

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
OLD DOMINION UNIVERSITY  
NORFOLK, VIRGINIA 23529-0247

LADDER  
317248  
11/18

**A CONSERVATIVE APPROACH FOR FLOW FIELD  
CALCULATIONS ON MULTIPLE GRIDS**

By

Monchai Kathong, Graduate Research Assistant

and

Surendra N. Tiwari, Principal Investigator

Progress Report

For the period ended March 31, 1988

Prepared for the  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23665-5225

Under

Cooperative Agreement NCC1-68

Dr. Robert E. Smith, Jr., Technical Monitor

ACD-Computer Applications Branch

(NASA-CR-185313) A CONSERVATIVE APPROACH  
FOR FLOW FIELD CALCULATIONS ON MULTIPLE  
GRIDS Progress Report, period ended 31 Mar.  
1988 (Old Dominion Univ.) 119 p CSCL 20D

N89-26173

Unclas  
G3/34 0217248

April 1988

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
OLD DOMINION UNIVERSITY  
NORFOLK, VIRGINIA 23529-0247

**A CONSERVATIVE APPROACH FOR FLOW FIELD  
CALCULATIONS ON MULTIPLE GRIDS**

By

Monchai Kathong, Graduate Research Assistant

and

Surendra N. Tiwari, Principal Investigator

Progress Report

For the period ended March 31, 1988

Prepared for the  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23665-5225

Under  
Cooperative Agreement NCC1-68  
Dr. Robert E. Smith, Jr., Technical Monitor  
ACD-Computer Applications Branch

Submitted by the  
Old Dominion University Research Foundation  
P. O. Box 6369  
Norfolk, Virginia 23508-0369

April 1988

## FOREWORD

This is a progress report on the research project "Numerical Solutions of Three-Dimensional Navier-Stokes Equations for Closed-Bluff Bodies." Specific efforts were directed in the area of "A Conservative Approach for Flow Field Calculations on Multiple Grids."

The period of performance on this research was July 15, 1987 through March 31, 1988. This work was supported by the NASA Langley Research Center through Cooperative Agreement NCC1-68. The Cooperative Agreement was monitored by Dr. Robert E. Smith, Jr., of the Analysis and Computation Division (Computer Applications Branch), NASA Langley Research Center, MS/125.

## TABLE OF CONTENTS

	Page
TABLE OF CONTENTS.....	i
LIST OF TABLES.....	ii
LIST OF FIGURES.....	iii
LIST OF SYMBOLS.....	iv
 Chapter	
1. INTRODUCTION.....	1
2. GRID GENERATION.....	8
2.1 Introduction.....	8
2.2 Transfinite Interpolation.....	13
2.3 Mapping Type and singularities.....	18
2.4 Multiple Grids.....	20
2.5 Brief Discussion on Conservative Rezoning Algorithm.....	26
2.6 Applications to Two and Three Dimensional Hyperbolic Equations.....	28
3. APPLICATIONS TO THE AERODYNAMICS CONFIGURATIONS.....	29
3.1 Governing Equations.....	31
3.2 Spatial Finite-Volume Discretization.....	34
3.3 Artificial-Viscosity Model.....	41
3.3.1 Nonlinear artificial-viscosity term.....	43
3.3.2 Linear artificial-viscosity term.....	43
3.4 Boundary Conditions.....	44
3.4.1 Solid Walls.....	45
3.4.2 Coordinate cuts.....	47
3.4.3 Inflow/outflow boundaries.....	47
3.4.4 Interface Conditions.....	50

	Page
3.5 Numerical Time Integration.....	52
3.6 Local Time Step Scaling.....	52
4. RESULTS AND DISCUSSIONS.....	57
4.1 Sphere and Slender Body.....	57
4.2 A Butter-Wing Configuration.....	58
6. CONCLUSIONS.....	61
REFERENCES.....	63

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
2.1 Boundary-fitted coordinates system.....	74
2.2 Physical v.s. computational domain.....	75
2.3 The computational domain with $f(u,v,w)$ specified on plane of constant $u,v$ or $w$ .....	76
2.4 Example of an out-of-surface derivative of $f$ .....	77
2.5 The computational domain with $f(u,v,w)$ and a finite number of out-of-surface derivatives of $f$ specified on the six boundaries.....	78
2.6 Natural mapping types for a 3D airplane wing.....	79
2.7 Structure of the 0-0 mapping for a wing-body configuration.....	80
2.8 Position of singular lines for the 0-0 mapping.....	81
2.9 The avoidance of singularity at the nose of the airplane by leaving boundary surface open.....	82
2.10 Zoning of multiply connected region.....	83
2.11 Grid Patching v.s. Grid Embedding Techniques.....	84
2.12 Overlap areas arising from the superposition of the old mesh and the new mesh.....	85
3.1 A typical volume element.....	86
3.2 An example of solid wall boundary condition.....	87
3.3 An example of coordinate cut boundary condition.....	88
3.4 The number of boundary conditions at inflow and outflow based on the ingoing characteristic variables $\phi^m, m=1, \dots, 4$ . ....	89
3.5 A typical interface between two grids.....	90
3.6 Evaluation of fluxes at the interface of grid no. 2.....	91
4.1 Multiple grid system for a sphere.....	92
4.2 Multiple grid system for a slender body.....	93

<u>Figure</u>	<u>Page</u>
4.3 Wall pressure coefficient along center line of a sphere at $M=0.2$ and zero degree angle of attack.....	94
4.4 Wall pressure coefficient along center line of a slender body at $M=1.5$ and zero degree angle of attack.....	95
4.5 Physical model of a Butler wing.....	96
4.6 Multiple grid system for a Butler wing configuration.....	97
4.7 Wall pressure coefficient along center line of a Butler wing configuration at $M=3.5$ and zero degree angle of attack.....	98
4.8 Wall pressure coefficient at different cross-section of a Butler wing configuration at $M=3.5$ and ten degree angle of attack.....	100
4.9 Wall pressure coefficient along center line of a Butler wing configuration at $M=2.5$ and zero degree angle of attack.....	104
4.10 Wall pressure coefficient at different cross-section of a Butler wing configuration at $M=2.5$ and ten degree angle of attack.....	105
5.1 A fighter-aircraft configuration.....	109

## LIST OF SYMBOLS

$A_n$	Area of new mesh cell
$A_{no}$	Overlapped area arising from the superposition of the old mesh and the new mesh
$A_o$	Area of old mesh cell
$A_p$	Area of a polygon
$E$	Total energy
$F$	Vector of fluxes (mass, momentum, energy) in x direction
$G$	Vector of fluxes (mass, momentum, energy) in y direction
$H$	Vector of fluxes (mass, momentum, energy) in z direction
$J$	Jacobian of coordinate transformation
$M$	Mach number
$N_p$	Number of sides of a particular polygon
$Q$	Vector of state variables
$Q_n$	A quantity in the new mesh cell
$Q_o$	A quantity in the old mesh cell
$\vec{a}_k(v,w)$	Known function on surfaces of constant u.
$\vec{b}_k(u,w)$	Known function on surfaces of constant v.
$\vec{c}_k(u,v)$	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 $\equiv H/A$
$p$	pressure
$q_o$	Volumetric density of $Q_o$
$t$	time

$u, v, w$	$x, y, z$ component of velocity; 3 parameter specifying function $\bar{f}$
$VOL_{ijk}$	Volume of a volume element
$x, y, z$	Cartesian coordinate system
$\alpha_k(\beta), \beta_k(v), \gamma_k(w)$	Univariate blending functions
$\delta_{k\ell}$	Kronecker Delta
$\rho$	Density
$\Omega$	An arbitrary region
$\partial\Omega$	Boundary of an arbitrary region
$\gamma$	Specific gas ration $\equiv \frac{c_p}{c_v}$
$\mu_I, \mu_J, \mu_K$	Average operators
$\lambda$	Eigenvalue
$\xi, \eta, \zeta$	Curvilinear coordinate system

# A CONSERVATIVE APPROACH FOR FLOW FIELD CALCULATIONS ON MULTIPLE GRIDS

By

M. Kathong<sup>1</sup> and S. N. Tiwari<sup>2</sup>

## SUMMARY

In the computation of flow fields about complex configurations, it is very difficult to construct body-fitted coordinate systems. 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 follows the 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. Some steady state solutions of the Euler equations are presented and discussed.

---

<sup>1</sup>Graduate Research Assistant, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, VA 23508.

<sup>2</sup>Eminent Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, VA 23508.

## 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. The problem of producing required freestream conditions in the test section of this 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

the limitations on the method. The 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. Good surveys on the approach can be found in [1-13]. Also, the foundations upon which the whole field is built are now reasonably well covered in text books [14-24]. 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.

In the CFD calculation, the continuum problem of the differential equations is projected to some finite-dimensional space for the dependent and independent variables and then by solving the resulting discrete equations 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, i.e., to create a grid. This brings about a so-called "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 can be divided into two categories. They 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 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. Detail on grid generation procedures are described in Chap. 2. Surveys of the methods including textbooks on grid generation procedure can also be found in references [25-30].

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 single grid or single-block grid, offers the simplicity of the computational domain 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 "multiple grids" or "zonal grid" approach can be categorized into two groups: grid patching and grid embedding. 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 embedding approach. Multiple grids approach is becoming more common as the complexity of the configuration being considered in CFD is increased. However, the approach results in 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 subdomain (or 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 embedding 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 is discussed in detail in Section 2.5.

The viscous Navier-Stokes equations are the ultimate equation to be solved in CFD. However, Navier-Stokes flow simulation are presently still in the stage of research. A success in 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 on the Navier-Stokes equations. Thus, the problems of computer storage and computational time including the uncertainties of turbulent modeling that arise in Navier-Stokes calculation can be eliminated. Solutions of the Euler equations, though inviscid, give the correct phenomena to many of 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 Sect. 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 a 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 discretized Euler equations. The purpose of adding these terms is to impose an entropy condition which is required to eliminate the non-physical shocks. Also, the addition of the artificial dissipation terms helps to eliminate 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 are 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 sphere at low Mach number and to the flow over a slender body at the supersonic Mach number are discussed in Sec. 4.1. Results obtained from these applications are very encouraging. Section 4.2 describes the application of the multiple grid approach to the flow over a Butler-Wing configuration. Again, good results have been obtained. The next step is to apply the approach to the internal/external flow on a fighter-aircraft.

## 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 and 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 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 system.

Any partial 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, and normal derivatives on the bodies can be represented using only finite difference between grid points on coordinate lines, without need of any interpolation, even though the coordinate system is not orthogonal at the boundary. 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 mesh 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 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 [31]. Figure 2.2 illustrates physical domain vs. computational domain.

Two primary categories for arbitrary coordinate generation have been developed. They 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 system may be elliptic 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 system is involved. Thus, the primary advantages of algebraic methods are speed and directness. Major disadvantages 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 [32]. For instance, the results of Shang [33] 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 [34] can be used to prevent nonsmooth boundary behavior (e.g., slope discontinuities) from propagating inward. Transfinite interpolation described by Gordon and Hall [35] in the early 1970's 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. For aerodynamic applications, Eriksson [36] and Rizzi and Eriksson [37] 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 [30,34] developed by Peter Eiseman 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 [38,39] but most notably for computing grids about turbine cascades. The two-boundary technique [40-42] 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 cubic interpolation, surface derivations 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 Cauchy-Riemann boundary conditions. The earliest successful development was formally reported by Winslow [43], who started with a Laplace system subjected to Dirichlet boundary conditions. Thompson, et al. [44] added periodic boundary conditions

to produce branch cuts for various topological configurations and who also 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. [45]. Without forcing terms, Mastin and Thompson [46] were able to show that the two-dimensional system analytically defined nonsingular transformation. Conformal mapping methods can also be included in the elliptic methods. Mehta and Lavan [47] 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 in Sampath [48], Wu et al. [49], and Napolitano et al. [50]. 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 has been given by Starius [51], and one which was well suited to body concavity has been presented by Stager and Sorenson [52]. 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 [53-55]. 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 [56] have combined these two schemes into a single scheme that takes advantages of the two but eliminates the drawbacks of each. 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 (i) defining an accurate mathematical description of all solid surfaces in the computational domain and ii) 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.

In this study, algebraic approach has been taken due to its speed and directness. Two boundary technique [40-42] has been used to obtain boundary surfaces. The interior grid points have been obtained by applying the transfinite interpolation technique. Some detail of transfinite interpolation is given in the next section. Details of two boundary technique can be found in Ref. [40].

## 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 this 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 [57] and subsequently extended by Gordon [58]. One of the earliest 2D grid generation applications using transfinite interpolation is described in Gordon and Hall [35]. A few examples of 3D applications are the works of Gerhard [59], Anderson and Spradley [60], and Spradley et al. [61]. 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 [36,62] and Eriksson and Rizzi [63] 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 that this feature provides 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 giving 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 mesh generation system.

The theory of transfinite interpolation is a very general concept of multivariate interpolation and is outlined 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

$$\alpha_k(u_\ell) = \delta_{k\ell} \quad ; \quad \beta_k(v_\ell) = \delta_{k\ell} \quad ; \quad \gamma_k(w_\ell) = \delta_{k\ell}$$

where

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

is needed to interpolate between these given planes.

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

$$\vec{f}_1(u, v, w) = \sum_{k=1}^p \alpha_k(u) \vec{a}_k(v, w)$$

$$\vec{f}_2(u, v, w) = \vec{f}_1(u, v, w) + \sum_{k=1}^q \beta_k(v) [\vec{b}_k(u, w) - \vec{f}_1(u, v_k, w)]$$

$$\vec{f}(u, v, w) = \vec{f}_2(u, v, w) + \sum_{k=1}^r \gamma_k(w) [\vec{c}_k(u, v) - \vec{f}_2(u, v, w_k)] \quad (2.1)$$

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 minimum input geometry data, a generalized transfinite interpolation procedure 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

$$\begin{aligned} \frac{\partial^n \vec{f}}{\partial u^n}(u_k, v, w) &= \vec{a}_k^{(n)}(v, w) ; & k &= 1, 2 \\ n &= 0, 1, 2, \dots, p_k \end{aligned}$$

$$\begin{aligned} \frac{\partial^n \vec{f}}{\partial v^n}(u, v_k, w) &= \vec{b}_k^{(n)}(u, w) ; & k &= 1, 2 \\ n &= 0, 1, 2, \dots, q_k \end{aligned}$$

$$\begin{aligned} \frac{\partial^n \vec{f}}{\partial w^n}(u, v, w_k) &= \vec{c}_k^{(n)}(u, v) ; & k &= 1, 2 \\ n &= 0, 1, 2, \dots, r_k \end{aligned}$$

which is simply the specification of  $\vec{f}$  and on 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

$$\begin{aligned}
 \alpha_k^{(n)}(u) &; \quad k = 1, 2 & n = 0, 1, 2, \dots, p_k \\
 \beta_k^{(n)}(v) &; \quad k = 1, 2 & n = 0, 1, 2, \dots, q_k \\
 \gamma_k^{(n)}(w) &; \quad k = 1, 2 & n = 0, 1, 2, \dots, r_k
 \end{aligned}$$

which have to satisfy the conditions

$$\frac{\partial^m}{\partial u^m} \delta_k^{(n)}(u_\ell) = \delta_{k\ell} \delta_{nm}$$

$$\frac{\partial^m}{\partial v^m} \beta_k^{(n)}(v_\ell) = \delta_{k\ell} \delta_{nm}$$

$$\frac{\partial^m}{\partial w^m} \gamma_k^{(n)}(w_\ell) = \delta_{k\ell} \delta_{nm}$$

The generalized transfinite interpolation algorithm is then written

$$\vec{f}_1(u, v, w) = \sum_{k=1}^2 \sum_{n=0}^{p_k} \alpha_k^{(n)}(u) \vec{a}_k^{(n)}(v, w)$$

$$\vec{f}_2(u, v, w) = \vec{f}_1(u, v, w) + \sum_{k=1}^2 \sum_{n=0}^{q_k} \beta_k^{(n)}(v) \cdot [\vec{b}_k^{(n)}(u, w) - \frac{\partial^n}{\partial v^n} \vec{f}_1(u, v_k, w)] \quad (2.2)$$

$$\vec{f}_3(u, v, w) = \vec{f}_2(u, v, w) + \sum_{k=1}^2 \sum_{n=0}^{r_k} \gamma_k^{(n)}(w) \cdot [\vec{c}_k^{(n)}(u, v) - \frac{\partial^n}{\partial w^n} \vec{f}_2(u, v, w_k)]$$

The function  $\vec{f}$  now defines a transformation from the region  $u_1 < u < u_2$ ,  $v_1 < v < v_2$ ,  $w_1 < w < w_2$  in  $u, v, w$  space to some arbitrarily shaped region of  $x, y, z$  space.

Generally speaking, the method of transfinite interpolation, is a very simple and straightforward concept that offers virtually unlimited possibilities; but for any particular application, it is necessary to supply a certain amount of geometry data to obtain a certain degree of control of the transformation. It is up to the user to balance the requirements of minimum input geometry data and maximum control.

### 2.3 Mapping Type and Singularities

It is clear from previous sections that the first stage of 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 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 0-0 is the most efficient for such a configuration [64]. The notation 0-0 is to be interpreted as "type 0 in the chordwise direction, type 0 in the spanwise direction", using the 2D notation shown in Fig. 2.6. Figure 2.7 illustrates the 0-0 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

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. 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). Grid singularities are undesirable but very often unavoidable and any practical finite difference scheme must be able to cope with them. If the physics near a singularity is not of primary interest, the finite difference solutions can be obtained in this region providing that 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 section 3.1.

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 from the viewpoint of 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, reference [64] has shown that this type of mapping is not as efficient as other types of mapping (for example O-O type). This fact can be explained that the C-H mapping 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 single-block grids, i.e., grids that map the physical domain onto a "slab" in the computational domain. This type of mapping is very desirable due to its simplicity. However, for very complicated geometries it can be difficult to generate single-block grid that are 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 to machine zero 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 [65], for example see Fig. 2.10. This approach, called "multiple grids" or "zonal grids" approach (the terms "zone" or "block" is also used interchangeably), falls into two catagories: grid patching and grid embedding. The approach subdivides a complicated domain into subdomains which can accomodate 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 [66,67], or with continuous lines slope [64,68], or discontinuity in slope [69], or perhaps not align each other at all. Robert and Lee [69] combine the subdomain grid in such a manner that the

resultant global grid is continuous across patch boundaries. Moreover, 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. [70] have demonstrated a patched grid technique for solution of the thin layer Navier-Stokes 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 [71-74] also illustrate the use of patched grid approach. The grid embedding 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 embedding approach have been studied by Starius [75,76], Kreiss [77], and Mastin and McConnaughey [78]. The practical application of overlapping grids to the solution of problems in computational fluid dynamics has been demonstrated by Atta [65], Thompson [79], Steger and Buning [80], and Benek et al. [81]. Steger et al. [82,83] have applied the grid embedding technique to an airfoil/flap in incompressible flow [82] and in subsonic, compressible flow [83]. Atta and Vadyak [84] have obtained a potential solution for a wing/nacelle geometry. These studies have demonstrated that the technique can be applied to subsonic flow. However, for the transonic flight regime Benek et al. [83] 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 [85] indicate that for a different grid geometry and algorithm, these problems may not be too severe. Figure 2.11 illustrates an example of grid patching v.s. grid embedding. Early efforts to predict multiple-component configurations are based on the transonic small disturbance formulation [86-88]. Efforts to

predict the flow field about a complete aircraft configuration using a single grid approach have been made by Yu [89]. 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 grid 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 the multiple grids approach will also result in 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 near the body and inviscid potential in the outer field. Chaderjian and Steger [90] 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.

A common difficulty to the multiple grid approach is constructing 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. This so-called "interface conditions" is required to guarantee the convergence to a weak solution of the governing equations if the algorithm converges. The multiple grid approach results in new boundary 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 [91] have shown that shock waves crossing embedded grid boundaries can become ill defined and convergence is generally degraded when the interpolation procedure is used. They have also shown 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 systems 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 [92]. In a multiple grid calculation, it is important that the interforces are also treated in a conservative manner so that the discontinuities can move freely across the interfaces [93]. The need for conservative grid interfaces is also illustrated in ref. [83].

The question of conservation when switching between two different grids or numerical schemes has been considered by several authors. Warming and Beam [94] have derived transition operators for switching conservatively between MacCormack's method and a second order upwind scheme. Hesseinius and Pulliam [95] 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 [96] has developed conservative zonal

interface conditions for zonal grids which share a common grid line, and has provided accurate calculations demonstrating the shock capturing ability of the zonal grids with a discontinuity crossing zones. Combier et al. [97] 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. [98] have presented results obtained metric discontinuous grids; the integration scheme used is the Osher upwind scheme. Reference [82] provides the results obtained on overlapping grids in conjunction with the stream function approach.

In patched grid approach, conservation can be easily maintained at the patch 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 effect transfer between zones, 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 zones. Other disadvantages of grid embedding approach, beside that of interpolation, are: i) it is difficult to maintain global conservation ii) the accuracy and convergence speed of the calculation seems to depend on the degree of overlapping of the zones 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 in three dimensions 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 mesh to another which was first described by Dukowicz [99]. Ramshaw [100] has suggested a procedure in 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 this technique is feasible for applications to realistic aerodynamic configurations.

The grid generation of multiple grid does not in principle differ from the generation of 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 [101] and Erkişon et al. [102] have 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 in Refs. 101 and 102. 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. Detail on the treatment of the conditions at the interface is given in sec. 3.3.

## 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 below. This method was first described in Ref. [99]. Reference [100] 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.

By far the most common type of generalized mesh is the arbitrary quadrilateral mesh, which is convenient to work with because it has the same simple topological and logical structure as a square or rectangular mesh. The basic concept of the algorithm is simple. Consider Fig. 2.12, two grid surfaces are overlapping each other in some fashion. The conserved quantities  $Q_{ij}$  of the original grid surface ( $A_{ij}$  is the area of each surface mesh) is to be transferred to another grid surface in which  $A_{ij}$  is the area of each surface mesh,  $Q_{ij}$  is denoted as the transferred quantity in each of these later surface mesh. Thus,  $Q_{ij}$  can be computed by

$$Q_{ij} = \sum (Q_{ok\ell}) \frac{(A_{nok\ell})}{A_{ok\ell}} = \sum (q_{ok\ell})(A_{nok\ell}) \quad (2.3)$$

Where  $A_{nok\ell}$  is the portion of the area  $A_{ij}$  which contains in the area  $A_{ok\ell}$  and the summation is up to the number of the original surface meshes contained in  $A_{ij}$ . And  $q_{ok\ell} = Q_{ok\ell}/A_{ok\ell} \equiv$  Volumetric density of  $Q_{ok\ell}$ , is assumed to be constant. The task now, is to find  $A_{nok\ell}$  and the number of original surface meshes contained in  $A_{ij}$ . The area of the polygon in 2D plane is given by [103].

