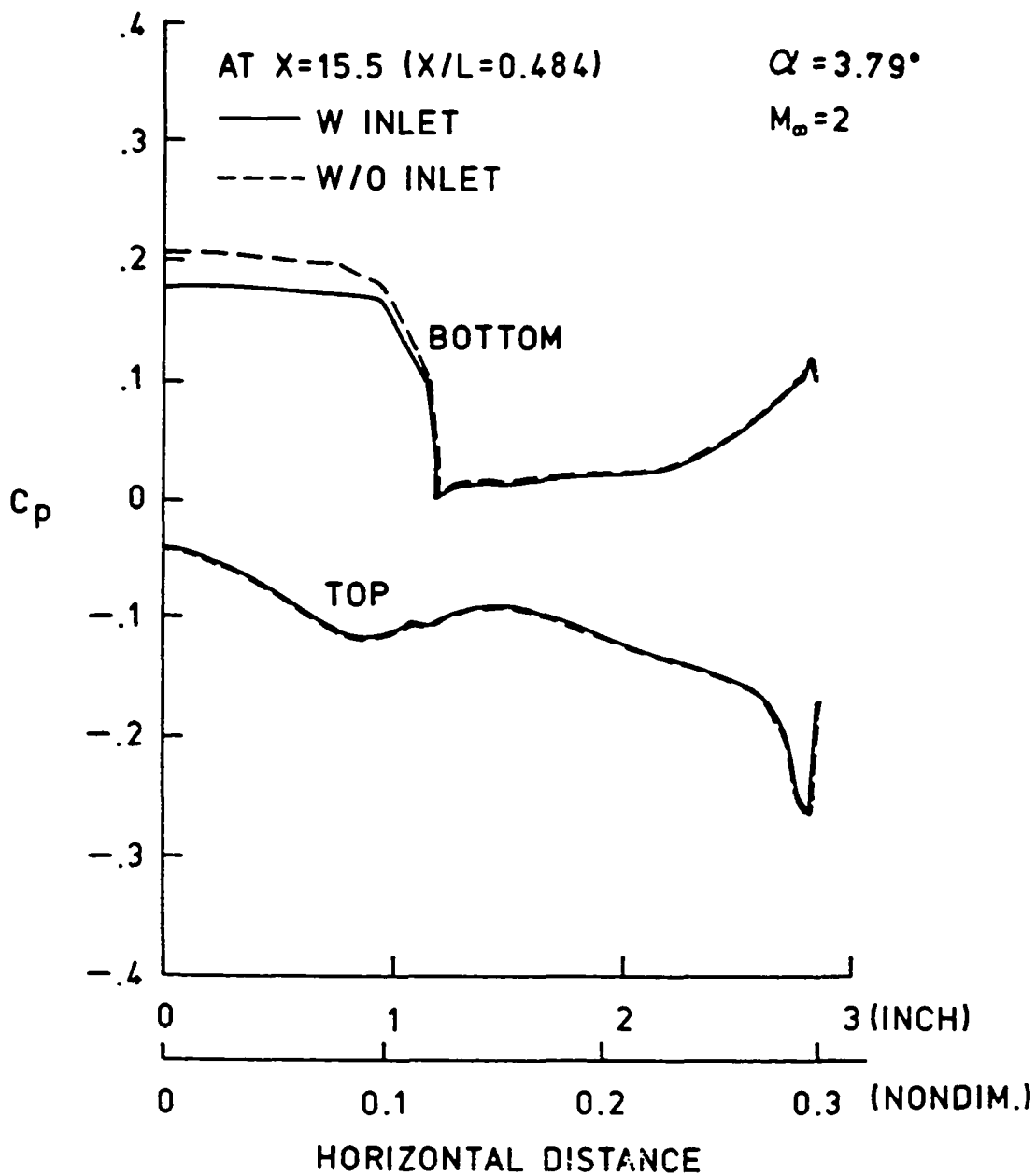


Fig. 4.34 Spanwise pressure distribution at the inlet intake ($\alpha = 3.79^\circ$, no wake).

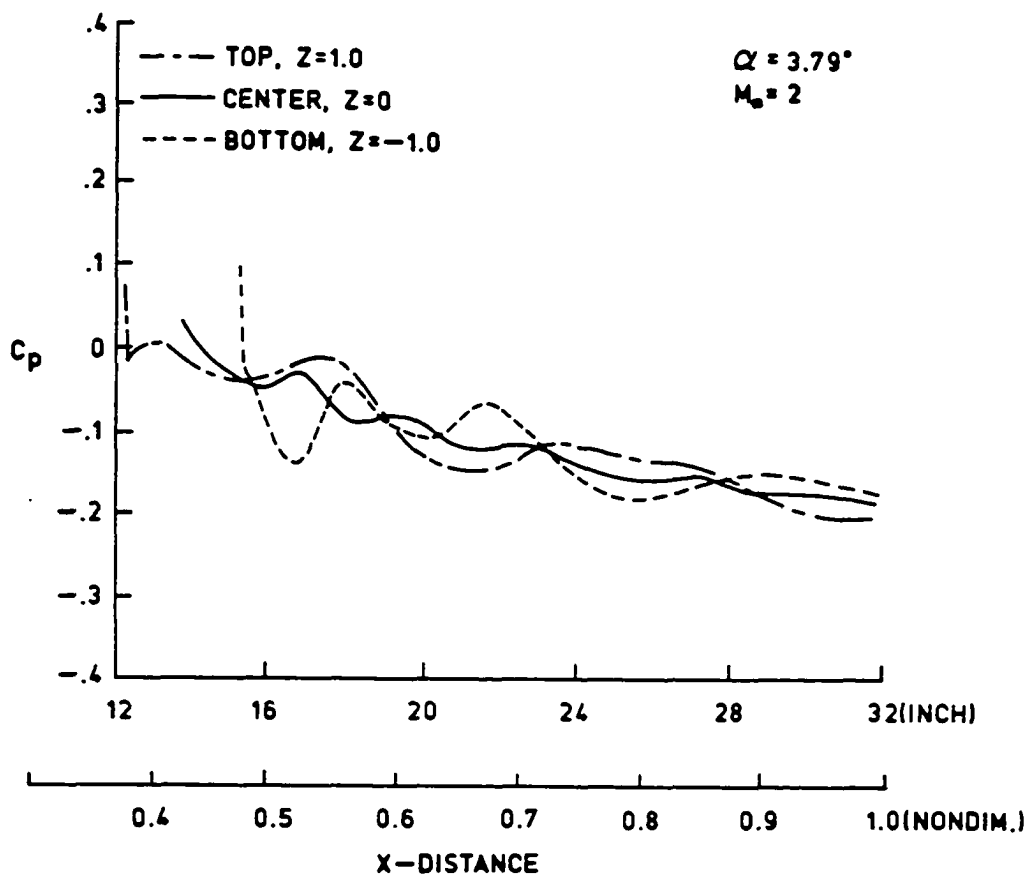


Fig. 4.35 Pressure distribution inside the inlet ($\alpha = 3.79^\circ$, no wake).

illustrates the similar features as in the previous case. The expansion of the flow and the existence of shock waves inside the inlet are clearly visible.

Figures 4.36 and 4.37 display contours of pressure coefficient on the top and bottom surfaces for flow at 7° angle of attack. On the top wing surface, there is a strong indication of vortical flow in the pressure distribution. Also, on the bottom surface, the high pressure region resulting from the bulging has moved forward on the bottom of the wing. A high pressure due to choking can be observed at the inlet intake (Fig. 4.38). This pressure is even higher than that of the previous two cases due to the higher angle of attack. Contours of the pressure coefficient on various cross-cuts are also shown in Fig. 4.39. The plots of spanwise pressure distribution (Fig. 4.40) indicate similar comparison as in the previous case.

The results for 10° angle of attack are shown in Figs. 4.41- 4.45. A vortical motion is evident on the top wing surface. The choking of the flow at the inlet intake is evident from the contour plots of Fig. 4.43. Figure 4.44 illustrates pressure coefficient contours on various cross-cuts. A comparison of the results presented in Fig. 4.45 also indicates the influence of the flow inside the inlet on the external flow in the region near the inlet intake. This demonstrates that the combination of the implementation of boundary conditions and the simulation of flow inside the inlet produces different results. In obtaining these results, numerical oscillations were noticed near the leading edge of the inner wing. Such oscillations were observed also in the study of Ref. 152. Convergence to a steady state solution appears to be normal at first. The residuals (difference of solution between two consecutive time steps) reach a minimum values and start a periodical phenomena. This phenomena prevents the solution from reaching a steady state. Further investigation indicates that the oscillation only occurs at a certain region, i.e., region just inside the top leading edge of the 70° swept portion of the wing. Elsewhere, the residuals are well below the convergence criterion. The investigation in [152] has shown that the use

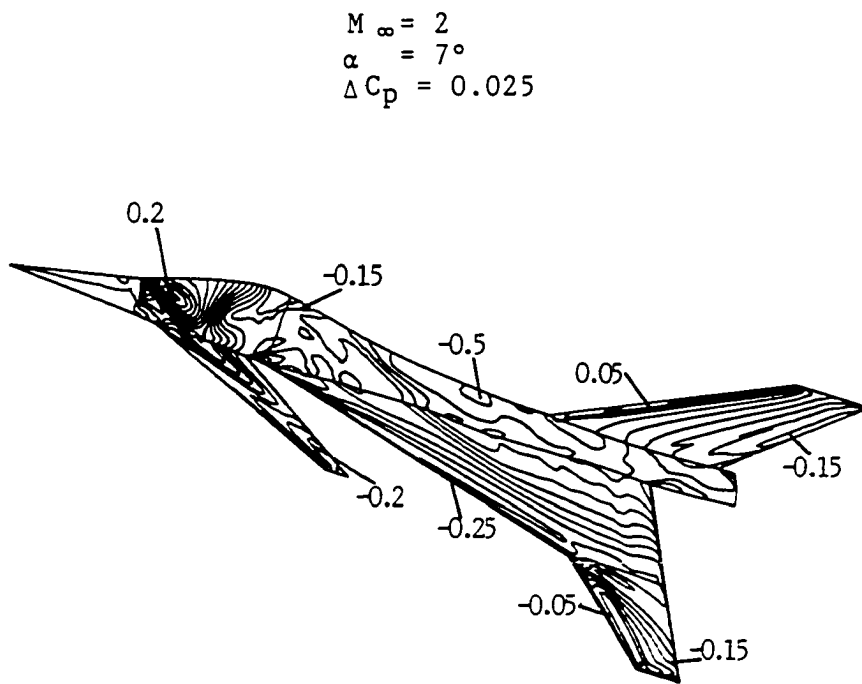


Fig. 4.36 C_p contours on the top surface ($\alpha = 7^\circ$, no wake).

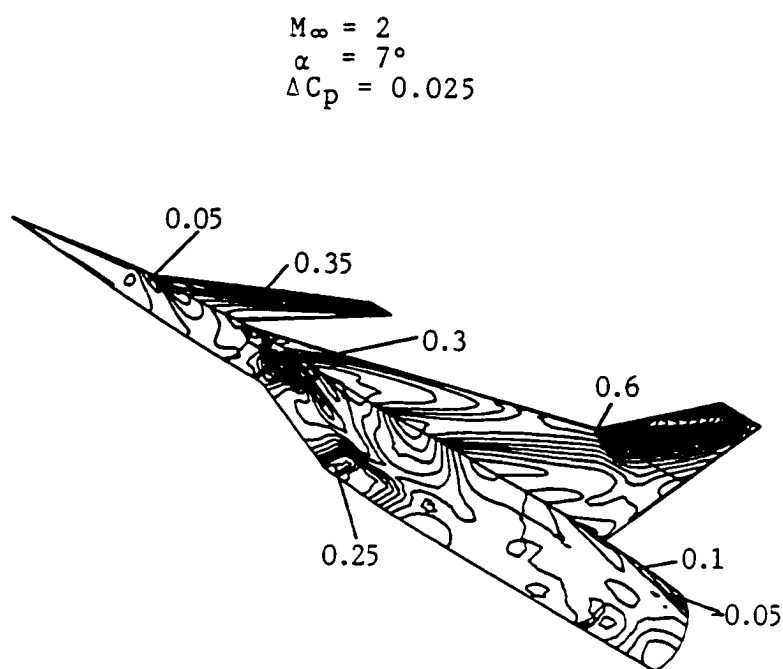


Fig. 4.37 C_p contours on the bottom surface ($\alpha = 7^\circ$, no wake).

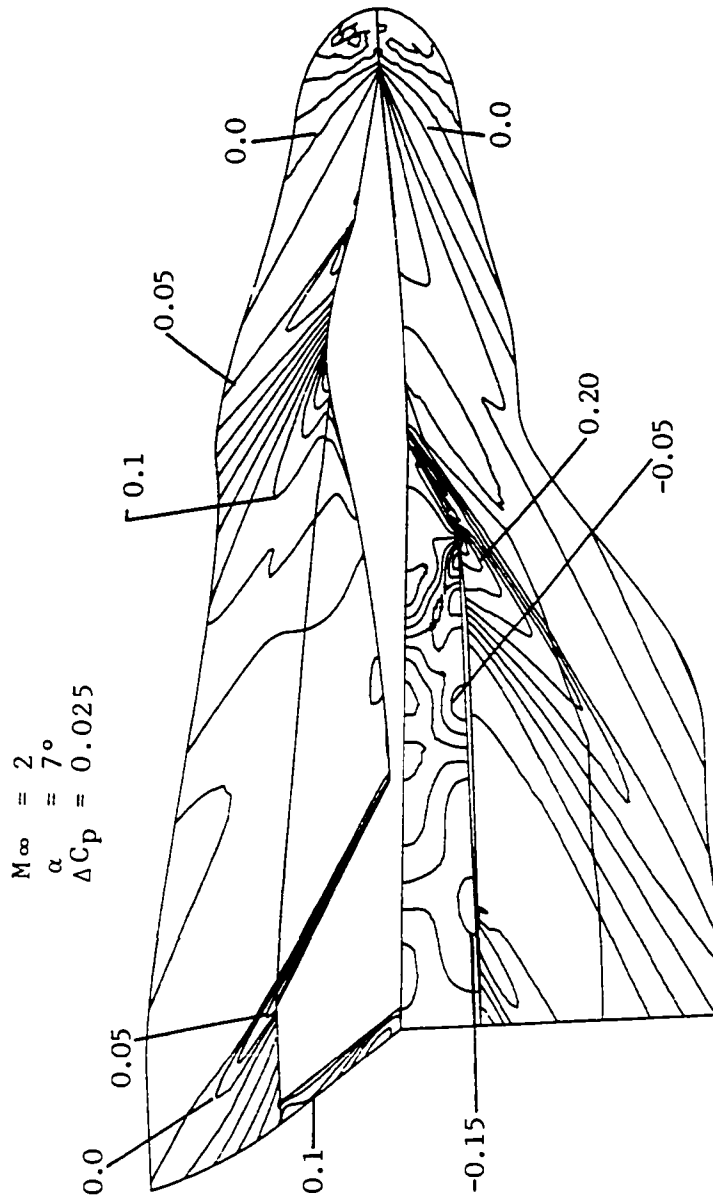


Fig. 4.38 C_p contours on the symmetry plane ($\alpha = 7^\circ$, no wake).

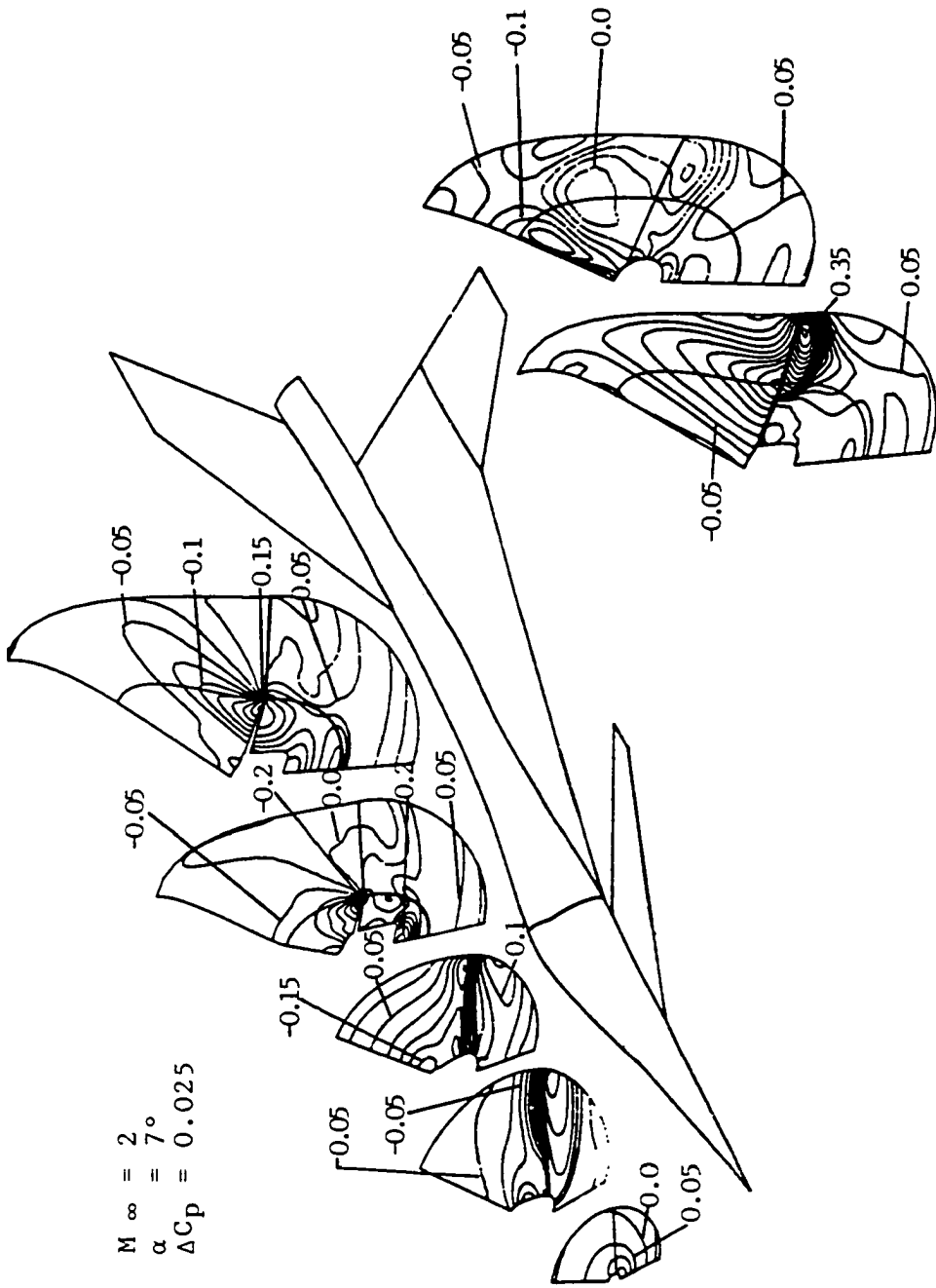


Fig. 4.39 C_p contours at various cross cut ($\alpha = 7^\circ$, no wake).

