Spring 1994

Unsteady Flow Simulations About Moving Boundary Configurations Using Dynamic Domain Decomposition Techniques

Guan-Wei Yen
Old Dominion University

Recommended Citation

Yen, Guan-Wei. "Unsteady Flow Simulations About Moving Boundary Configurations Using Dynamic Domain Decomposition Techniques" (1994). Doctor of Philosophy (PhD), dissertation, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/zfx4-8p98
https://digitalcommons.odu.edu/mae_etds/296

UNSTEADY FLOW SIMULATIONS ABOUT MOVING BOUNDARY CONFIGURATIONS USING DYNAMIC DOMAIN DECOMPOSITION TECHNIQUES

by

Guan-Wei Yen

B.E., June 1984, Aeronautical Engineering Department
National Cheng-Kung University, Taiwan, R.O.C.

M.S. August 1989, Mechanical Engineering Department
Old Dominion University

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

DOCTOR OF PHILOSOPHY
AEROSPACE ENGINEERING
OLD DOMINION UNIVERSITY
May 1994

Approved by

Oktay Baysal (Director)

Colin P. Britcher

David S. Miller

Osama A. Kandil

ABSTRACT

Guan-Wei Yen
Old Dominion University, 1994
Director: Dr. Oktay Baysal

A computational method is developed to solve the coupled governing equations of an unsteady flowfield and those of rigid-body dynamics in six degrees-of-freedom (6-DOF). This method is capable of simulating the unsteady flowfields around multiple component configurations with at least one of the components in relative motion with respect to the others. Two of the important phenomena that such analyses can help us to understand are the unsteady aerodynamic interference and the boundary-induced component of such a flowfield. By hybridizing two dynamic domain decomposition techniques, the grid generation task is simplified, the computer memory requirement is reduced, and the governing equations of the rigid-body dynamics are simplified with certain assumptions. Three dimensional, unsteady Navier-Stokes equations are solved on each of the subdomains by a fully-vectorized, finite-volume, upwind-biased, and approximately-factored method. These equations are solved on the composite meshes of hybrid subdomain grids that can move with respect to each other. Hence, the present method combines the advantages of an efficient, geometrically conservative, minimally and automatically dissipative algorithm with the advantages and flexibility of the domain decomposition techniques. Several measures that reduce the numerical error are studied and compared with the exact solution of a moving normal shock in a tube. This solution compares very well with the analytic solution of the isentropic equations. It is concluded, that as a miminum measure, the connectivity of nonconservative overlapped scheme needs to be second-order accurate for spatial and temporal discretizations, as well as for the moving subdomain interpolations. Furthermore, the CFL numbers should be restricted to below unity, if affordable, for flows with high flow gradients. The method is further scrutinized by simulating the flow past a sinusoidally pitching airfoil, and the flow past a sinusoidally pitching and plunging airfoil. The results of the former case are successfully compared with the experimental data. The final two-dimensional case is the separation of a store from an airfoil along a prescribed path. As the first three dimensional case, the flowfield past an oscillating cylinder near a vertical wall is simulated. Prior to coupling it with the flowfield equations, the 6-DOF trajectory method is validated by successfully comparing the path it predicts with the one used in a captive trajectory testing. Finally, a rigid-body dynamics method is used to predict the aerodynamically determined trajectory of a store dropped from its initial position under a wing. The results of the present investigation contribute to the understanding of the unsteady aerodynamic interference and the boundary-induced component of such a flowfield. However, its main contribution is the newly proposed computational method for flows involving relative boundary motions.

TABLE OF CONTENTS

APPENDICES

ACKNOWLEDGMENTS

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

Special thanks are also extended to my committee members Dr. Osama A. Kandil, Dr. Colin P. Britcher, and Mr. David S. Miller for their valuable suggestions in the dissertation preparation. I also wish to thank my previous group partners V. R. Lessard and K. Fouladi, and current group partners K. P. Singh and J. C. Newman III.

Further, I would like to express the highest love, gratitude and appreciation to my parents, wife, and other members of my family for their support and encouragement.

This work is supported by NASA Langley Research Center under the Grant NAG-1-1150.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A_i : i^{th} Flux Jacobian matrix

AOA : angle-of-attack (degree)

$\bar{a}$: acceleration vector

a_∞ : speed of sound

C_A : axial force coefficient

C_N : normal force coefficient

C_S : side force coefficient

C_{mr} : rolling moment coefficient

C_{mp} : pitching moment coefficient

C_{my} : yawing moment coefficient

C_p : pressure coefficient

C_L : lift coefficient

CFL : Courant-Friedrichs-Lewy condition

DDT : domain decomposition techniques

D^3T : dynamic domain decomposition techniques

DOF : degree-of-freedom

D_+, D_- : differential operators

d : reference length

e_t : total energy

E_i : i^{th} inviscid flux vectors

E_{vi} : i^{th} viscous flux vectors

F : force

G_i : i^{th} subdomain grids

GCL : Geometry conservation law

I : unit diagonal matrix

$\bar{\bar{I}}$: inertia matrix

J : Jacobian matrix of transformation

k : reduced frequency

M : moment

Mach number

m : mass

$\vec{n}$: normal vectors

p : nondimensional pressure

p_{incr} : increment of nondimensional pressure

p_{min} : minimum nondimensional pressure

p_{max} : maximum nondimensional pressure

p_{∞} : nondimensional pressure for freestream

$\vec{Q}$: vector of conserved variables

$\vec{q}$: vector of primitive variables

$\vec{r}$, $\vec{R}$, $\vec{x}$: position vectors

s : cell surface area

t : time - in Cartesian coordinates

T : temperature

$\vec{u}$: fluid velocity vector

$\vec{U}$: contravariant velocity vector

V : cell volume

$\vec{v}$: rigid-body velocity vector

x_i : i^{th} Cartesian coordinate

x, y, z : non-inertia coordinate systems

X, Y, Z: inertia coordinate system

GREEK SYMBOL

α : angle of attack in degree

α, β, γ: Euler angles (degree)

τ : time - in generalized curvilinear coordinate systems

γ : specific heat ratio

ρ : nondimensional density

ξ^i : i^{th} generalized curvilinear coordinates

$\bar{\omega}$: angular velocity vector

$\uparrow\downarrow$: up and down motions

Chapter 1

INTRODUCTION

Computational fluid dynamics (CFD) plays an increasingly important role in aerospace applications, because the advances in computer hardware and software have made it possible to solve complex flow problems using CFD. The objective of CFD is to solve the system of differential equations that govern the flow field with discrete and/or integral methods. Hence, efficient solvers which are capable of solving the governing equations of fluid motion by integral, finite-difference, finite-volume and finite-element techniques, have evolved.

An essential task of CFD algorithms is the construction of an adequate grid (mesh) on which the governing equations can be solved in a finite form. A major difficulty in CFD for a complex three-dimensional configuration is how to generate an appropriate grid. The term complex configuration can be defined as any physical domain in which there are single or multiple bodies of nonsmooth, joint or disjoint geometries. A few examples of complex flow domains are the flow around a whole aircraft, a wing and a fuselage, a wing and a nacelle, a store and a cavity, a complete stage of a turbomachine, an external-internal nozzle, or a store inside/outside a cavity. Due to the complexities of these configurations, constructing a single body fitted grid is either a difficult task or may result in a skewed mesh, which in turn results in an erroneous solution. Body-fitted and boundary confirming curvilinear grids are desirable, because of the advantage of implementing the surface boundary conditions accurately. Also, a proper surface oriented coordinate system enables coordinate-related approximations to the equations of motions for arbitrary complex geometries. Hence, it becomes increasingly difficult to locally control the orthogonality, volume variations, cell aspect ratios, and other grid measures, which affect the accuracy of the solution as the geometric complexity increases. Other difficulties may arise when clustering the grids for the regions where the flow variables and their gradients change rapidly, or when one of the subcomponent has relative motion with respect to other subcomponents. Several different approaches have been used by researchers to reduce this complexity and ease the grid generation efforts: for example, dynamic domain decomposition, unstructured grids, Lagrangian coordinate description, arbitrary Lagrangian-Eulerian description, and grid adaptation.