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

where the summation is over all the sides of  $p$ , and  $\epsilon_s^p$  is either +1 or -1 according as  $p$  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. The overlapped areas are polygons  $p$  whose sides are segments of the old-mesh lines and the new-mesh lines. The number of sides of each type, and total number of sides, will be different for different overlapped areas. Each side 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 mesh, into the cells of the new mesh. It would be inefficient and difficult, to automate in a computer, to naively sweep over the overlap areas directly. Instead, Ramshaw [100] 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  $(x_2^s, y_2^s)$ .

If the side  $s$  is a segment of the old mesh then the quantity  $Q$  in the new mesh cell containing side  $s$  is therefore 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 mesh then the contribution to cell on the left is  $\Delta_s^N = \frac{1}{2} q_0 (x_1^s y_2^s - x_2^s y_1^s)$  while the contribution to the cell on its right is just  $-\Delta_s^N$ , where  $q_0$  is the volumetric density of the old mesh cell in which side  $s$  lies.

Adding  $\Delta_s^O$  and  $\Delta_s^N$  for each of new mesh cell yields the quantity  $Q$  contained in each of the new-mesh cell.

## 2.6 Application to Two and Three Dimensional 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 as,

$$\text{in two dimensional space} \quad q_t + a q_x + b q_y = 0 \quad (2.6)$$

$$\text{and in three dimensional space} \quad q_t + a q_x + b q_y + c q_z = 0 \quad (2.7)$$

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

$$\text{and} \quad q = f(x-at, y-bt)$$

$$q = f(x-at, y-bt, z-ct)$$

for the two and three dimensional space, respectively.

In two dimensional case, the equation 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 "flux") from one grid to another. The three dimensional equation is more suitable as the model equation of the equations of motion. The domain is divided into two subdomains which are independent from each other. 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. Results have been compared with the solutions from single grid calculations. Satisfactory results have been obtained. Details of this study can be found in Ref. [104].

### 3. APPLICATION TO THE AERODYNAMIC CONFIGURATIONS

The ultimate equations to be solved in CFD 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 [105] full potential equations [106-109] and Euler equations [110-115], 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 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 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 three-dimensional flows. The 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, 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 economic than methods based on the Navier-Stokes equations. A number of efficient and reliable numerical schemes have been developed for the Euler equations [110-115]. Even though viscous terms are neglected, solutions to Euler equations agree well with the experimental results. Shock waves captured in this model agree with the Rankine-Hugoniot relations regardless of their strength. And, more importantly, 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 aircraft and missiles where shock waves are strong, to more complex non-uniform shear flows past wings. Details of the Euler equations are given in section 3.1.

The Euler solution procedure is based on a center finite-volume scheme with explicit Runge-Kutta time stepping [116]. This type of scheme was first used by Jameson et al. [117], but the present scheme differs significantly from this 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 application [118-121]. The finite-volume scheme is described in section 3.2.

In most instances the solution to the first order steady state equations is desired. The steady state Euler equations change 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 from some initial guessed flow field through time until an asymptotic steady state is achieved. 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 difference and fourth difference damping terms are added to the Euler equation. The fourth difference terms are global and linear whereas the pressure-controlled second-difference 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.

Generally, to reach a steady state solution requires a large number of iteratives and a long computational time [122]. 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 solution. This concept is introduced in Sec. 3.6. The explicit three-stage Runge-Kutta integration scheme is also addressed in Sec. 3.5.

### 3.1 Governing Equations

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

where

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

$$Q = \begin{matrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{matrix}$$

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

and

$\rho$  = density

$u$  =  $x$  - component of velocity

$v$  =  $y$  - component of velocity

$w$  =  $z$  - component of velocity

$E$  = total energy = internal energy + kinetic energy (in the absence of Potential Energy) =  $\rho e + 0.5 (u^2 + v^2 + w^2)$

or in the integral form

$$\frac{d}{dt} \int_{\Omega} Q \, dx dy dz = \oint_{\Omega} (n_x \cdot F + n_y \cdot G + n_z \cdot H) \, ds = 0 \quad (3.2)$$

where  $\Omega$  = arbitrary finite region

$n_x$  =  $x$  component of normal vector at the boundary of the region

$n_y$  =  $y$  component of normal vector at the boundary of the region

$n_z$  =  $z$  component of normal vector at the boundary of the region

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

$$p = (\gamma - 1) \rho e \quad (3.3)$$

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

An assumption has been made, in writing Eqs. (3.1) and (3.2), that the fluid has undergone no external forces. It can be shown that the system of conservation laws given by Eqs. (3.1) or (3.2) is hyperbolic [17]. 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  $\Omega$  and then shrinking  $\Omega$  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  $\Omega$ . 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 solutions 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 angle of attack. Although secondary vortices brought about by viscous effect, on the leeward side of the wing are not modeled, their effects on the primary vortices are small [123]. 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 (Crocco's theorem) [124]. 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 [125] for the flow around an airfoil with a sharp trailing edge. However, some numerical evidence prevails 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 [126] 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 channel 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.

### 3.2 Spatial Finite-Volume Discretization

Development of a method to solve the 3D Euler equations has been made [37,116,127-129]. 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 or global coordinate transformation, and its toleration of grid singularities because the flow equations are balanced only within the cells of the grid [130], 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 a Kutta condition [125]. In contrast, separated flows are obtained by space marching methods only if a 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 mesh all of a given grid (see Fig. 3.1), i.e.

$$\frac{d}{dt} \int_{\Omega_{i,j,k}} \vec{Q} \, dx dy dz + \oint_{\partial\Omega_{i,j,k}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) \, ds = 0 \quad (3.4)$$

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

By the mean-value theorem, Eq. (3.4) becomes

$$VOL_{i,j,k} \frac{d\vec{Q}}{dt}_{i,j,k} + \delta_I \tilde{F}_{i,j,k} + \delta_J \tilde{G}_{i,j,k} + \delta_K \tilde{H}_{i,j,k} = 0 \quad (3.5)$$

where  $\delta_I \tilde{F}_{i,j,k} = \tilde{F}_{i+1/2,j,k} - \tilde{F}_{i-1/2,j,k}$

$$\delta_J \tilde{G}_{i,j,k} = \tilde{G}_{i,j+1/2,k} - \tilde{G}_{i,j-1/2,k} \quad (3.6)$$

$$\delta_K \tilde{H}_{i,j,k} = \tilde{H}_{i,j,k+1/2} - \tilde{H}_{i,j,k-1/2}$$

and

$$\begin{aligned}
 \tilde{F}_{i+1/2,j,k} &= SIX_{i+1/2,j,k} \cdot \vec{F}_{i+1/2,j,k} + SIY_{i+1/2,j,k} \cdot \vec{G}_{i+1/2,j,k} \\
 &\quad + SIZ_{i+1/2,j,k} \cdot \vec{H}_{i+1/2,j,k} \\
 \tilde{F}_{i-1/2,j,k} &= SIX_{i-1/2,j,k} \cdot \vec{F}_{i-1/2,j,k} + SIY_{i-1/2,j,k} \cdot \vec{G}_{i-1/2,j,k} \\
 &\quad + SIZ_{i-1/2,j,k} \cdot \vec{H}_{i-1/2,j,k} \\
 \tilde{G}_{i,j+1/2,k} &= SIX_{i,j+1/2,k} \cdot \vec{F}_{i,j+1/2,k} + SIY_{i,j+1/2,k} \cdot \vec{G}_{i,j+1/2,k} \\
 &\quad + SIZ_{i,j+1/2,k} \cdot \vec{H}_{i,j+1/2,k} \\
 \tilde{G}_{i,j-1/2,k} &= SIX_{i,j-1/2,k} \cdot \vec{F}_{i,j-1/2,k} + SIY_{i,j-1/2,k} \cdot \vec{G}_{i,j-1/2,k} \\
 &\quad + SIZ_{i,j-1/2,k} \cdot \vec{H}_{i,j-1/2,k} \\
 \tilde{H}_{i,j,k+1/2} &= SIX_{i,j,k+1/2} \cdot \vec{F}_{i,j,k+1/2} + SIY_{i,j,k+1/2} \cdot \vec{G}_{i,j,k+1/2} \\
 &\quad + SIZ_{i,j,k+1/2} \cdot \vec{H}_{i,j,k+1/2} \\
 \tilde{H}_{i,j,k-1/2} &= SIX_{i,j,k-1/2} \cdot \vec{F}_{i,j,k-1/2} + SIY_{i,j,k-1/2} \cdot \vec{G}_{i,j,k-1/2} \\
 &\quad + SIZ_{i,j,k-1/2} \cdot \vec{H}_{i,j,k-1/2}
 \end{aligned} \tag{3.7}$$

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

$$\begin{aligned}
[\vec{F}(\vec{Q})]_{i+1/2,j,k} &= \vec{F}(\mu_I \vec{Q}_{i+1/2,j,k}) \\
[\vec{F}(\vec{Q})]_{i,j+1/2,k} &= \vec{F}(\mu_J \vec{Q}_{i,j+1/2,k}) \\
[\vec{F}(\vec{Q})]_{i,j,k+1/2} &= \vec{F}(\mu_K \vec{Q}_{i,j,k+1/2}) .
\end{aligned} \tag{3.8a}$$

Expressions similar to eq. (3.8a) are obtained for  $\vec{G}$  and  $\vec{H}$ . The average operator,  $\mu$ , is defined as

$$\begin{aligned}
\mu_I \psi_{i+1/2,j,k} &= 1/2 (\psi_{i+1,j,k} + \psi_{i,j,k}) \\
\mu_J \psi_{i,j+1/2,k} &= 1/2 (\psi_{i,j+1,k} + \psi_{i,j,k}) \\
\mu_K \psi_{i,j,k+1/2} &= 1/2 (\psi_{i,j,k+1} + \psi_{i,j,k}) .
\end{aligned} \tag{3.9}$$

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

$$\begin{aligned}
[\vec{F}(\vec{Q})]_{i+1/2,j,k} &+ \mu_I \vec{F}(\vec{Q}_{i+1/2,j,k}) \\
[\vec{F}(\vec{Q})]_{i,j+1/2,k} &= \mu_J \vec{F}(\vec{Q}_{i,j+1/2,k}) \\
[\vec{F}(\vec{Q})]_{i,j,k+1/2} &= \mu_K \vec{F}(\vec{Q}_{i,j,k+1/2})
\end{aligned} \tag{3.8b}$$

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

If the flux function were linear, alternatives (3.8a) and (3.8b) would be equivalent. For non-linear flux, only scheme (3.8b) provides the correct jump in  $\vec{Q}$  across the shock. Thus, in this study, each term in Eq. (3.7) is defined as

$$F_{i+1/2,j,k} = 1/2 (F_{i+1,j,k} + F_{i,j,k})$$

$$F_{i,j+1/2,k} = 1/2 (F_{i,j+1,k} + F_{i,j,k})$$

$$F_{i,j,k+1/2} = 1/2 (F_{i,j,k+1} + F_{i,j,k}) .$$

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

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

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

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

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

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

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

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

$$\begin{aligned}
S1Y_{i,j,k+1/2} = & 1/2 ((z_{i-1/2,j+1/2,k+1/2} - z_{i+1/2,j-1/2,k+1/2}) \\
& \cdot (x_{i+1/2,j+1/2,k+1/2} - x_{i-1/2,j-1/2,k+1/2}) \\
& - (x_{i-1/2,j+1/2,k+1/2} - x_{i+1/2,j-1/2,k+1/2}) \\
& \cdot (z_{i+1/2,j+1/2,k+1/2} - z_{i-1/2,j-1/2,k+1/2}))
\end{aligned}$$

$$\begin{aligned}
S1Z_{i,j,k+1/2} = & 1/2 ((x_{i-1/2,j+1/2,k+1/2} - x_{i+1/2,j-1/2,k+1/2}) \\
& \cdot (y_{i+1/2,j+1/2,k+1/2} - y_{i-1/2,j-1/2,k+1/2}) \\
& - (y_{i-1/2,j+1/2,k+1/2} - y_{i+1/2,j-1/2,k+1/2}) \\
& \cdot (x_{i+1/2,j+1/2,k+1/2} - x_{i-1/2,j-1/2,k+1/2}))
\end{aligned}$$

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 (3.5) together with (3.8b) 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 equation. Its difference stencil is that of a finite-difference scheme, but it differs in that cell-averaged 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. (3.5) 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 consideration.

Eriksson [130] 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 alteration the finite-difference scheme is unstable even if the singularities is 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. The important aspect of the finite-volume approach is that it suits well with 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 [131]. 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 [132]. 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 literatures has been developed 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 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: (i) 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, equation (3.5), can be written as

$$\frac{d}{dt} \vec{Q}_{ijk} = \vec{F}_C(\vec{Q}_{ijk}) + \vec{F}_D(\vec{Q}_{ijk}) = \vec{F}(\vec{Q}_{ijk}) \quad (3.10)$$

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

$$f_{ijk} = \chi \{ \delta_I [s_I (\vec{Q}_{ijk}) \delta_I] + \delta_J [s_J (\vec{Q}_{ijk}) \delta_J] + \delta_K [s_K (\vec{Q}_{ijk}) \delta_K] \} \vec{Q}_{ijk} \quad (3.11)$$

where  $\chi$  is a constant and  $s_I, s_J$  and  $s_K$  are coefficients that depend on the solution field through the pressure according to

$$s_I = \|\delta_I^2 LP_{ijk}\|, \quad s_J = \|\delta_J^2 LP_{ijk}\| \text{ and } s_K = \|\delta_K^2 LP_{ijk}\|,$$

where  $\delta_I^2, \delta_J^2$  and  $\delta_K^2$  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  $LP_{ijk} = \text{Log}(p_{i,j,k})$ .

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 Term

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

$$DQ_{ijk} = -\gamma(\delta_I^4 + \delta_J^4 + \delta_K^4) Q_{ijk}, \quad (3.12)$$

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

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

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

and  $\gamma$  is a constant. Linear extrapolation is used at the boundary cells.

### 3.4 Boundary Conditions

For the computation of many fluid dynamic problems more difficulty is encountered in satisfying the boundary conditions than in balancing the differential equations at the interior points of the flow field. This is so 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 [133], still some way 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 [124] have presented a mathematical theory for the practical application of local absorbing boundary conditions at artificial boundaries. Their 'First Approximation' is adapted to this study.

Four distinct types of boundary conditions: conditions at solid walls, periodic conditions across coordinate cuts, flow into or out of the artificial boundary, 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, Eq. (3.4), thus is simplified to

$$\int_{\text{solid wall}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = \int_{\text{solid wall}} \vec{S} ds \quad (3.13)$$

where

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

Equation (3.13) is used to derive the contribution from those mesh walls which coincide with a solid wall. For example, if  $j = 1/2$  is denoted for these mesh cell walls (Fig. 3.2) then Eq. (3.13) is approximated by

$$\int_{\text{solid wall}} (\hat{n}_x \cdot \vec{F} + \hat{n}_y \cdot \vec{G} + \hat{n}_z \cdot \vec{H}) ds = \begin{matrix} 0 \\ \text{SJX}_{i, 1/2, k} \cdot p_{i, 1/2, k} \\ \text{SJY}_{i, 1/2, k} \cdot p_{i, 1/2, k} \\ \text{SJZ}_{i, 1/2, k} \cdot p_{i, 1/2, k} \\ 0 \end{matrix} \quad (3.14)$$

where

$$\begin{aligned} \text{SJX}_{i, 1/2, k} = & 1/2 ((y_{i+1/2, 1/2, k-1/2} - y_{i-1/2, 1/2, k+1/2}) \\ & \cdot (z_{i+1/2, 1/2, k+1/2} - z_{i-1/2, 1/2, k-1/2}) \\ & - (z_{i+1/2, 1/2, k-1/2} - z_{i-1/2, 1/2, k+1/2}) \\ & \cdot (y_{i+1/2, 1/2, k+1/2} - y_{i-1/2, 1/2, k-1/2})) \end{aligned}$$

$$\begin{aligned} \text{SJY}_{i, 1/2, k} = & 1/2 ((z_{i+1/2, 1/2, k-1/2} - z_{i-1/2, 1/2, k+1/2}) \\ & \cdot (x_{i+1/2, 1/2, k+1/2} - x_{i-1/2, 1/2, k-1/2}) \\ & - (x_{i+1/2, 1/2, k-1/2} - x_{i-1/2, 1/2, k+1/2}) \\ & \cdot (z_{i+1/2, 1/2, k+1/2} - z_{i-1/2, 1/2, k-1/2})) \end{aligned}$$

$$\begin{aligned} \text{SJZ}_{i, 1/2, k} = & 1/2 ((x_{i+1/2, 1/2, k-1/2} - x_{i-1/2, 1/2, k+1/2}) \\ & \cdot (y_{i+1/2, 1/2, k+1/2} - y_{i-1/2, 1/2, k-1/2}) \\ & - (y_{i+1/2, 1/2, k-1/2} - y_{i-1/2, 1/2, k+1/2}) \\ & \cdot (x_{i+1/2, 1/2, k+1/2} - x_{i-1/2, 1/2, k-1/2})) \end{aligned}$$

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

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

### 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 [123] 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

$$\frac{\partial \vec{Q}}{\partial t} + A \frac{\partial \vec{Q}}{\partial x} = 0 \quad (3.15)$$

where  $\vec{Q}$  represents the characteristic variables and the Jacobian matrix  $A$  can be written as

$$A = \begin{array}{c|ccc} & 0 & \alpha & \beta & \epsilon \\ \hline \alpha c^2 - uU + k\alpha V^2 & \alpha(1-k)u + U & \beta u - k\alpha v & \epsilon u - k\alpha w \\ \beta c^2 - vU + k\beta V^2 & \alpha v - k\beta u & \beta(1-k)v + U & \epsilon v - k\beta w \\ \epsilon c^2 - wU + k\epsilon V^2 & \alpha w - k\epsilon u & \beta w - k\epsilon v & \epsilon(1-k)w + U \end{array},$$

$$\text{where } \alpha = \vec{s} \cdot \hat{e}_x; \beta = \vec{s} \cdot \hat{e}_y; \epsilon = \vec{s} \cdot \hat{e}_z$$

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

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

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

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

$$\text{where } a_{\pm} = 1/2 kU \pm \left[ \frac{1}{4} k^2 U^2 + c^2 (\alpha^2 + \beta^2 + \epsilon^2) \right]^{1/2}$$

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. (3.15):

$$\frac{\partial}{\partial t} \vec{\phi} + \Lambda \frac{\partial \vec{\phi}}{\partial x} = 0 \quad \text{where } \vec{\phi} = T^{-1} \vec{Q}, \quad \Lambda = T^{-1} A T = \begin{array}{ccccc} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array} \begin{array}{c} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{array} \quad (3.16)$$

After the intermediate variables

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

$$\xi = \alpha^2 + \beta^2 + \epsilon^2, \quad Q_{\pm} = kU - a_{\pm}, \quad R_{\pm} = \epsilon Q_{\pm} - \zeta kw, \quad p_{\pm} = kwa_{\pm} + \epsilon c^2$$

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

$$T = \begin{array}{cc} \begin{array}{c} k\tilde{U} \\ kvU - \beta(kv^2 + c^2) \\ -kuU + \alpha(kv^2 + c^2) \\ 0 \end{array} & \begin{array}{c} 0 \\ \tilde{V} \\ \tilde{W} \\ \tilde{U} \end{array} & \begin{array}{c} R_+ \\ uR_+ + \alpha p_+ \\ vR_+ + \beta p_+ \\ wR_+ - k(\alpha u + \beta v)a_+ \\ -(\alpha^2 + \beta^2)c^2 \end{array} & \begin{array}{c} R_- \\ uR_- + \alpha p_- \\ vR_- + \beta p_- \\ wR_- - k(\alpha u + \beta v)a_- \\ -(\alpha^2 + \beta^2)c^2 \end{array} \end{array}$$

and

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

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  $Q = T\phi$ . At outflow it is  $\phi^{(3)}$  that is given the values of undisturbed flow, and  $\phi^{(1)}$ ,  $\phi^{(2)}$  and  $\phi^{(4)}$  are extrapolated from the computational field.

### 3.3.4 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 conservative approach which offers the conservation of fluxes at the interfaces. The conservative treatment at the interfaces may be 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 the common boundary (interface) between two grids as shown in figure 3.5. The application of the finite volume approach to the cell  $(i,j, NK-1)$  requires the integrated fluxes,  $H_{ij}^{(1)} = h_{ij}^{(1)} \cdot A_{ij}^{(1)}$  at the cell face  $k = NK - 1/2$ , where  $A_{ij}^{(1)}$  is the cell surface area at  $k = NK - 1/2$ . For the interior cell, fluxes at the cell walls are computed by taking the average of the flux functions evaluated at cell centers, i.e.,  $h_{i,j,k+1/2}^{(1)} = 1/2 (h_{i,j,k} + h_{i,j,k+1})$ . At the cell face  $k = NK - 1/2$  of grid 1,  $h_{i,j,NK}$  which contain in

grid 2 are needed. They are computed by interpolating the flux functions evaluates at cell centers of grid 2.

For grid 2, the integrated fluxes  $H_{i,j,-1/2}^{(2)}$  are required for the computations of cell  $k=1$ . They are obtained by applying the conservative rezoning algorithm described previously. To illustrate how the algorithm is applied, consider an interface between two grids as shown in figure 3.6. Here the subscript  $k$  is dropped since the interface lies on the cell wall of constant  $k$ . The conservative treatment at the interface requires that the integrated fluxes going through an area  $A_{k\ell}^{(2)}$  be the same for both grids, i. e.,  $H_{k\ell}^{(1)} = H_{k\ell}^{(2)}$ . From Fig. 3.6,  $H_{k\ell}^{(1)}$  can be evaluated as

$$H_{k\ell}^{(1)} = \sum_{n=1}^p H_{ij}^{(1)} \cdot \frac{A_{k\ell ij}^{(2)}}{A_{ij}^{(1)}} = \sum_{n=1}^p h_{ij}^{(1)} \cdot A_{k\ell ij}^{(2)} \quad (3.17)$$

where  $A_{k\ell ij}^{(2)}$  is the portion of the area  $A_{k\ell}^{(2)}$  which contain in the area  $A_{ij}^{(1)}$  and  $p$  is the number of the surface mesh of grid 1 which contain in the area  $A_{k\ell}^{(2)}$ . In this manner the conservative rezoning algorithm can be applied to compute

$$H_{k\ell}^{(2)} = H_{k\ell}^{(1)} = \sum_{n=1}^p h_{ij}^{(1)} \cdot A_{k\ell ij}^{(2)} \quad (3.18)$$

since  $h_{ij}^{(1)}$  are known from the previous interface treatment for grid 1.