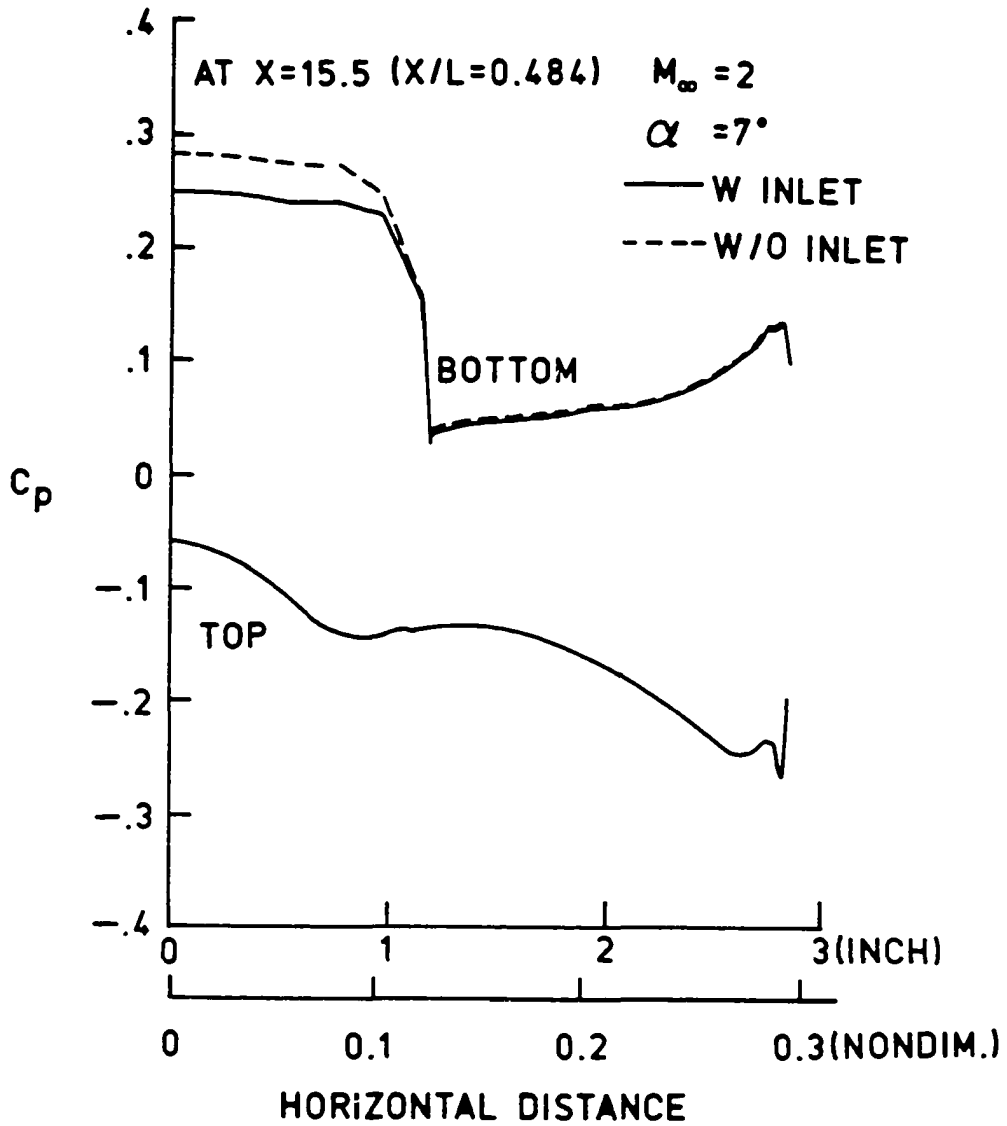


Fig. 4.40 Spanwise pressure distribution at the inlet intake ($\alpha = 7^\circ$, no wake.)

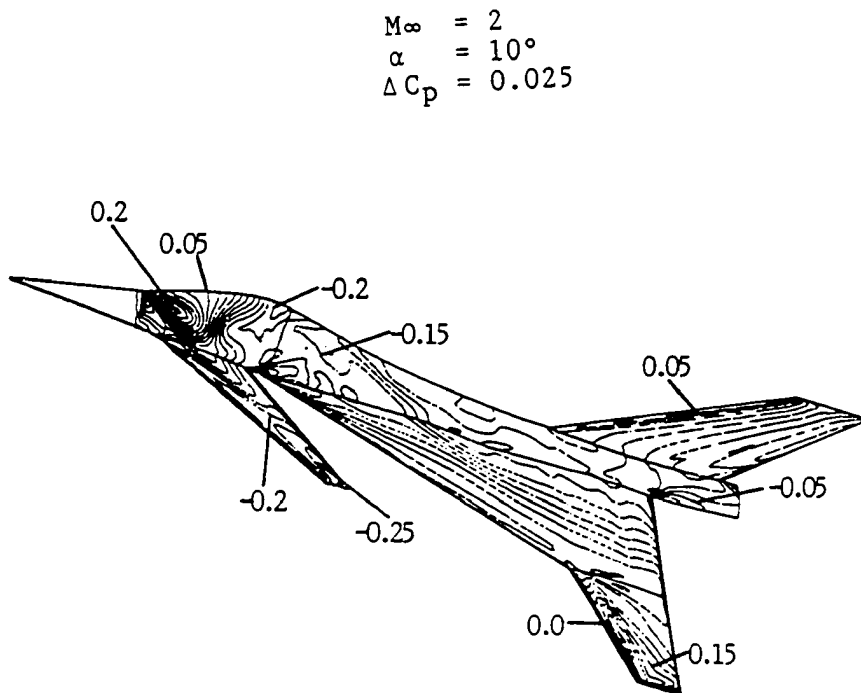


Fig. 4.41 C_p contours on the top surface ($\alpha = 10^\circ$, no wake).

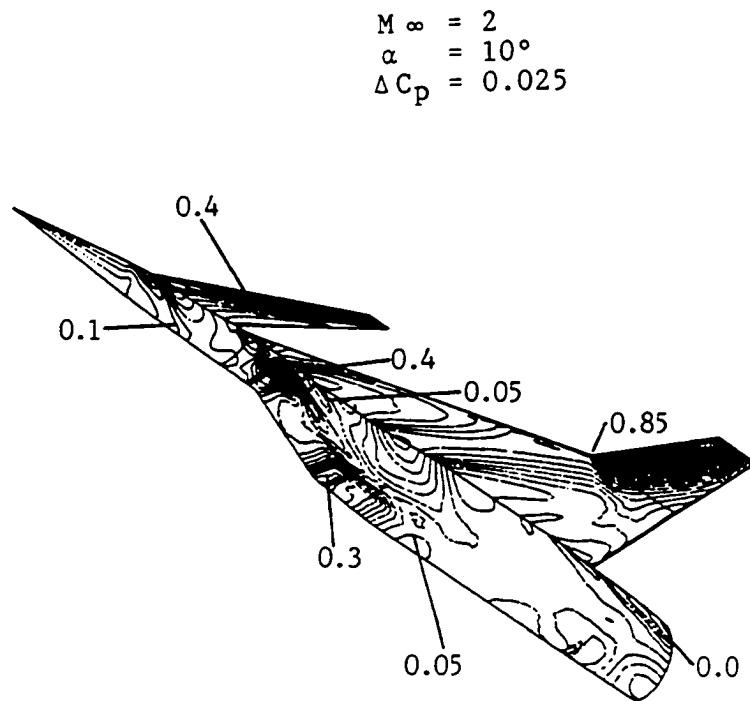


Fig. 4.42 C_p contours on the bottom surface ($\alpha = 10^\circ$, no wake).

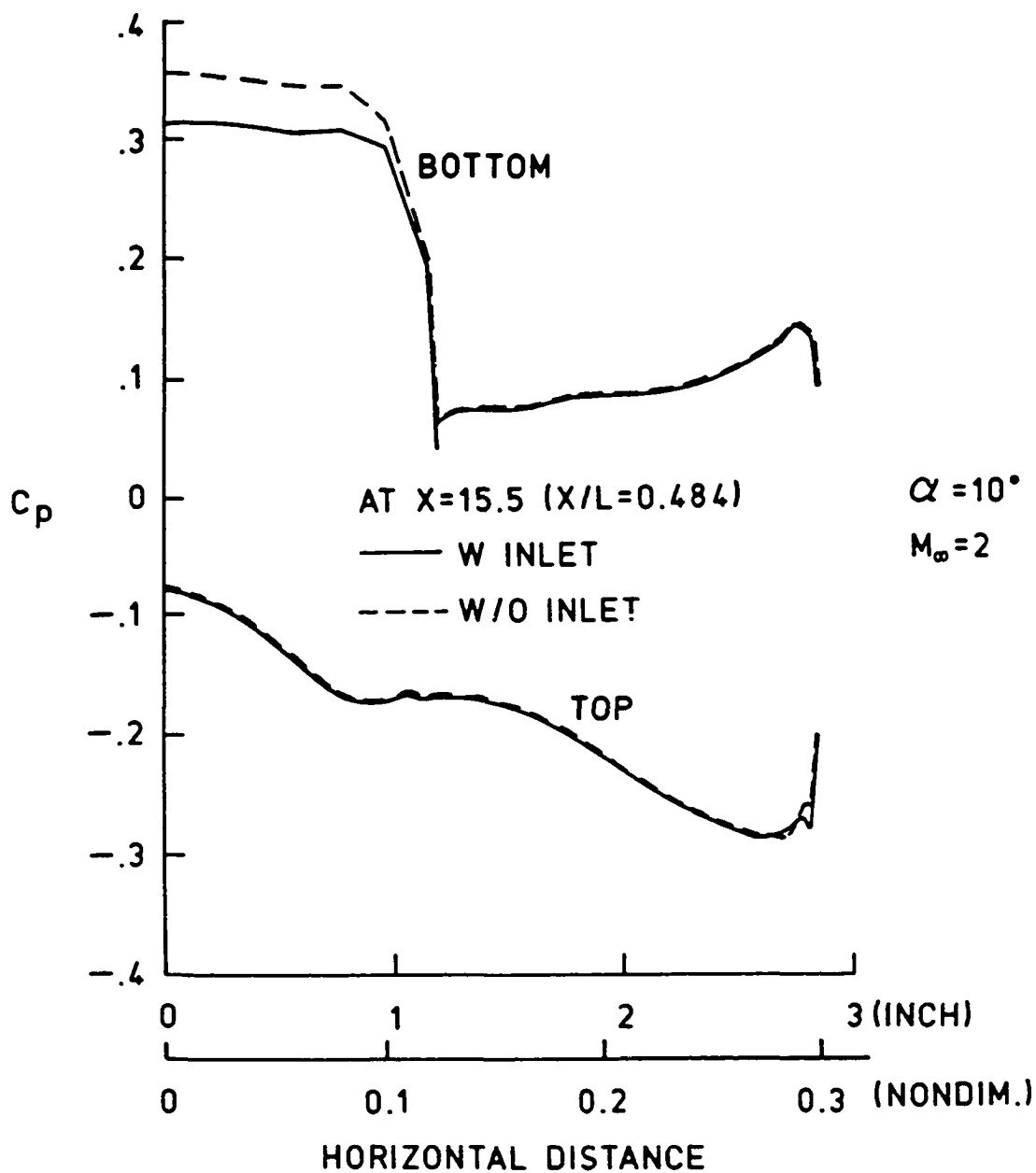


Fig. 4.43 Spanwise pressure distribution at the inlet intake ($\alpha = 10^\circ$, no wake).

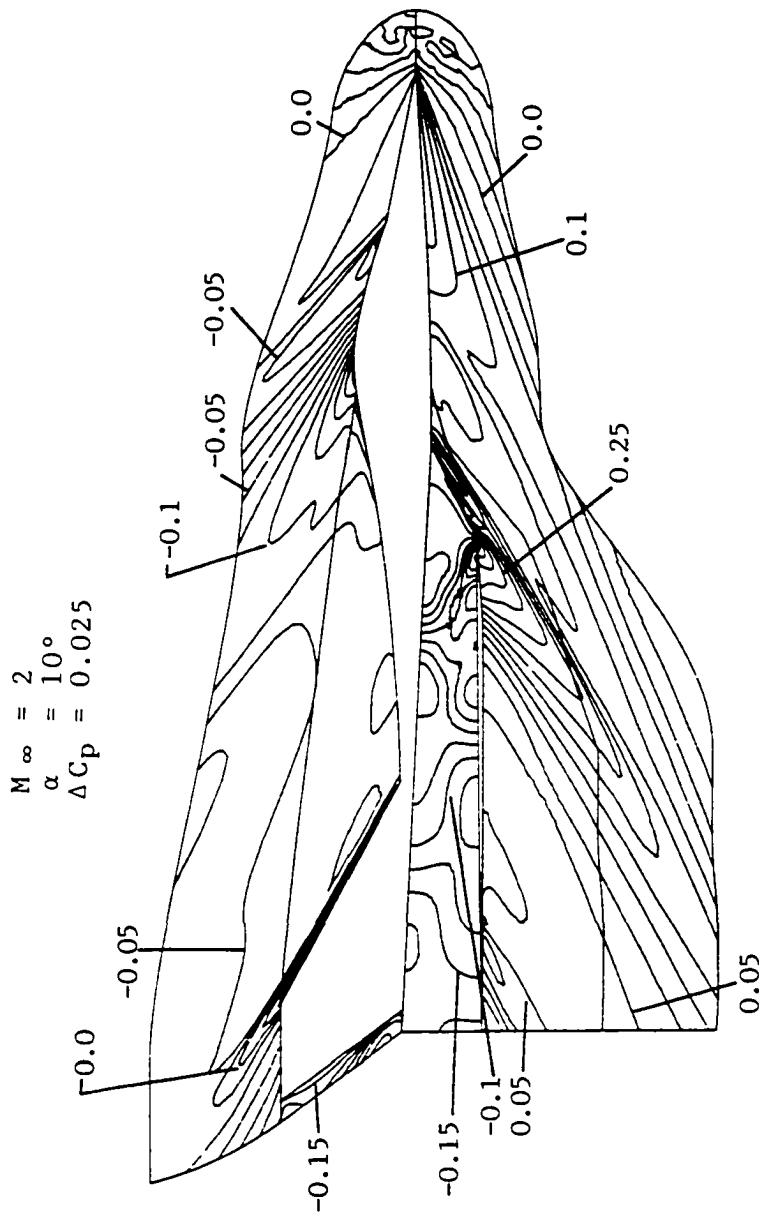


Fig. 4.44 C_p contours on the symmetry plane ($\alpha = 10^\circ$, no wake).

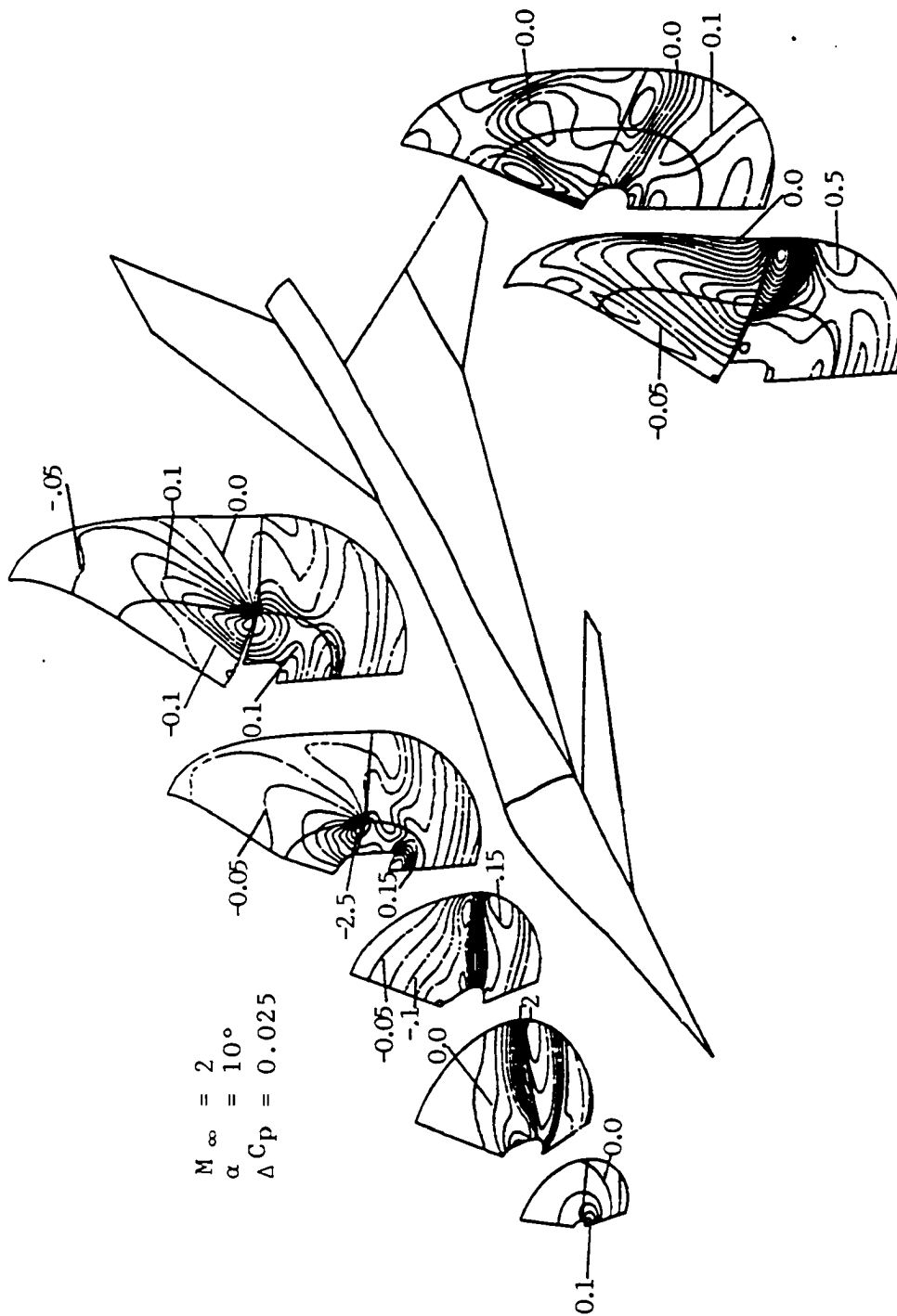


Fig. 4.45 C_p contours at various cross cuts ($\alpha = 10^\circ$, no wake).

of minimum time step everywhere could solve this problem. The local time-step scaling, however, is still used to obtain all steady state solutions presented in this study. This is because the oscillation is not caused by the use of the conservative interface condition which is the primary objective of the present study. Moreover, the use of minimum time stepping is quite expensive. This periodical phenomenon has been clearly explained in [152] and will not be repeated here.