1.1 Literature Survey

The following section lists some of the pioneering studies on overlapped and degenerate zonal methods and their applications to some flow problems. Also, the computational investigations on unsteady flows and moving boundaries are reviewed in section 1.1.2. Different frame of reference descriptions, that is, Lagrangian, Eulerian, and Arbitrary Lagrangian-Eulerian frames, will also be reviewed in section 1.1.2.

1.1.1 Domain Decomposition Techniques

There are three basic types of domain decomposition methods: multiblock, zonal (or patched), and overlapped methods. A fourth method, the degenerate zonal method, can be derived to have partially the features of the multiblock and partially those of zonal methods. However, the current research is primarily focused on the overlapping method with some additional effort to incorporate the degenerate zonal method.

The overlapping method entails dividing the flow domain into regions that overlap or share common physical and computational space [1]. Within the overlap region, the grids communicate through data transferance by an interpolation procedure. This method allows the subdomains to be non-disjoint so that one mesh may be embedded completely or partially within another. This procedure permits each subdomain to be meshed independently with no requirements of continuous grid lines across boundaries. Because each subdomain grid is independent of another, the grid generation task is greatly reduced for complicated flow regions. Each subdomain mesh can be created using different grid generation techniques suitable for that particular domain. This is especially beneficial for subdomains that require high grid densities.

There are several drawbacks of using the overlapped scheme [1]. However, with careful treatment, most of the problems can be partially or completely alleviated. The disadvantages are the following: (i) the technique requires an overlapped region between subdomains that may not be always possible, as in the case of extremely small gap between two solid surface, (ii) the accuracy of boundary data transferance depends on the interpolation procedure, whether it is conservative or nonconservative, and (iii) the accuracy and convergence speed of the solution indirectly depend on the degree of overlapping of the grids relative to the size of the subdomains.

The degenerate zonal method is a relatively new technique, which is derived from the convergence accelerating multigrid method. It has the simplicity and suitability in refining the grid wherever necessary. It is required that only every *n-th* line normal to the grid interface is contiguous. If $n=2$, then the line that is not contiguous bisects the distance between the neighboring two contiguous lines. This concept is a natural extension from the generation of coarse-fine grids necessary in the multigrid convergence acceleration methods. The intergrid communication is very simple for this method. Details of these methods will be discussed in Chapter 3.3.

One of the earlier applications of grid overlapping methods was presented by Atta [2]. He developed a method for constructing a two-dimensional grid system for a finite difference method to solve the full potential equations and obtained a solution for transonic flow about a configuration with multiple components. The computational model was a two component configuration that consisted of an airfoil embedded in rectangular boundaries. The results showed that the accuracy and convergence speed of an implicit approximate factorization scheme depended on the extent of the overlap region and the size of each subdomain. This technique was later extended to three-dimensional flows by Atta and Vadyak [3]. Also, the method was applied to a wing-pylon-nacelle configuration. The transfer of information between different subdomains within the overlap regions was done by a trivariate interpolation polynomial based on a linear Taylor series expansion.

Steger et al. [4] developed a grid overlapping technique called the "CHIMERA" scheme. This early study was restricted to the finite difference solution of two-dimensional, linearized flows. Later, this method was extended for three-dimensional Euler and Navier-Stokes [5-6] solution of complex flows. This scheme involves the automatic connection of multiple, overlaid grids, and the use of different solution procedures for different subdomain grids. In this scheme, a global grid covers the entire flow domain, and one or more minor grid is then overlaid on the global grid to solve the secondary features of the configuration, such as flaps, nacelles, or stores. The minor grids can be fully or partially overlapped without requiring the grid that is subsequently excluded from the flow solution in the global grid. The information transfers between the different subdomains are communicated through trilinear interpolation within the overlap regions. The overlapping scheme has been proven to be very flexible and successful on various complex configurations. However, it was shown that some errors may arise when the high flow gradient passes through the grid boundaries [7].

Suhs [8] used the CHIMERA scheme to attack the three-dimensional cavity flow at subsonic and supersonic speed. Despite the inappropriateness of the thin layer approximation of Navier-Stokes equations for the unsteady cavity flow, this study further exemplifies the versatility of the CHIMERA scheme for complex flow problems.

Chesshire and Henshaw [9-11] have developed a technique for the generation of curvilinear composite overlapped grids and the numerical solution of partial differential equations on them. Continuity conditions through interpolations are imposed at the overlapped boundaries. This code, CMPGRD, can generate composite two and three dimensional grids with any number of component grids. These composite grids can have second or higher order of interpolation accuracy; therefore, the overall accuracy can be made about the same order as the discretization error. However, the higher order interpolations require a larger overlapped region between subdomains and considerably more calculations. The CMPGRD program is capable of automatically generating the sequence of coarser grids for a multigrid algorithm flow solver.

Baysal et al. [12-14] incorporated the multigrid scheme with the CHIMERA method, and accommodated these modifications to a finite-volume flow solver. A steady-state cylinder was placed next to a flat plate, with the overlapped and upwind, finite-volume flow solver and the interference flowfield has been studied. The details of the modification of CHIMERA and application cases are given in [1]. Then, the overlapped scheme has been incorporated with other domain decomposition techniques, namely, the multiblock and zonal methods, to optimize the strength of the *hybrid* domain decomposition methods [15-21]. The computational simulations of a missile with fins and a sting, either inside or outside a cavity, have been performed. The comparison of computational and experimental results showed good agreement. The details of these application cases and modified algorithms are given in [22].

Krist, et al. [23] extended the multigrid method to refine the grids in certain higher flow gradient regions. This technique is called the "degenerate zonal method" and it was demonstrated for a delta wing flowfield. The shock that passed through the degenerate zonal region had much better resolution than the portion outside of that region. However, this method was only performed for static cases, because the residual number interpolation between different grid levels were restricted to steady state solutions.

The overlapping method, which gives the most freedom for grid generation, uses the nonconservative intergrid communication. Therefore, the degree of conservation for the overlapped grids will be one of the topics to be investigated. A preliminary study was done by Berger [24] on a general procedure, for deriving conservative interface conditions that give weak solutions to the differential equation, if they converge on one and two dimensional overlapped grids. The conservative interface condition can be derived using a finite volume approach and balancing the spatial flux at the interface. An alternative to conceiving the spatial flux across overlapped boundaries is to conserve the time flux, Q, of the cell center at the boundaries. The time-flux conservation approach has been found to maintain the conservative properties at the boundaries within truncation errors [25]. The conservation of time flux is accomplished by interpolation to the cell centers of one grid assuming a weighted variation of time flux with the cells of the other grids. However, to find the cell volume weighted variations for three dimensional grids is a geometrically complicated procedure.

1.2.2 Unsteady Problems with Moving Boundaries

Several approaches have successfully been applied for the moving boundary problems. These can be classified as based on frame of reference (such as Eulerian and Lagrangian frame of reference) and type of grid (such as structured and unstructured grid).

The unstructured grid method [26] has successfully been employed in certain static and unsteady cases. The unstructured grid approach discretizes the flowfield by triangular or tetrahedral elements with nodes placed at the vertices. Discretizing the domain by such elements gives flexibility in grid generation about complex geometries. Also, the unstructured grid method is primarily used with finite-element or finite volume methods and has made remarkable progress in the last decade. This method is one of the most promising approaches that can resolve the problem of unsteady flowfield with complex geometries. However, there are two disadvantages for the unstructured grids. First, this method requires an extra amount of computer memory for the cell connectivity information. Secondly, it is a difficult task to generate unstructured grids in the proximity of a solid surface, where for example, clustering is used for viscous solutions. Also, for CFD applications, the finite-difference and finite-volume methods for structured grids are more mature and more efficient when compared to the finite-element method. Hence, at least for the present time, the unstructured approach may become less desirable for the CFD practitioners.