### 3.5 Numerical Time Integration

The complete semi-discrete scheme, Eq. (3.10), including all boundary conditions defines a unique system of nonlinear ordinary differential equations

$$u_t = F(u) \quad ; \quad u(0) = u_0 \quad (3.19)$$

which must be integrated in time by numerical means. The explicit three-stage Runge-Kutta scheme presented by Gary [134] is used in this study. This scheme is defined by the following algorithm:

$$\begin{aligned} u^*(t_{n+1}) &= u(t_n) + \Delta t F(u(t_n)) \\ u^{**}(t_{n+1}) &= u(t_n) + \frac{1}{2} \Delta t F(u(t_n)) + \frac{1}{2} \Delta t F(u^*(t_{n+1})) \\ u(t_{n+1}) &= u(t_n) + \frac{1}{2} \Delta t F(u(t_n)) + \frac{1}{2} \Delta t F(u^{**}(t_{n+1})) \end{aligned} \quad (3.20)$$

It can be shown that this scheme, when applied to the semi-discretized Euler Eqs. (3.10) is second-order accurate and is stable with a CFL-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 approach is that the properties of these schemes 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 so because time step used in explicit methods is restricted to a maximum value according to the CFL (Courant-Friedrichs-Lewy) condition [135]. 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 the applications where only steady state solutions are desired, and true time accuracy is of no concern, it has been found [136] that the use of the "local time step" scaling is highly beneficial to accelerate the convergence to steady state solutions. 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:

$$\vec{Q}_t + \vec{F}(\vec{Q})_x + \vec{G}(\vec{Q})_y = 0 \quad (3.21)$$

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 (3.21) can be transformed into the computational space as

$$(J\vec{Q})_t + (y \vec{F}_\eta - x \vec{G}_\eta)_\xi + (-y \vec{F}_\xi + x \vec{G}_\xi)_\eta = 0 \quad (3.22)$$

where  $J$  = Jacobian of transformation =  $x_\xi y_\eta - x_\eta y_\xi$ .

Equation (3.22) is integrated over the region  $D_{ij}$  and can be written as

$$\frac{d}{dt} \iint_{D_{ij}} \vec{Q} J d\xi d\eta - \oint_{\partial D_{ij}} [(y \vec{F}_\eta - x \vec{G}_\eta) d\xi + (-y \vec{F}_\xi + x \vec{G}_\xi) d\eta] = 0 \quad (3.23)$$

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 same notation have been defined as

$$x_{i,j} = x(\xi_i, \eta_j) \quad ; \quad y_{i,j} = y(\xi_i, \eta_j) \quad , \quad \vec{Q}_{i,j} = \vec{Q}(x_{i,j}, y_{i,j}, t)$$

$$\vec{F}_{i,j} = \vec{F}(\vec{Q}_{i,j}) \quad , \quad \vec{G}_{i,j} = \vec{G}(\vec{Q}_{i,j})$$

$$\delta_{\xi} \vec{Q}_{i,j} = \vec{Q}_{i+1/2,j} - \vec{Q}_{i-1/2,j}, \quad \delta_{\eta} \vec{Q}_{i,j} = \vec{Q}_{i,j+1/2} - \vec{Q}_{i,j-1/2}$$

$$\mu_{\xi} \vec{Q}_{i,j} = 1/2 (\vec{Q}_{i+1/2,j} + \vec{Q}_{i-1/2,j}), \quad \mu_{\eta} \vec{Q}_{i,j} = 1/2 (\vec{Q}_{i,j+1/2} + \vec{Q}_{i,j-1/2})$$

Equation (3.23) can be approximated by

$$\begin{aligned} \frac{d}{dt} \vec{Q}_{i,j} \cdot \iint_{D_{i,j}} J d\xi d\eta + \delta_{\eta} [-(\delta_{\xi} y_{i,j})(\mu_{\eta} \vec{F}_{i,j}) + (\delta_{\xi} x_{i,j})(\mu_{\eta} \vec{G}_{i,j})] \\ + \delta_{\xi} [(\delta_{\eta} y_{i,j})(\mu_{\xi} \vec{F}_{i,j}) - (\delta_{\eta} x_{i,j})(\mu_{\xi} \vec{G}_{i,j})] = 0 \end{aligned} \quad (3.24)$$

Equation (3.24) is a semi-discrete centered scheme for the two dimensional Euler equation. To derive the local time step scaling for equation 3.24, first the transformation matrices are freezed at mesh point (i,j), i.e.

$$\begin{aligned} x_{\xi}(\xi, \eta) \approx x_{\xi}(\xi_i, \eta_j) = \tilde{x}_{\xi}; \quad x_{\eta}(\xi, \eta) \approx x_{\eta}(\xi_i, \eta_j) = \tilde{x}_{\eta} \\ y_{\xi}(\xi, \eta) \approx y_{\xi}(\xi_i, \eta_j) = \tilde{y}_{\xi}; \quad y_{\eta}(\xi, \eta) \approx y_{\eta}(\xi_i, \eta_j) = \tilde{y}_{\eta} \\ J(\xi, \eta) = x_{\xi}(\xi, \eta) y_{\eta}(\xi, \eta) - x_{\eta}(\xi, \eta) y_{\xi}(\xi, \eta) \approx \tilde{x}_{\xi} \tilde{y}_{\eta} - \tilde{x}_{\eta} \tilde{y}_{\xi} = \tilde{J} \end{aligned} \quad (3.25)$$

Thus, Eq. (3.23) becomes

$$\begin{aligned} \tilde{J} \Delta \xi \Delta \eta \frac{d}{dt} \vec{Q}_{i,j} - 1/2 \Delta \xi \tilde{y}_{\xi} (\vec{F}_{i,j+1} - \vec{F}_{i,j-1}) + 1/2 \Delta \xi \tilde{x}_{\xi} (\vec{G}_{i,j+1} - \vec{G}_{i,j-1}) \\ - 1/2 \Delta \eta \tilde{y}_{\eta} (F_{1+1,j} - F_{1-1,j}) - 1/2 \Delta \eta \tilde{x}_{\eta} (G_{1+1,j} - G_{1-1,j}) = 0 \end{aligned} \quad (3.26)$$

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

$$\begin{aligned} \vec{F}(\vec{Q}) \approx \vec{F}(\vec{Q}_{i,j}) + \tilde{A} \cdot (\vec{Q} - \vec{Q}_{i,j}); \quad \vec{G}(\vec{Q}) \approx \vec{G}(\vec{Q}_{i,j}) + \tilde{B} \cdot (\vec{Q} - \vec{Q}_{i,j}) \\ \tilde{A} = \frac{\partial \vec{F}}{\partial \vec{Q}}(\vec{Q}_{i,j}); \quad \tilde{B} = \frac{\partial \vec{G}}{\partial \vec{Q}}(\vec{Q}_{i,j}) \end{aligned} \quad (3.27)$$

where  $\tilde{A}$  and  $\tilde{B}$  are the Jacobian matrices evaluated at  $\vec{Q}_{i,j}$  with the aid from Eq. (3.27), Eq. (3.26) can be written as

$$\frac{d}{dt} \vec{Q}_{i,j} + \frac{1}{2\Delta\eta\tilde{J}} (-\tilde{y}_{\xi} \tilde{A} + \tilde{x}_{\xi} \tilde{B}) (\vec{Q}_{i,j+1} - \vec{Q}_{i,j-1}) + \frac{1}{2\Delta\xi\tilde{J}} (\tilde{y}_{\eta} \tilde{A} - \tilde{x}_{\eta} \tilde{B}) (\vec{Q}_{i+1,j} - \vec{Q}_{i-1,j}) = 0 \quad (3.28)$$

which can be treated by Fourier analysis. Let

$$\vec{Q}_{ij}(t) = \hat{Q}_{ij}(t) = \hat{Q}_l^{st} (\hat{l}^{\theta_1})^i (\hat{l}^{\theta_2})^j \quad (3.29) \quad -\pi < \theta_1 < \pi, \quad -\pi < \theta_2 < \pi \quad (3.29)$$

be the solution of Eq. (3.28)

Substitution of Eq. (3.29) into Eq. (3.28) yields the eigenvalue problem

$$[sI + \frac{\hat{l} \sin(\theta_1)}{\Delta\xi\tilde{J}} (\tilde{y}_{\eta} \tilde{A} - \tilde{x}_{\eta} \tilde{B}) + \frac{\hat{l} \sin(\theta_2)}{\Delta\eta\tilde{J}} (-\tilde{y}_{\xi} \tilde{A} + \tilde{x}_{\xi} \tilde{B})] \hat{Q} = 0 \quad (3.30)$$

A nontrivial solution to Eq. (3.30) can be found only if

$$\det \left[ \left( \frac{\hat{y}_{\eta} \sin(\theta_1)}{\Delta\xi\tilde{J}} - \frac{\tilde{y}_{\xi} \sin(\theta_2)}{\Delta\eta\tilde{J}} \right) \tilde{A} + \left( \frac{-\tilde{x}_{\eta} \sin(\theta_1)}{\Delta\xi\tilde{J}} + \frac{\tilde{x}_{\xi} \sin(\theta_2)}{\Delta\eta\tilde{J}} \right) \tilde{B} - (-\hat{l}sI) \right] = 0 \quad (3.31)$$

Since the original conservation law, Eq. (3.21) is hyperbolic, the matrices  $\tilde{A}$  and  $\tilde{B}$  by definition satisfy the condition

$$\det (\alpha \tilde{A} + \beta \tilde{B} - \lambda I) = 0 \Rightarrow \{\lambda_p(\alpha, \beta)\}_{p=1}^n \text{ all real} \quad (3.32)$$

$\alpha, \beta \text{ real}$

Thus, the eigenvalues  $s_p(\theta_1, \theta_2)$  of Eq. (3.31) are all purely imaginary and given by

$$s_p(\theta_1, \theta_2) = -i\lambda_p(\alpha, \beta) \quad ; \quad p = 1, \dots, n \quad (3.33)$$

where

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

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

For the Euler equations the Jacobian matrices  $\tilde{A}$  and  $\tilde{B}$  are known analytically as well as the eigenvalues  $\lambda_p(\alpha, \beta)$ . Thus, the local spectral radius at mesh 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. (3.24) is, then defined by

$$\left( \iint_{D_{ij}} J d\xi d\eta \right) \rho_{i,j} \frac{d}{dt} \vec{Q}_{i,j} + \delta_\eta \left[ -(\delta_\xi y_{i,j})(\mu_\eta \vec{F}_{i,j}) + (\delta_\xi x_{i,j})(\mu_\eta \vec{G}_{i,j}) \right]$$

$$+ \delta_\xi \left[ (\delta_\eta y_{i,j})(\mu_\xi \vec{F}_{i,j}) - (\delta_\eta x_{i,j})(\mu_\xi \vec{G}_{i,j}) \right] = 0 \quad (3.34)$$

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.

## 4. RESULTS AND DISCUSSIONS

### 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. The simplicity of the sphere, both from grid generation point of view and flow field calculation, gives an advantage to the understanding of the approach. 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. However, 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. Solutions for flows over both configurations have been obtained at zero-degree angle of attack. 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. The entire flowfield is divided into two subdomain at about the center of the configurations. Grids in both subdomains are of O-O type with different number of grid points, i.e.,  $21 \times 21 \times 17$  v.s.  $33 \times 33 \times 17$ . Figure 4.1 and 4.2 illustrate grids used for sphere and slender body respectively. 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 with solutions obtained by multiple grid

calculations. Wall pressure coefficient at the center line are compared in Fig. 4.3 and 4.4. Good agreement can be seen clearly from the comparisons. Even though, these comparisons are made for only simple cases but confidence in using the concept of multiple grids should be built up. Indeed, solutions to flow over these configurations can be obtained for various flow conditions and number of grid points. The same agreement is expected to be obtained. 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 [137]. 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 section 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

$$\text{major axis (i.e., span)} = x/\beta \quad 0 < x < L \quad (4.1a)$$

$$\begin{aligned} \text{minor axis (i.e., thickness on} \\ \text{center line)} &= x/\beta \quad 0 < x < 0.2 L \quad (4.1b) \end{aligned}$$

$$= \frac{x}{\beta} \left[ 1 - \left( \frac{x - 0.2L}{0.8x} \right)^4 \right] \quad 0.2L < x < L \quad (4.1c)$$

$$\text{where } \beta^2 = M_\infty^2 - 1$$

Butler [137] 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 [138,139] has obtained experimental results for a Butler Wing with varying Mach number and angle of attack. Abolhassani et al. [140] 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  $\sin^{-1}(1/3.5) = 16.602^\circ$ . The model is mounted in the tunnel by attaching a string support of 12.5 mm. diameter to the lower surface.

Even though solutions can be obtained using a single grid system, multiple grid solutions on a Butler Wing suit well for the purpose of this study. This is because grid of O type is suitable for the front part of the configuration while H-type grid seems to be a better choice near the trailing edge. The multiple grid system is composed of three subdomain grids 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 Butler Wing configuration illustrates the usefulness of the multiple grids approach where grid topology is changed from one subdomain grid to another.

Wall pressure coefficient at various flow conditions are plotted in Fig. 4.3-4.10. In all plots, a solid line indicates wall pressure coefficient obtained from multiple grid calculations. Plots of wall pressure coefficient along the center line at 3.5 Mach number and zero degree angle of attack are shown in Fig. 4.3. In Fig. 4.7a, comparisons are made with ref. [136-139]. Discrepancies near the nose region exist because 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 [138] has made similar conclusion about these

discrepancies. Good agreement for the comparison between multiple grid solutions and solutions obtained from a single grid calculation is shown in Fig. 4.7b. This indicates that the use of multiple grids does not cause the discrepancies in Fig. 4.7a.

The computed pressure coefficients for ten degrees angle of attack are plotted and compared with the Refs. 138 and 140 in Fig. 4.8. 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  $x/c \approx 0.5$  and  $0.7$ . 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. 138. Figure 4.9 shows results at zero degree angle of attack, whereas, results at ten degrees angle of attack is shown in Fig. 4.10. The same conclusion as in the case of 3.5 Mach number can be made.

The results obtained, thus far, demonstrate that the use of multiple grid 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. This study may not yet indicate the usefulness or necessities of the multiple grid 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.

## 5. CONCLUSIONS

Solutions of Euler flow over aerodynamics configurations on multiple grid 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 grid is also addressed. The solutions obtained from this study illustrate a promising future of multiple grids approach. It should be stressed, once more, that the main purpose of this study is 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 grid 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 any 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, 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 grid 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 e.t.c. Good characteristics of a grid system is much easier to obtain in multiple grid approach as compared to a single grid approach. Experience from this study has indicated that efforts to be made to obtain a converged solution as a multiple grid

system is not as much as those to be made on a single grid system. Arguments can be made, however, that configurations used in this study are yet too simple to make any conclusion about this advantage of the multiple grid approach. More studies will have to be made to confirm it.

The next step, which is under investigation, is to apply this approach to simulate the internal/external flow around a fighter-aircraft configuration (Fig. 5.1). This simulation should illustrate the usefulness of the multiple grid approach, since it becomes necessary to use different grid topologies between the internal and external flow. The findings from this study is expected to yield a significant contribution to the field of CFD. After the solutions of the Euler equations is successfully obtained, the applications to the Navier-Stokes equations, including chemical reacting and turbulent flows (providing that proper turbulent modelling is implemented), should be possible.

## REFERENCES

1. 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.
2. Krause, E., "Computational Fluid Dynamics: Its Present Status and Future Direction", Computers and Fluids, Vol. 13, No. 3, 1985, pp. 239-269.
3. 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.
4. Orszag, S. A. and Israeli, M., "Numerical Simulation of Viscous Incompressible Flows," Annual Review of Fluid Mechanics, Vol. 6, 1974, pp. 281-318.
5. 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.
6. Eiseman, P. R., "Grid Generation for Fluid Mechanics Computations," Annual Review of Fluid Mechanics, Vol. 17, 1985, pp. 487-522.
7. Forrest, A. R., "Computational Geometry," Proceedings of the Royal Society of London, A 321, February 1971, pp. 187-195.
8. MacCormack, R. W. and Lomax, H., "Numerical Solution of Compressible Viscous Flows," Annual Review of Fluid Mechanics, Vol. 11, 1979, pp. 289-316.
9. Jameson, A., "The Evaluation of Computational Methods in Aerodynamics," Journal of Applied Mechanics, Vol. 50, December 1983, pp. 1052-1070.
10. Oswatitsch, K. and Rues, D. (Eds.) Symposium Transsonicum II, Springer-Verlag, Berlin/Meidelberg/New York, 1976, pp. 362-574.
11. Turkel, E., "Progress in Computational Physics", Computers and Fluids, Vol. 11, No. 2, 1983, pp. 121-144.
12. 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.
13. 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.
14. Sod, G. A., "Numerical Methods in Fluid Dynamics," Initial and Boundary-Value Problem, Cambridge University Press, London, 1985.
15. Holt, M., Numerical Methods in Fluid Dynamics, Computational Physics Series, 2nd ed., Springer-Verlag, New York/Heidelberg/Berlin, 1984.

16. Peyret, R. and Taylor, T. D., Computational Methods for Fluid Flow, Computational Physics Series, Springer-Verlag, New York/Heidelberg/Berlin, 1983.
17. Richtmyer, R. D. and Morton, K. W., Difference Methods for Initial-Value Problems, Interscience, New York, 1967.
18. Richtmyer, R. D., Principles of Advanced Mathematical Physics, Vol. I, Springer-Verlag, New York, 1978.
19. Serrin, J., Encyclopedia of Physics, Fluid Dynamics I, Ed. by S. Flügge, Vol. VIII/1. Springer-Verlag, Berlin, 1959.
20. Roache, P. J., Computational Fluid Dynamics, Hermosa, Albuquerque, New Mexico, 1972.
21. Patankar, S. V. and Spalding, D. B., Heat and Mass Transfer in Boundary Layers, 2nd ed., Intertext, New York, 1970.
22. Hockney, R. W. and Jesshope, C. R., Parallel Computers, Adam Hilger, Bristol, 1983.
23. Gentzsch, W., Vectorization 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. Thompson, J. F., Warsi, Z.U.A., and Mastin, C. W., Numerical Grid Generation - Foundations and Applications, North-Holland, New York, 1985.
26. Thompson, J. F., Warsi, Z.U.A., and Mastin, C. W., "Boundary-Filled Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, Vol. 47, No. 1, July 1982, pp. 1-108.
27. 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.
28. Smith, R. E. (Ed.) "Numerical Grid Generation Techniques," NASA CP-2166, October 1980.
29. 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.
30. Eiseman, P. R. "Geometric Methods in Computational Fluid Dynamics," ICASE Report No. 80-11, ICASE/NASA Langley Research Center, Hampton, Virginia, April 1980.

31. 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.
32. Marstin, C. W. and Thompson, J. F., "Errors in Finite-Difference Computations on Curvilinear Coordinate System," MSSU-EIRS-ASE-80-4, Mississippi State University, 1980.
33. 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.
34. Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," Journal of Computational Physics, Vol. 33, No. 1, October 1979, pp. 307-338.
35. 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.
36. Eriksson, L. -E., "Three-Dimensional Spline-Generated Coordinate Transformations for Grid Around Wing-Body Configurations," Numerical Grid Generation Techniques, NASA CP-2166, September 1980, pp. 253-264.
37. 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.
38. Eiseman, P. R., "Orthogonal Grid Generation" in Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, New York, 1982, pp. 193-233.
39. 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.
40. 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.
41. Smith, R. E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Navier-Stokes Equations," NASA TM-83123, May 1981.
42. 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.
43. 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.

44. 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.
45. 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.
46. 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.
47. 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.
48. 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.
49. Wu, J. C., El-Refail, M., and Tekondis, S. G., "Solutions of the Unsteady Two-Dimensional Compressible Navier-Stokes Equations Using the Integral Representation Method," AIAA Paper No. 81-6046, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January 12-15, 1981.
50. 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.
51. Starius, G., "Construction of Orthogonal Curvilinear Meshs by Solving Initial Value Problems," Numerische Mathematik, Vol. 28, Fasc. 1, 1977, pp. 25-48.
52. 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.
53. 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.
54. Nakamura, S., "Non-Iterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations," Proc. Eighth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, Berlin, 1982.
55. Edwards, T. A., "Three-Dimension Grid Generation Using Parabolic Partial Differential Equations," AIAA Paper No. 85-0485, AIAA 23rd Aerospace Scientific Meeting, Reno, Nevada, January 1985.

56. 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.
57. 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.
58. Gordon, W. J., "Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation," SIAM Journal of Numerical Analysis, Vol. 8, No. 1, March 1971, pp. 158-177.
59. Gerhard, M. A., "A General Purpose Mesh Generator for Finite Element Codes," M-101, Lawrence Livermore Lab., University of California, Livermore, California, April 1979.
60. Anderson, P. G., and Spradley, L. W., "Finite Difference Grid Generation by Multivariate Blending Function Interpolation, Numerical Grid Generation Techniques," NASA CP-2166, September 1980, pp. 143-156.
61. 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.
62. 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.
63. 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, Viviani, H., ed., Vieweg Verlag, Paris, October 1981.
64. 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.
65. Atta, E., "Component-Adaptive Grid Interfacing," AIAA paper No. 81-0382, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, January 1981.
66. 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.
67. Weatherill, N. P. and Forsey, C. R., "Grid Generation and Flow Calculations for Complex Aircraft Geometries Using a Multi-Block Scheme," AIAA Paper No. 84-1615, AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.
68. Thomas, P. D., "Composite Three-Dimensional Grid Generated by Elliptic System," AIAA Journal, Vol. 20, No. 9, September 1982, pp. 1195-1202.

69. Rubbert, P. E. and Lee, K. D., "Patched Coordinate Systems," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, p. 235-252.
70. 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.
71. 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.
72. 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.
73. 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.
74. 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.
75. Starius, G., "Composite Mesh Difference Methods for Elliptic Boundary Value Problems," Numerische Mathematik, Vol. 28, Fasc. 2, 1977, pp. 243-258.
76. Starius, G., "On Composite Mesh Difference Methods for Hyperbolic Differential Equations," Numerische Mathematik, Vol. 35, Fasc. 3 1980, pp. 241-255.
77. Kreiss, B., "Construction of a Curvilinear Grid," SIAM Journal on Scientific and Statistical Computing, Vol. 4, No. 2, June 1983, pp. 270-279.
78. 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.
79. Thompson, D. S., "A Mesh Embedding Approach for Prediction of Transonic Wing/Body/Store Flow Fields," NASA CP-2201, October 1981, pp. 403-427.
80. Steger, J. A. and Buning, P. G., "Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers," NASA TM-86676, February 1985.
81. Benek, J. A., Buning, P. G. and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper No. 85-1523-, Proceeding AIAA 7th Computational Fluid Dynamic Conference, Cincinnati, Ohio, July 15-17, 1985.