Variations in the pressure distribution inside the inlet at the plane of symmetry are illustrated in Figs. 4.46–4.48 for four different angles of attack along top walls, bottom walls, and center line of the inlet, respectively. These figures indicate similar features, such as the expansion of the flow and the existence of shocks inside the inlet, among flows at different angles of attack. The effect of increasing angle of attack is seen to increase the pressure inside the inlet. Variations in the pressure coefficient along the centerline of the symmetry plane from the free-stream to the exit plane of the inlet are shown in Fig. 4.49 for different angles of attack. The variations indicate a similar pattern for each angle of attack. Shock waves generated at the nose of the aircraft and near the inlet intake (due to choking) are clearly visible. The expansion of flows inside the inlet and interaction of shock waves from the walls are clearly evident. It is seen that pressure increases with increasing angle of attack.

4.4.4 Zero Degree Angle of Attack (With Wake Region)

The pressure coefficient contours on the top and bottom surfaces of the fuselage, canard, and wing are shown in Figs. 4.50 and 4.51. These contours are similar to those obtained for the case without the wake except near the end of the fuselage. A lower pressure (as compared to flows without wake region) is observed in this region due to simulation of the flow in wake region. The influence of flow in the wake region is clearly seen in Fig. 4.52. The spanwise pressure distribution from

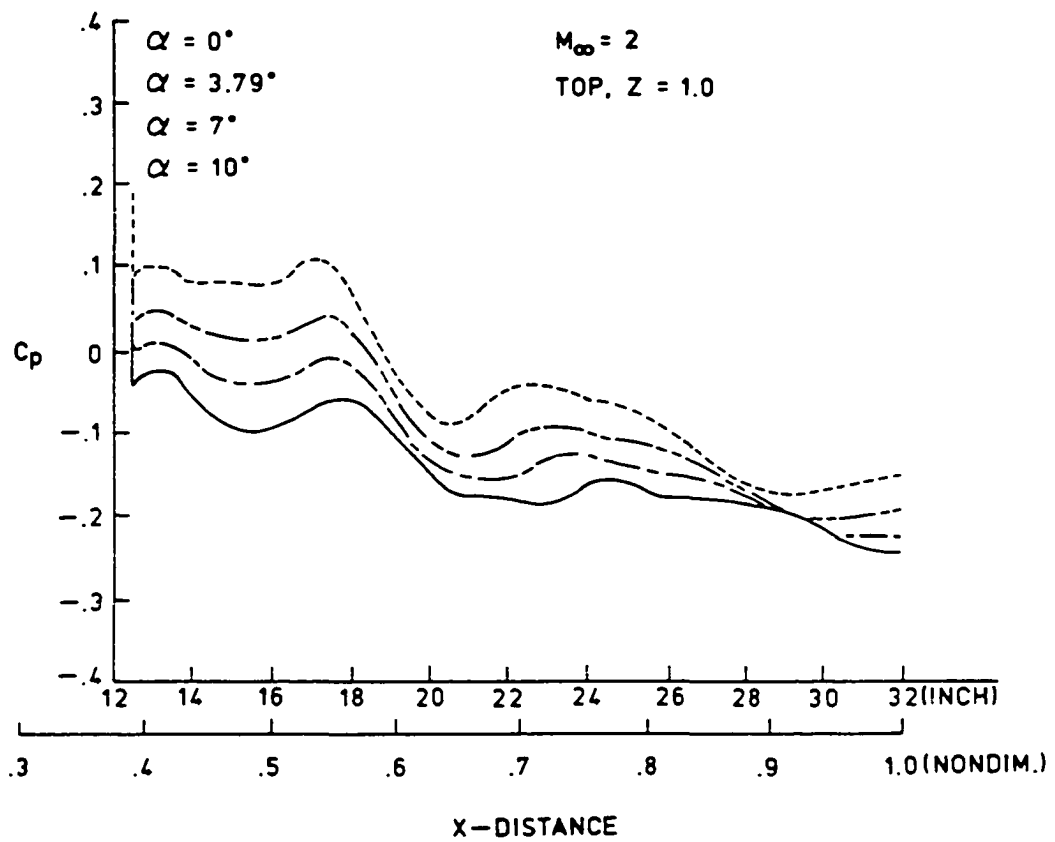


Fig. 4.46 Pressure distribution along top wall of the inlet.

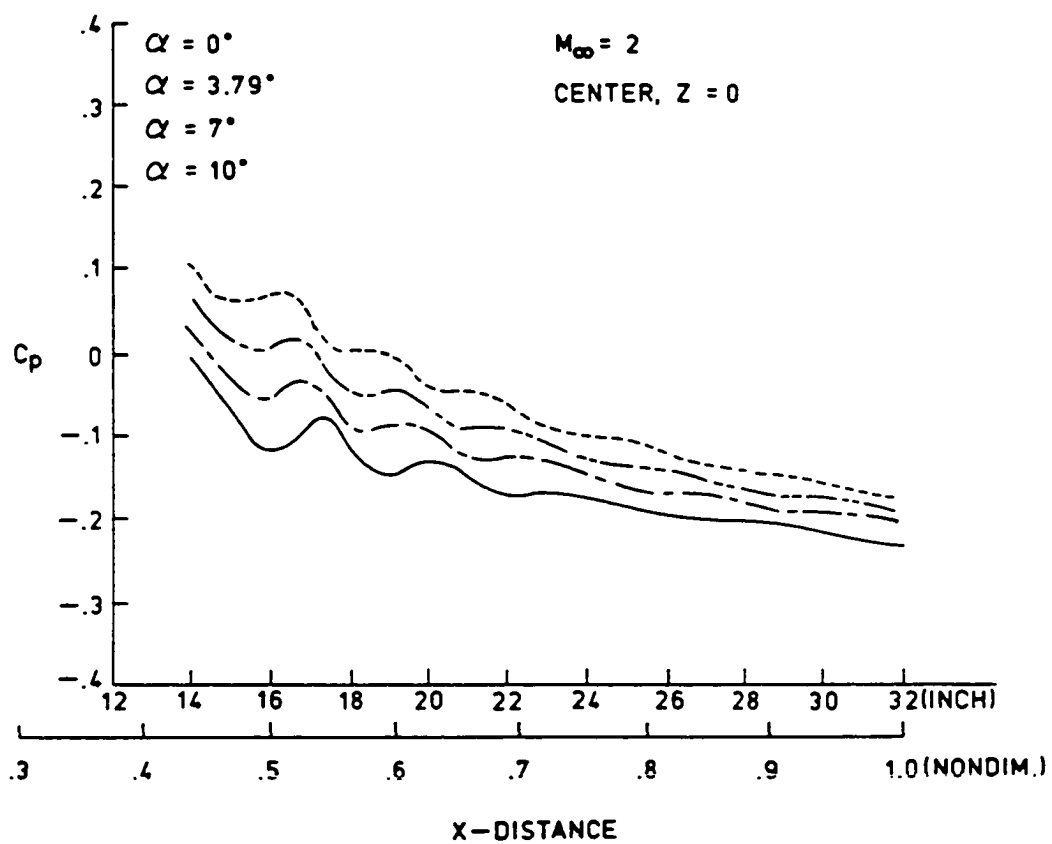


Fig. 4.47 Pressure distribution along center of the inlet.

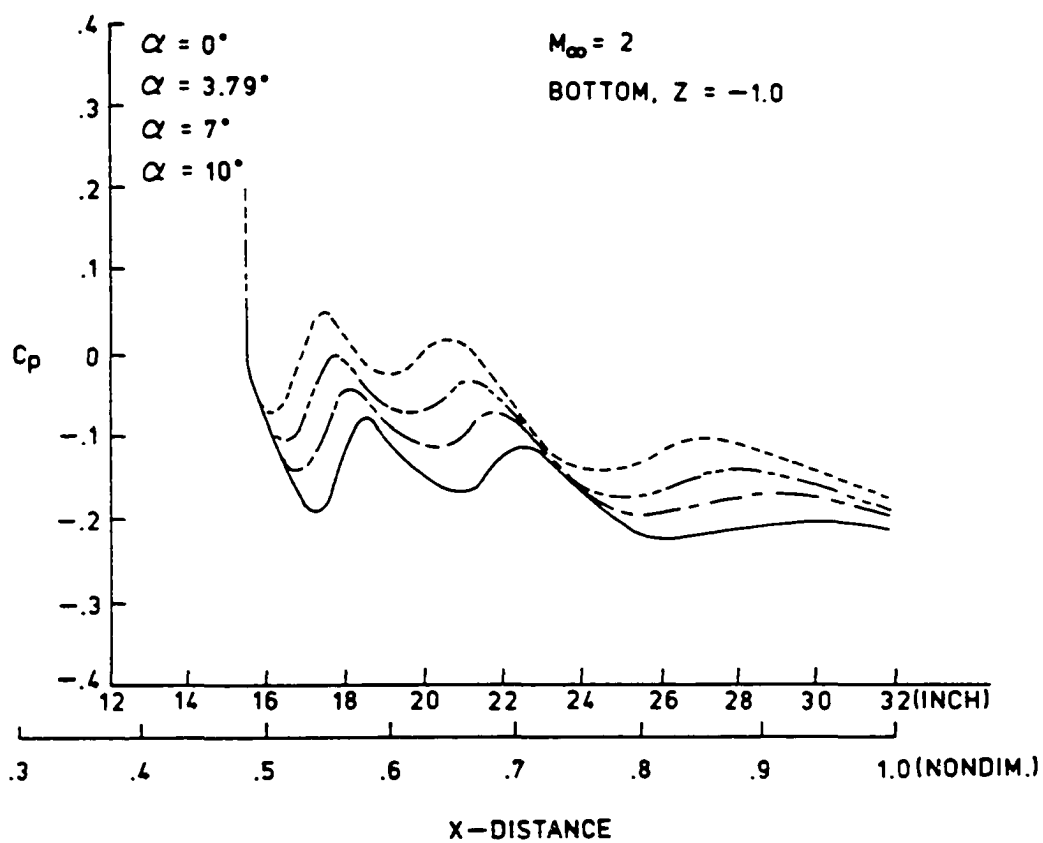


Fig. 4.48 Pressure distribution along bottom walls of the inlet.

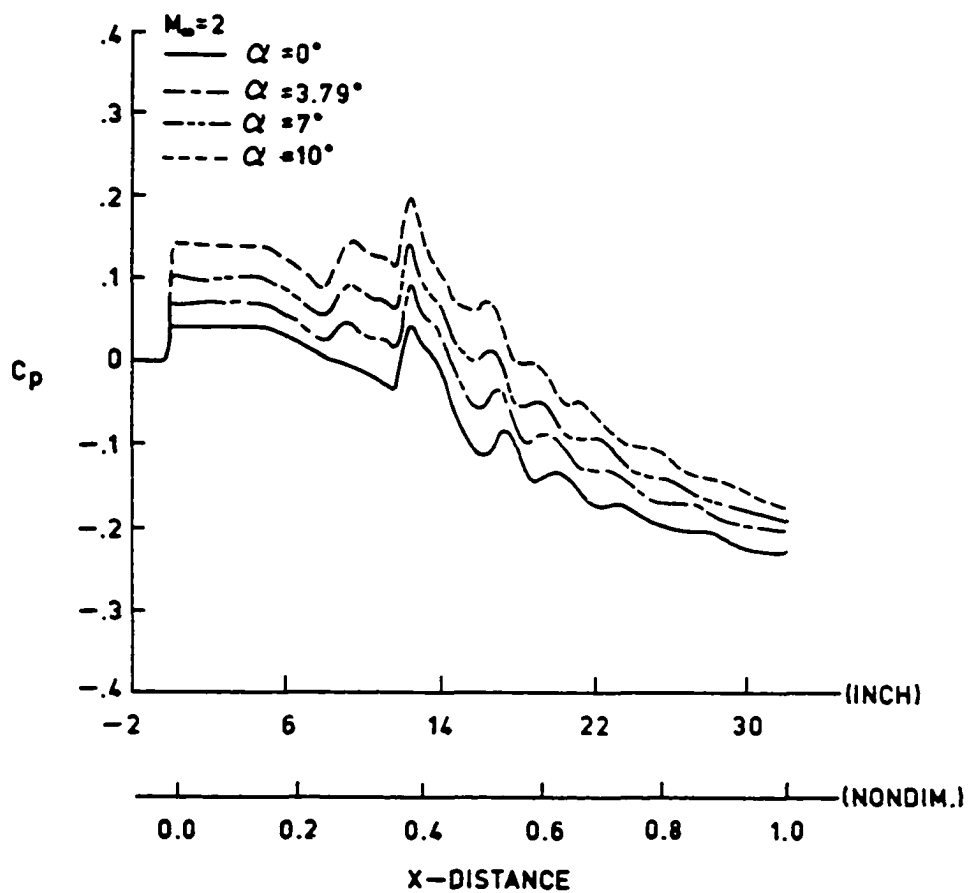


Fig. 4.49 Pressure distribution from freestream to exit plane of the inlet for various angles of attack.

$$\begin{aligned}M_{\infty} &= 2 \\ \alpha &= 0^{\circ} \text{ (With Wake)} \\ \Delta C_p &= 0.025\end{aligned}$$

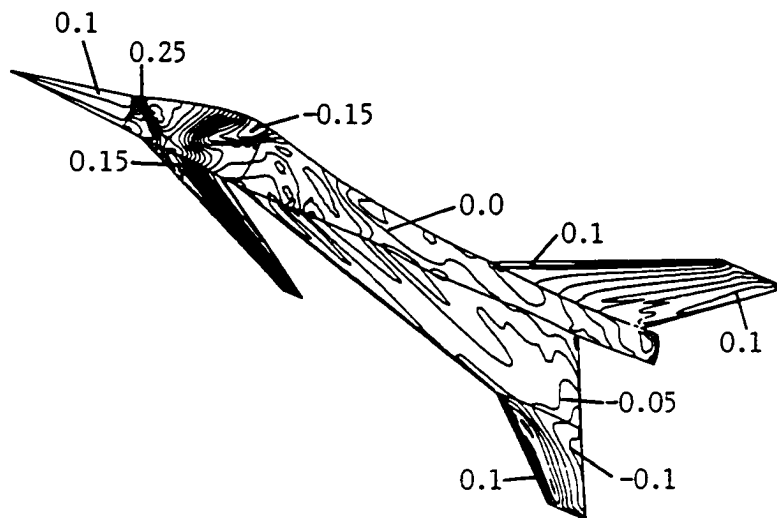


Fig. 4.50 C_p contours on the top surface ($\alpha = 0^{\circ}$, with wake).

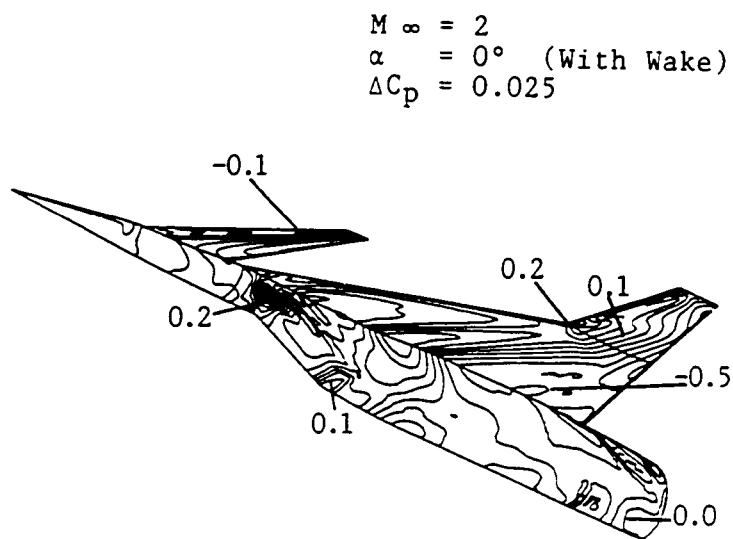


Fig. 4.51 C_p contours on the bottom surface ($\alpha = 0^\circ$, with wake).