Lagrangian methods [27] have also been successfully demonstrated in certain cases. These methods are characterized by a coordinate system that moves with the fluid. Hence, each computational cell always contains the same fluid elements. These methods have special value for moving boundary problems for three reasons: (1) they permit material interfaces to be specially delineated and precisely followed; (2) they allow interface boundary conditions to be easily applied; (3) the detailed structures embedded in the flow can be zoned more finely than adjacent coarser structures [28]. These methods are not widely used, perhaps, because the governing equations always form an initial value problem. The two main problems with the Lagrangian methods are mesh tangling and numerical inaccuracy due to highly irregular meshes. The first problem arises because a mesh of fixed topology quickly becomes singular in flows undergoing large distortions. With the help of rezoning and reconnection methods, both methods of reconstructing the mesh involve reapportioning the conservative variables among the affected computational cells, and this represents a reappearance of the undesired convective flux characteristic of Eulerian methods.

Eulerian methods [12] are characterized by a coordinate system that is either stationary in the fixed reference frame or is moving in a certain prescribed manner to accommodate the continually changing shape of the solution domain. The fluid travels between different computational cells even when the grid moves, because grid movements are not related to the motion of the fluid. The main advantage of this approach in contrast to the Lagrangian methods, is that the fluid can undergo arbitrarily large distortions without loss of accuracy. Also, these methods are much easier to formulate. Consequently, most of the current CFD researchers are using the Eulerian methods to describe the flow motion. However, the Eulerian approach has several negative numerical effects on the solution accuracy [29]; (1) fluid particles are free to cross the control volume (or grid line), therefore, convecting with them numerical mixing and diffusion across the cell interface; (2) the numerical diffusion is only associated with the error resulting from approximating the convective terms; (3) a contact/slip or shear layer is smeared increasingly with corresponding increase time and distance.

In the Eulerian methods, the primitive or conserved variables are represented by their global fixed coordinates; however, the same variables that are written in the Lagrangian frame are represented by their local coordinates. As a result, density and pressure regardless of the subdomain grid on which they are computed are always represented by their unique and global reference coordinates, and the velocity vectors can easily be transferred between the global frame and the individual local frames through simple formulas (Chapters 2.3 and 2.5). Therefore, the data transfer between different component grids with the Eulerian description is straightforward, and the local frames can be chosen to be either inertial or non-inertial (dynamic). In the Lagrangian method, the data transfer process needs the information on the local grid speed to account for the relative velocity of different components. Hence, the Eulerian methods are suitable for the dynamic domain decomposition techniques (D^3T), and can be easily applied to dynamic problems.

Dougherty et al. [30] performed computations for unsteady transonic flows, where two objects were moved along prescribed paths. A two-dimensional, inviscid, unsteady code was used to study the CHIMERA scheme for an application to this type of problem. In this study, only the minor mesh moves with respect to the major global mesh. The results indicate that allowing one mesh to move with respect to another does not adversely affect the time accuracy of an unsteady flow. The results of the moving mesh scenario show the importance of overlapped/embedded schemes. Dougherty and Kuan [31] later extended this method to the three-dimensional analysis of an elliptic body near a flat plate, where this elliptic body was moving along prescribed paths: a simple downward translation, a pitch up, and a rotation.

Meakin and Suhs [32] extended the CHIMERA scheme for time accurate simulation of multiple aerodynamic bodies with relative motion. This method featured the unsteady CHIMERA technique and an implicit, approximately factored, finite difference scheme for the time dependent, thin-layer Navier-Stokes equations. Also, a concept of "smart search" has been introduced into the code [33] which can save CPU time for overlapped reconnections. Several configurations were considered as test cases and two configurations were used to demonstrate the modified method. The first case was a store released from its original position below a wing. The second case was the separation of an integrated space shuttle from its solid rocket booster. Initially, converged steady state solutions for both configurations were obtained and used as the initial conditions for the unsteady computations. In both cases, the released objects were moved along prescribed paths. However, Meakin has further improved the method to complex configurations with aerodynamically determined motions [34-35]; namely, a finned store that separated from a wing and pylon, and the tilt rotor motion of a V-22 aircraft configuration.

Another promising method for the unsteady flow problems with moving boundaries is the unstructured grid approach. For example, Batina [26] resolved a moving airfoil with unstructured dynamic meshes; Singh et al. [36] simulated the flowfield of a store separating from a wing. However, because of the deformation of the dynamic meshes incurred in the moving procedure, geometry conservation becomes a prerequisite in this approach, which in turn requires extra computer time and memory. The Geometry Conservation Law (GCL) was proposed by Thomas and Lombard [37] in 1978. Their contention was, if the grid was changing, then a constraint on the way the metrics were differenced had to be satisfied to prevent additional errors from being introduced into the solution. To improve the method, a differential GCL was formulated that governs the spatial volume element under an arbitrary mapping. The GCL was solved numerically along with the flow conservation laws using conservative difference operators. Also, Vinokur [38] and Obayashi [39] have further addressed this topic with the example that is described for non-inertial frames of reference.

Tamura and Fujii [40] applied the conservation law to the discretization methods not only in space but also in time. Comparisons of non-conservative methods, GCL method, and GCL with time conservation modification, will be further explained in Section 2.5.

The present investigation is focused on the Eulerian frames of reference. Hence, only a few example references with Lagrangian approaches will be highlighted in this section. The Lagrangian formulation of the governing equations can be found in [27]. To reduce the distortion of the mesh, it is necessary to introduce a new mesh and to transfer information from the old mesh to the new one. The implementation of the continuous rezoning process is referred to as Arbitrary-Lagrangian-Eulerian (ALE) method [41]. This method has been successfully applied to a two-dimensional store separation from a cavity by Lohner [42].

Kandil et al. [43-44] derived the Euler and Navier-Stokes equations in relative frame of reference. The new governing equations preserved the conservation form. During the time-accurate stepping, the "*Navier-displacement*" equations were used to solve the grid deformation sequentially. If only rigid-body motions are assumed to occur, the proposed moving-frame of reference formulation eliminates the need for computing the grid motion at each time step. This method can also be used for a single object with small deformations (e.g. delta wing deformation) with accurate and efficient results.

Liou [29] proposed a modified version of Lagrangian description called the "extended Lagrangian method." This method avoids the inaccuracy incurred due to geometry and variable interpolations used by the previous Lagrangian methods. It automatically adapts to flow features without resorting to clustering, thereby maintaining rather uniform grid spacing throughout and for large time steps.

1.2 Background and Rationale

The present research focuses on the study of unsteady flowfields that involve multiple component configurations with at least one of the components in relative motion with respect to the others. Two of the important phenomena that such analyses can help to understand are the unsteady aerodynamic interference and the boundary induced component of the flowfield. Even though the majority of computational flow analyses performed to this date have involved steady flows about stationary bodies, most of the physically realistic flow problems are unsteady in characteristic. The unsteady flow physics, even without the multi-body interference, is currently a topic of active research since it is not understood as well as for steady flows. Also, the generated unsteady flowfield often behaves significantly differently due to the aerodynamic interference of such multiple bodies. Typical examples of this unsteady flow include the periodic motion or random vibration of aircraft wings, rotor/stator interaction in turbomachinery, relative motions of propeller blades or moving flaps with respect to a complete aircraft, store separation from aircraft, and the solid rocket booster separation from the space shuttle.

To simplify the analysis, the "steady flow past a stationary object" assumption is often used in engineering calculations. Depending on the particular problem in question, this may be a reasonable approach. However, it becomes not only difficult to choose the assumed freestream flow velocity but it also renders unacceptable results, if the problem in question involves a multicomponent configuration with at least one of the components engaged in a motion different from the other components.