82. 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.
83. 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.
84. 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, June 1982.
85. Dougherty, F. C., "Development of a Chimera Grid Scheme with Application to Unsteady Problem," Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, Stanford, California, May 1985.
86. 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.
87. 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.
88. 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.
89. 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.
90. Chanderjan, N. M. and Steger, J. L., "A Zonal Approach for the Steady Transonic Simulation of Inviscid Rotational Flow," AIAA Paper No. 83-1927, AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts, July 1983.
91. Eberhardt, S. and Baganoff, D., "Overset Grids in Compressible Flow," AIAA Paper No. 85-1524, Proceeding AIAA 7th Computational Fluid Dynamic Conference, Cincinnati, Ohio, July 15-17, 1985, pp. 332-339.
92. Lax, P., Hyperbolic System of Conservation Laws and the Mathematical Theory of Shock Waves, Society for Industrial and Applied Mathematics, Philadelphia 1972.
93. Berger, M. J., "On Conservation at Grid Interfaces," ICASE Report No. 84-43, ICASE/NASA Langley Research Center, Hampton, Virginia, September 1984.

94. 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.
95. Hessenius, 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.
96. 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.
97. Cambier, L., Ghazzi, W., Veuillot, J. P., and Viviand, H., "Une Approche par Domaines pour le calcul d'Ecoulements Compressibles," Cinquieme Colloque International sur les Methodes de calcul, France, December 1981.
98. Rai, M. M., Chakravarthy, S. R., and Hessenius, K. A., "Zonal Grid Calculations Using the Osher Scheme," Computers and Fluids, Vol. 12, No. 3, 1984, pp. 161-173.
99. Dukowicz, J. K., "Conservative Rezoning (Remapping) for General Quadrilateral Meshes," Journal of Computational Physics, Vol. 54, No. 3, June 1984, pp. 411-424.
100. Ramshaw, J. D., "Conservative Rezoning Algorithm for Generalized Two-Dimensional Meshes," Journal of Computational Physics, Vol. 59, No. 2, June 1985, pp. 193-199.
101. 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.
102. 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.
103. Rektorys, K. (Ed.), Survey of Applicable Mathematics, Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1969, p. 207.
104. Kathong, M. and Tiwari, S. N., "Conservative Finite Volume Solutions of a Linear Hyperbolic Transport Equation in Two and Three Dimensions Using Multiple Grids," Progress Report, NCC1-68, Department of Mechanical Engineering and Mechanics, College of Engineering and Technology, Old Dominion University, Norfolk, Virginia, July 1987.
105. Hafez, M. and Lovell, D., "Numerical Solution of Transonic Stream Function Equation," AIAA Journal, Vol. 21, No. 3, March 1983, pp. 327-335.
106. Murman, E. M. and Cole, J. D., "Calculation of Plane, Steady Transonic Flows," AIAA Journal, Vol. 9, No. 1, January 1971, pp. 114-121.

107. 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.
108. 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.
109. 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.
110. 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.
111. 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 1, 1971.
112. Denton, J. D., "A Time Marching Method for Two- and Three-Dimensional Blade-to-Blade Flow," Aeronautical Research Council Report and Memorandum, United Kingdom, 1975.
113. Moretti, G., "The  $\lambda$  - Scheme" Computers and Fluids, Vol. 7, No. 3, September 1979, pp. 191-205.
114. 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.
115. 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.
116. 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.
117. 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.
118. 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.
119. Rizzi, A. and Eriksson, L. -E., "Computation of Inviscid Incompressible Flow with Rotation," Journal of Fluid Mechanics, Vol. 143, October 1985, pp. 275-312.

120. Eriksson, L. -E., "Simulation of Inviscid Flow Around Airfoils and Cascades Based on the Euler Equations," FAA TN-1985-20, The Aeronautical Research Institute of Sweden, Stockholm, Sweden, 1985.
121. 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.
122. 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.
123. 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.
124. Anderson, J. D., Modern Compressible Flow, McGraw-Hill, New-York, 1982, pp. 165-166.
125. Karamcheti, K., Principles of Ideal-Fluid Aerodynamics, Krieger, Malabar, Florida, 1966, pp. 390-393.
126. Engquist, B. and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," Mathematics of Computation, Vol. 31, July 1977, pp. 629-651.
127. Schmidt, W. and Jameson, A., "Recent Developments in Finite-Volume Time-Dependent Techniques for Two and Three Dimensional Transonic Flows," in Computational Fluid Dynamics, Lecture Series Notes VKI-LS-1982-04, von Karman Institute for Fluid Dynamics, Brussels, 1982.
128. Rizzi, A. and Eriksson, L. -E., "Explicit Multistage Finite Volume Procedure to Solve the Euler Equations for Transonic Flow," in Computational Fluid Dynamics, Lecture Series Notes 1983-04, von Karman Institute for Fluid Dynamics, Brussels, 1983.
129. 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.
130. Eriksson, L. -E., "A Study of Mesh Singularities and Their Effects on Numerical Errors," FFA TN1984-10, The Aeronautical Research Institute of Sweden, Stockholm, Sweden, 1984.
131. Lomax, H., "Some Prospects for the Future of Computational Fluid Dynamics," AIAA Journal, Vol. 20, No. 8, August 1982, pp. 1033-1043.
132. MacCormack, R. W. and Paullay, A. J., "The Influence of the Computational Mesh on Accuracy for Initial Value Problems with Discontinuous or Monunique Solutions," Computers and Fluids, Vol. 2, Nos. 3-4, December 1974, pp. 339-361.

133. 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.
134. Gary, J., "On certain Finite-Difference Schemes for Hyperbolic Systems," Mathematics of Computation, Vol. 18, No. 85, January 1964, pp. 1-18.
135. Courant, R., Friedrichs, K. O., and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik," Mathematische Annalen, Vol. 100, 1928, pp. 32-74.
136. Denton, J. D., "Time-Marching Methods for Turbomachinery Flow Calculation," Numerical Methods in Applied Fluid Dynamics, Ed. by B. Hunt, Academic Press, London, 1980.
137. 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.
138. 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.
139. Squire, L. C., "Measured Pressure Distribution and Shock Shapes on a Simple Delta Wing," Cambridge University, Department of Engineering, CUED/Aero/tr 9, 1979.
140. 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.

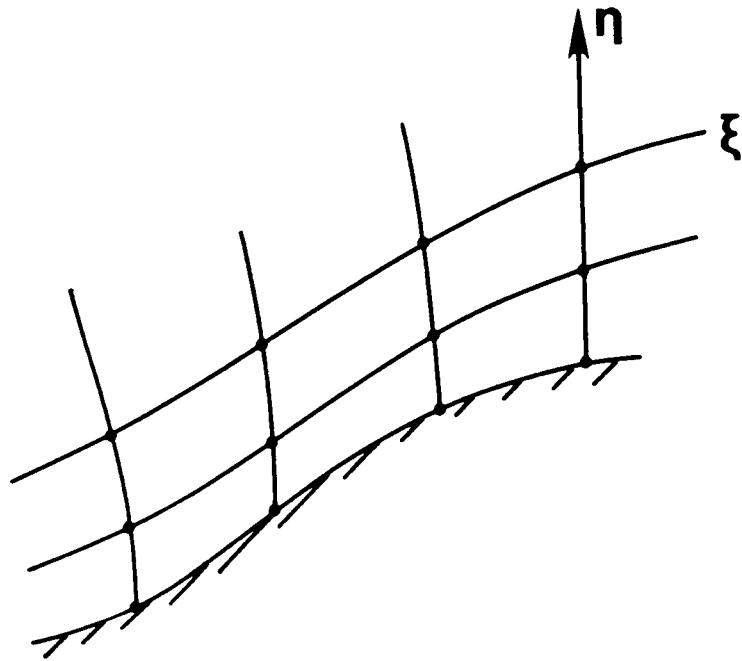
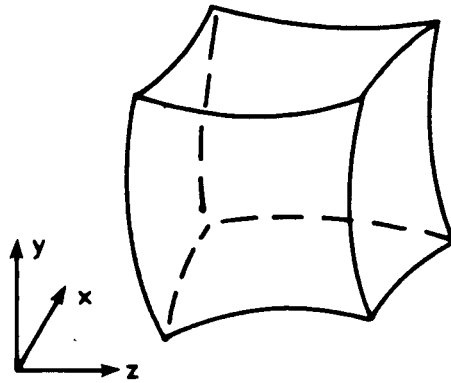
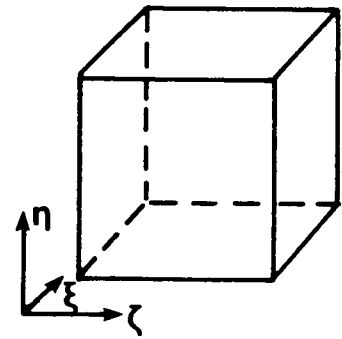


Fig. 2.1 Boundary-fitted coordinate system.



(a) Physical domain



(b) Computational domain

Fig. 2.2 Physical versus Computational domain.

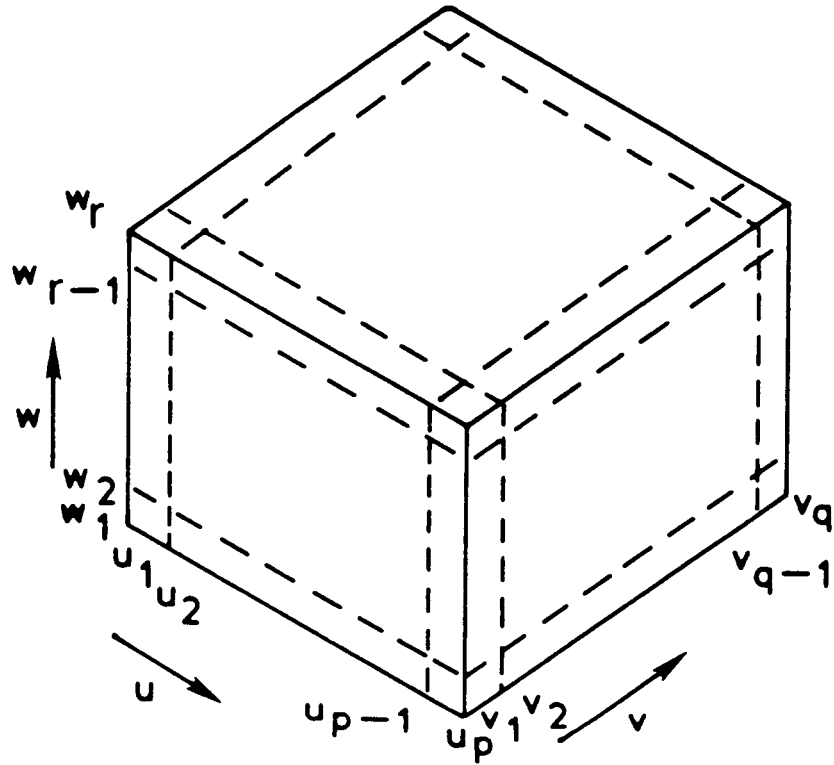


Fig. 2.3 A computational domain where  $\vec{f}$  is known on certain planes.

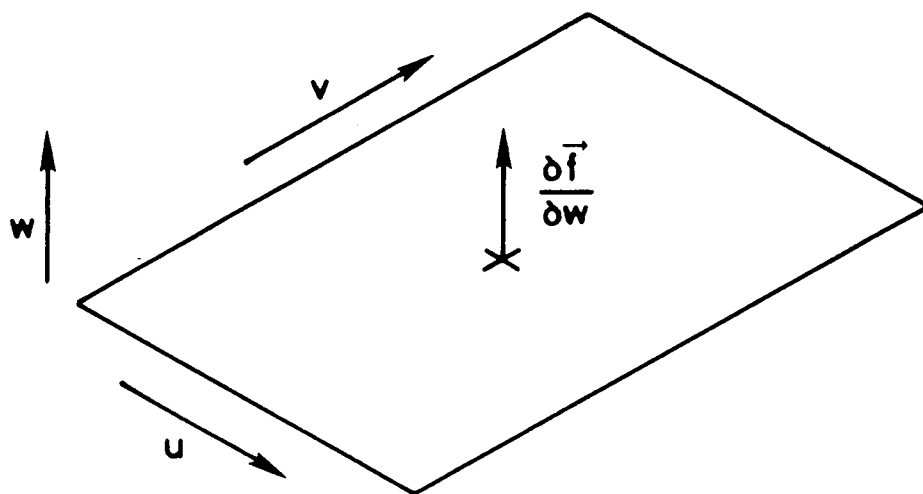


Fig. 2.4 An out-of-surface derivative of  $\vec{f}$ .

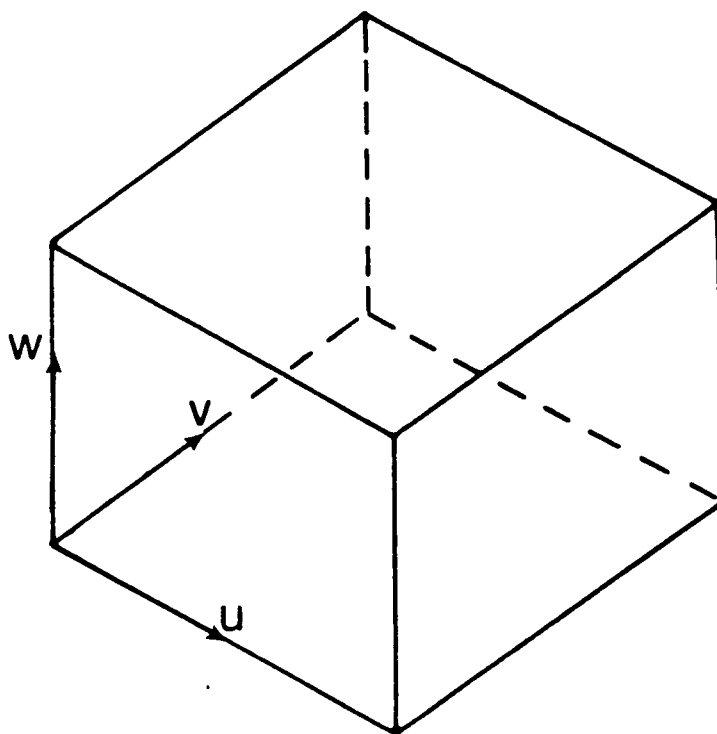


Fig. 2.5 A computational domain where  $\frac{\partial f}{\partial n}$  are specified on the outer surface.

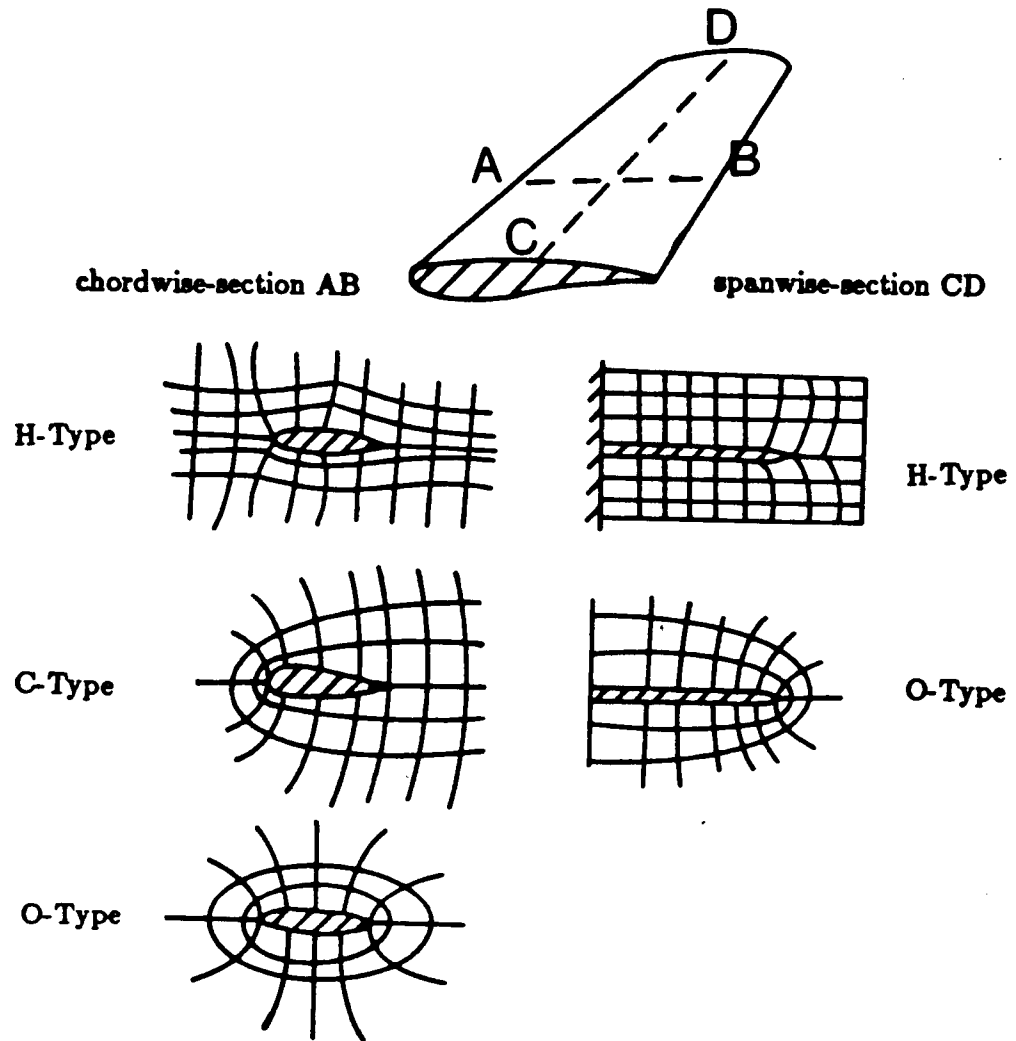


Fig. 2.6 Mapping types for a 3D airfoil.

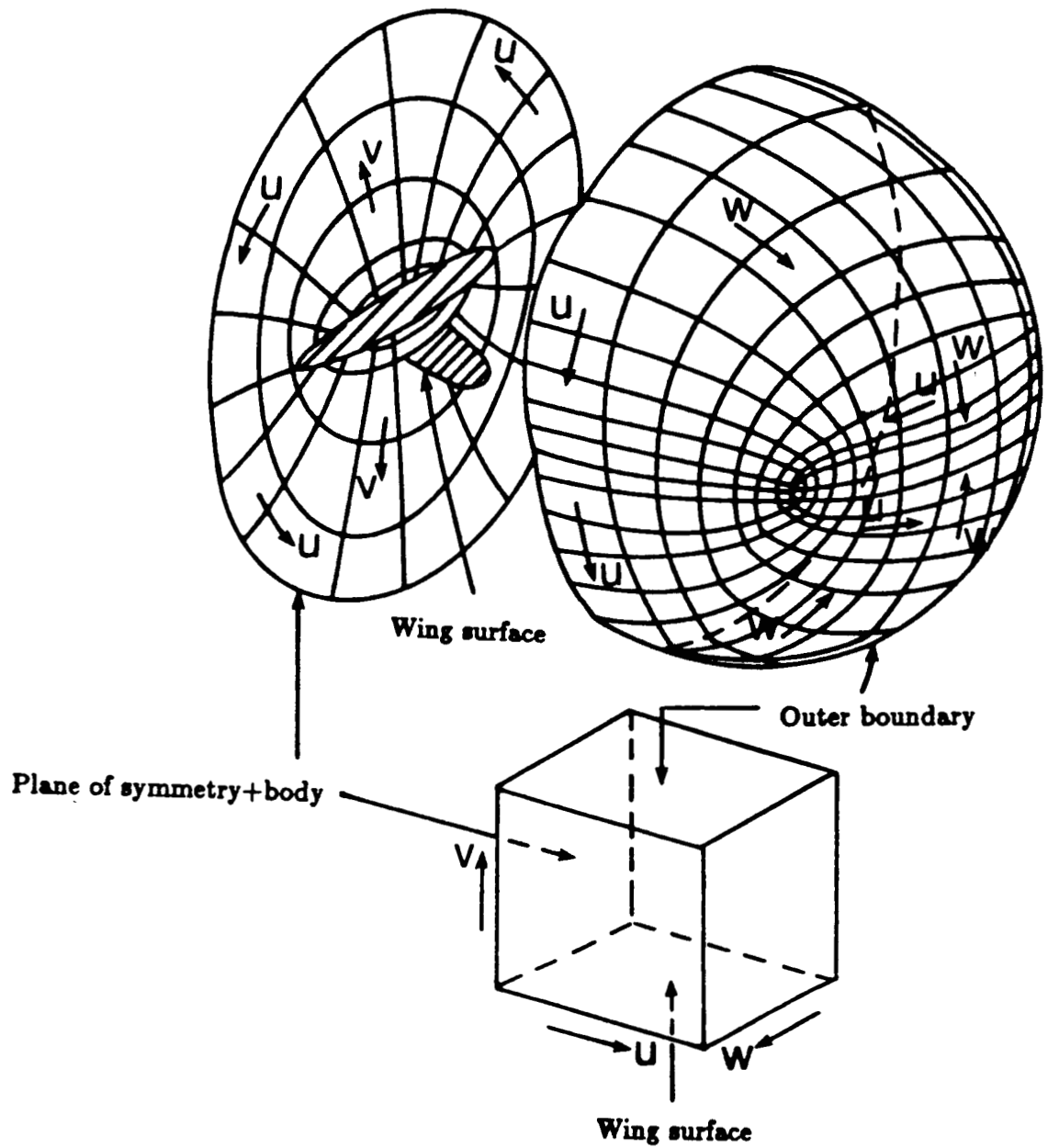


Fig. 2.7 An O-O mapping for a wing-fuselage configuration.