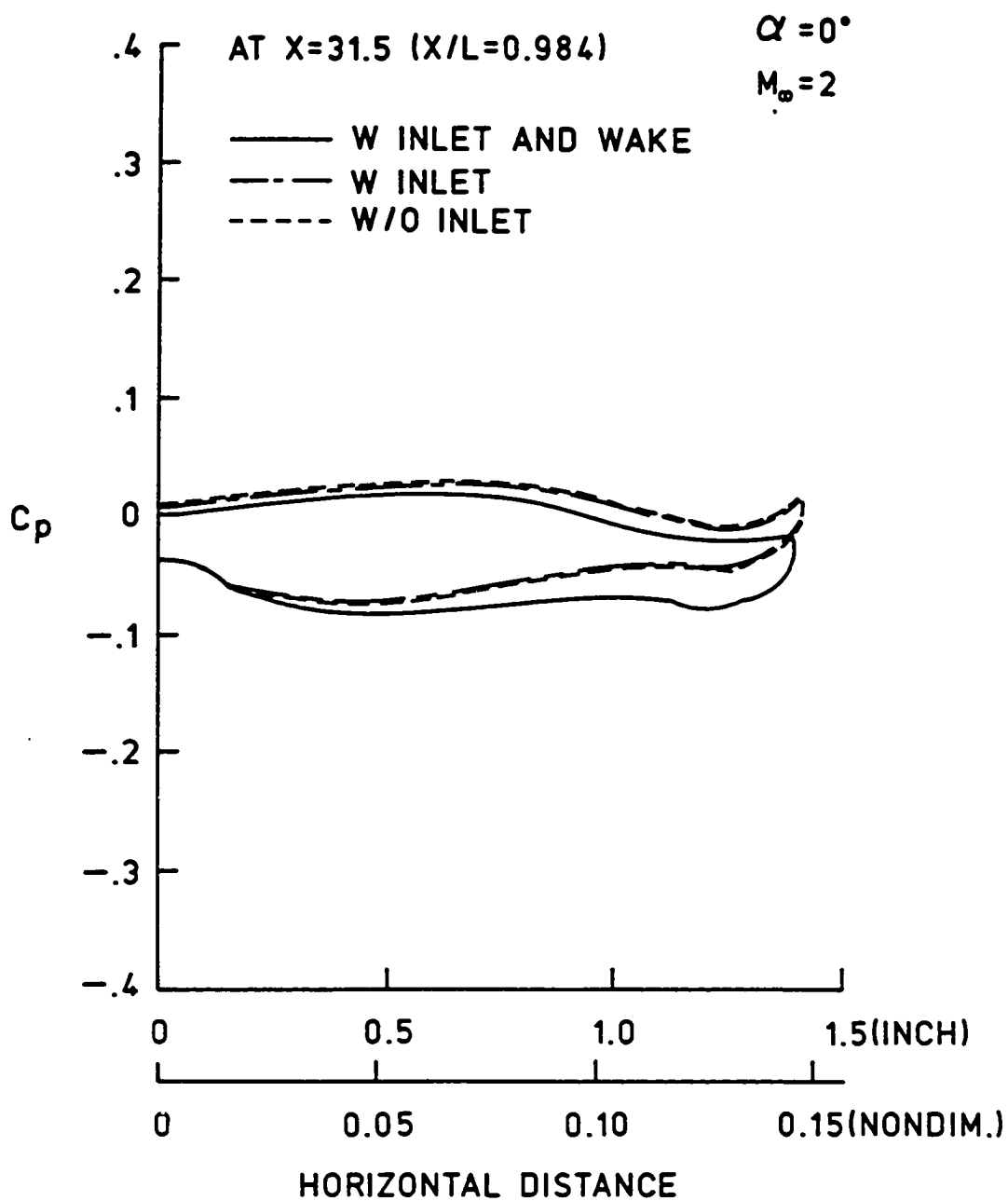


Fig. 4.52 Spanwise pressure distribution near the end of the fuselage ($\alpha = 0^\circ$, with wake).

three different solutions are compared at 98% of the configuration ($x = 31.5$ inch). The solid line represents the case of flow with the inlet and wake region. The other two curves represent the cases of flow without the wake region. The label 'w inlet' on the top right of the figure means with flow in the inlet but without flow in the wake region whereas 'w/o inlet' means without flow in the inlet or in the wake region [152]. The word 'w inlet and wake' simply means with flow in the inlet and in the wake region. At this particular location, the two solutions without flow in the wake region agree very well. However, they indicate a higher pressure than that obtained for the flow with wake region. This is because the outflow boundary condition has to be given right at the boundary of the domain (at the end of the fuselage for the case without wake region). As mentioned in Sec. 3.4.3, this condition arises because the computational domain has to be closed for practical reasons. The farther this boundary moves away from the configuration, the computational model is closer to the real situation. The primary reason that the previous studies do not extend the domain beyond the end of the fuselage is due to the difficulty in generating a grid in that region. Moreover, the condition at the exit plane of the inlet is unknown. The jet afterbody interaction can also influence the flow in the region near the rear fuselage. Indeed, it is the ultimate goal of this study to overcome such difficulties and to be able to model the flow as close to the real situation as possible. The pressure coefficient contours at the plane of symmetry (Fig. 4.53) and at various cross-cuts (Fig. 4.54) indicate a strong interaction between the jet and afterbody flows. Contours in the front of the configuration display the same features as seen in the case of flow without wake region (Sec.4.4.2). In the wake region, the flow coming out of the exit plane of the inlet (jet) has lower pressure but higher velocity than the external flow around the airplane (afterbody flow). As a result of this interaction, the internal flow gets compressed and the external flow is expanded. The shock waves emanating from the rear fuselage meet farther downstream and

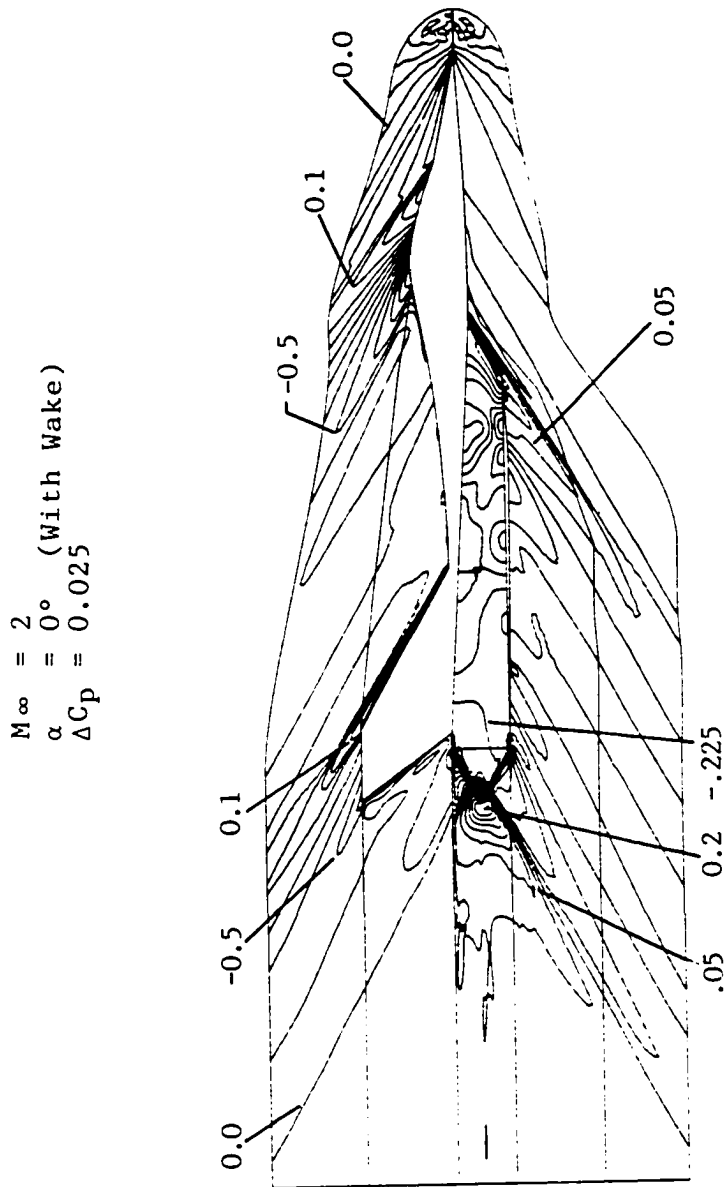


Fig. 4.53 C_p contours on the symmetry plane ($\alpha = 0^\circ$, with wake).

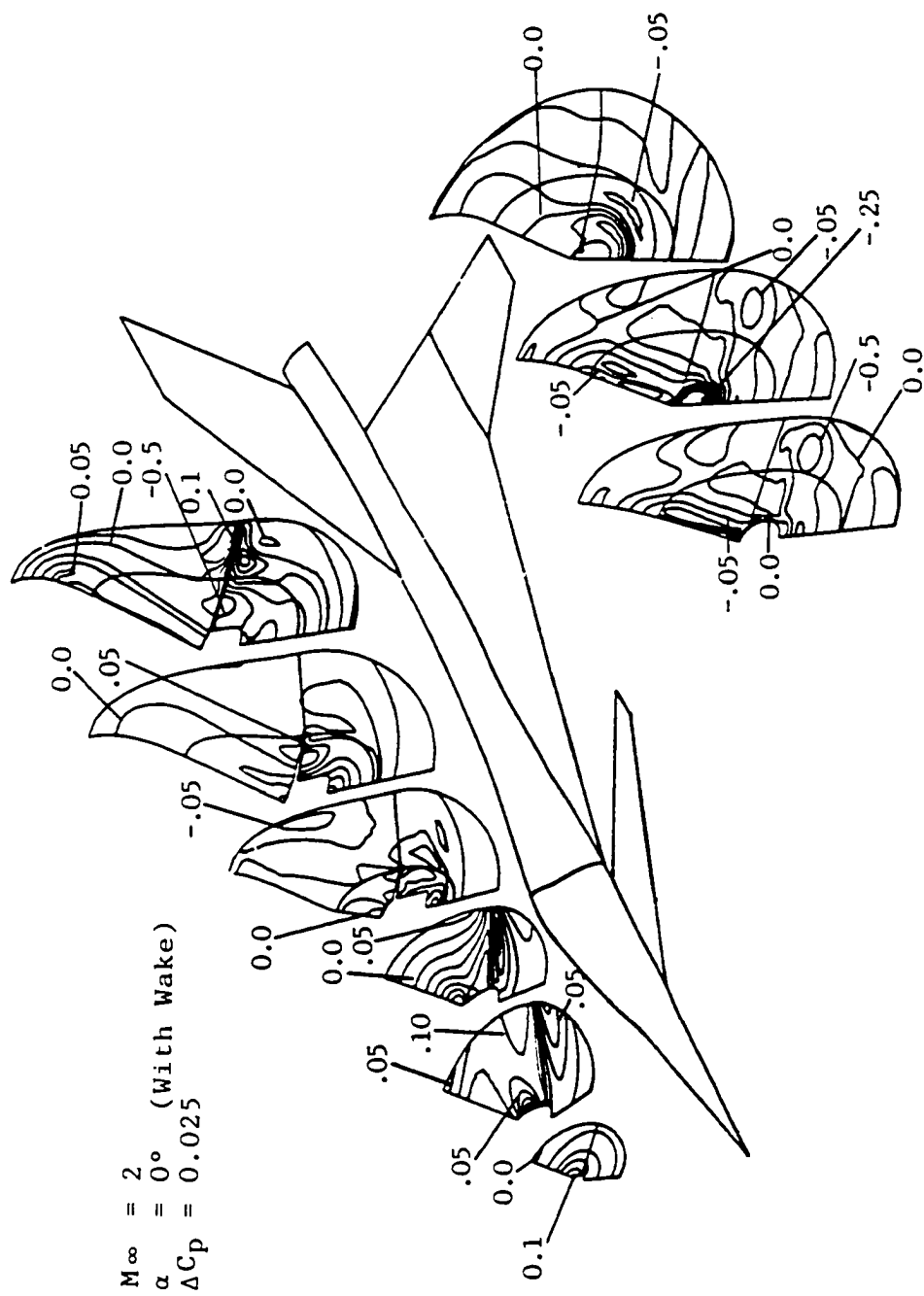


Fig. 4.54 C_p contours at various cross cuts ($\alpha = 0^\circ$, with wake).

reflect off each other. The flows are not symmetric about the middle of the jet because the top of the configuration includes the wings and vertical fin. It should be mentioned that accuracies cannot be claimed for the solutions of flow in the wake region. Realistically, flow in this region becomes turbulent and the Navier-Stokes equations with an appropriate turbulent modeling must be used. This study, however, has taken a step toward that goal since it is now possible to generate grids to cover both internal and external regions.

The pressure coefficient variations at the plane of symmetry are illustrated in Fig. 4.55 for three different locations. The plots start at the inlet intake and end at the end of the domain in the wake region. Inside the inlet, i.e., $12 \leq x \leq 32$ (or $0.4 \leq \frac{x}{L} \leq 1.$), the results are the same as those of the case without the grid in the wake region (Fig. 4.29). The flow expands and shock waves reflect from the inlet walls. At the exit plane of the inlet, the internal flow starts interacting with the external flow from the top and bottom of the inlet. The jet coming out of the center of the inlet, however, does not interact with the afterbody flow until a little farther distance downstream. These are indicated by the jumps of pressures. The flow enters the inlet supersonically and is accelerated as it goes through the inlet. Thus, the flow exits the inlet supersonically with a rather high velocity. This velocity is much higher than that of the afterbody flow. Thus, there exist oblique shock waves and vortical flow in this region. The shock waves meet farther downstream and produce a high pressure region behind them. Thus, the pressure jump at the center line occurs a little farther downstream and its magnitude is relatively higher. It should be noted that the flow in the wake region displays a similar feature as the classical free jet exhausting from a supersonic nozzle. Finally, the pressure distribution from the free-stream to the end of the domain in the wake region at the center of the symmetry plane is shown in Fig. 4.56. This shows the trend in pressure variation for the entire region.

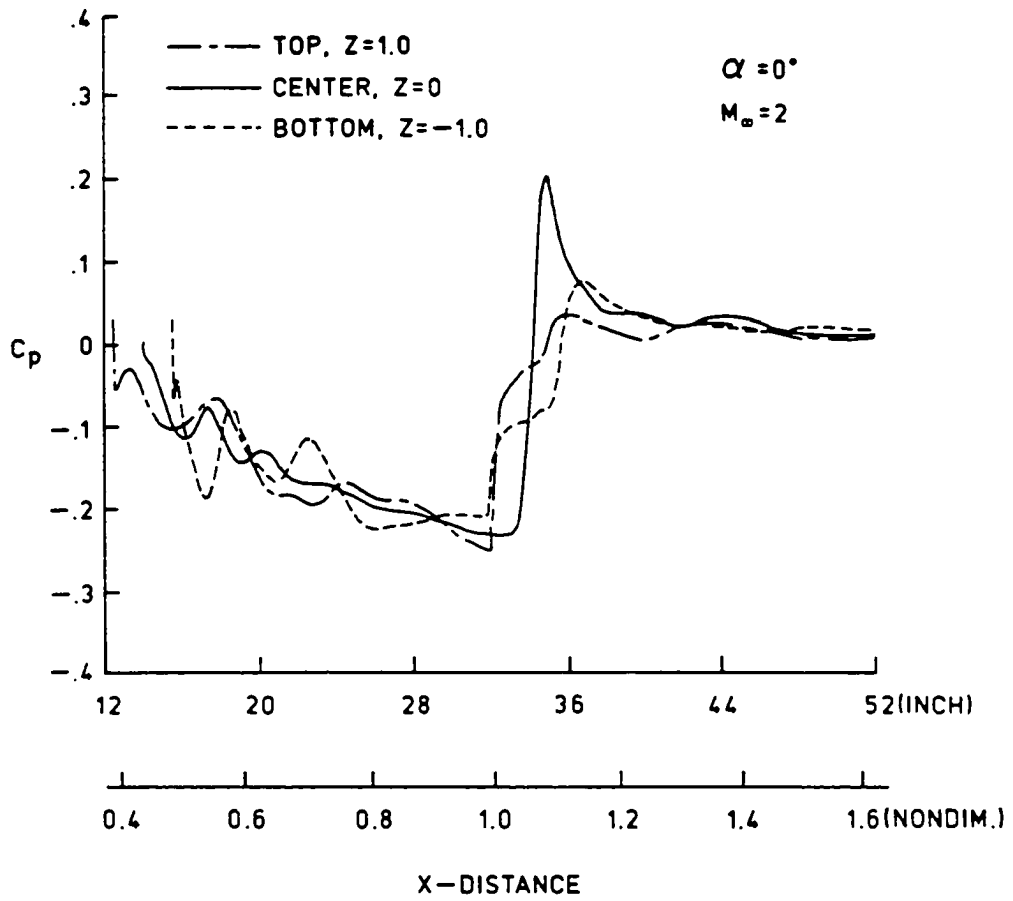


Fig. 4.55 Pressure distribution from the inlet intake to end of the wake region.

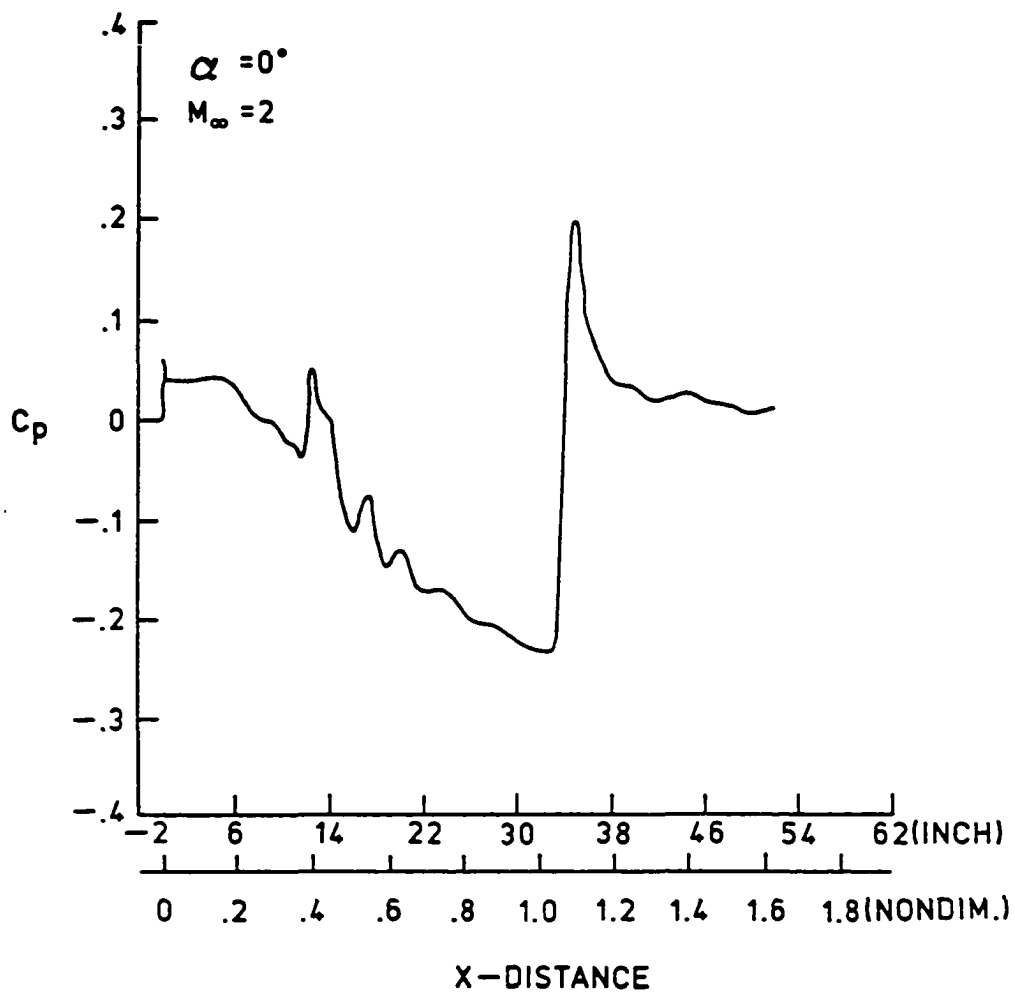


Fig. 4.56 Pressure distribution from freestream to end of the wake region.