In setting up a problem of this class, there are three levels of assumptions one can make for the freestream and solid-surface interaction:

(i) All components are engaged in the same rigid-body motion and by assigning the velocity of this motion to the freestream one can assume all the objects to be stationary. Relative motions of the components are assumed instantaneously frozen; i.e., this becomes an animation if a series of these analyses is performed. Such computations provide insight into only the static component of the inherent aerodynamic interference, in addition to other underlying physical phenomena,

(ii) All components are engaged in the same rigid-body motion, but this time the complete computational grid wrapped around the configuration is assigned this motion during the analyses. This requires unsteady and dynamic-grid calculations. The motion-induced flow component can be captured with such calculations provided that the dominant contribution is not due to the relative motion of a component.

(iii) Each component is assigned its own rigid-body motion and unsteady computations are performed for these dynamic objects. However, the rigid-body motion of each component is either known or assumed to be known so that they can be prescribed as given for the computations. Such calculations require remeshing the computational domain as the components move relative to each other or assigning each component its own grid, which move with the component and communicate with the other component grids.

The analysis will become closer to the physics of the problem yet more complex as the level of these assumptions is increased. Also, each level has its own apparent domain of applicability. For the class of problems, where the rigid-body motion of each component is neither known a priori nor can easily be guessed, but is determined by the generated instantaneous flowfield, the aforementioned levels of assumptions may be rather compromising. Necessity for a higher level of simulations has been recognized and addressed by other investigations. This is the impetus of the present research effort.

The present investigation is focused on the dynamic domain decomposition techniques, abbreviated herein as D^{3}T. The three principle elements of D^{3}T are: (i) Division of the flow region into simpler subdomains within which grids are independently or semi-independently generated using existing grid generation schemes. Hence, different levels of governing equations or different solution methods can be used for different subdomains. (ii) The subdivision of the subdomain's motion. (iii) D^{3}T is a domain block-processing scheme where only the data corresponding to particular subdomain is required to reside in the core memory of the computer at one particular time. Thus, at least theoretically, the block-processing technique permits the use of unlimited global grid sizes and lends itself very efficiently to parallel processing.

With the assistance of D^{3}T, the local origin of each subdomain can be chosen arbitrarily and independently. Hence, the equations of rigid-body motion and moments of inertia may be much simplified with a careful choice of the local origins. However, the most critical element of the D^3T is the communication between different subdomains. The communication or data transferance between subdomain boundaries is accomplished by some type of interpolation method of either conservative or nonconservative nature.

1.3 Present Work

The objectives of the present investigation can be grouped into two categories, as those for the computational methodology and those for the unsteady aerodynamics. These objectives are met by modifying an existing CFD algorithm [1, 22, 45], which is capable of obtaining unsteady solutions for flows around stationary multiple bodies. These objectives are:

1. Developing dynamic domain decomposition techniques to decompose the flow domain into simpler subdomains from existing methods for stationary objects [1].
2. Modifying an existing higher order accurate, implicit, approximately-factored, upwind-biased, finite-volume method [45] which has previously been extended for static domain decomposition techniques [22] to accommodate these dynamic domain decomposition techniques.
3. Coupling a rigid-body-dynamics methodology to the present CFD method in order to determine the trajectory of a flying or a falling rigid body.
4. Studying some of the numerical errors encountered in solving unsteady flows via demonstrative problems: the flows past airfoils engaged in prescribed motions, quasi-two-dimensional shock tube problems, and a 2-D store separating from an airfoil along a prescribed path.

5. Investigating the unsteady flowfield of interfering objects in motion through the demonstrative example of a secant-ogive-cylinder oscillating near a vertical wall.

6. Investigating the interference flowfield of a store falling from a wing, where the trajectory of the store motion is determined by the instantaneous aerodynamic loads.

The logical sequence of the dissertation is as follows. Chapter 2 conveys the basic formulation of both fluid and rigid body dynamics. That is, the governing equations and corresponding initial and boundary conditions, and their solution algorithms for both formulations are discussed in this chapter. The dynamic domain decomposition techniques (D^3T) are described in Chapter 3. The characteristics of D^3T are explained in this chapter. Some of the important contributors to the numerical error of the present computational algorithm are studied in Chapter 4. Several numerical experiments are performed to isolate and identify some of the consequences of each one of these error components, then demonstrate several suggested measures to control them. Four cases of unsteady airfoil flows are demonstrated and discussed in Chapter 5. Chapter 6 provides the extension of the method to three-dimensional multiple objects. The computed results for an oscillating cylinder near a vertical wall are discussed in this chapter. Chapter 7 contains the extension of the method to simulate a store separating from a wing. Details of the "on-line" D^3T and the trajectory predictions are also discussed by demonstrating them through this case. Some conclusions and recommendations based on the demonstration cases are given in Chapter 8.

Chapter 2

GOVERNING EQUATIONS AND SOLUTION METHODS

2.1 Governing Equations of Fluid Flow

The unsteady, Reynolds-averaged, thin-layer Navier-Stokes equations of air flow are written in conservation form and in time-dependent curvilinear coordinates [44],

$$\xi^m = \xi^m(x_1, x_2, x_3, t) \tag{2.1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial E_1}{\partial \xi^1} + \frac{\partial E_2}{\partial \xi^2} + \frac{\partial (E_3 - E_{3v})}{\partial \xi^3} = 0 \tag{2.2}$$

where

$$Q = [\, \rho, \rho u_1, \rho u_2, \rho u_3, e_t \,]^t / J \tag{2.3}$$

$$\begin{aligned} E_m = \frac{1}{J}\, [\rho U_m, \ \rho u_1 U_m + \partial_1 \xi^m p\, , \rho u_2 U_m + \partial_2 \xi^m p\, , \\ \rho u_3 U_m + \partial_3 \xi^m p\, , U_m(e_t + p) - \xi^m_t p \ ]^t \end{aligned} \tag{2.4}$$

$$(E_v)_3 = \frac{1}{J}\left[0, \ \partial_k \xi^3 \tau_{k1}, \ \partial_k \xi^3 \tau_{k2}, \ \partial_k \xi^3 \tau_{k3}, \ \partial_k \xi^3 (u_r \tau_{kr} - q_k)\right]^t \tag{2.5}$$

The contravariant velocity expression is

$$U_m = \partial_k \xi^m \, u_k + \partial_t \xi^m \tag{2.6}$$

and the transformation Jacobian is defined by

$$J = \frac{\partial(\, \xi^1 \, , \xi^2 \, , \xi^3 \,)}{\partial(\, x_1 \, , x_2 \, , x_3 \,)} \, . \tag{2.7}$$

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

$$p = (\gamma - 1)\left[e_t - \frac{\rho\left(u_1^2 + u_2^2 + u_3^2 \right)}{2} \right] \tag{2.8}$$

The shear stresses are given by

$$\partial_k \xi^m \tau_{k1} = \frac{M_\infty \mu}{Re}\left[\left(\partial_k \xi^m \partial_1 \xi^r - \frac{2}{3} \partial_1 \xi^m \partial_k \xi^r \right) \frac{\partial u_k}{\partial \xi^r} + \partial_k \xi^m \partial_k \xi^r \frac{\partial u_1}{\partial \xi^r} \right] \tag{2.9}$$

The expression for $\partial_k \xi^m \tau_{k2}$ and $\partial_k \xi^m \tau_{k3}$ are obtained analogous to Eq. (2.9) with the subscript 1 replaced by 2 and 3, respectively. The heat flux term in Eq. (2.5) is given as

$$\partial_k \xi^m q_k = \frac{M_\infty \mu}{Re} \frac{1}{(\gamma - 1) Pr} \partial_k \xi^m \frac{\partial T}{\partial \xi^r} \tag{2.10}$$

In Eq. (2.2), the conserved variables of vector Q are normalized with the freestream values of density, the local speed of sound and the molecular viscosity. The Sutherland formula is used to determine the molecular viscosity. Reynolds stresses are modeled with a modified Baldwin-Lomax turbulence model reported in [46-48].

$$\rho = \frac{\tilde{\rho}}{\tilde{\rho}_\infty} \;;\; u_1 = \frac{\tilde{u}_1}{\tilde{a}_\infty} \;;\; u_2 = \frac{\tilde{u}_2}{\tilde{a}_\infty} \;;\; u_3 = \frac{\tilde{u}_3}{\tilde{a}_\infty} \;;\; e_t = \frac{\tilde{e}_t}{\tilde{\rho}_\infty \tilde{a}_\infty^2}. \tag{2.11}$$

Also used in the non-dimensionalization is the reference length L.