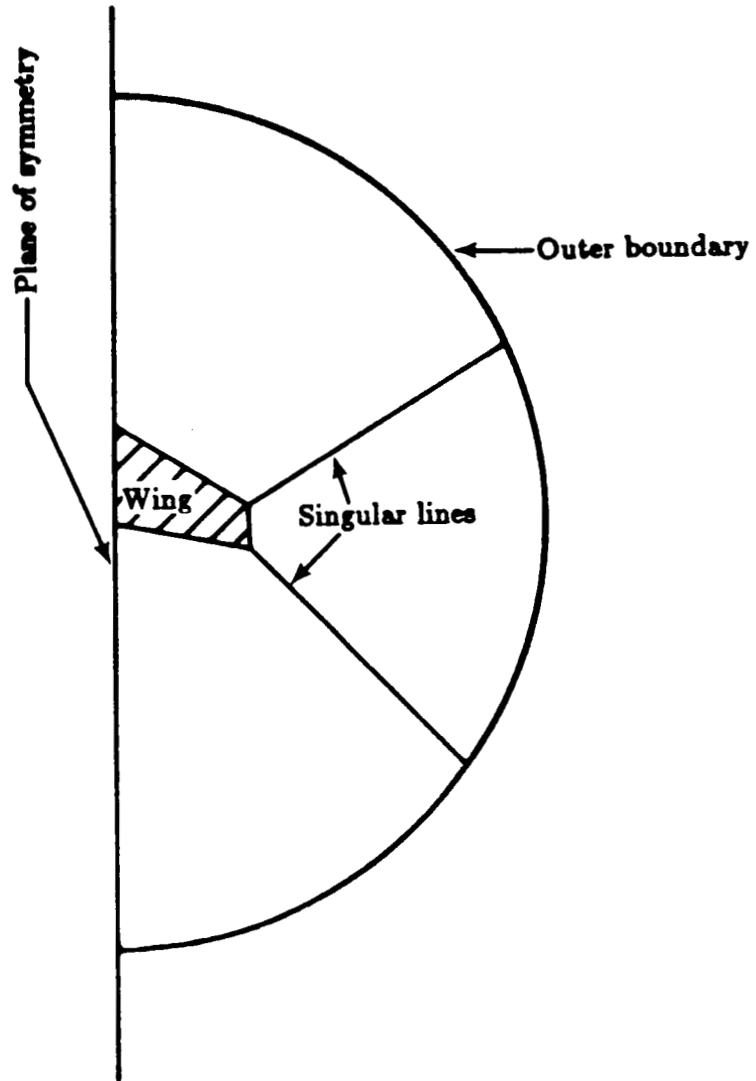


Fig. 2.8 Singularity lines for the O-O mapping.

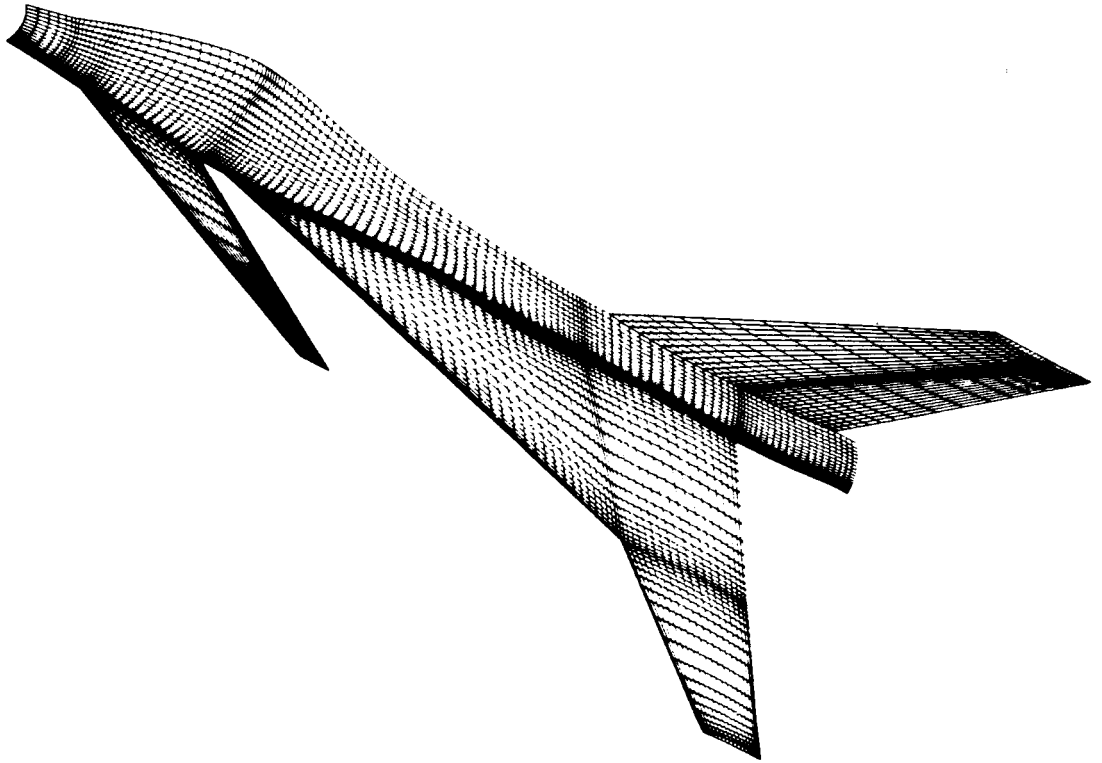


Fig. 2.9 The boundary surface is opened to avoid dealing with singularities.

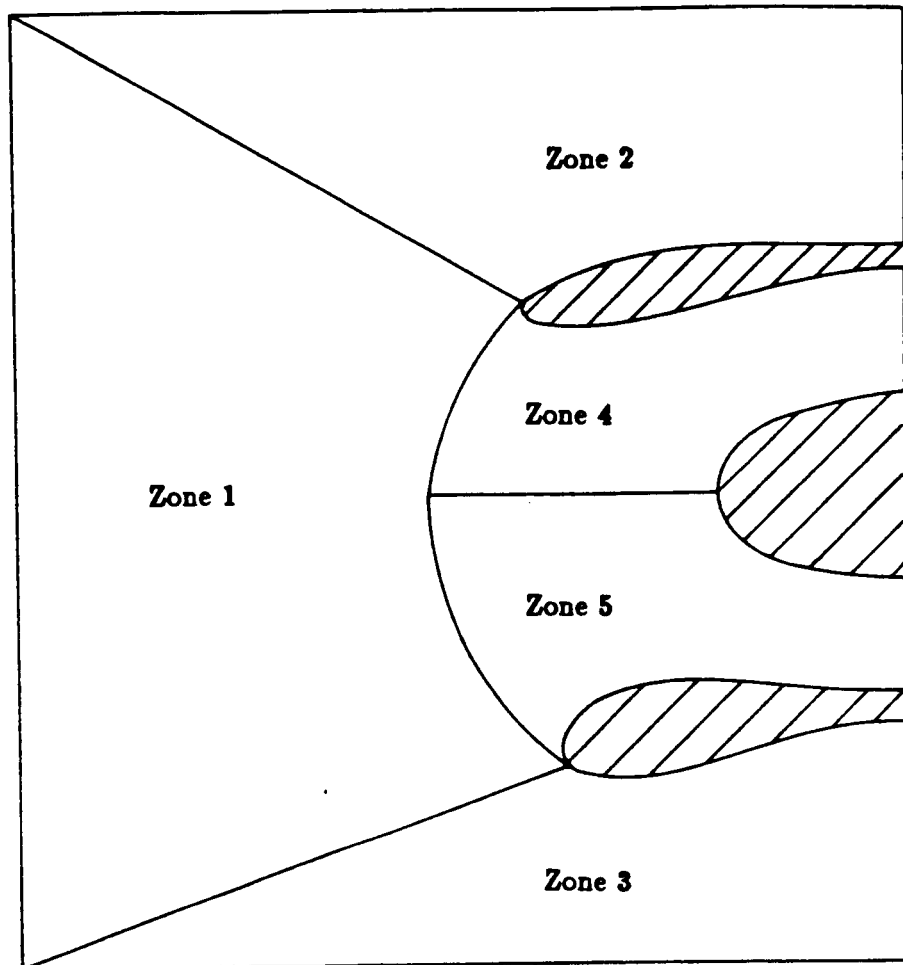
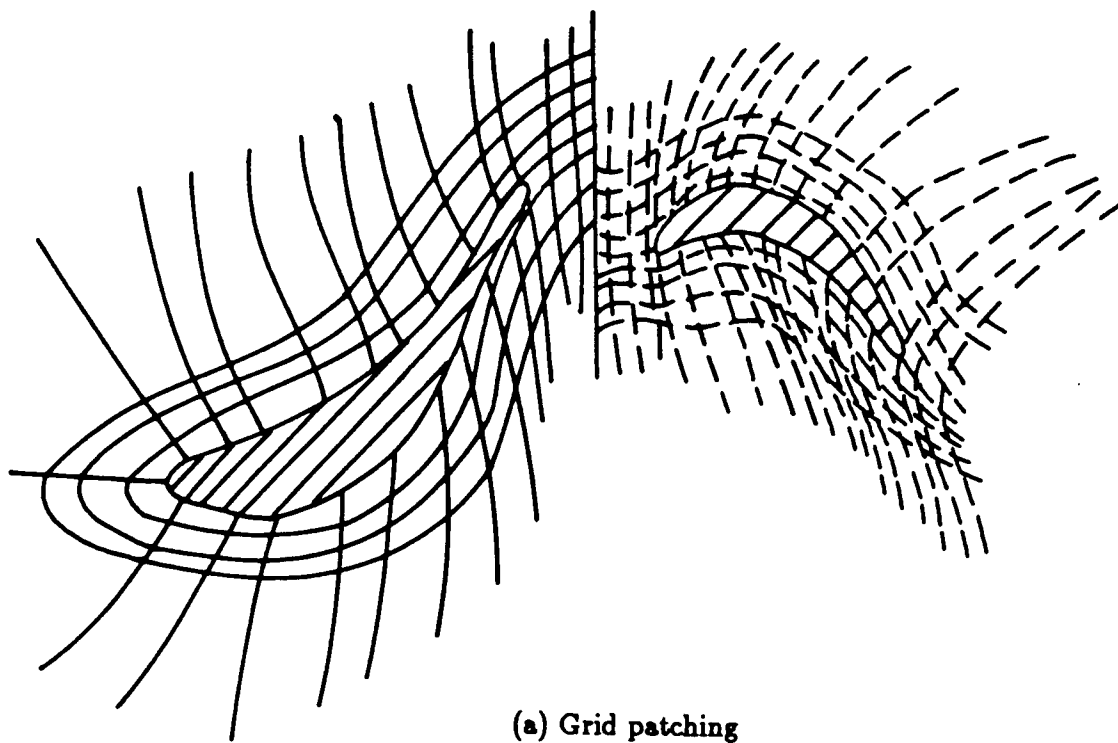
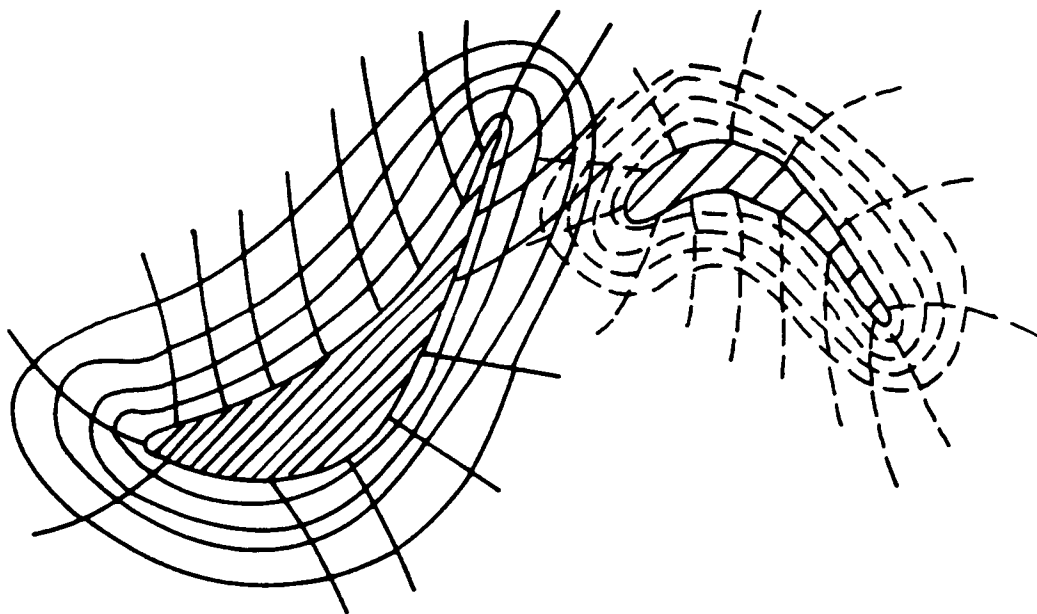


Fig. 2.10 Zoning of multiple-connected region.



(a) Grid patching



(b) Grid overlapping

Fig. 2.11 Grid patching versus grid overlapping.

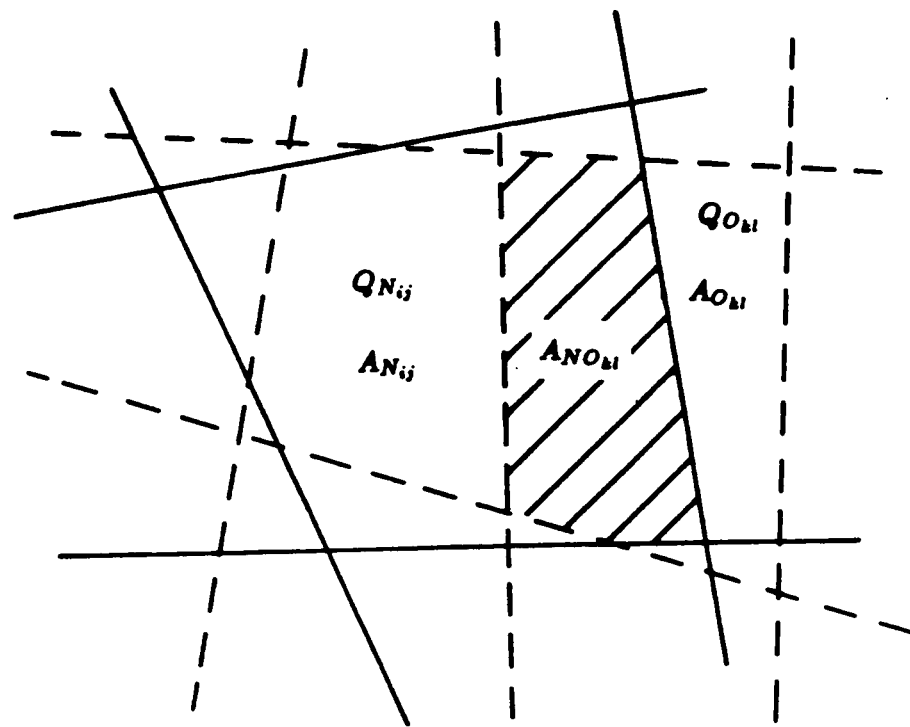


Fig. 2.12 Patched surfaces.

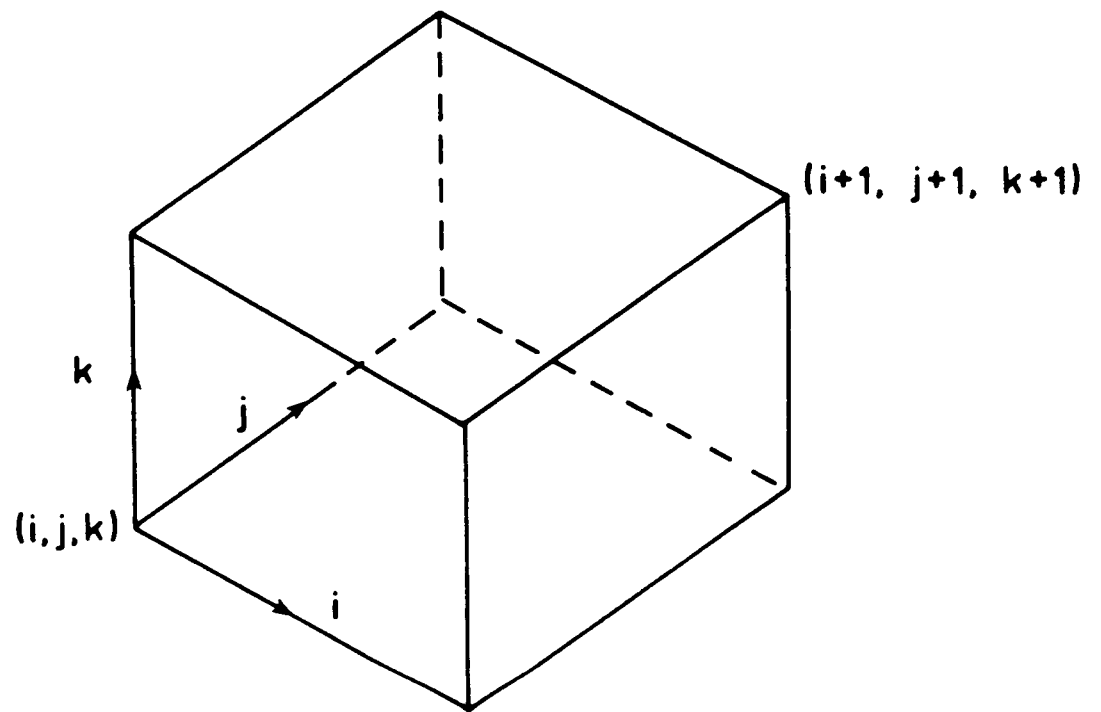


Fig. 3.1 A typical volume element.

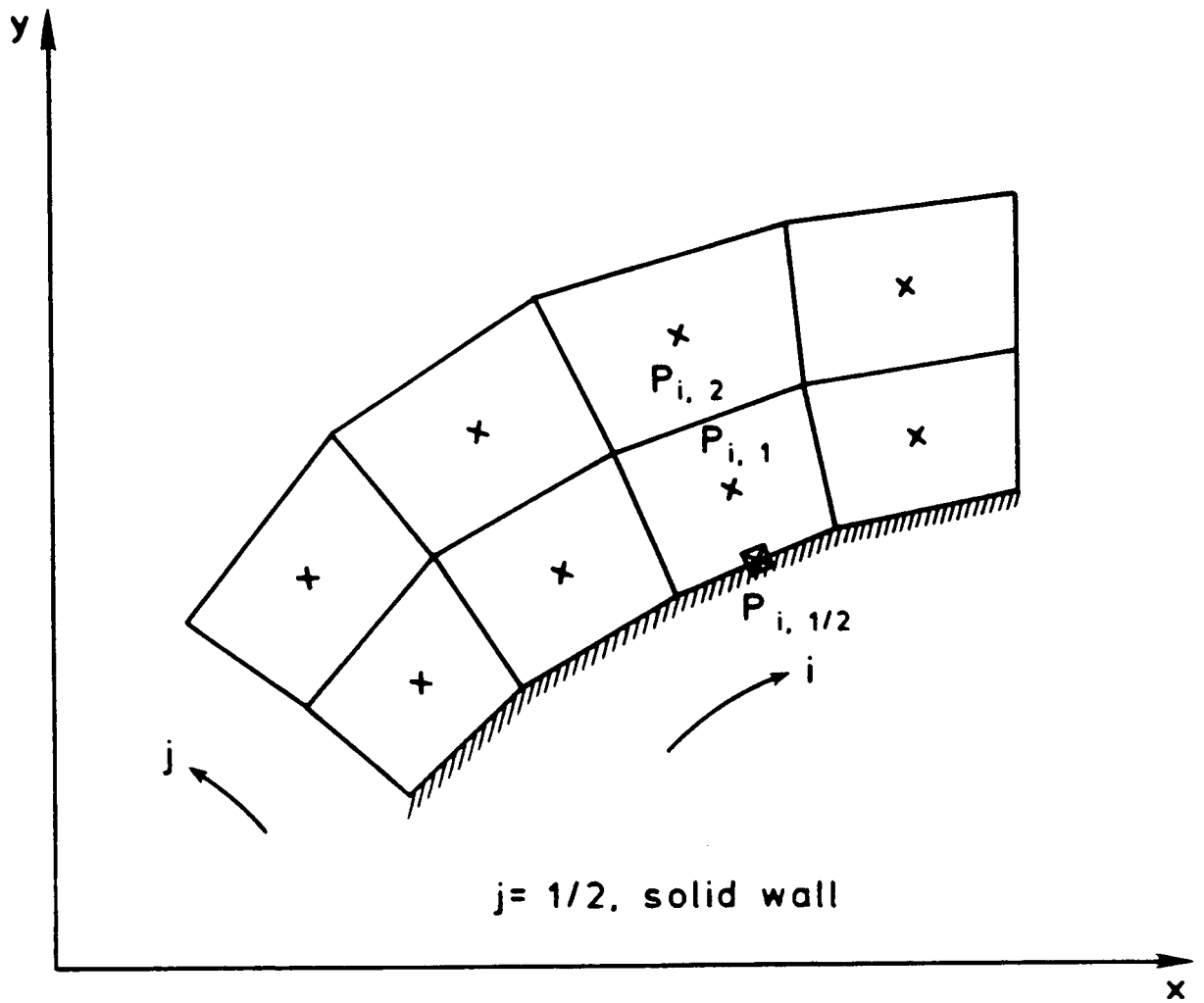


Fig. 3.2 A solid wall boundary.

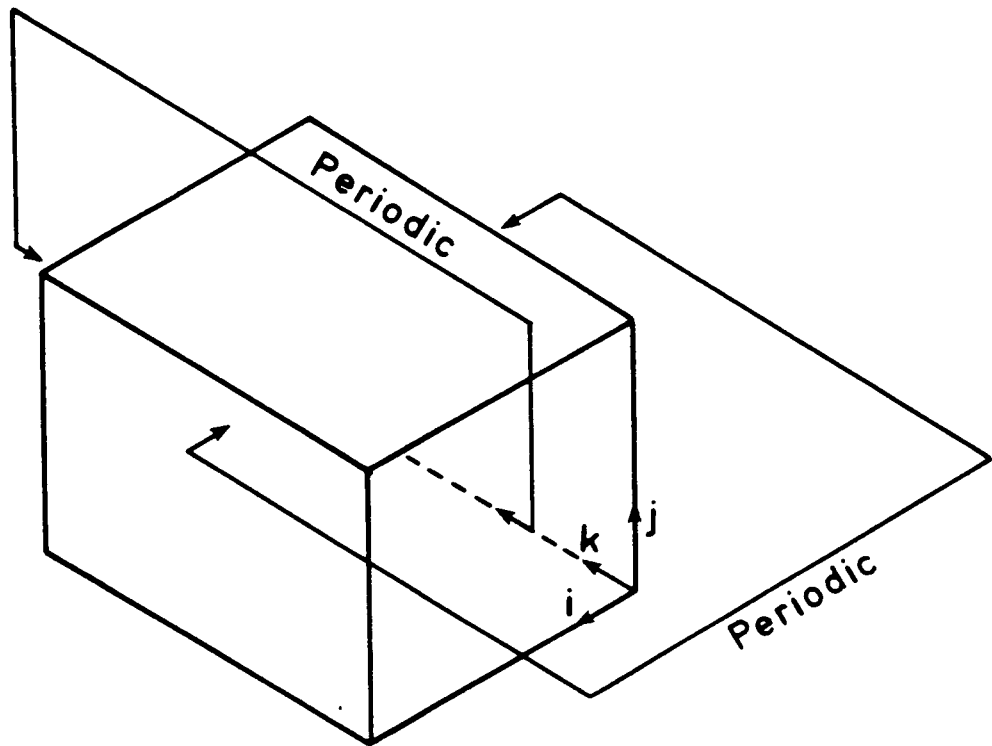


Fig. 3.3 Coordinate cut boundaries.