Chapter 5

CONCLUSIONS

Solutions of Euler equations for flows over aerodynamics configurations on multiple grids systems are presented. Some details on grid generation techniques and solution procedures are discussed. The concept of conservative treatment at the interfaces of various subdomain grids is also addressed. The solutions obtained from this study illustrate a promising future for the multiple grids approach. It should be stressed, once more, that the main purpose of this study has been to determine whether the use of multiple grids approach is feasible on these configurations, not to determine the characteristics of the flows. The use of multiple grids approach, however, should give the correct characteristics of such flows, as this study indicates. Thus, a single grid system can be constructed to solve for solutions on some of the configurations considered in this study. The solutions obtained from a single grid calculation can be used as references to compare with those obtained from a multiple grids calculations. This fact should not underestimate the usefulness of multiple grids approach. In some instances, for example the fighter aircraft configuration (Sec. 4.4), the construction of a single grid system may not be possible at all. Even for simple configurations where a single grid system can be constructed, the use of multiple grids approach eliminates the difficulties that arise in the grid generation procedures. Also, experiences have shown that the use of multiple grids approach enhances the solution procedures. For example, the convergence to steady state of the solution to the equations of motion depends on many factors. One factor

which is very important is the characteristics of a grid system, i.e., grid spacing, grid orthogonalities etc.. Good characteristics of a grid system are much easier to obtain in the multiple grids approach as compared to a single grid approach. Experience from this study has indicated that efforts made to obtain a converged solution with a multiple grids system is not as much as with a single grid system.

The achievements of this study can be summarized as the following. First, software for computing the conservative interface conditions has been developed. Then, the concept of multiple grids including the conservative interface conditions has been tested by obtaining the solutions to the hyperbolic equations. Satisfactory results have been obtained. The use of the approach to obtain the solutions for flows about simple configurations illustrates that no significant errors have been produced. Finally, the usefulness of the approach have been shown by applying it to obtain the solutions for flows about a complex configuration where the use of a single grid system may not be possible.

Many directions can be taken for future studies. One of the obvious extension of this study is to consider the same configuration as that in Sec. 4.4 and obtain results for different flight regimes. Another possibility is to change the inlet description and put some objects, such as turbine blades, to model some kind of engine. An ambitious extension of this study will be to use the Navier-Stokes equations with an appropriate turbulent model and obtain results for chemically reacting flows through the inlet. However, it is important to note that this suggestion does not imply that the use of the approach will solve all the problem encountered in obtaining solutions for these cases. Of course, to obtain a reasonable solution for any flow condition relies not only on a grid system but also the reasonable model equations. If the complex phenomenon is not modeled properly, it would not yield a reasonable solution no matter how good the grid system is. Nonetheless, this study offers a method which can be used to eliminate one of many difficulties in CFD. It

has been proved that the generation of grids about a geometrically complex configuration is no longer a problem. The only restriction of the method is that it can only be applied to the planar interfaces. The method is not restricted to any flow phenomena. Thus, it can be used to obtain solutions for any flows. However, the realism of such solutions depends also on other factors, such as model equations, etc.. It should also be mentioned that the conservative rezoning algorithm was originally described to solve the problem in the Lagrangian coordinates. This study is believed to be the first to apply it to the problem in the Eulerian coordinates. Recently, others researchers have picked up the idea presented in this study to solve problems in the Eulerian coordinates. Hopefully, this study provide a significant contribution to the field of CFD.

REFERENCES

1. Forrest, A. R., "Computational Geometry," *Proceedings of the Royal Society of London, A* 321, February 1971, pp. 187-195.
2. Orszag, S. A. and Israeli, M., "Numerical Simulation of Viscous Incompressible Flows," *Annual Review of Fluid Mechanics*, Vol. 6, 1974, pp. 281-318.
3. Oswatitsch, K. and Rues, D. (Eds.), *Symposium Transsonicum II*, Springer-Verlag, Berlin/Meidelberg/New York, 1976, pp. 362-574.
4. MacCormack, R. W. and Lomax, H., "Numerical Solution of Compressible Viscous Flows," *Annual Review of Fluid Mechanics*, Vol. 11, 1979, pp. 289-316.
5. Turkel, E., "Progress in Computational Physics," *Computers and Fluids*, Vol. 11, No. 2, 1983, pp. 121-144.
6. Jameson, A., "The Evaluation of Computational Methods in Aerodynamics," *Journal of Applied Mechanics*, Vol. 50, December 1983, pp. 1052-1070.
7. Thompson, J. F. "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," *Applied Numerical Mathematics*, Vol. 1, January 1985, pp. 3-27.
8. MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA paper No. 85-0032, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 1985.
9. Shang, J. S., "An Assessment of Numerical Solutions of the Compressible Navier-Stokes Equation," *Journal of Aircraft*, Vol. 22, No. 5, May 1985, pp. 353-370.
10. Krause, E., "Computational Fluid Dynamics: Its Present Status and Future Direction," *Computers and Fluids*, Vol. 13, No. 3, 1985, pp. 239-269.
11. Eiseman, P. R., "Grid Generation for Fluid Mechanics Computations," *Annual Review of Fluid Mechanics*, Vol. 17, 1985, pp. 487-522.
12. Jameson, A. "Successes and Challenges in Computational Aerodynamics," AIAA paper No. 87-1184, Proceedings of the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 9-11, 1987, pp. 1-35.
13. Rizzi, A. and Engquist, B. "Selected Topics in the Theory and Practice of Computational Fluid Dynamics," *Journal of Computational Physics*, Vol. 72, No. 1, September 1987, pp. 1-69.
14. Serrin, J., *Encyclopedia of Physics, Fluid Dynamics*, Ed. by S. Flugge, Vol. VIII/1. Springer-Verlag, Berlin, 1959.

15. Richtmyer, R. D. and Morton, K. W., *Difference Methods for Initial-Value Problems*, Interscience, New York, 1967.
16. Patankar, S. V. and Spalding, D. B., *Heat and Mass Transfer in Boundary Layers*, 2nd ed., Intertext, New York, 1970.
17. Roache, P. J., *Computational Fluid Dynamics*, Hermosa, Albuquerque, New Mexico, 1972.
18. Richtmyer, R. D., *Principles of Advanced Mathematical Physics*, Vol. I, Springer-Verlag, New York, 1978.
19. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.
20. Peyret, R. and Taylor, T. D., *Computational Methods for Fluid Flow*, Computational Physics Series, Springer-Verlag, New York/Heidelberg/ Berlin, 1983.
21. Hockney, R. W. and Jesshope, C.R., *Parallel Computers*, Adam Hilger, Bristol, 1983.
22. Holt, M., *Numerical Methods in Fluid Dynamics*, Computational Physics Series, 2nd ed., Springer-Verlag, New York/Heidelberg/Berlin, 1984.
23. Gentzsch, W., *Vecterization of Computer Programs with Application to Computational Fluid Dynamics*, Vieweg, Braunschweig, 1984.
24. Anderson, D.A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York 1984.
25. Sod, G. A., "Numerical Methods in Fluid Dynamic," *Initial and Boundary Value Problem*, Cambridge University Press, London, 1985.
26. Eiseman, P. R. "Geometric Methods in Computational Fluid Dynamics," ICASE Report No. 80-11, ICASE/NASA Langley Research Center, Hampton, Virginia, April 1980.
27. Smith, R. E. (Ed.) "Numerical Grid Generation Techniques," NASA CP-2166, October 1980.
28. Thompson, J. F. (Ed.), *Numerical Grid Generation*, Proceedings of a Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Their Use in the Numerical Solution of Partial Differential Equations, North-Holland, New York, April 1982.
29. Thompson, J. F., Warsi, Z.U.A., and Mastin, C. W., "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," *Journal of Computational Physics*, Vol. 47, No.1, July 1982, pp.1-108.
30. Thompson, J. F., Warsi, Z.U.A., and Mastin, C. W., *Numerical Grid Generation-Foundations and Applications*, North-Holland, New York, 1985.
31. Smith, R. E., "Algebraic Mesh Generation for Large Scale Viscous-Compressible Aerodynamic Simulation," *Large Scale Scientific Computation*, Ed. by S. V. Porter, Academic Press, New York, 1986, pp.237-269.

32. Thompson, J. F., "Numerical Solution of Flow Problems Using Body-Fitted Coordinate Systems," Lecture Series in Computational Fluid Dynamics, von Karman Institute for Fluid Dynamics, Belgium 1978, in *Computational Fluid Dynamics*, Ed. by W. Kollmann, Hemisphere, New York, 1980.
33. Mastin, C. W. and Thompson, J. F., "Errors in Finite-Difference Computations on Curvilinear Coordinate System," Mississippi State University, MSSU-EIRS-ASE-80-4, 1980.
34. Shang, J. S., "Numerical Simulation of Wing-Fuselage Interference," AIAA Paper No. 81-0068, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January 12-15, 1981.
35. Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," *Journal of Computational Physics*, Vol. 33, No. 1, October 1979, pp.307-338.
36. Gordon, W. J., and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Application to Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol.7, No. 4, 1973, pp.461-477.
37. Eriksson, L. E., "Three-Dimensional Spline-Generated Coordinate Transformations for Grid Around Wing-Body Configurations," in Numerical Grid Generation Techniques, NASA CP-2166, September 1980, pp. 253-264.
38. Rizzi, A., and Eriksson, L. E., "Transfinite Mesh Generation and Damped Euler Equation Algorithm for Transonic Flow Around Wing-Body Configurations," AIAA Paper No. 81-0999, Proceedings of the AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, California, June 22-23, 1981, pp. 43-68.
39. Eiseman, P. R., "Orthogonal Grid Generation" in *Numerical Grid Generation*, Ed. by J. F. Thompson, North-Holland, New York, 1982, pp. 193-233.
40. Eiseman, P. R., "Three-Dimensional Coordinate About Wings," AIAA Paper No. 79-1461, Proceedings of the 4th AIAA Computational Fluid Dynamics Conference, Williamsburg, Virginia, July 23-25, 1979, pp. 166-174.
41. Smith, R. E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Navier-Stokes Equations," Ph.D. Dissertation, Old Dominion University, Norfolk, Virginia, May 1981.
42. Smith, R. E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Navier-Stokes Equations," NASA TM-83123, May 1981.
43. Smith, R. E. and Weigel, B. L., "Analytic and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation," AIAA Paper No.80-0192, AIAA 18th Aerospace Sciences Meeting, Pasadena, California, January 1980.
44. Winslow, A. M., "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh," *Journal of Computational Physics*, Vol.1, No. 2, November 1966, pp. 149-172.
45. Thompson, J. F., Thames, F. C., and Mastin, C. W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, No. 3, July 1974, pp. 299-319.

46. Thompson, J. F., Thames, F. C., and Mastin, C. W., "TOMCAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 24, No. 3, July 1977, pp.274-302.
47. Mastin, C. W. and Thompson, J. F., "Elliptic Systems and Numerical Transformations," *Journal of Mathematical Analysis and Applications*, Vol. 62, No. 1, January 1978, pp. 52-62.
48. Mehta, U. B and Lavan, Z., "Starting Vortex, Separation Bubbles and Stall: a Numerical Study of Laminar Unsteady Flow around an Airfoil," *Journal of Fluid Mechanics*, Vol.67, Part 2, January 1975, pp. 227-256.
49. Wu, J. C. and Sampath,S., "A Numerical Study of Viscous Flow around an Airfoil," AIAA Paper No. 76-337, AIAA 9th Fluid and Plasma Dynamics Conference, San Diego, California, July 14-16, 1976.
50. Wu, J. C., El-Refaee, M., and Lekoudis, S. G., "Solutions of the Unsteady Two-Dimensional Compressible Navier-Stokes Equations Using the Integral Representation Method," AIAA Paper No. 81-0046, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January 12-15, 1981.
51. Napolitano, M., Werle, M. J., and Davis, R. T., "Numerical Solutions for High Reynolds Number Separated Flow Past a Semi-Infinite Compression Corner," *Computers and Fluids*, Vol. 7, No. 3, September 1979, pp. 165-175.
52. Starius, G., "Construction of Orthogonal Curvilinear Meshes by Solving Initial Value Problems," *Numerische Mathematik*, Vol. 28, Fasc. 1, 1977, pp.25-48.
53. Steger, J. L. and Sorenson, R. L., "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates," NASA CP-2166, September 1980, pp. 413-478.
54. Nakamura, S., "Marching Grid Generation Using Parabolic Partial Differential Equations, in *Numerical Grid Generation*, Ed. by J. F. Thompson, North-Holland, New York, 1982, pp. 775-786.
55. Nakamura, S., "Non-Iterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations," Proceeding of the Eighth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, Berlin, 1982.
56. Edwards, T. A., "Three-Dimension Grid Generation Using Parabolic Partial Differential Equations," AIAA Paper No. 85-0485, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 1985.
57. Nakamura, S. and Suzuki, M., "Non-Iterative Three Dimensional Grid Generation Using a Parabolic-Hyperbolic Hybrid Scheme," AIAA Paper No.87-0277, AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 1987.
58. Berglind, T., "A Comparison of Single-Block and Multi- Block Grids around Wing-Fuselage Configurations," FFA TN-1986- 42, The Aeronautical Research Institute of Sweden (FFA), Stockholm, Sweden, 1986.
59. Smith, R. E., "Algebraic Grid Generation about Wing- Fuselage Bodies," paper presented at the 15th Congress International Council of the Aeronautical Sciences, London, UK, September 1986.

60. Coons, S. A., "Surfaces for Computer Aided Design of Space Forms," MAC-TR-41, Project MAC, Design Division, Department of Mechanical Engineering, M.I.T., Cambridge, Massachusetts, June 1967.
61. Gordon, W. J., "Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation," *SIAM Journal on Numerical Analysis*, Vol. 8, No. 1, March 1971, pp.158-177.
62. Gerhard, M. A., "A General Purpose Mesh Generator for Finite Element Codes," M-101, Lawrence Livermore Lab., University of California, Livermore, California, April 1979.
63. Anderson, P. G., and Spradley, L. W., "Finite Difference Grid Generation by Multivariate Blending Function Interpolation," in *Numerical Grid Generation Techniques*, NASA CP-2166, September 1980, pp. 143-156.
64. Spradley, L. W., Stalnaker, J. F., and Ratliff, A. W., "Solution of the Three-Dimensional Navier-Stokes Equations on a Vector Processor," *AIAA Journal*, Vol. 19, No. 10, October 1981, pp. 1302-1308.
65. Eriksson, L. E., "Generation of Boundary-Conforming Grids around Wing-Body Configurations Using Transfinite Interpolation," *AIAA Journal*, Vol. 20, No. 10, October 1982, pp. 1313-1320.
66. Eriksson, L. E. and Rizzi, A. W., "Computation of Vortex Flow around Wings Using the Euler Equations," *Proceedings of the Fourth GAMM-Conference on Numerical Methods in Fluid Mechanics*, Ed. by H. Viviani, Vieweg Verlag, Paris, October 1981.
67. Eriksson, L. E., "Practical Three-Dimensional Mesh Generation Using Transfinite Interpolation," *SIAM Journal on Scientific and Statistical Computing*, Vol. 6, No.3, July 1985, pp. 712-741.
68. Atta, E., "Component-Adaptive Grid Interfacing," AIAA paper No. 81-0382, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January 1981.
69. Miki, K. and Takagi, T., "A Domain Decomposition and Overlapping Method for the Generation of Three-Dimensional Boundary-Fitted Coordinate Systems," *Journal of Computational Physics*, Vol. 53, No.2, February 1984, pp. 319-330.
70. Weatherill, N. P. and Forsey, C. R., "Grid Generation and Flow Calculations for Complex Aircraft Geometries Using a Multi-Block Scheme," AIAA Paper No. 84-1665, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.
71. Thomas, P. D., "Composite Three-Dimensional Grid Generated by Elliptic System," *AIAA Journal*, Vol. 20, No. 9, September 1982, pp. 1195-1202.
72. Rubbert, P. E. and Lee, K. D., "Patched Coordinate Systems," in *Numerical Grid Generation*, Ed. by J. F. Thompson, North-Holland, 1982, pp. 235-252.
73. Kathong, M., Smith, R. E., and Tiwari, S. N., "A Conservative Approach for Flow Field Calculation on Multiple Grids," AIAA Paper No. 88-0224, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11-14, 1988.

74. Lasinski, T. A., Andrews, A. E., Sorenson, R. L., Chaussee, D. S., Pulliam, T. H., and Kutler, P., "Computation of the Steady Viscous Flow Over a Tri-Element Augmented Wing Airfoil," AIAA Paper No. 82-0021, January 1982.
75. Szema, K. Y., Chakravarthy, S. R., Riba, W. T., Byerly, J., and Dresser, H. S., "Multi-Zone Euler Marching Technique for Flow over Single and Multi-Body Configurations," AIAA Paper No. 87-0592, AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 1987.
76. Rai, M. M., "An Implicit, Conservative, Zonal-Boundary Scheme for the Euler Equations," AIAA Paper No. 85-0488, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 14-17, 1985.
77. Rai, M. M., "A Relaxation Approach to Patched-Grid Calculations with the Euler Equations," AIAA Paper No. 85-0295, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 14-17, 1985.
78. Venkatapathy, E. and Lombard, C. K., "Flow Structure Capturing on Overset Patched Meshes," AIAA Paper No. 85-1690, AIAA 18th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Cincinnati, Ohio, July 16-18, 1985.
79. Starius, G., "Composite Mesh Difference Methods for Elliptic Boundary Value Problems," *Numerische Mathematik*, Vol. 28, Fasc. 2, 1977, pp. 243-258.
80. Starius, G., "On Composite Mesh Difference Methods for Hyperbolic Differential Equations," *Numerische Mathematik*, Vol. 35, Fasc. 3, 1980, pp. 241-255.
81. Kreiss, B., "Construction of a Curvilinear Grid," *SIAM Journal on Scientific and Statistical Computing*, Vol. 4, No. 2, June 1983, pp. 270-279.
82. Mastin, C. W. and McConnaughey, H. V., "Computational Problems on Composite Grids," AIAA Paper No. 84-1611, AIAA 17th Fluid Dynamics, Plasma Dynamics and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.
83. Thompson, D. S., "A Mesh Embedding Approach for Prediction of Transonic Wing/Body/Store Flow Fields," NASA CP-2201, October 1981, pp. 403-427.
84. Steger, J. A. and Buning, P. G., "Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers," NASA TM-86676, February 1985.
85. Benek, J. A., Buning, P. G., and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper No. 85-1523, Proceeding of the AIAA 7th Computational Fluid Dynamic Conference, Cincinnati, Ohio, July 15-17, 1985.
86. Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," Mini Symposium on Advances in Grid Generations, ASME Applied Bioengineering and Fluids Engineering Conference, Houston, Texas, June 1983.
87. Benek, J. A., Steger, J. L., and Dougherty, F. C., "A Flexible Grid Embedding Technique with Application to the Euler Equations," AIAA Paper No. 83-1944, Proceeding of the AIAA 6th Computational Fluid Dynamics Conference, July 1983, pp. 373-382.
88. Atta, E. H. and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flow About Aircraft Configurations," AIAA Paper No. 82-1017, AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma, and Heat Transfer Conference, St. Louis, Missouri, June 7-11, 1982.