2.2 Solution Method

In the present research, an upwind, finite volume method is used to solve the fluid governing equations of the fluid flow. Some details of the finite volume method are given in Appendix A.1.

A single step temporal scheme [49] for advancing the solution of Eq. (2.2) is

$$\Delta Q^n = \frac{\mu\Delta t}{1+\nu}\frac{\partial}{\partial t}\Delta Q^n + \frac{\Delta t}{1+\nu}\frac{\partial}{\partial t}Q^n + \frac{\nu}{1+\nu}\Delta Q^{n-1} + \vartheta\left[\left(\mu - \frac{1}{2} - \nu\right)\Delta t^2, \Delta t^3\right] \quad (2.12)$$

where $Q^n = Q(n\Delta t)$ and $\Delta Q^n = Q^{n+1} - Q^n$. The general time differencing formula Eq. (2.12), with the appropriate choice of the parameters μ and ν, reproduce two and three level implicit schemes as listed in Table (2.1). Applying Eq. (2.2) to integrate the time term of Eq. 2.12 yields

$$\Delta Q^n = \frac{\mu\Delta t}{1+\nu}\left[-\delta_{\xi^i}\left(\Delta E_i^n\right) + \delta_{\xi^3}\left(\Delta E_{3v}^n\right)\right] + \frac{\Delta t}{1+\nu}\left[-\delta_{\xi^i}\left(E_i^n\right) + \delta_{\xi^3}\left(E_{3v}^n\right)\right] + \frac{\nu}{1+\nu}\Delta Q^{n-1} \quad (2.13)$$
$$+\text{H.O.T.}$$

Obviously, the spatial derivatives of Eq. (2.13) are nonlinear terms that would be difficult in solving the algebraic equations for ΔQ^n. A time linearized equation with the same temporal accuracy as Eq. (2.13) can be approximated if we use the Taylor series expansion

$$E_i^{n+1} = E_i^n + \frac{\partial E_i^n}{\partial Q}(\Delta Q) + \frac{1}{2}\frac{\partial^2 E_i^n}{\partial Q^2}(\Delta Q)^2 + \vartheta\left(\Delta t^3\right), \quad (2.14)$$

or

$$\Delta E_i^n = A_i^n \Delta Q^n + \vartheta\left(\Delta t^2\right) \quad (2.15)$$

where A_i is the Jacobian matrix $\frac{\partial E_i}{\partial Q}$. Likewise

$$\Delta E_{3v}^n = \left(\frac{\partial E_{3v}}{\partial Q}\right)^n \Delta Q^n + \vartheta\left(\Delta t^2\right) \quad (2.16)$$

If the approximations (2.15) and (2.16) are introduced into (2.14), we obtain

$$\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^1}\left(\partial_Q E_1\right)+\Delta_{\xi^2}\left(\partial_Q E_2\right)+\Delta_{\xi^3}\left(\partial_Q E_3\right)-\Delta^2_{\xi^3}\left(\partial_Q E_{3v}\right)\right]\right\}\Delta Q^n$$
$$=\frac{\Delta t}{1+\nu}\left(-\Delta_{\xi^i}E_i^n+\delta^2_{\xi^3}E^n_{3v}\right)+\frac{\nu}{1+\nu}\Delta Q^{n-1}+\vartheta\left[\left(\mu-\frac{1}{2}-\nu\right)\Delta t^2,\ \Delta t^3,\ \Delta\xi_i^3\right] \qquad (2.17)$$

where $\Delta_{\xi i}$, $\Delta^2_{\xi i}$ denote upwind and central differences, respectively. Eq. (2.17) forms a system of algebraic equations. The elements of the Jacobian matrix are banded and the bandwidth depends on the grid size and choice of spatial differencing method. For multidimensional flows, the left hand side is a very large sparse matrix with a large bandwidth. Then, the exact inversion of the matrix is very costly due to the large number of operations and computer memory required. An approximate factorization scheme (AF) splits the implicit operator into a sequence of lower bandwidth Jacobian matrix. The spatially-factored form of Eq. (2.17) which retains the temporal accuracy, conditionally, can be obtained as follows:

$$\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^1}\left(\partial_Q E_1\right)\right]^n\right\}\times\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^2}\left(\partial_Q E_2\right)\right]^n\right\}$$
$$\times\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^3}\left(\partial_Q E_3\right)-\Delta^2_{\xi^3}\left(\partial_Q E_{3v}\right)\right]^n\right\}\Delta Q=\text{R.H.S.}(2.12)+\vartheta(\Delta t^3) \qquad (2.18)$$

Thus a spatially factored algorithm [49] for Eq. (2.17) but linear in ΔQ^n is

$$\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^1}\left(\partial_Q E_1\right)\right]^n\right\}\times\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^2}\left(\partial_Q E_2\right)\right]^n\right\}$$
$$\times\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^3}\left(\partial_Q E_3\right)-\Delta^2_{\xi^3}\left(\partial_Q E_{3v}\right)\right]^n\right\}\Delta Q$$
$$=-\frac{\Delta t}{1+\nu}\left[\Delta_{\xi^1}E_1^n+\Delta_{\xi^2}E_2^n+\Delta_{\xi^3}E_3^n-\Delta^2_{\xi^3}\left(\partial_Q E_{3v}\right)^n\right] \qquad (2.19)$$
$$+\frac{\nu}{1+\nu}\Delta Q^{n-1}+\vartheta\left[\left(\mu-\frac{1}{2}-\nu\right)\Delta t^2,\ (lf+1)\Delta t^3,\ \Delta\xi_i^3\right],$$

Denoting the right-hand-side of Eq. (2.19) as R.H.S., the left-hand-side may be rearranged to facilitate the alternating-direction-implicit (ADI) algorithm,

$$\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^1}\left(\partial_Q E_1\right)\right]^n\right\}\Delta Q^{**} = -(\mathrm{R.H.S.})^n \tag{2.20}$$

$$\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^2}\left(\partial_Q E_2\right)\right]^n\right\}\Delta Q^{*} = \Delta Q^{**} \tag{2.21}$$

$$\begin{aligned}\left\{I+\frac{\mu\Delta t}{1+\nu}\left[\Delta_{\xi^3}\left(\partial_Q E_3\right)-\Delta^2_{\xi^3}\left(\partial_Q E_{3v}\right)\right]^n\right\}\Delta Q &= \Delta Q^{*}\\ Q^{n+1} &= Q^n+\Delta Q\end{aligned} \tag{2.22}$$

This scheme requires the solution of a 5*5 block tridiagonal system. It has the advantage of being fully vectorizable. The last term of the Eq.(2.19) indicates the order of magnitude of the leading truncated term, that is, the truncation error. The method is second-order accurate in space. The temporal accuracy, however, depends on the magnitude of the neglected terms in linearization and in approximate factorization (represented by the parameter *lf*), as well as the values of the free parameters μ and ν. For μ=1/2 and ν=1, the temporal truncation error becomes $\vartheta\left[(lf+1)\Delta t^3\right]$, then, this method is second-order accurate temporally. This algorithm is implemented by modifying the general purpose CFD code CFL3D [45] for improved time accuracy.

Roe's flux difference splitting [50] is used to construct the spatial fluxes for the convective and pressure terms. Some details of this method are highlighted in Appendix A.2. Also, to reduce the computational time, Pulliam's [51] diagonalization for the Roe's flux difference splitting scheme, details of which are given in Appendix A.3, can be used.