C-2

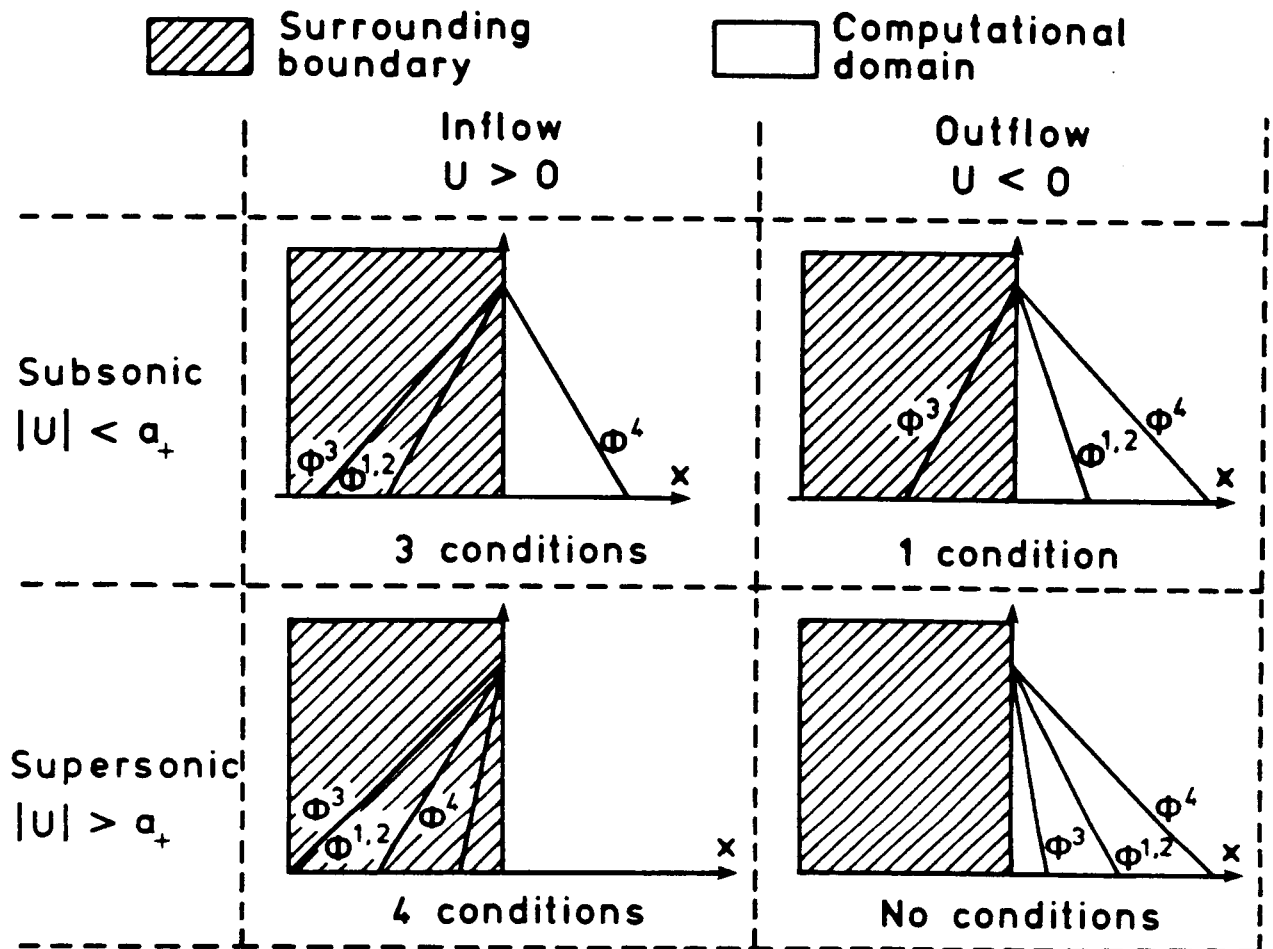


Fig. 3.4 Conditions at the inflow/outflow boundaries.

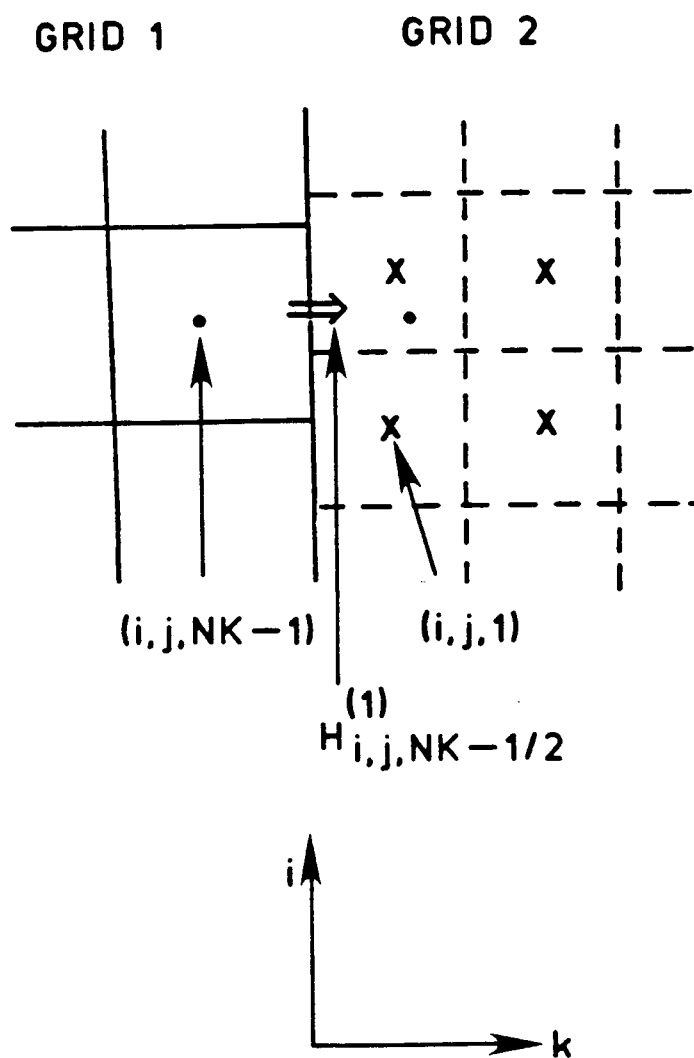


Fig. 3.5 A typical interface.

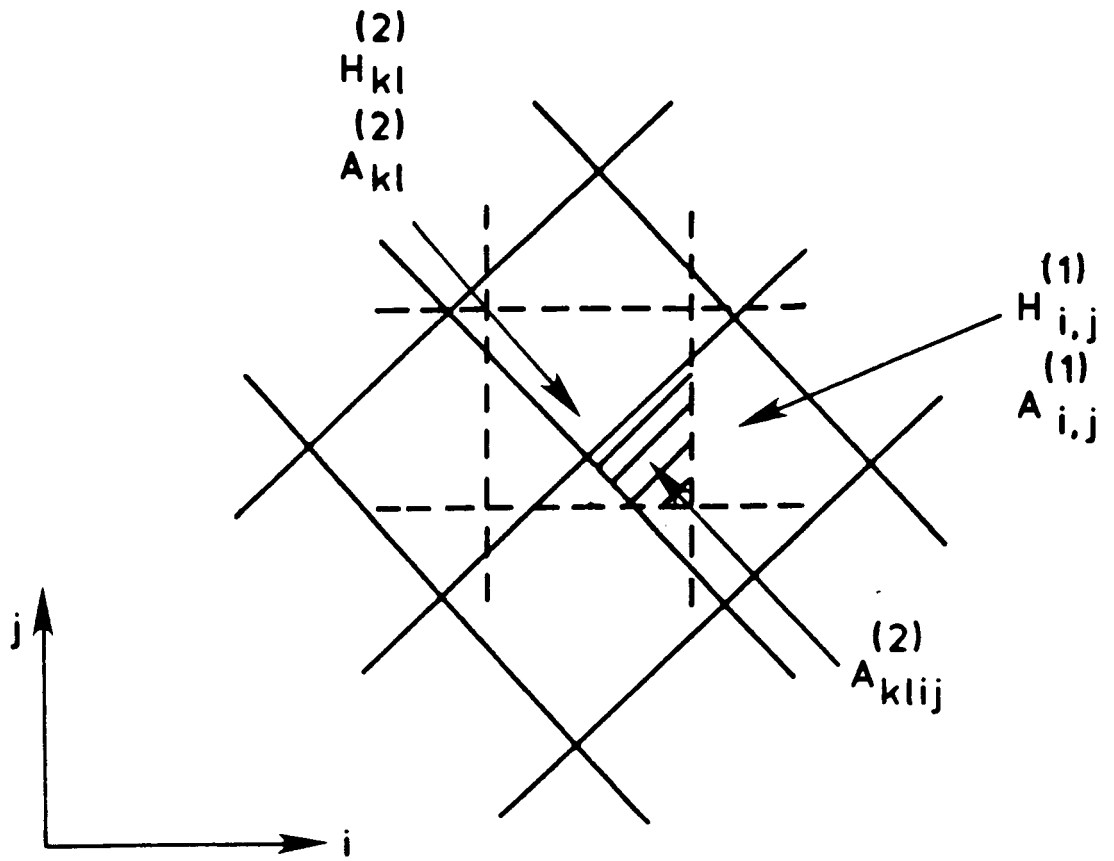
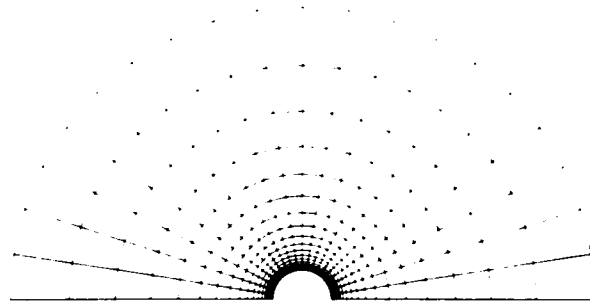
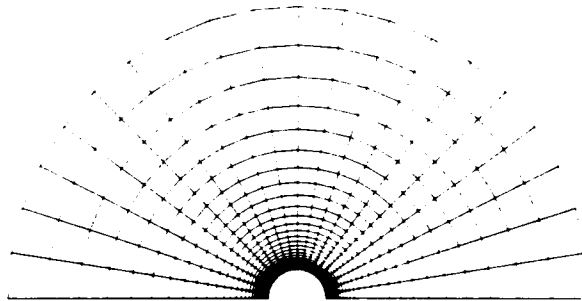


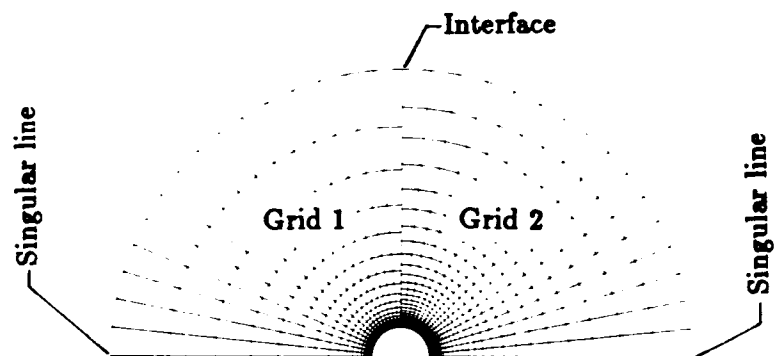
Fig. 3.6 A typical patched interface.



(a) Grid 1

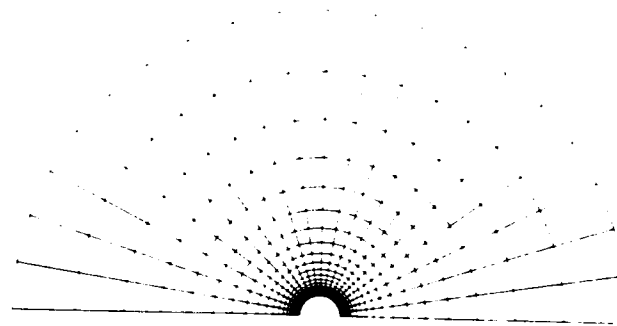


(b) Grid 2

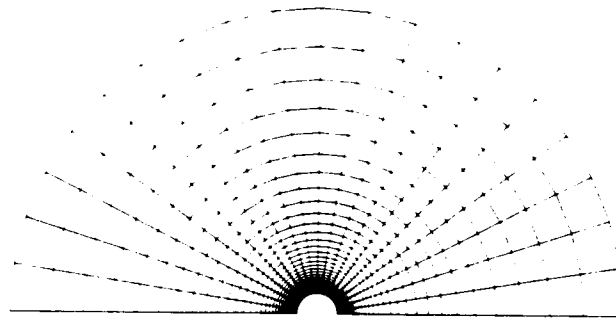


(c) Symmetry plane

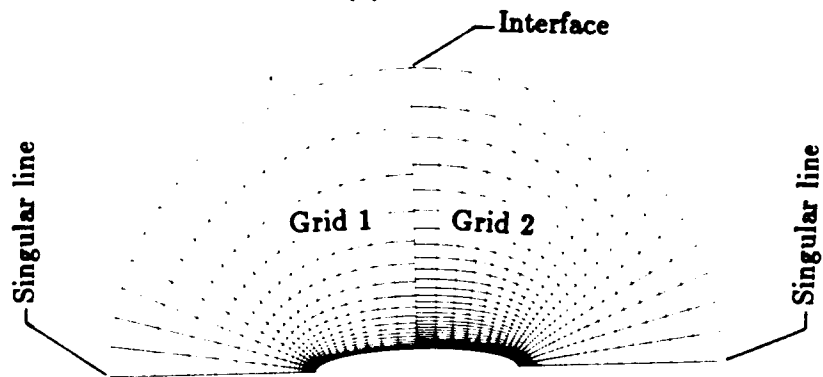
Fig. 4.1 Grid system for a sphere.



(a) Grid 1



(b) Grid 2



(c) Symmetry plane

Fig. 4.2 Grid system for a slender body.

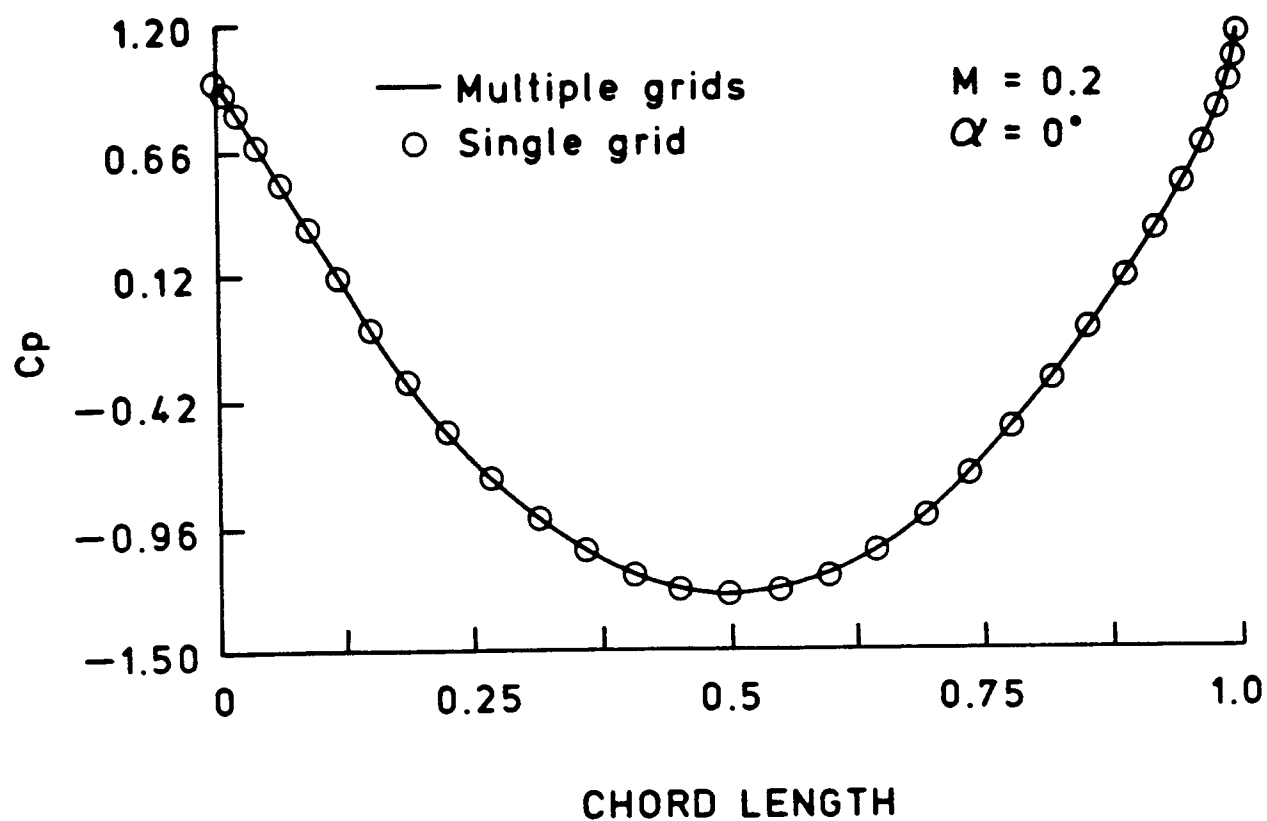


Fig. 4.3 The pressure coefficient on the surface of a sphere.

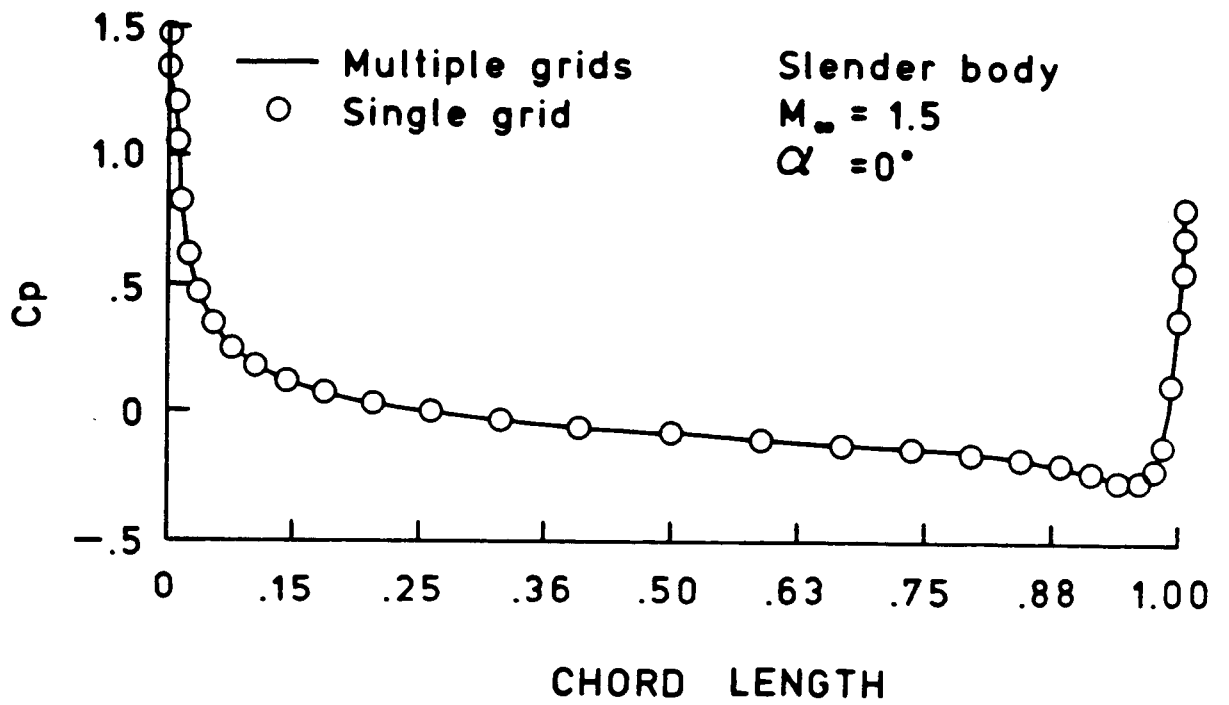


Fig. 4.4 The pressure coefficient on the surface of a slender body.