89. Dougherty, F. C., "Development of a Chimera Grid Scheme with Application to Unsteady Problem," Ph.D. Thesis, Department of Aeronautics and Astronautic, Stanford University, Stanford, California, May 1985.
90. Boppe, C. W. and Stern, M. A., "Simulated Transonic Flows for Aircraft with Nacelles, Pylons, and Winglets," AIAA Paper No. 80-0130, AIAA 18th Aerospace Sciences Meeting, Pasadena, California, January 14-16, 1980.
91. Shankar, V. and Malmuth, N. D., "Computational and Simplified Analytical Treatment of Transonic Wing-Fuselage-Pylon-Store Interactions," AIAA paper no. 80-0127, AIAA 18th Aerospace Sciences Meeting, Pasadena, California, January 14-16, 1980.
92. Srokowski, A. J., Shrewsbury, G. A., and Lores, M. E., "A Transonic Mutual Interference Program for Computing the Flow about Wing-Pylon/Nacelle Combinations," AIAA paper No. 80-1333, AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, July 14-16, 1980.
93. Yu, N. J., "Transonic Flow Simulation for Complex Configurations with Surface Fitted Grids," AIAA paper No. 81-1258, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, California, June 23-25, 1981.
94. Chanderjan, N. M. and Steger, J. L., "A Zonal Approach for the Steady Transonic Simulation of Inviscid Rotational Flow," AIAA Paper No. 83-1927, Proceeding of the AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts, July 1983.
95. Eberhardt, S. and Baganoff, D., "Overset Grids in Compressible Flow," AIAA Paper No. 85-1524, Proceeding of the AIAA 7th Computational Fluid Dynamic Conference, Cincinnati, Ohio, July 15-17, 1985, pp. 332-339.
96. Lax, P., *Hyperbolic System of Conservation Laws and the Mathematical Theory of Shock Waves*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1972.
97. Berger, M. J., "On Conservation at Grid Interfaces," ICASE/NASA Langley Research Center, Hampton, Virginia, ICASE Report No.84-43, September 1984.
98. Warming, R. and Beam, R., "Upwind Second-Order Difference Schemes and Applications in Aerodynamic Flows," *AIAA Journal*, Vol. 14, No. 9, September 1976, pp. 1241-1249.
99. Hennesius, K. and Pulliam, T., "A Zonal Approach to Solution of the Euler Equations," AIAA Paper No. 82-0969, AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, June 7-11, 1982.
100. Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA Paper No. 84-0164, AIAA 22nd Aerospace Sciences Meeting, Reno, Nevada, January 9-12, 1984.
101. Cambier, L., Ghazzi, W., Veillot, J. P., and Viviand, H., "Une Approche par Domaines pour le calcul d'écoulements Compressibles," Cinquieme Colloque International sur les Methodes de calcul, France, December 1981.

102. Rai, M. M., Chakravarthy, S.R., and Hennesius, K. A., "Zonal Grid Calculations Using the Osher Scheme," *Computers and Fluids*, Vol. 12, No. 3, 1984, pp. 161-173.
103. Dukowicz, J. K., "Conservative Rezoning (Remapping) for General Quadrilateral Meshes," *Journal of Computational Physics*, Vol. 54, No.3, June 1984, pp. 411-424.
104. Ramshaw, J. D., "Conservative Rezoning Algorithm for Generalized Two-Dimensional Meshes," *Journal of Computational Physics*, Vol. 59, No. 2, June 1985, pp.193-199.
105. Eriksson, L. E., "Flow Solution on a Dual-Block Grid around an Airplane," *Computer Methods in Applied Mechanics and Engineering*, Vol.64, Nos. 1-3, October 1987, pp. 79-93.
106. Rektorys, K. (Ed.), *Survey of Applicable Mathematics*, Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1969, p. 207.
107. Kathong, M. and Tiwari, S. N., "Conservative Finite Volume Solutions of a Linear Hyperbolic Transport Equation in Two and Three Dimensions Using Multiple Grids," Department of Mechanical Engineering and Mechanics, College of Engineering and Technology, Old Dominion University, Norfolk, Virginia, Progress Report, NCC1-68, July 1987.
108. Hafez, M. and Lovell, D., "Numerical Solution of Transonic Stream Function Equation," *AIAA Journal*, Vol. 21, No. 3, March 1983, pp. 327-335.
109. Murman, E. M. and Cole, J. D., "Calculation of Plane, Steady Transonic Flows," *AIAA Journal*, Vol.9, No. 1, January 1971, pp. 114-121.
110. Jameson, A. "Iterative Solution of Transonic Flow Over Airfoils and Wings Including Flows at Mach 1," *Communications on Pure and Applied Mathematics*, Vol.27, No. 3, May 1974, pp. 283-304.
111. Hafez, M., South, J., and Murman, E., "Artificial Compressibility Methods for Numerical Solution of Transonic Full Potential Equations," *AIAA Journal*, Vol. 17, No. 8, August 1979, pp. 838-844.
112. Jameson, A., "Transonic Potential Flow Calculations in Conservation Form," Proceedings of the AIAA Second Computational Fluid Dynamics Conference, Hartford, Connecticut, June 19-20, 1975, pp.148-161.
113. MacCormack, R. L., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper No. 69-354, AIAA Hypervelocity Impact Conference, Cincinnati, Ohio, April 30-May 2, 1969.
114. McDonald, P. W., "The Computation of Transonic Flow Through Two Dimensional Gas Turbines Cascades," ASME Paper 71-GT-89, ASME Gas Turbine Conference and Products Show, Houston, Texas, March 28-April, 1971.
115. Denton, J. D., "A Time Marching Method for Two- and Three-Dimensional Blade-to-Blade Flow," Aeronautical Research Council Reports and Memoranda, No. 3775, United Kingdom, 1975.
116. Moretti, G., "The λ - Scheme" *Computers and Fluids*, Vol. 7, No. 3, September 1979, pp. 191-205.

117. Chakravarthy, S. R. Anderson, D. A., and Salas, M. D., "The Split-Coefficient Matrix Method for Hyperbolic Systems of Gas Dynamic Equations," AIAA Paper No. 80-0268, AIAA 18th Aerospace Sciences Meeting, Pasadena, California, January 22-23, 1980.
118. Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," AIAA Paper No. 81-1025, AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, June 22-23, 1981.
119. Rizzi, A. and Eriksson, L. E., "Computational of Flow around Wings Based on the Euler Equations," *Journal of Fluid Mechanics*, Vol. 148, March 1984, pp. 45-71.
120. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper No. 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, California, June 22-25, 1981.
121. Eriksson, L. E. and Rizzi, A., "Computation of Vortex Flow around a Canard/Delta Combination," *Journal of Aircraft*, Vol. 21, No. 11, November 1984, pp. 858-865.
122. Rizzi, A. and Eriksson, L. E., "Computation of Inviscid Incompressible Flow with Rotation," *Journal of Fluid Mechanics*, Vol.153, October 1985, pp. 275-312.
123. Eriksson, L. E., "Simulation of Inviscid Flow around Airfoils and Cascades Based on the Euler Equations," FFA TN-1985-20, The Aeronautical Research Institute of Sweden (FFA), Stockholm, Sweden, 1985.
124. Eriksson, L. E., "Simulation of Transonic Flow in Radial Compressors," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, Nos. 1-3, October 1987, pp. 95-111.
125. Viviand, H. "Pseudo-Unsteady Methods for Transonic Flow Computation," Proceedings of the 7th International Conference on Numerical Methods in Fluid Dynamics, Ed. by W. Reynolds, Lecture Notes in Physics, Springer Verlag, 1981.
126. Rizetta, D. P. and Shang, J. S., "Numerical Simulation of Leading Edge Vortex Flows," AIAA Paper No. 84-1544, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.
127. Anderson, J. D., *Modern Compressible Flow*, McGraw-Hill, New-York, 1982, pp. 165-166.
128. Karamcheti, K., *Principles of Ideal Fluid Aerodynamics*, Krieger, Malabar, Florida, 1966, pp. 390-393.
129. Engquist, B. and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, July 1977, pp. 629-651.
130. Schmidt, W. and Jameson, A., "Recent Developments in Finite-Volume Time-Dependent Techniques for Two and Three Dimensional Transonic Flows," in Computational Fluid Dynamics, von Karman Institute for Fluid Dynamics, Brussels, Lecture Series Notes VKI-LS-1982-04, 1982.

131. Rizzi, A. and Eriksson, L. E., "Explicit Multistage Finite Volume Procedure to Solve the Euler Equations for Transonic Flow," in *Computational Fluid Dynamics*, von Karman Institute for Fluid Dynamics, Brussels, Lecture Series Notes VKI-LS-1983-04, 1983.
132. Rizzi, A., Eriksson, L. E., Schmidt, W., and Hitzel, S., "Numerical Solutions of the Euler Equations Simulating Vortex Flows around Wings," AGARD-CPP-342, Advisory Group for Aeronautical Research and Development, Paris, 1983.
133. Eriksson, L. E., "A Study of Mesh Singularities and Their Effects on Numerical Errors," FFA TN-1984-10, The Aeronautical Research Institute of Sweden (FFA), Stockholm, Sweden, 1984.
134. Lomax, H., "Some Prospects for the Future of Computational Fluid Dynamics," *AIAA Journal*, Vol. 20, No. 8, August 1982, pp. 1033-1043.
135. MacCormack, R. W. and Paullay, A. J., "The Influence of the Computational Mesh on Accuracy for Initial Value Problems with Discontinuous or Nonunique Solutions," *Computers and Fluids*, Vol. 2, Nos. 3-4, December 1974, pp.339-361.
136. Rizzi, A. and Inouye, M., "Time Split Finite-Volume Method for Three-Dimensional Blunt-Body Flow," *AIAA Journal*, Vol.11, No. 11, November 1973, pp. 1478-1485.
137. Hennesius, K. A. and Rai, M. M., "Three-Dimensional, Conservative, Euler Computations Using Grid Systems and Explicit Methods," AIAA Paper No. 86-1081, AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics, and Lasers Conference, Atlanta, Georgia, May 12- 14, 1986.
138. Newman, W. M. and Sproull, R. F., *Principle of Interactive Computer Graphics*, McGraw-Hill, New York, 1979, pp. 69-72.
139. Walters, R. W., Reu, T., McGrory, W. D., Thomas, J. L., and Richardson, P. F., "Longitudinally-Patched Grid Approach with Applications to High Speed Flows," AIAA Paper No. 88-0715, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11-14, 1988.
140. Gary, J., "On certain Finite-Difference Schemes for Hyperbolic Systems," *Mathematics of Computation*, Vol. 18, No. 85, January 1964, pp.1-18.
141. Courant, R., Friedrichs, K. O., and Lewy, H., "Über die partiellen Differenzgleichungen der mathematischen Physik," *Mathematische Annalen*, Vol. 100, 1928, pp.32-74.
142. Denton, J. D., "Time-Marching Methods for Turbomachinery Flow Calculation," *Numerical Methods in Applied Fluid Dynamics*, Ed. by B. Hunt, Academic Press, London, 1980.
143. Goldstein, S., *Modern Developments in Fluid Dynamics*, Vol. I, Dover, New York, 1965, pp. 25.
144. Moretti, G. and Abbett, M., "A Time-Dependent Computational Method for Blunt Body Flows," *AIAA Journal*, Vol. 4, No. 12, December 1966, pp. 2136-2141.

145. Butler, D. S., "The Numerical Solution of Hyperbolic System of Partial Differential Equations in Three Independent Variables," *Proceedings of The Royal Society, Series A, Mathematical and Physical Sciences*, No.1281, Vol. 255, April 1960, pp. 232-252.
146. Squire, L. C., "Measured Pressure Distributions and Shock Shapes on a Simple Delta Wing," *The Aeronautical Quarterly*, Vol. 32, Part 3, August 1981, pp.188-198.
147. Squire, L. C., "Measured Pressure Distribution and Shock Shapes on a Simple Delta Wing," Cambridge University, Department of Engineering, CUED/Aero/tr 9, 1979.
148. Abolhassani, J. S., Smith, R. E., and Tiwari, S. N., "Numerical Solutions of Navier-Stokes Equations for a Butler Wing," AIAA Paper No.87-0115, AIAA 25th Aerospace Sciences Meeting, January 1987.
149. Jameson, A. and Baker, T. J., "Multigrid Solution of the Euler Equations for Aircraft Configurations," AIAA Paper No. 84- 0093, January 1984.
150. Berger, M. J. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," Department of Mechanical and Aerospace Engineering Report No. 1633, Princeton University, Princeton, New Jersey, October 1983.
151. Eriksson, L. E., "Euler Solutions on O-O Grids around Wings Using Local Refinement," Proceeding of the 6th GAMM Conference on Numerical Methods in Fluid Mechanics, Ed. by D. Rues and W. Kordulla, Vieweg Verlag, Gottingen, September 25-27, 1985.
152. Eriksson, L. E., Smith, R. E., Wiese, M. R., and Farr, N., "Arid Generation and Inviscid Flow Computation about Cranked-Winged Airplane-Geometries," AIAA paper No. 87-1125, Proceedings of the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 9-11, 1987.
153. Karman, S. L., Jr., Steinbrenner, J. P., and Kisielewski, K. M., "Analysis of the F-16 Flow Field by a Block Grid Euler Approach," paper presented at the 58th Meeting of the Fluid Dynamics Panel Symposium on Applications of Computational Fluid Dynamics in Aeronautics, Aix-En-Provence, France, April 7-10, 1986.
154. Fritz, W. and Leicher, S., "Numerical Solution of 3-D Inviscid Flow Fields around Complex Aircraft Configurations," paper presented at the 15th Congress International Council of the Aeronautical Sciences, London, UK, September 1986.
155. Szema, K. Y. and Shankar, V., "Validation of a Full Potential Method for Combined Yaw and Angle of Attack," AIAA Paper No. 86-1834, June 1986.
156. Szema, K. Y. and Chakravarthy, S. R., "Multi-Zone Euler Marching Technique for Flow Over Single and Multi-Body Configurations," AIAA Paper No. 87-0592, January 1987.
157. Hom, K. W. and Ticatch, L. A., "Investigation of an Advanced Supersonic Fighter Concept Including Effects of Horizontal Tail and Canard Control Surfaces Over a Mach Number Range From 1.6 to 2.5," NASA TP-2526, 1986.

Appendix A

THE IMPLEMENTATION OF THE CONSERVATIVE REZONING ALGORITHM

In Sec. 2.5, the concept of the conservative rezoning algorithm is described only briefly. It is felt that more details of the algorithm including its implementation must also be given. This appendix is the appropriate place to do so. For illustration, consider a patched planar interface shown in Fig. A.1. The dash lines represent grid lines of the old grid cell with the known quantities, i.e., Q_O 's, which are constant within each grid cell. The solid lines represent grid lines of the new grid cell with the unknown quantities, i.e., Q_N 's, to be obtained from Q_O 's conservatively. It is clearly seen from Fig. A.1 that the quantity within a particular new grid cell can be obtained conservatively according to the equation,

$$Q_N = \sum_{i=1}^{N_P} (Q_{O_i}) \frac{A_{NO_i}}{A_{O_i}} = \sum_{i=1}^{N_P} (q_{O_i}) (A_{NO_i}) \quad (\text{A.1})$$

where A_{NO_i} is the portion of the area A_{O_i} which overlaps with the area A_N . The quantity N_P is the number of A_{NO_i} 's that is contained in the area A_N , and $q_{O_i} = \frac{Q_{O_i}}{A_{O_i}}$. It can be seen that A_{NO_i} are polygons with various number of sides. According to Sec. 2.5 the area of a polygon in a two dimensional plane is given by

$$A_p = \frac{1}{2} \sum_{s=1}^{N_s} \epsilon_s^p (x_1^s y_2^s - x_2^s y_1^s) \quad (\text{A.2})$$

where (x_1^s, y_1^s) and (x_2^s, y_2^s) are coordinates of the two end points of a particular line segment s , N_s is the number of sides of a particular polygon and ϵ_s^p is +1 or -1