2.3 Governing Equations of Rigid Body Motion

The method described so far produces the solution of a flowfield around a static body or a body engaged in a prescribed motion; that is, it can be represented by the equations of fluid dynamics. To extend the method for a flow past a body engaged in a motion which is not known a priori, the connection between the force field computed by the fluid dynamics and the rigid-body kinematics needs to be established.

In order to be oriented for the frames of references, consider a fluid particle P in the inertial domain XYZ with origin at O. $\bar{r}$ denotes the position vector of P. The non-inertial frame of reference xyz has it origin O' with $\bar{R}$ as the position vector of O' relative to the origin of the global domain XYZ (Fig. 2.1). The XYZ system is fixed in space but the xyz system translates and rotates relative to it [52]. Then the position vectors are related by

$$\bar{r} = \bar{R} + \bar{x}, \tag{2.23}$$

where $\bar{x}$ is the position vector of P relative to O'. Differentiating with respect to time, the absolute velocity of P is

$$\bar{v} = \dot{\bar{r}} = \dot{\bar{R}} + (\dot{\bar{x}})_r + \bar{\omega} \times \bar{x}. \tag{2.24}$$

In order to explain the meaning of terms on the right side of Eq. (2.24), the point P' needs to be defined. This can be a rigid grid point of the subdomain xyz, which is coincident with P at the time of observation but fixed in the xyz system. Then $\dot{\bar{R}}$ is the absolute velocity of O', $\bar{\omega}$ is the absolute rotation rate of the xyz system, and $\bar{\omega} \times \bar{x}$ is the velocity of P' relative to O' as viewed by a nonrotating observer. $(\dot{\bar{x}})_r$ is the velocity of P relative to O' as viewed by an observer rotating with the xyz system. It should be noted that the $(\dot{\bar{x}})_r$ is zero for a rigid-body motion.

Differentiating Eq. (2.24) once more with respect to time, the absolute acceleration of P can be derived as

$$\vec{a} = \dot{\vec{v}} = \ddot{\vec{R}} + \dot{\vec{\omega}} \times \vec{x} + \vec{\omega} \times (\vec{\omega} \times \vec{x}) + (\ddot{\vec{x}})_r + 2\vec{\omega} \times (\dot{\vec{x}})_r . \tag{2.25}$$

where $\ddot{\vec{R}}$ is the absolute acceleration of O'. The term $\dot{\vec{\omega}} \times \vec{x}$ is similar in nature to the tangential acceleration, whereas the term $\vec{\omega} \times (\vec{\omega} \times \vec{x})$ represents a centripetal acceleration. The term $(\ddot{\vec{x}})_r$ is the acceleration of the point P relative to the xyz system, and the term $2\vec{\omega} \times (\dot{\vec{x}})_r$ is known as the Coriolis acceleration.

The general expression to describe the rotational motion (rotation equations) of a rigid body can be written as

$$\vec{M} = \bar{\bar{I}}\dot{\vec{\omega}} + \vec{\omega} \times \left(\bar{\bar{I}}\vec{\omega}\right), \tag{2.26}$$

where the moment $\vec{M}$, inertia $\bar{\bar{I}}$, and the angular velocity $\vec{\omega}$ have the following components:

$$\begin{gathered} \vec{M} = \left(M_x, M_y, M_z\right) \\ \bar{\bar{I}} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \\ \vec{\omega} = (\omega_x, \omega_y, \omega_z) \end{gathered} \tag{2.27}$$

A considerable simplification can be made in the general rotational equations of motion if the xyz coordinate axes are chosen to coincide with the principal axes and the local origin is located at the center of mass. With this choice of body axes, all cross-products of inertia vanish and moments of inertia are time independent, hence, Eq. (2.26) can be reduced to

$$\begin{aligned} M_x &= I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_y\omega_z \\ M_y &= I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_x\omega_z \\ M_z &= I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_y\omega_x \end{aligned} \tag{2.28}$$

These are known as the Euler's equations of rigid-body rotation.

The translational equations of a rigid body are represented by a coordinate system fixed on the body. The equations can be written as

$$\bar{F} = m\dot{\bar{v}} = m\left(\dot{\bar{v}}\right)_r + m(\bar{\omega} \times \bar{v}), \tag{2.29}$$

where m is the mass of the rigid body.

2.4 Solution Method for Rigid Body Motion

The aerodynamic forces and moments are derived by integrating the instantaneous surface pressure field and their moments applied at the local origin of the non-inertial frame of reference:

$$\bar{F} = m\bar{g} - \int_s p \cdot dS_m , \quad m = 1,\ 2 \text{ and } 3 , \tag{2.30}$$

$$\bar{M} = -\int_s \left[\left(\bar{r} - \bar{r}_{cg} \right) \times d\bar{F} \right]. \tag{2.31}$$

But, all the aerodynamic coefficients are obtained relative to the inertial frame of reference adopted for the fluid dynamics equations. Hence, Eqs. (2.30) and (2.31) need to be transformed from the inertial XYZ system to the non-inertial xyz system.

By applying the general rotation equations of rigid-body dynamics, the trajectory (or path) of a body motion can be determined as a function of the aerodynamic coefficients, its weight and its inertial characteristics. Therefore, transformation relations are required between the inertial and non-inertial frames of reference. Let r_j' and r_j represent the vector quantities in the non-inertial and inertial frames of references, respectively, then the transformation is $r_j' = c_{ij}\, r_j$, where c_{ij} is the directional cosine tensor:

$$\overline{\overline{C}} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ -\cos(\beta)\sin(\gamma) + \sin(\alpha)\sin(\beta)\cos(\gamma) & \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) & \sin(\alpha)\cos(\beta) \\ \sin(\alpha)\sin(\gamma) + \cos(\alpha)\sin(\beta)\cos(\gamma) & -\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix} \qquad (2.32)$$

where α, β and γ are the Euler angles of the non-inertial frame with respect to the inertial frame.

Let M_x, M_y, and M_z be the moments represented in the body's principal axes, and M_X, M_Y, and M_Z be the same moments but represented with respect to the fixed axes (inertial frame), then the transformation relation can be expressed as follows:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} C_{ij} \end{bmatrix} \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix} \qquad (2.33)$$

A similar relation needs to be used for the force equations.

In order to update the velocity and position vectors of the body in time, Eqs. (2.28) are solved simultaneously to obtain the angular acceleration. Then, the angular acceleration is integrated in time using a fourth order Runge-Kutta method to obtain the angular velocity field. Using the obtained angular velocities, Eqs. (2.29) may now be solved simultaneously to obtain the translational accelerations. These accelerations are then integrated to give the translational velocity field. However, the time integrals of ω_x, ω_y, and ω_z do not correspond to any physical angles which might be used to give the orientation of the body. Hence, the Euler angles for the new position are obtained by solving the next group of differential equations simultaneously,

$$\begin{aligned}\dot{\alpha} &= \left[\omega_y \sin(\gamma) + \omega_z \cos(\gamma)\right]\sec(\beta)\\ \dot{\beta} &= \omega_y \cos(\gamma) - \omega_z \sin(\gamma)\\ \dot{\gamma} &= \omega_x + \left[\omega_y \sin(\gamma) + \omega_z \cos(\gamma)\right]\tan(\beta)\end{aligned} \tag{2.34}$$

Hence, the angular velocity is integrated in time by using a fourth order Runge-Kutta method to obtain the Euler angles.

Then, the velocities and angular velocities must be transformed back to the inertial reference frame,

$$\begin{bmatrix}\omega_X\\ \omega_Y\\ \omega_Z\end{bmatrix} = \left[C_{ij}\right]^T \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{2.35}$$

A similar transformation needs to be performed for the velocity field. The new position vector of the rigid body can be found by using Newton's law of motion,

$$\begin{aligned}\vec{R}^{n+1} &= \vec{R}^{n} + \vec{V}_{\text{translation}} \cdot \Delta t\\ \vec{r}^{n+1} &= \vec{R}^{n+1} + \left[C_{ij}\right]^T \cdot \vec{x}\end{aligned} \tag{2.36}$$

where the velocity vector is

$$\vec{V} = \vec{V}_{\text{translation}} + \vec{\omega} \times \vec{x}. \tag{2.37}$$

All the variables shown in Eqs. 2.36-2.37 are represented in the inertial frame of reference XYZ.