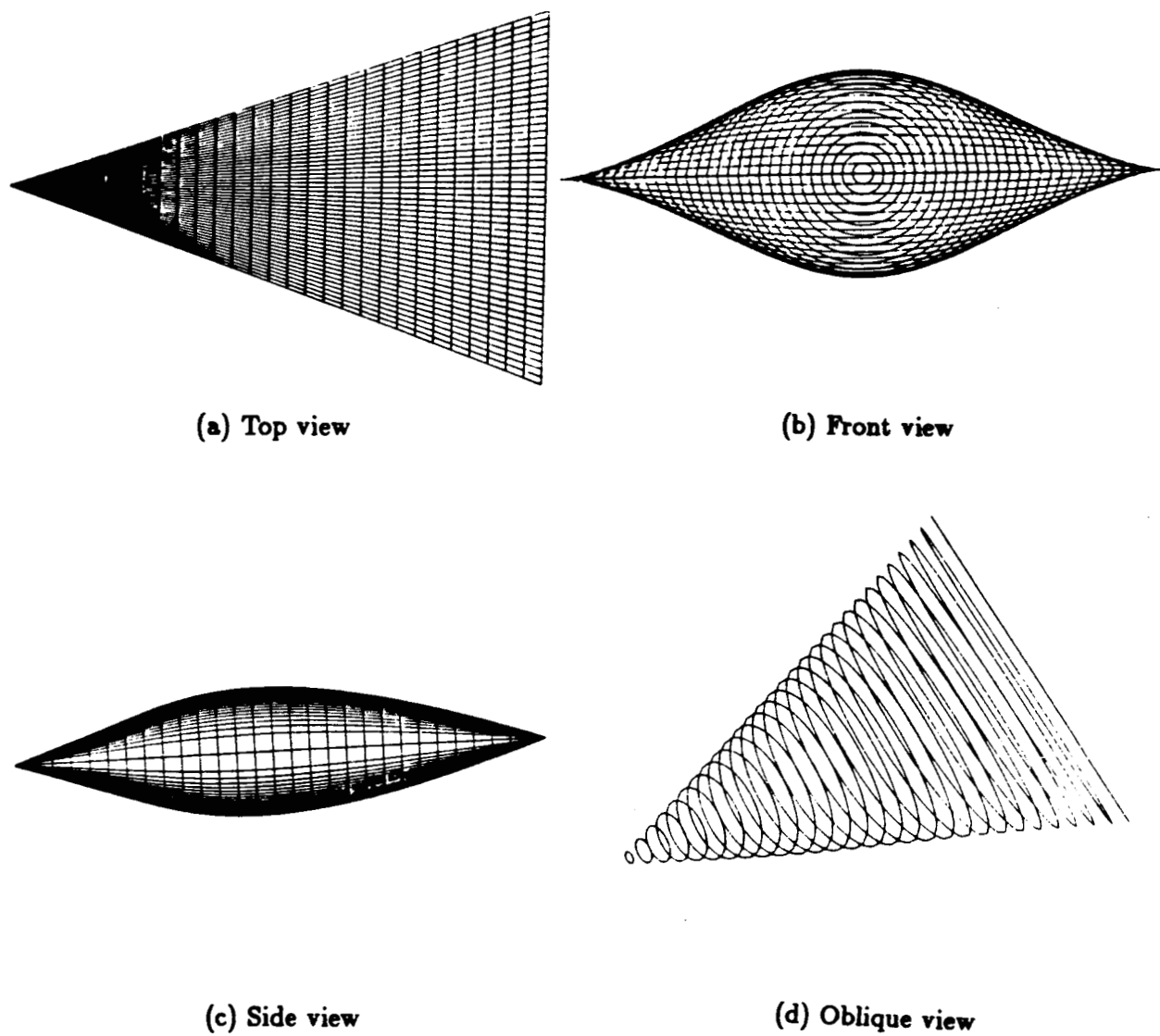
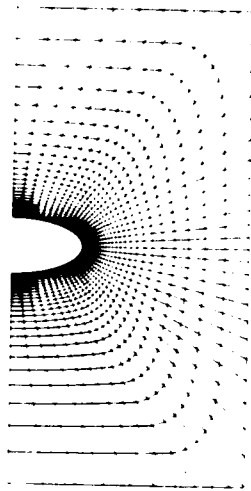
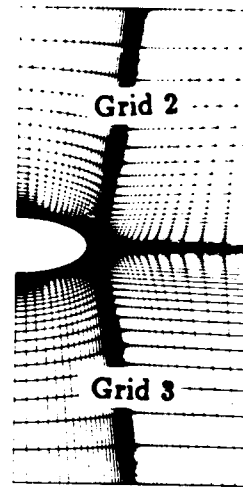


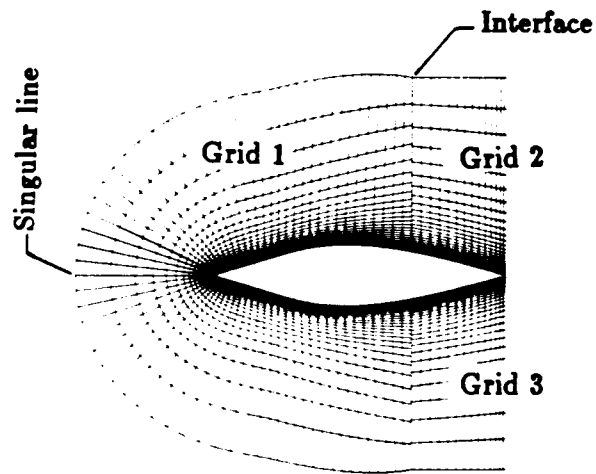
Fig. 4.5 Physical model of a Butler-Wing.



(a) O-Grid



(b) H-Grid



(c) Symmetry plane

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

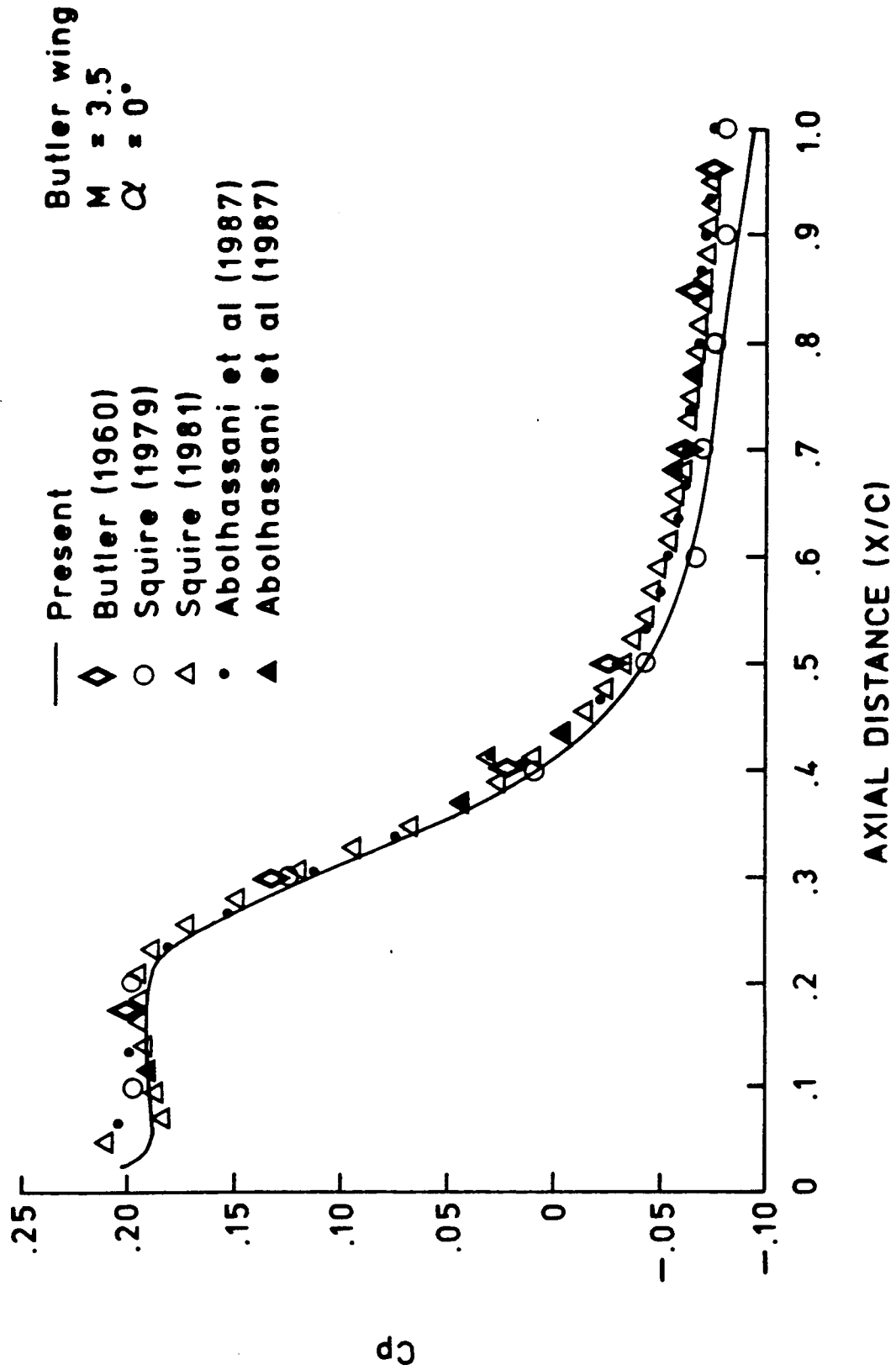


Fig. 4.7(a) The pressure coefficient on the surface of a Butler- Wing ( $M = 3.5, \alpha = 0^\circ$ ).

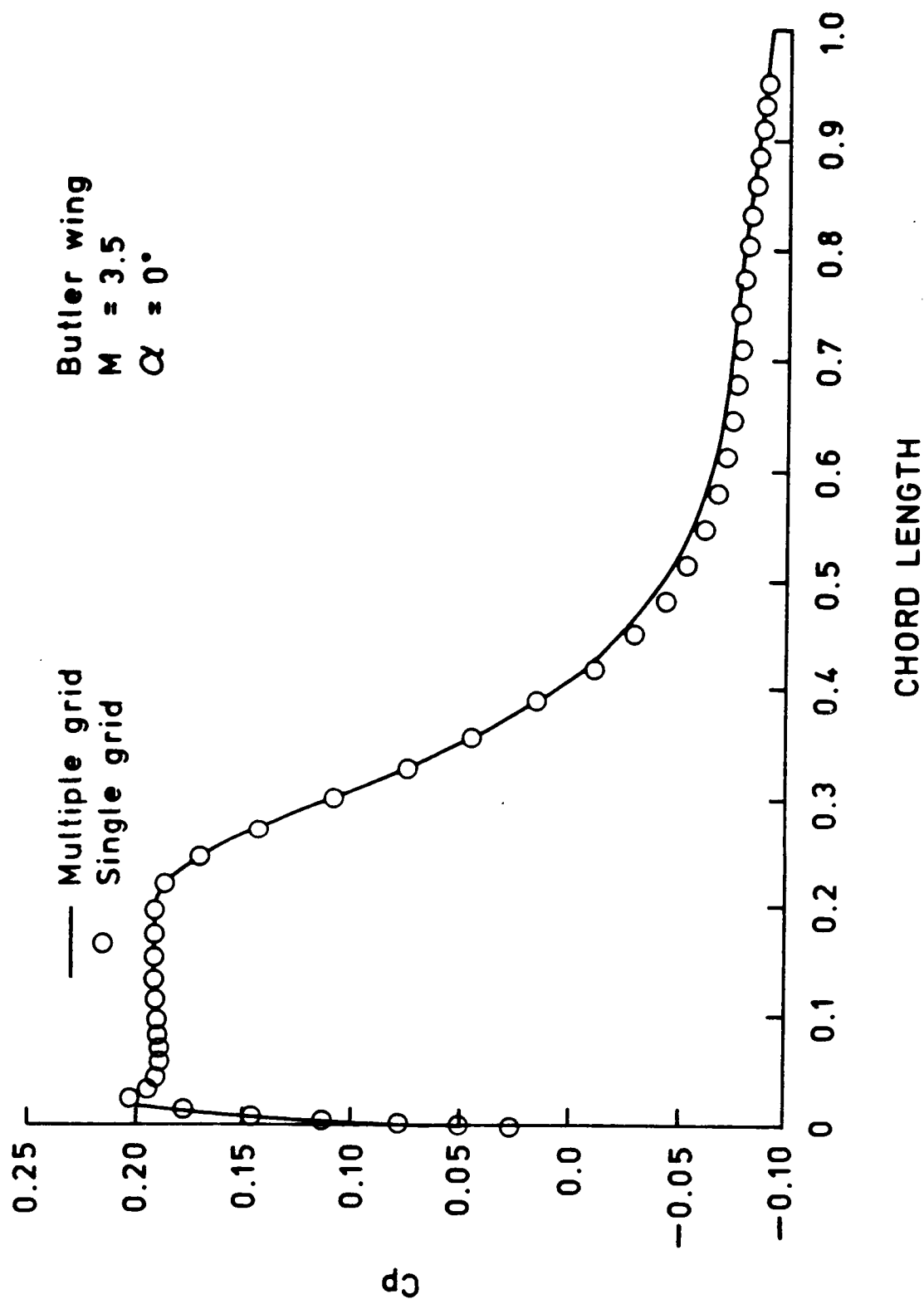
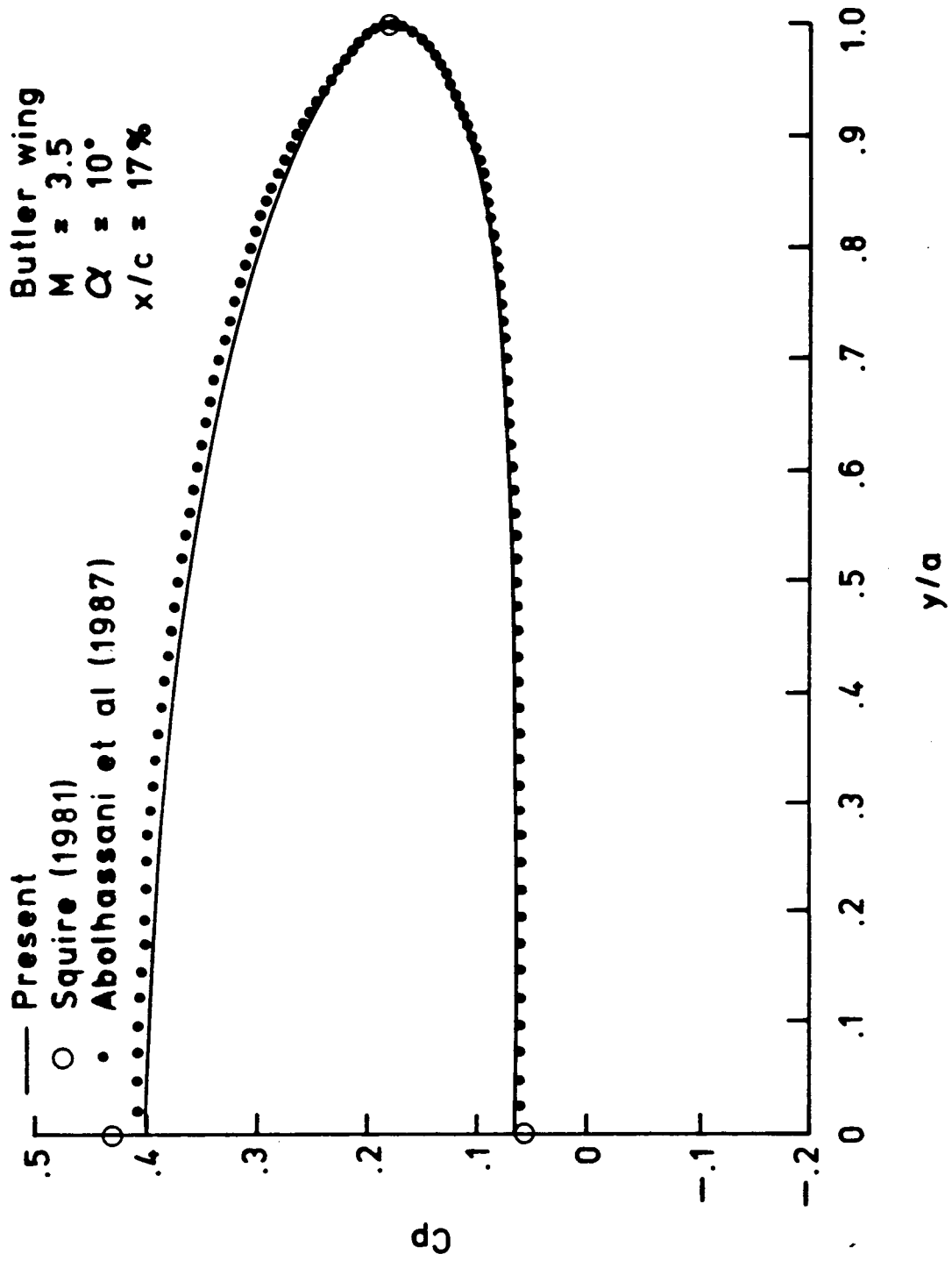
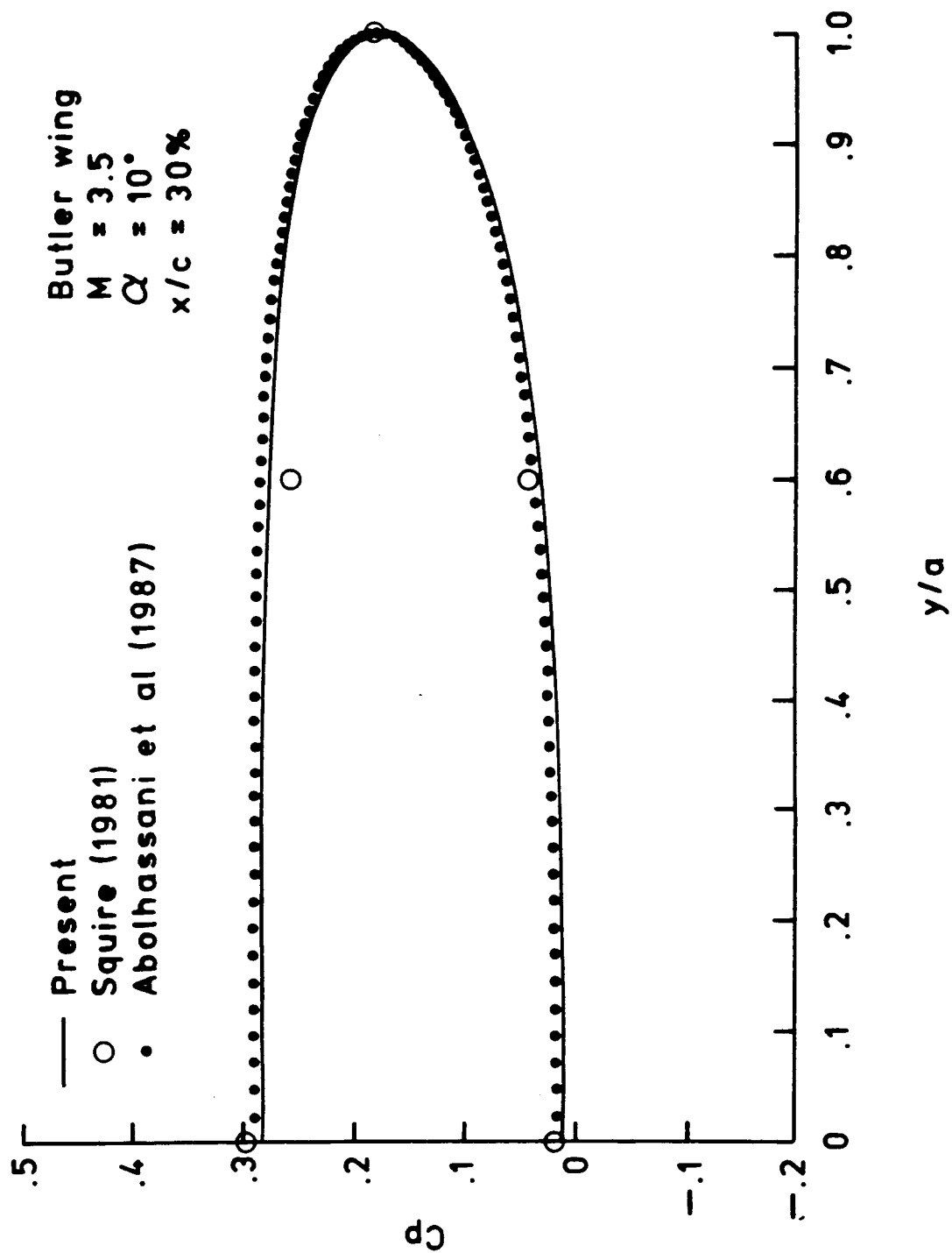


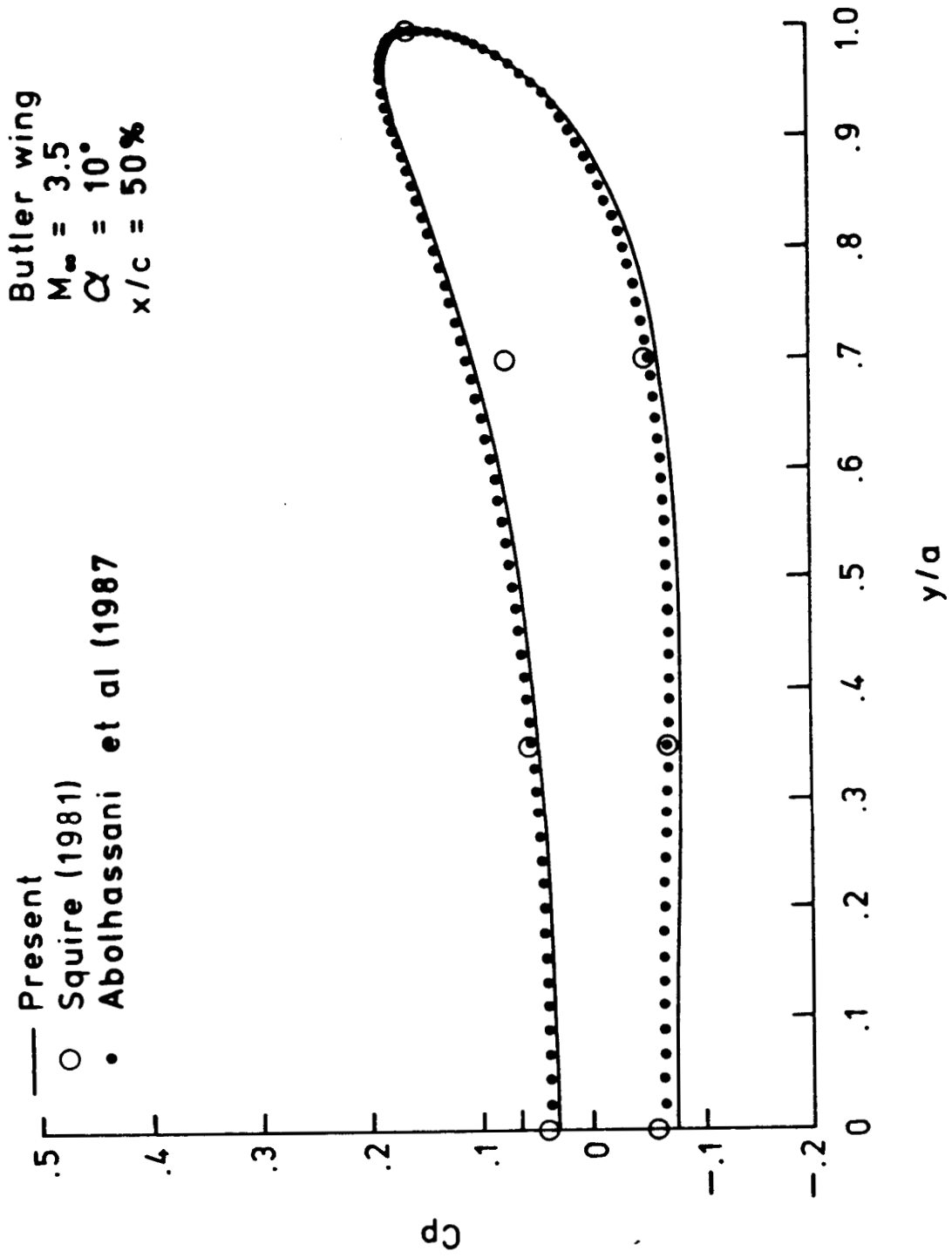
Fig. 4.7(b) Comparison of  $C_p$  with the single grid calculation.