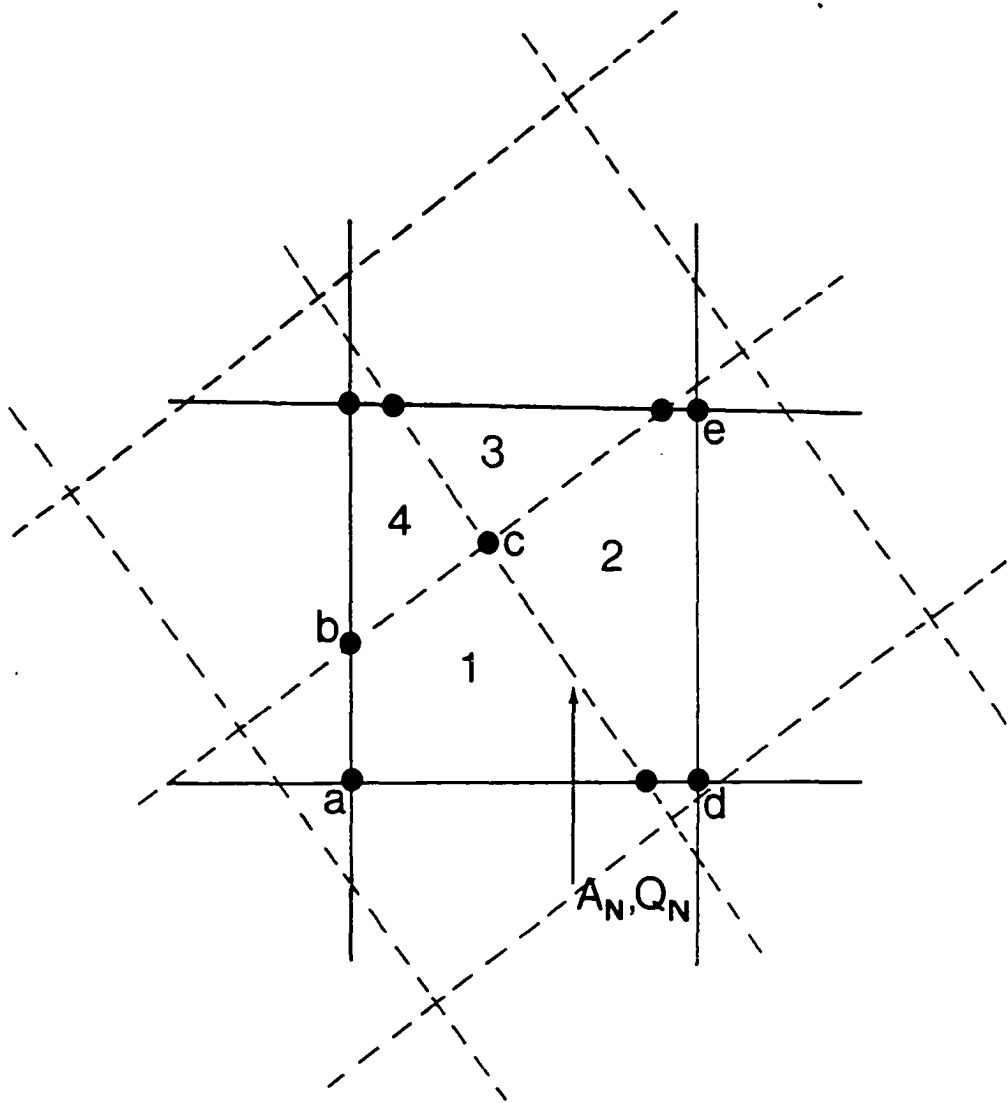


Fig. A.1 A patched planar interface.

according to whether the polygon lies on the left or right, respectively, of the line segment s when one travels from (x_1^s, y_1^s) to (x_2^s, y_2^s) . Thus, the quantity contained in the area A_N , shown in Fig. A.1, is obtained via Eq.(A.1) as

$$Q_N = q_{O_1}A_{NO_1} + q_{O_2}A_{NO_2} + q_{O_3}A_{NO_3} + q_{O_4}A_{NO_4}. \quad (\text{A.3})$$

Substituting Eq.(A.2) into Eq.(A.3) yields

$$\begin{aligned} Q_N &= \frac{1}{2}q_{O_1} \sum_{s=1}^{N_s} \epsilon_s^p(x_1^s y_2^s - x_2^s y_1^s)^{(1)} \\ &+ \frac{1}{2}q_{O_2} \sum_{s=1}^{N_s} \epsilon_s^p(x_1^s y_2^s - x_2^s y_1^s)^{(2)} \\ &+ \frac{1}{2}q_{O_3} \sum_{s=1}^{N_s} \epsilon_s^p(x_1^s y_2^s - x_2^s y_1^s)^{(3)} \\ &+ \frac{1}{2}q_{O_4} \sum_{s=1}^{N_s} \epsilon_s^p(x_1^s y_2^s - x_2^s y_1^s)^{(4)} \end{aligned} \quad (\text{A.4})$$

As mentioned in Sec. 2.4, it is practical and efficient to compute the quantity Q_N by sweeping through grid lines. It can be seen that Q_N is linearly dependent on the quantities q_O 's and the coordinate points (x, y) . That is, Eq.(A.4) has a so-called 'associative' property. As a consequence, it is immaterial whether Q_N is computed at once or it is computed by accumulating the contribution from each segment as grid lines are swept through. The contribution of each line segment to a particular quantity Q_N can be computed as the following. Let's assume that all intersection points among grid lines have been found and indicated by "•" in Fig. A.1. As the old grid lines are swept through, the line segment "bc" is the co-segment of the polygon A_{NO_1} on its right and the polygon A_{NO_4} on its left. So, the contribution of "bc" to the quantity Q_N are $-\frac{1}{2}q_{O_1}(x_b y_c - x_c y_b)$ and $\frac{1}{2}q_{O_4}(x_b y_c - x_c y_b)$ according to the above convention regarding the parameter " ϵ ". Thus, the contribution of "bc" to the quantity Q_N can be written as

$$\Delta_{bc}^O = \frac{1}{2}(q_{O_4} - q_{O_1})(x_b y_c - x_c y_b) \quad (\text{A.5})$$

Equation (A.5) is the same as Eq.(2.5) where q_{O_4} and q_{O_1} are written as q_L and q_R since they are on the left and right of the segment "bc" respectively.

Let's move on to sweeping through the new grid lines (solid lines in Fig. A.1). For example, it can be seen that the line segment "ab" and "de" belong to the polygons A_{NO_1} and A_{NO_2} respectively. The contribution of "ab" to the quantity Q_N is, thus,

$$\Delta_{ab}^N = -\frac{1}{2}q_{O_1}(x_a y_b - x_b y_a) \quad (\text{A.6})$$

since the polygon A_{NO_1} lies on the right of "ab". And the contribution of "de" to the quantity Q_N is

$$\Delta_{de}^N = \frac{1}{2}q_{O_2}(x_d y_e - x_e y_d) \quad (\text{A.7})$$

since the polygon A_{NO_2} lies on the left of "de".

Equations (A.6) and (A.7) are the same as Eq.(2.6) in Sec. 2.5. Note that each new line segment is also a co-segment of two polygons, one is A_N , and another is a polygon to its right or left. Thus, by sweeping through a new grid line, the contribution of each line segment, as in Eq.(A.6) or (A.7), can be added to two polygons at the same time, but only with opposite sign of Δ_s^N .

It is obvious that the tasks, now, are finding the intersection points of grid lines and locating a polygon to which each line segment belongs. These are not trivial and can be done in several ways. Finding the intersection points may not seem to be quite involved since it can be done by solving a pair of equations of straight lines (note that a line segment can be approximated by a straight line). However, it can be a little complicated to cover all possible cases. These cases include a line of infinite slope (vertical line), two lines parallel to each other, two lines lie on top of each other, etc.. Note also that these grid lines have the finite length, not infinite length as in the general equations of straight lines. The next task, namely that of locating a polygon to which each line segment belongs, can also be accomplished by several methods. First of all, a line segment can be represented

by its midpoint. Three distinct methods for determining whether a point is located within a particular polygon are given below. There are several other methods but these three methods have been experienced and extensively investigated by the author.

The first method is illustrated in Fig. A.2. Here, the particular polygon is assumed to be a quadrilateral since it has the same topology as a regular grid cell. If a point P lie inside the polygon, the summation of all angles which P makes with the vertices of the polygon must be equal to 2π radian. If the summation is less than 2π radian, the point P lie outside the polygon. These angles can be computed by using the law of cosine. This is the drawback of the method. This is because computing a trigonometric function on a computer takes more time than performing an arithmetic operation (plus, minus, e.t.c.). Moreover, ambiguity arises when the point P lies close to a side of the polygon.

The second method is borrowed from an interpolation procedure. Each quadrilateral can be divided into two triangles as shown in Fig. A.3. Each of these triangles can be mapped onto a unit triangle as seen in Fig. A.4. It can be shown that any point $P(x,y)$ in the triangle can be represented by

$$\begin{aligned}x &= x_A + (x_B - x_A)\xi + (x_D - x_A)\eta \\y &= y_A + (y_B - y_A)\xi + (y_D - y_A)\eta\end{aligned}\tag{A.8}$$

where $0 \leq \xi \leq 1$, $0 \leq \eta \leq 1$ and $0 < \xi + \eta \leq 1$.

Equation (A.8) can be written as

$$\begin{bmatrix} (x_B - x_A) & (x_D - x_A) \\ (y_B - y_A) & (y_D - y_A) \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} x - x_A \\ y - y_A \end{bmatrix}.\tag{A.9}$$

If the conditions

$$0 \leq \xi \leq 1; 0 \leq \eta \leq 1 \text{ and } 0 < \xi + \eta \leq 1,$$

are met for either one of the two triangles, the point $P(x,y)$ lies inside of the quadrilateral. Otherwise it lies outside of the quadrilateral.

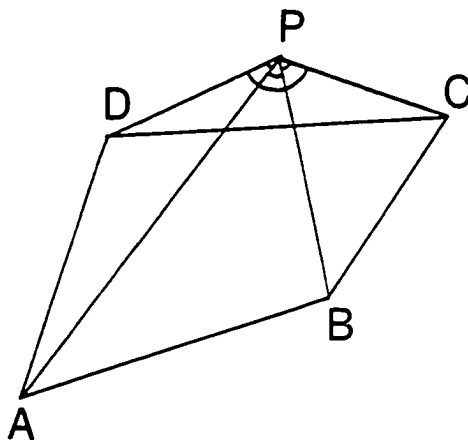
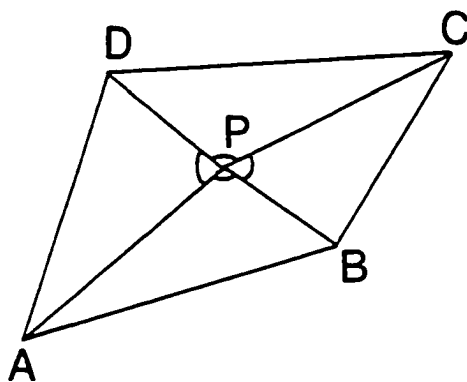


Fig. A.2 The first method for searching.

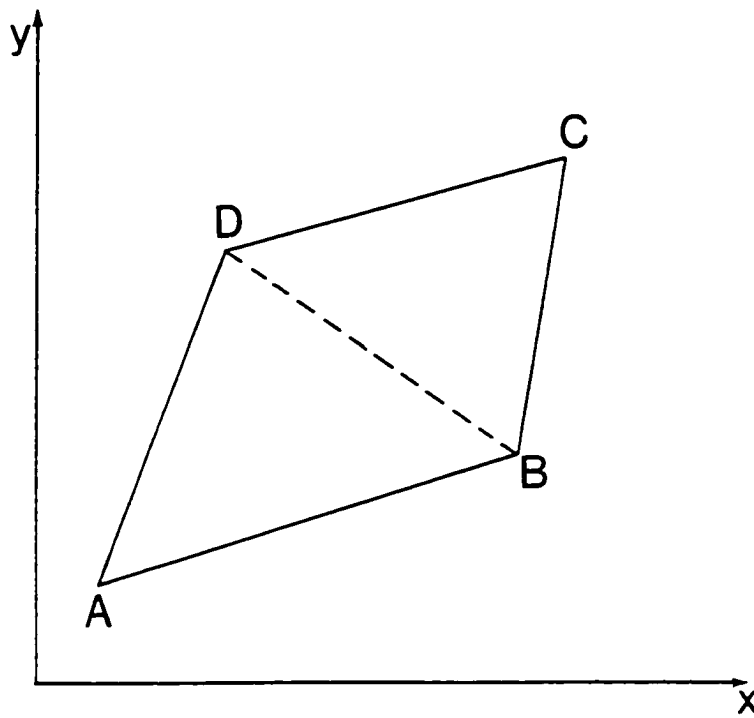


Fig. A.3 A quadrilateral is divided into two triangles.

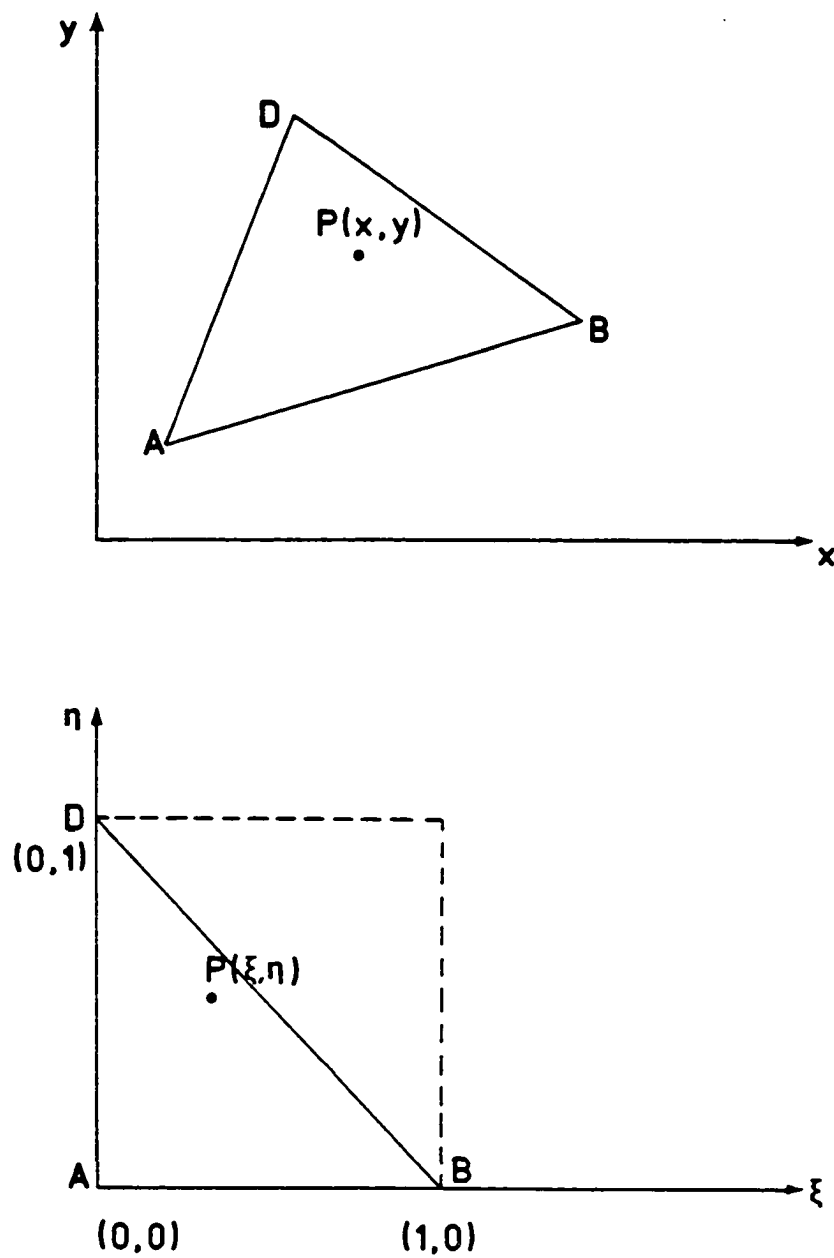


Fig. A.4 Mapping of a triangle onto a unit triangle.

The third method makes use of a vector algebra approach. This method tells whether a point $P(x,y)$ lies on the top, bottom, left, or right of a vertex of the quadrilateral. Thus, the first step is to locate a point (i,j) which is the co-vertex of four quadrilaterals. This can be done by finding a shortest distant between the point $P(x,y)$ and a grid point (i,j) as illustrated in Fig. A.5. Even though the previous two methods do not require this procedure, it is a good idea to do so since it can save some computational time. Let

$$\vec{V} = v_x \hat{e}_x + v_y \hat{e}_y$$

$$\vec{V}_1 = v_{1x} \hat{e}_x + v_{1y} \hat{e}_y$$

$$\vec{V}_2 = v_{2x} \hat{e}_x + v_{2y} \hat{e}_y$$

denote three vectors emanating from the point (i,j) to the point $P(x,y)$ and to the other two vertices of the quadrilateral.

From vector algebra, a vector can be represented by a linear combination of any two vectors which lie on the same plane as that vector (Fig. A.6). So, it can be written as

$$\vec{V} = a_1 \vec{V}_1 + a_2 \vec{V}_2 \quad (\text{A.10})$$

or

$$\begin{aligned} v_x \hat{e}_x + v_y \hat{e}_y &= a_1(v_{1x} \hat{e}_x + v_{1y} \hat{e}_y) + a_2(v_{2x} \hat{e}_x + v_{2y} \hat{e}_y) \\ &= (a_1 v_{1x} + a_2 v_{2x}) \hat{e}_x + (a_1 v_{1y} + a_2 v_{2y}) \hat{e}_y. \end{aligned} \quad (\text{A.11})$$

Equation (A.11) can be written in the matrix form as

$$\begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \quad (\text{A.12})$$

Once a_1 and a_2 are found, it can be determined whether the point $P(x,y)$ lies on the top, bottom, left, or right of the point (i,j) . This is illustrated in Fig. A.7.

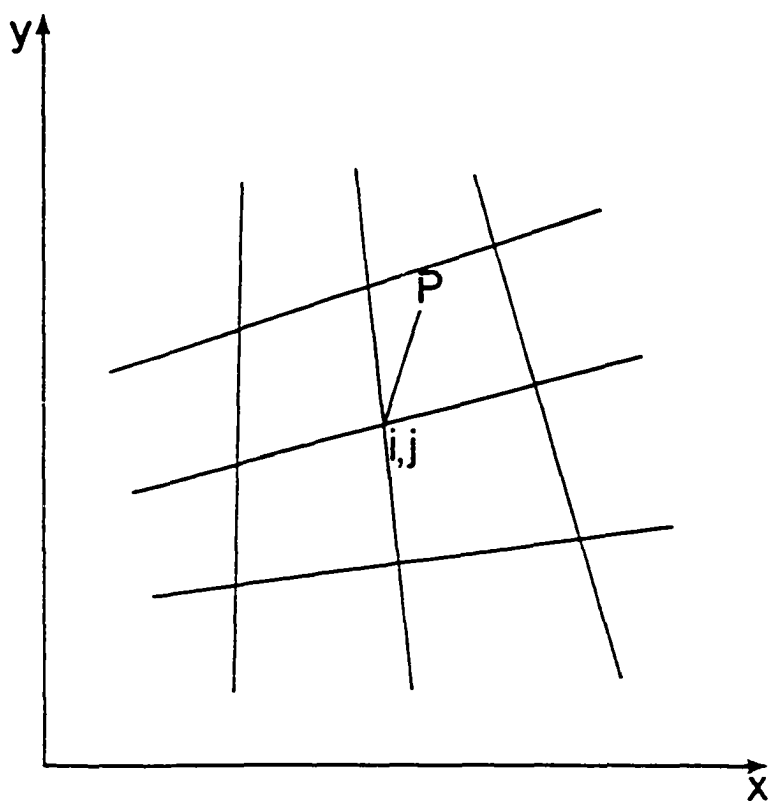


Fig. A.5 The shortest distance between a point P and a grid point (i,j) .