2.5 On Geometric Conservation Law

The integral form of a conservation law of Euler equations can be rewritten [38-39] as

$$\int_{V(t_2)} Q dV - \int_{V(t_1)} Q dV + \int_{t_1}^{t_2} \oint_{S(t)} \bar{E}\bar{n} \cdot d\bar{S} dt = 0 \tag{2.38}$$

where V(t) is the cell volume at time t and $\bar{n} \cdot d\bar{S}$ is a vector element of surface area with outwardly normal. The flux $\bar{E}$ can be written as

$$\bar{E} = (\bar{u} - \bar{v})Q = E_{st} - \bar{v}Q. \tag{2.39}$$

where $\bar{u}$ and $\bar{v}$ are flow particle and solid boundary speed, respectively. For the freestream conditions Q_∞ and E_{st_∞}, the geometric identities can be derived from Eq. (2.38) as follows:

$$Q_\infty [V(t_2) - V(t_1)] = -\bar{E}_{st_\infty} \cdot \int_{t_1}^{t_2} \oint_{S(t)} \bar{n} \cdot d\bar{S} dt + Q_\infty \int_{t_1}^{t_2} \oint_{S(t)} \bar{n} \cdot \bar{v} d\bar{S} \, dt. \tag{2.40}$$

For a stationary grid, the first geometric identity can be obtained as

$$\oint_{S(t)} \bar{n} \cdot d\bar{S} = 0. \tag{2.41}$$

Substituting Eq.(2.41) simplifies (2.40),

$$[V(t_2) - V(t_1)] = \int_{t_1}^{t_2} \oint_{S(t)} \bar{n} \cdot \bar{v} d\bar{S} \, dt. \tag{2.42}$$

This equation is essential for moving grids. To ensure the global conservation for numerical solutions, the differential form of Eq. (2.42) needs to be satisfied:

$$\frac{\partial V}{\partial t} = \oint_{S(t)} \bar{v} \cdot \bar{n} \, d\bar{S}. \tag{2.43}$$

To proceed with the surface integration on the right-hand side, the surface element velocity $\bar{v}$ needs to be defined more clearly. Therefore, Eq. (2.24) will be reused here,

$$\bar{v} = \dot{\bar{r}} = \dot{\bar{R}} + (\dot{\bar{x}})_r + \bar{\omega} \times \bar{x}. \tag{2.24}$$

where xyz system can be an inertial or a non-inertial frame. In the current derivation, P' is assumed to be a grid point in a rigid grid, and P is a grid point in a deforming grid that is temporarily overlapped and identical to P' grid (Fig. 2.1). The difference between P and P' is the change of a cell in time. From Eqs.(2.24) and (2.43),

$$\frac{\partial V}{\partial t} = \dot{\vec{R}} \cdot \left(\oint_S \vec{n} \cdot dS \right) + \vec{\omega} \times \left(\oint_S \vec{x} \cdot \vec{n} dS \right) + \left(\oint_S \vec{n} \cdot \left(\dot{\vec{x}} \right)_r dS \right) \qquad (2.44)$$

The first integral on the right hand side is zero due to Eq.(2.41). The second integral represents the rigid rotation (P' with respect to xyz system) of grid and does not affect the change of the cell volume. Therefore, Eq.(2.44) can be rewritten in the following differential forms

$$V_t = \left(S^{\xi_i} \cdot \vec{r}_t \right)_{\xi^i} , \qquad i = 1,\ 2,\ \text{and}\ 3 \qquad (2.45)$$

with the constraints

$$\left(S^{\xi_i} \right)_{\xi^i} = 0, \qquad i = 1,\ 2,\ \text{and}\ 3 \qquad (2.46)$$

Eq.(2.45) is the differential statement of the geometric conservation law (GCL).

Therefore, it can be concluded, that if there is no grid deformation (P always coincides with P' at any instant), then there is no need to consider the GCL equations. On the contrary, if there is a grid deformation (P has relative motion with respect to P'), then GCL needs to be evaluated. In the case of using, for example, the dynamic overlapped grids, there is no grid deformation and, therefore, GCL equations are not needed. However, when using dynamic unstructured grids [26, 36], grid cells are deformed and GCL needs to be satisfied by solving Eq. (2.45).

2.6 Initial and Boundary Conditions

The choice of initial and boundary conditions directly affect the accuracy of CFD methods. The initial conditions should correspond to the physical nature of the flow. In this dissertation, the initial conditions are chosen to be the freestream conditions or previous computational results.

The boundary conditions are specified explicitly in spite of the employed implicit CFD algorithm. The walls are considered to be impermeable and adiabatic. The pressure at the solid surfaces is evaluated using the momentum equations of the fluid flow [43],

$$\frac{\partial p}{\partial n} = \rho \bar{v} \cdot (\bar{v} \cdot \nabla \hat{n}) - \rho \hat{n} \cdot \bar{a}, \tag{2.47}$$

where $\bar{a}$ is obtained from Eq.(2.25), and the velocities for the inviscid surfaces are set equal to the contravariant velocities:

$$u_1 = U_1 \ , \ u_2 = U_2 \ , \ u_3 = U_3 \ , \ \frac{\partial T}{\partial n} = 0. \tag{2.48}$$

The velocities for the viscous surface are set equal to the velocity of the surface,

$$u_1 = v_1 \ , \ u_2 = v_2 \ , \ u_3 = v_3 \ , \ \frac{\partial T}{\partial n} = 0. \tag{2.49}$$

Finally, the density is calculated by employing the state equation.

The farfield boundary conditions are obtained by using the locally one-dimensional characteristic boundary conditions. For each cell, the velocity normal to the boundary and the speed of sound are calculated from the two locally one-dimensional Riemann invariants [22, 53] given by:

$$R^{\pm} = U \pm \frac{2}{\gamma - 1} a \ . \tag{2.50}$$

The invariants are constant along the characteristics defined by:

$$\left(\frac{d\xi}{dt}\right)^{\pm} = U \pm a \ . \tag{2.51}$$

The local normal velocity at the boundary is calculated by summing the two Riemann invariants, and the speed of sound is obtained by subtracting the two. The appropriate boundary conditions are specified after the magnitude of the contravariant Mach number and the directions of the contravariant velocity at each cell are checked:

$$|M_\xi| > 1;\ U < 0 \quad u_1 = u_{1\infty},\ u_2 = u_{2\infty},\ u_3 = u_{3\infty},\ T = T_\infty,\ p = p_\infty . \tag{2.52}$$

$$|M_\xi| > 1;\ U > 0 \quad \frac{\partial u_1}{\partial \xi^1} = 0,\ \frac{\partial u_2}{\partial \xi^1} = 0,\ \frac{\partial u_3}{\partial \xi^1} = 0,\ \frac{\partial T}{\partial \xi} = 0,\ \frac{\partial p}{\partial \xi} = 0 , \tag{2.53}$$

$$|M_\xi| < 1;\ U < 0 \quad u_1 = u_{1\infty},\ u_2 = u_{2\infty},\ u_3 = u_{3\infty},\ T = T_\infty,\ \frac{\partial p}{\partial \xi} = 0 . \tag{2.54}$$

$$|M_\xi| < 1;\ U > 0 \quad \frac{\partial u_1}{\partial \xi^1} = 0,\ \frac{\partial u_2}{\partial \xi^1} = 0,\ \frac{\partial u_3}{\partial \xi^1} = 0,\ \frac{\partial T}{\partial \xi} = 0,\ p = p_\infty , \tag{2.55}$$

For a supersonic inflow, all flow characteristic lines point toward the computational domain. Hence, the inflow boundary conditions are set equal to the freestream (Eq. 2.52).