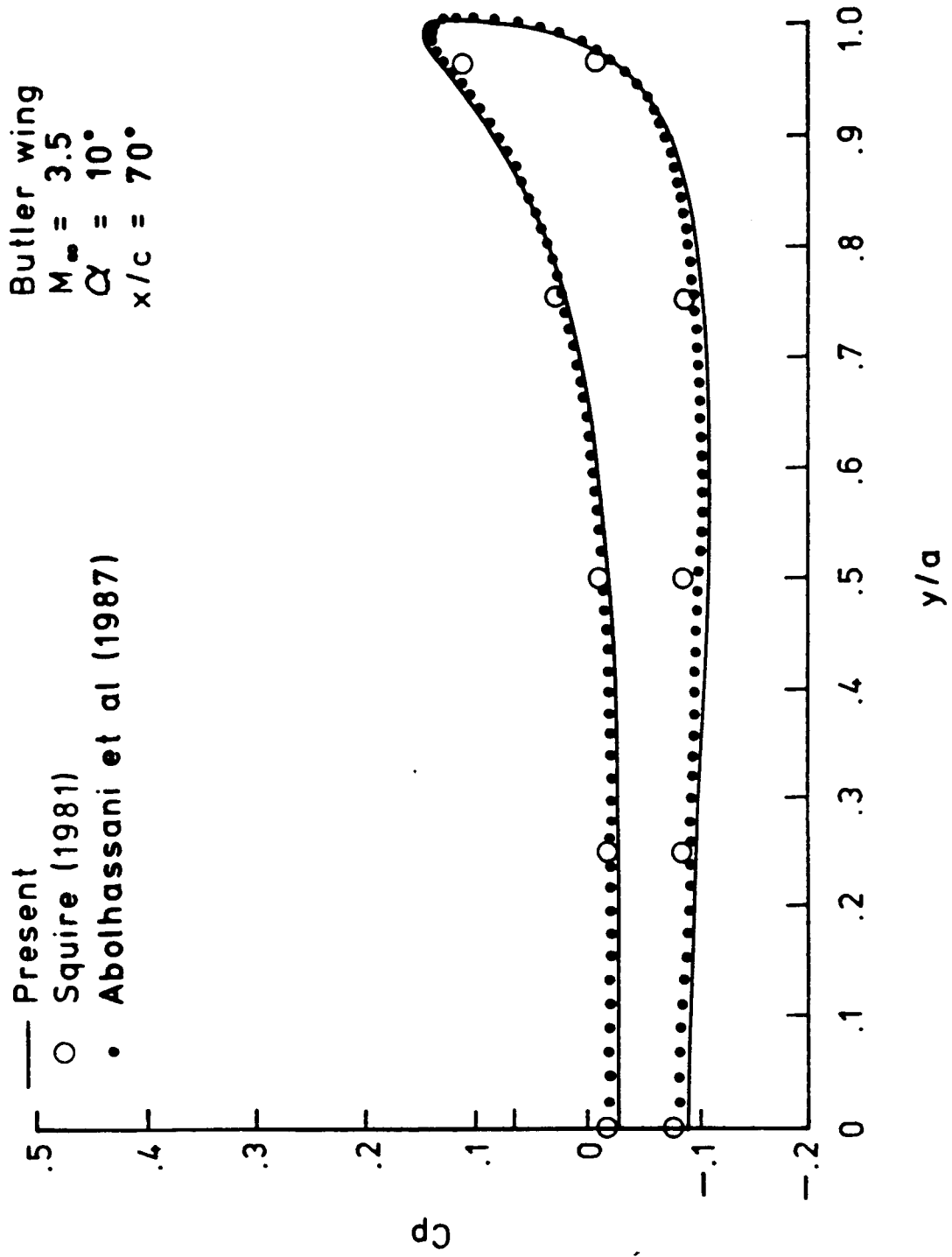


(a)  $x/c = 17\%$

Fig. 4.8(a)  $C_p$  on the surface of a Butler-Wing at different cross cut ( $M = 3.5, \alpha = 10^\circ$ ).

Fig. 4.8(b)  $x/c = 30\%$

Fig. 4.8 (c)  $x/c = 50\%$

Fig. 4.8 (d)  $x/c = 70\%$

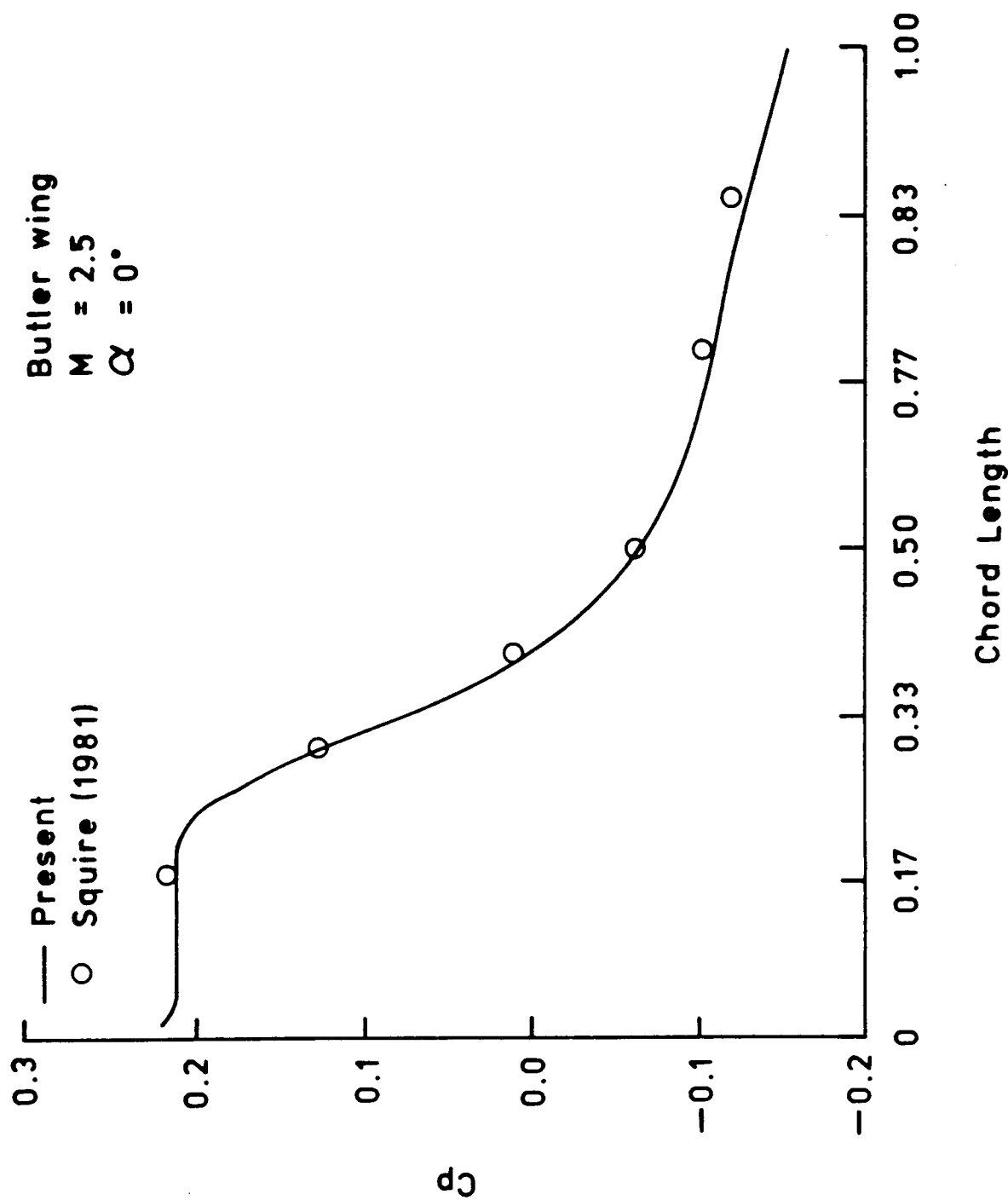


Fig. 4.9  $C_p$  on the surface of a Butler-wing ( $M = 2.5, \alpha = 0^\circ$ )

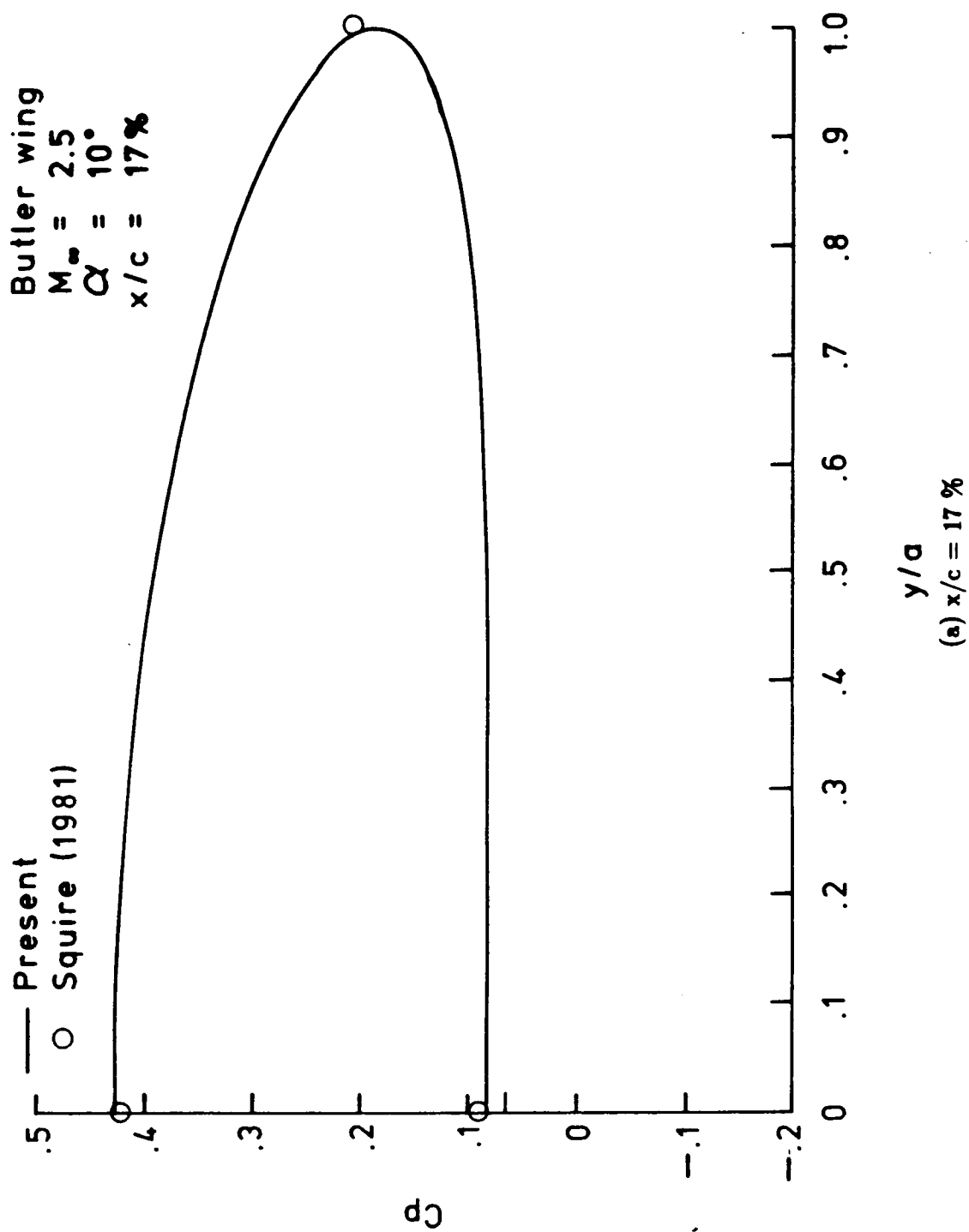
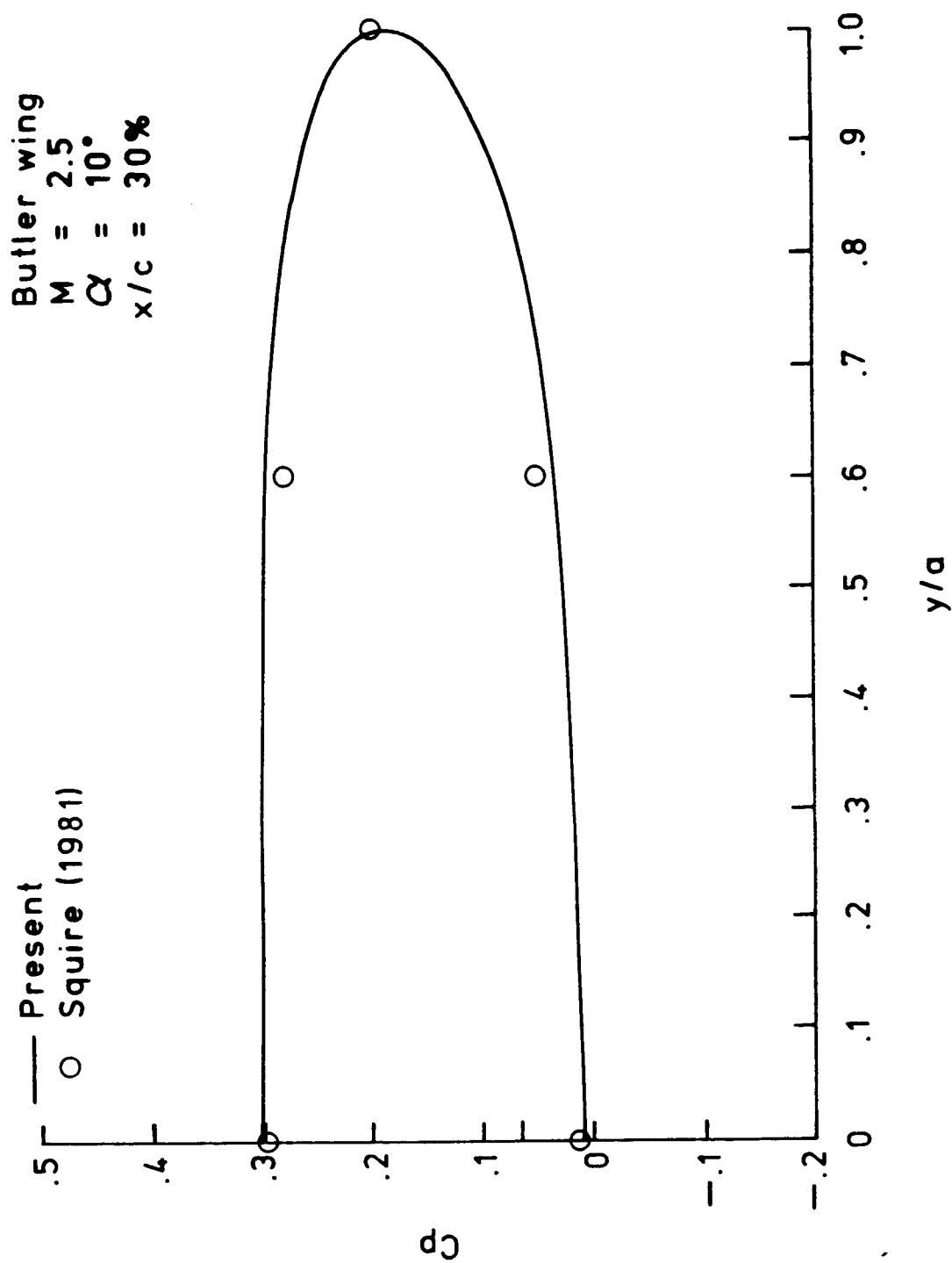
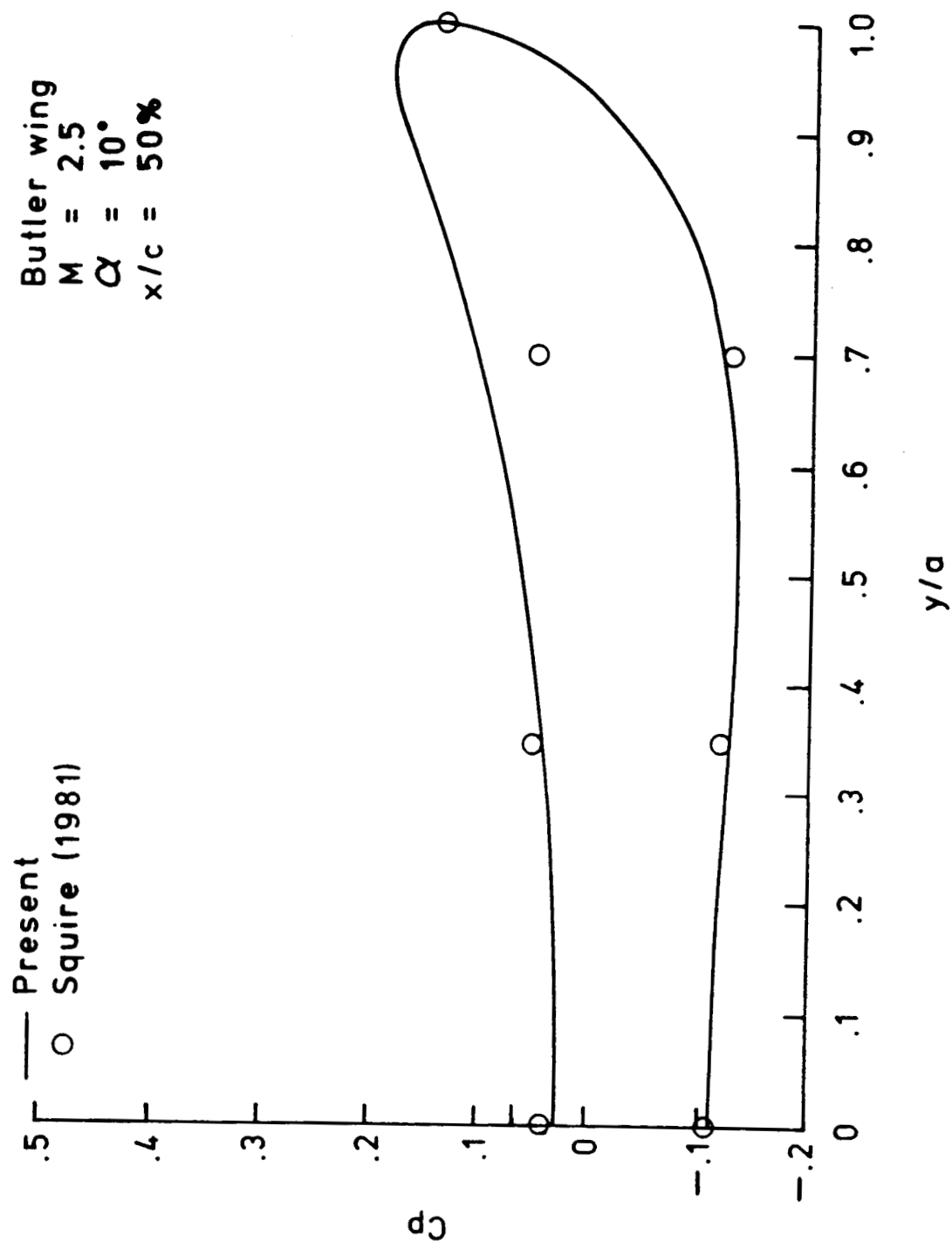


Fig. 4.10(a)  $C_p$  on the surface of a Butler-Wing at different cross cut  
 ( $M = 2.5, \alpha = 10^\circ$ )

Fig. 4.10(b)  $x/c = 30\%$

Fig. 4.10 (c)  $x/c = 50\%$

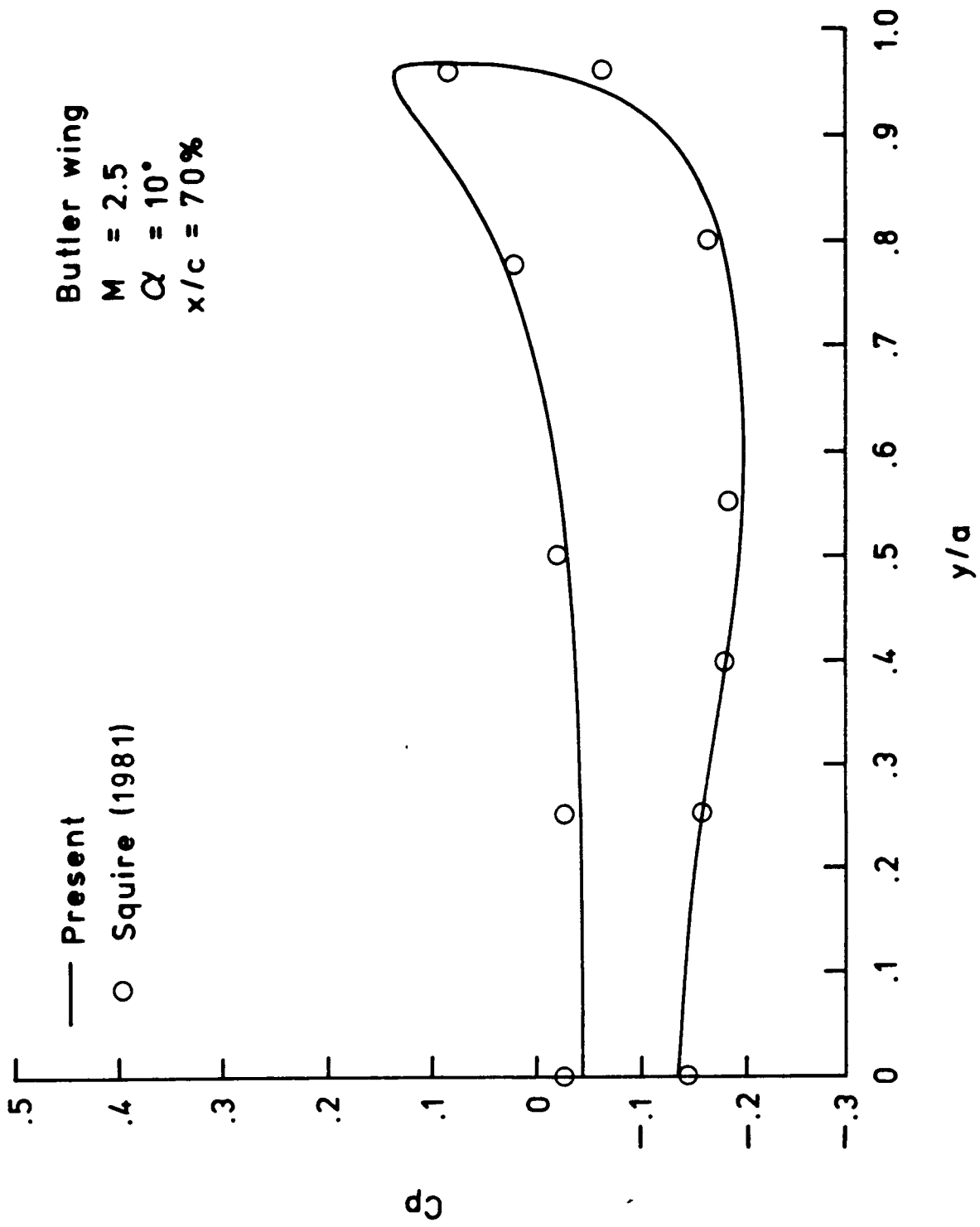
Fig. 4.10(d)  $x/c = 70\%$



Fig. 5.1 A fighter aircraft configuration.

ORIGINAL PAGE  
BLACK AND WHITE PHOTOGRAPH