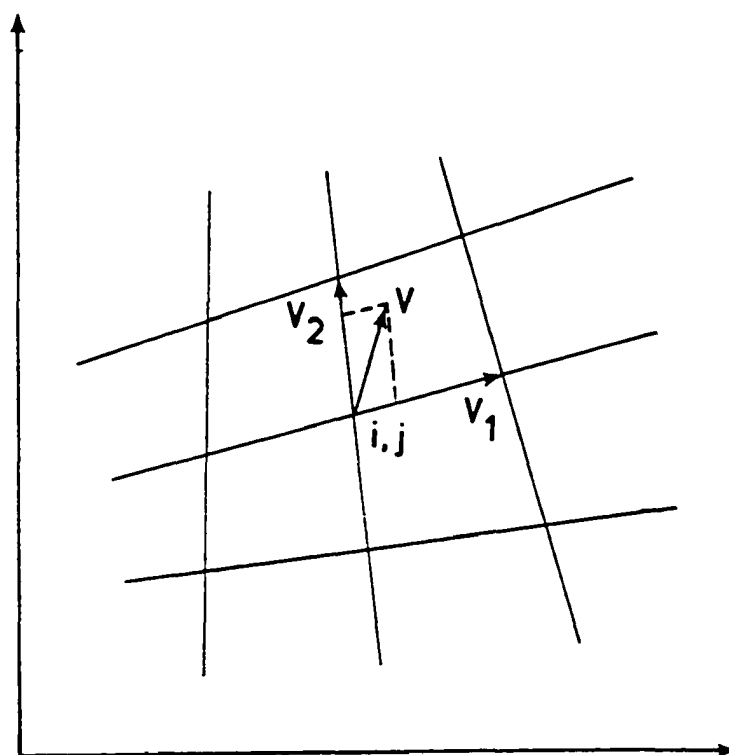


Fig. A.6 Three vectors emanating from a point (i,j) .

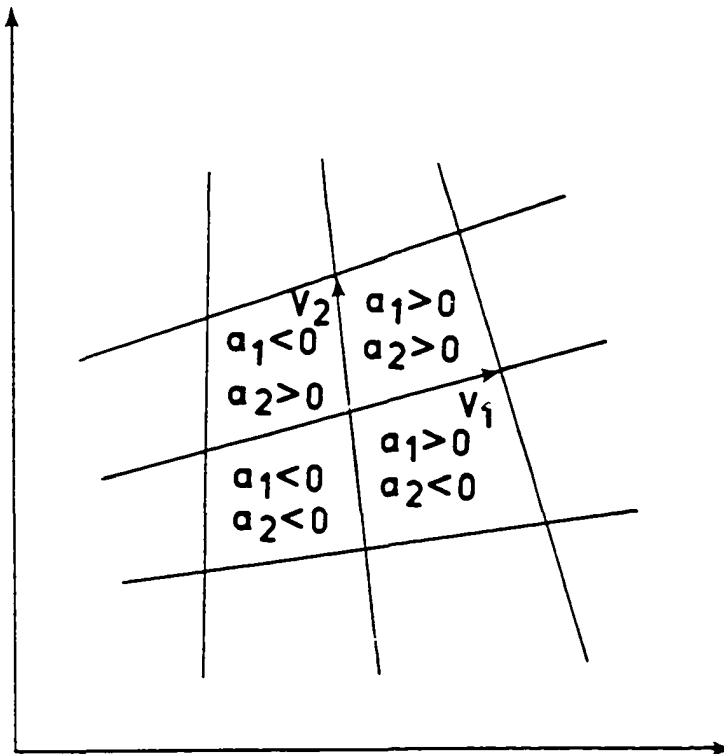


Fig. A.7 Conditions that determine in which grid cell the point P lies.

The third method is generally faster than the other two methods since less arithmetic operations are performed. The only drawback of this method is that the point $P(x,y)$ may not actually lie inside any of the four quadrilaterals found previously if grid lines have a large curvature. If this is the case, the method will indicate that the point $P(x,y)$ lies in one of the quadrilaterals which is incorrect. However, this problem can be prevented when care is exercised. All three methods can be extended into three dimensions, but the third method is less complicated and requires much less computational time.

At this point, it may not be clear as how the algorithm is implemented. Nonetheless, it is seen that all necessary tools in doing so have been given. It is the matter of utilizing them. The implementation of the algorithm is given below. The following procedure is extracted from the author's years of experience. There may be other ways to implement the algorithm more efficiently. However, it has been proven to the author, whose main background is in Mechanical Engineering (not in Computer Sciences), that the following procedure is sufficient and effective. The first step is to find all intersection points among grid lines. Thus, each grid line is divided into several line segments. The number of the line segments for each grid line varies according to the number of intersection it makes with the other grid lines. Then, each grid line is swept through. Each line segment is represented by its midpoint. For a line segment of the old grid line, a search routine is used to locate the area $A_{N_{ij}}$ in which the segment lies. Then the contribution Δ_s^O as in Eq.(A.5) is added to the quantity $Q_{N_{ij}}$. The quantities q_L and q_R in the formula are known quantities and can be identified simply by the index of this particular segment. For a line segment of the new grid line, a search routine is used to locate the area A_O in which the segment lies. This information tells which q_O to be used in Eq.(A.6) for computing Δ_s^N . The quantity Δ_s^N is, then, added to the quantity Q_N on the left of the line segment. At the same time the quantity Q_N on its right is subtracted by

Δ_i^N . These Q_N 's can be easily identified by the index of the line segment. This way all the quantities Q_N 's are filled by Δ_i^O and Δ_i^N as all grid lines are swept through. It is immaterial whether the old or new grid lines are swept through first. It can be shown that the method works no matter how grid lines are oriented. So, it can be applied at the interfaces where two (or more) subdomain grids of completely different topologies meet. This statement is also confirmed by the results in Chap. 4.

It should be mentioned that the conservative rezoning algorithm yields an ambiguous case if grid lines are coincident. This is because the method does not know which area A_N to put Δ_{bc}^O (Eq.(A.5)) into, when the old grid lines are swept through. Also, it is not sure of which quantity q_O to use in Eq.(A.6) when the new grid lines are swept through. This case can be avoided by moving the grid lines which lie on top of each other a little apart (say 1×10^{-7}). This does not add a significant error to the calculation.

Appendix B

THE CALCULATION OF CELL SURFACE AREA AND VOLUME

In Sec. 3.2, it was noted that the finite volume approach requires the calculation of the volume of the cell element and its surface areas. A method of doing so is illustrated in this appendix. Although any arbitrary grid can be used, the hexahedral cells have been found the most practical and widely used. So, the procedure to compute the volume of a hexahedral and its surface area is described here. An example of a general hexahedral is shown in Fig.B.1 where S_I , S_J , and S_K are vectors of the three of its surface areas in the curvilinear (i, j, k) coordinate. Figure B.2 illustrates a surface of the hexahedral. If the four vertices defining the surface are coplanar, its area is given by one half the cross product of its diagonal line segments,

$$\vec{S} = \frac{1}{2} \vec{R}_{31} \times \vec{R}_{42} \quad (\text{B.1})$$

If the vectors \vec{R}_{31} and \vec{R}_{42} are written as

$$\vec{R}_{31} = (x_3 - x_1)\hat{e}_x + (y_3 - y_1)\hat{e}_y + (z_3 - z_1)\hat{e}_z$$

and

$$\vec{R}_{42} = (x_4 - x_2)\hat{e}_x + (y_4 - y_2)\hat{e}_y + (z_4 - z_2)\hat{e}_z \quad ,$$

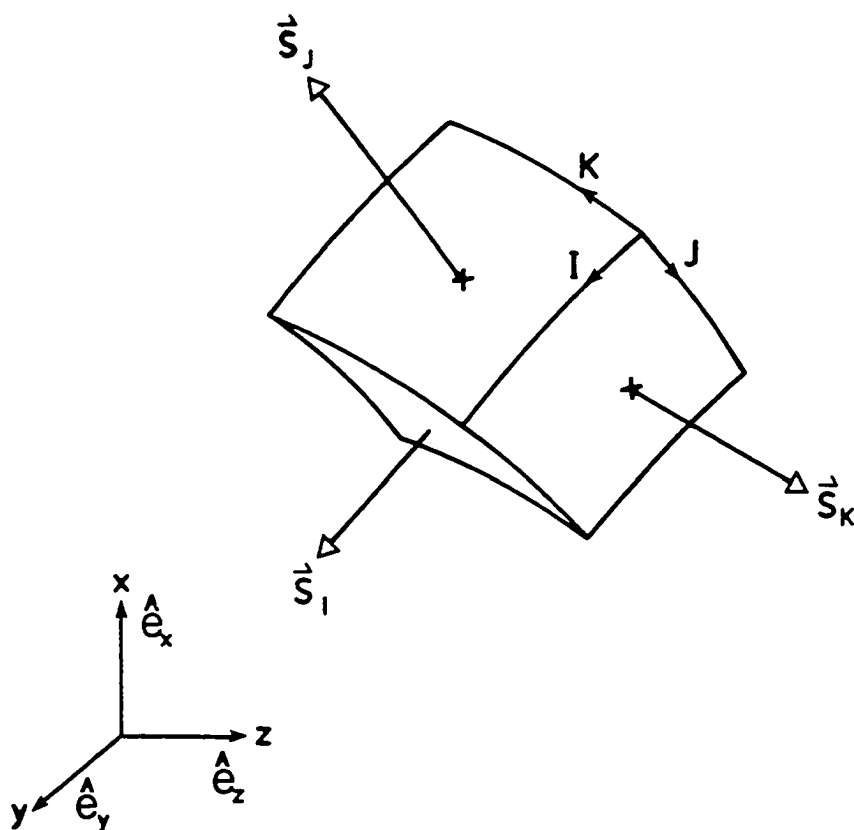


Fig. B.1 An arbitrary hexahedron.

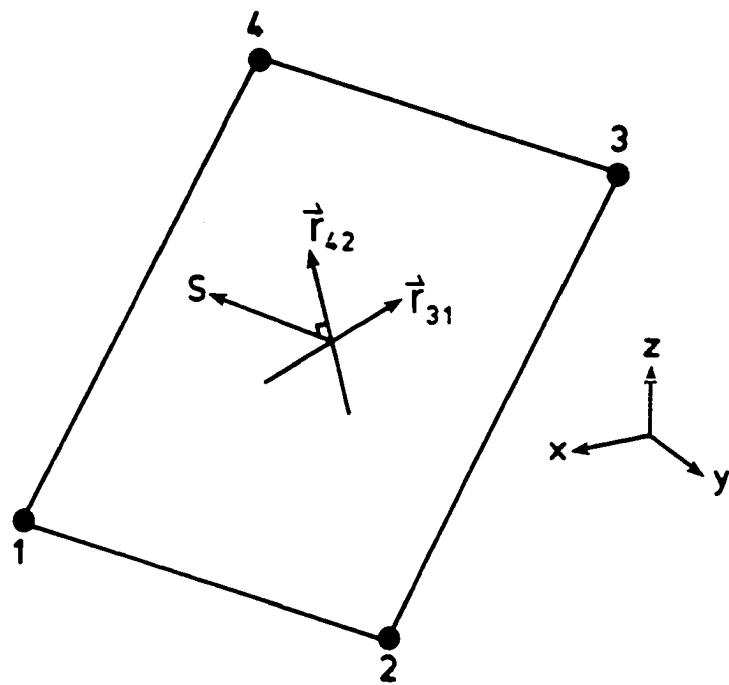


Fig. B.2 A surface of a hexahedral.

the vector \vec{S} can be computed as,

$$\vec{S} = \frac{1}{2} \begin{vmatrix} \hat{e}_x & \hat{e}_y & \hat{e}_z \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ (x_4 - x_2) & (y_4 - y_2) & (z_4 - z_2) \end{vmatrix} ,$$

or

$$\begin{aligned} \vec{S} &= \frac{1}{2} [(y_3 - y_1)(z_4 - z_2) - (z_3 - z_1)(y_4 - y_2)] \hat{e}_x \\ &- \frac{1}{2} [(x_3 - x_1)(z_4 - z_2) - (z_3 - z_1)(x_4 - x_2)] \hat{e}_y \\ &+ \frac{1}{2} [(x_3 - x_1)(y_4 - y_2) - (y_3 - y_1)(x_4 - x_2)] \hat{e}_z . \end{aligned}$$

This can be written as

$$\vec{S} = SIX\hat{e}_x + SIY\hat{e}_y + SIZ\hat{e}_z . \quad (\text{B.2})$$

If the index i, j, k are used in Fig. B.1, the formulas for SIX_{ijk} , SIY_{ijk} , and SIZ_{ijk} in Sec. 3.2 can be verified. Note that Eq.(B.1) is still a good approximation for the surface area even if the surface is non-planar.

The volume of a hexahedral is computed in the following way. A general hexahedral is composed of five tetrahedra (Fig. B.3), each of whose volume is determined by one sixth of the triple product, e.g. (Fig.B.3),

$$V_{1236} = \frac{1}{6} (\vec{R}_{21} \times \vec{R}_{31}) \cdot \vec{R}_{61} . \quad (\text{B.3})$$

If the vectors \vec{R}_{21} , \vec{R}_{31} , and \vec{R}_{61} are written as

$$\begin{aligned} \vec{R}_{21} &= (x_2 - x_1)\hat{e}_x + (y_2 - y_1)\hat{e}_y + (z_2 - z_1)\hat{e}_z \\ \vec{R}_{31} &= (x_3 - x_1)\hat{e}_x + (y_3 - y_1)\hat{e}_y + (z_3 - z_1)\hat{e}_z \\ \vec{R}_{61} &= (x_6 - x_1)\hat{e}_x + (y_6 - y_1)\hat{e}_y + (z_6 - z_1)\hat{e}_z , \end{aligned}$$

it can be shown that

$$V_{1236} = \frac{1}{6} \begin{vmatrix} (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ (x_6 - x_1) & (y_6 - y_1) & (z_6 - z_1) \end{vmatrix} .$$

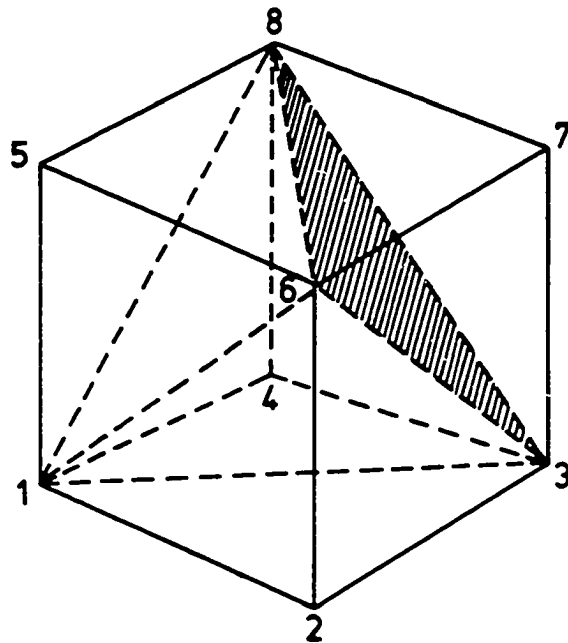


Fig. B.3 A hexahedron is composed of five tetrahedra.

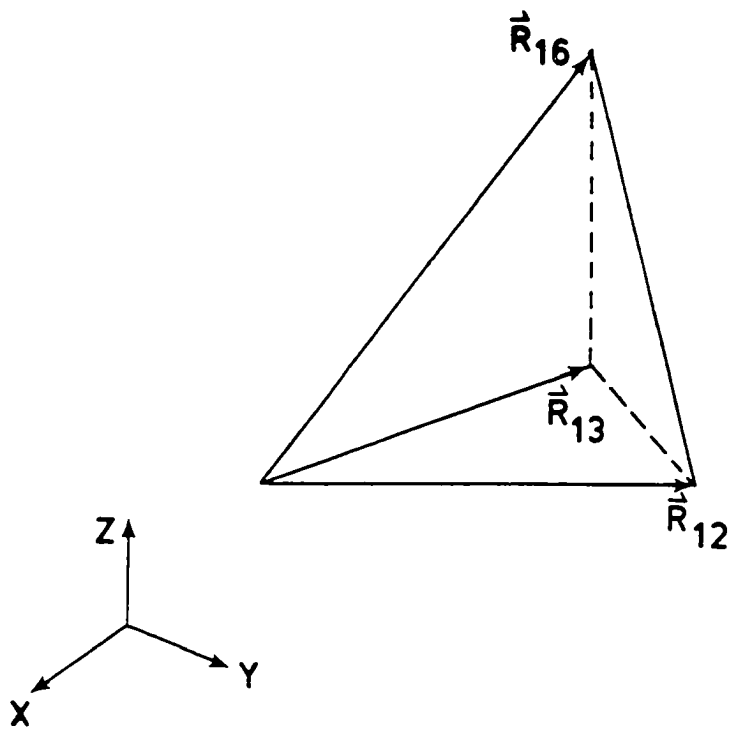


Fig. B.4 An arbitrary tetrahedron.

Thus, the volume of the hexahedral is the sum of the five constituent tetrahedra,
i.e.,

$$VOL = V_{1236} + V_{3867} + V_{3816} + V_{1685} + V_{1348} \quad . \quad (\text{B.4})$$

BIOGRAPHY

Monchai Kathong was born on June 25, 1957, in Bangkok, Thailand. He was raised in Bangkok, Thailand and graduated from Saint Dominic School in 1972. In January 1980, he received his Bachelor of Science degree in Engineering from the Royal Thai Naval Academy. He was commissioned an engineering officer aboard various ships in the Royal Thai Navy.

In August 1981, he received a scholarship from the Royal Thai Navy for the graduate study at University of Wisconsin. He received a degree of Master of science in Mechanical Engineering from University of Wisconsin in December 1982. He enrolled in the doctoral program in the Mechanical Engineering and Mechanics Department at Old Dominion University in August 1983. During his graduate work at Old Dominion University, he conducted his research at the Computer Applications Branch of the Analysis and Computation Division of NASA Langley Research Center.