For supersonic outflow, all flow characteristic lines point out of the computational domain. Hence, the outflow boundary conditions are obtained using a zeroth-order extrapolation (Eq. 2.53).

Reflection boundary conditions are used on the symmetry plane of the two-dimensional cases. For example, when the plane of symmetry is assumed in ξ^1 direction, then

$$u_1 = -u_1\ ,\ u_2 = u_2\ ,\ u_3 = u_3\ ,\ \frac{\partial T}{\partial \xi^1} = 0\ ,\ \frac{\partial p}{\partial \xi^1} = 0. \tag{2.56}$$

Table 2.1: Partial list of schemes contained in Eq. (2.12)

μ	ν	Scheme	Truncation error
1	0	Euler, implicit	$\vartheta\left(\Delta t^2\right)$
1/2	0	Trapezoidal, implicit	$\vartheta\left(\Delta t^3\right)$
1	1/2	Three-point-backward, implicit	$\vartheta\left(\Delta t^3\right)$

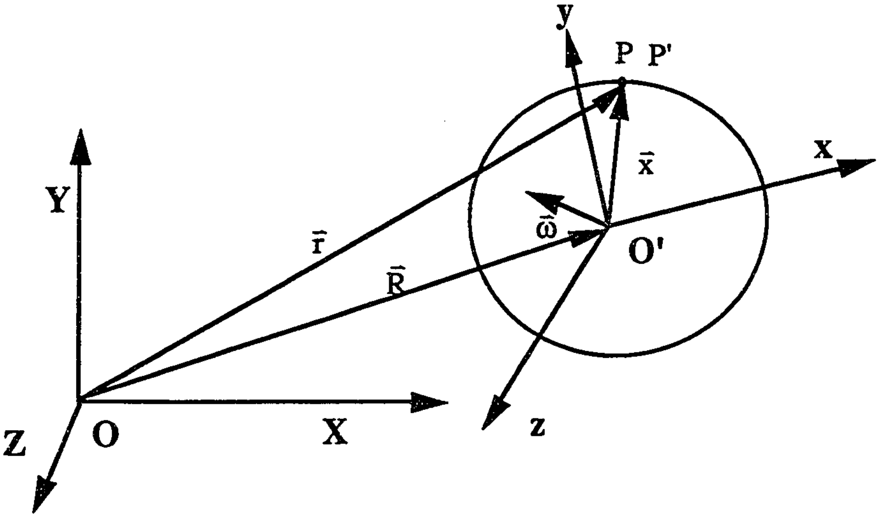


Fig. 2.1: The position vectors of points P and P' relative to a space-fixed inertial system and a moving (non-inertial) system.

Chapter 3

DYNAMIC DOMAIN DECOMPOSITION

This chapter illustrates the concept of extending *static* Domain Decomposition Techniques (DDT) to *Dynamic* Domain Decomposition Techniques (D^3T). Section 3.1 explains the basic concept of time dependent Jacobian transformation for coordinate systems. Section 3.2 contains the criterion of choosing D^3T and some characteristics of D^3T. The concepts and mathematic formulae of time-dependent degenerate zonal method are given in Section 3.3. Several options of interacting overlapped grids method and flow solver are described in Section 3.4. It can be optimized for the accuracy or efficiency. Section 3.5 gives a simple flowchart that outlines the interaction of these computational components.

3.1 Time-Dependent Coordinate Transformation

There are two apparent differences between a steady and an unsteady flow solution algorithm. First, there is a converged solution (residual approaches zero) for a steady-state case, which, of course, does not exist for an unsteady flow case. Secondly, in an unsteady flow calculation, the time becomes an independent variable; in other words, the time variable can not be ignored in the Jacobian transformation matrix. With the assumption

$$dt = d\tau , \tag{3.1}$$

the Jacobian matrix for unsteady flow can be written as

$$\begin{bmatrix} dt \\ dx_1 \\ dx_2 \\ dx_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ x_{1_\tau} & x_{1_{\xi^1}} & x_{1_{\xi^2}} & x_{1_{\xi^3}} \\ x_{2_\tau} & x_{2_{\xi^1}} & x_{2_{\xi^2}} & x_{2_{\xi^3}} \\ x_{3_\tau} & x_{3_{\xi^1}} & x_{3_{\xi^2}} & x_{3_{\xi^3}} \end{bmatrix} \begin{bmatrix} d\tau \\ d\xi^1 \\ d\xi^2 \\ d\xi^3 \end{bmatrix} . \qquad (3.2)$$

Consequently, the grid speed appears in the contravariant velocity for a generalized coordinate system. For example,

$$U_m = \partial_k \xi^m u_k + \partial_t \xi^m , \quad m, k = 1, 2, \text{and } 3, \qquad (3.3)$$

where the grid speed for the computational domain can be written as

$$\partial_t \xi^m = -\left(\frac{\partial x^j}{\partial \tau}\right) \cdot \nabla \xi^m , \quad m, j = 1, 2, \text{and } 3. \qquad (3.4)$$

The term $\left(\frac{\partial x^j}{\partial \tau}\right)$ is the grid speed for the physical domain, and it is a parameter of primary importance for unsteady aerodynamic problems involving moving boundaries [54-56].

Because the time variable gets involved in the unsteady calculation equal to Eq. (2.24), the individual component grid of the multicomponent configuration may contain different grid speeds. Therefore, they need careful treatment for the communication between different component grids [54].

3.2 Dynamic Domain Decomposition Techniques (D^3T)

In deciding on the type of domain decomposition for a problem with static boundaries, two major concerns are the geometry of the configuration and a priori knowledge of the flowfield. A proper choice of the scheme requires a compromise, so that generating each subdomain grid is an easy task, physical and intergrid boundary conditions can easily be implemented, the total number of cells is minimized, and proper resolution is enforced where significant flow gradients occur.

In the case of dynamic objects, an additional concern is the type of motion each nonstationary component of the configuration is engaged. Using D^3T, one can resolve a complex motion into its simple dynamic components and assign a subdomain grid for each component of the motion. For example, a two-component configuration with each component moving in 6-DOF, can have a composite of twelve subdomain grids, each of which moving in a single DOF. There are two major differences between the present D^3T and static domain decomposition. First, each subdomain grid can be moving. This is accounted for by the time-dependent curvilinear coordinate transformations, whereby the grid velocities are also determined. These nonlinear equations are solved for the nonlinear rigid-body dynamics and the flowfield around it. Secondly, the intergrid information transfer is time dependent. That is, all the searches and the bookkeeping for the interpolations or other modes of transfer are renewed and updated as functions of time. The accuracy and the efficiency of this process are the important issues in developing D^3T.

Since the governing equations are written in the space-fixed frame of reference, all the primitive variables are absolute everywhere and at anytime regardless of the subdomain grid on which they are computed. Therefore, they can be transferred from one subdomain grid to the other regardless of the relative motion of these grids.

3.3 Degenerate Zonal Method

This is a relatively new DDT [23, 54-55], which is derived from the convergence accelerating multigrid method. It has the simplicity and the suitability of refining the grid wherever necessary. It is required that only every *n-th* line normal to the grid interface is contiguous. If $n=2$, then the line that is not contiguous bisects the distance between the neighboring two contiguous lines. This concept is a natural extension from the generation of coarse-fine grids necessary in the multigrid convergence acceleration methods. The intergrid communication is very simple for this method. A schematic of the degenerate zonal grid scheme, when all the grids are static, is shown in Fig. 3.1 for two dimensions. This diagram represents a full refinement in both directions. The dashed lines define a finer mesh embedded completely within a coarser mesh depicted by the solid lines. The open circles are the cell-center locations of the finer embedded grid. Each of the grids is a rectangularly-ordered set of points; in the sketch, a portion of the flowfield is covered both by the embedded grid and a portion of the coarser grid. The grids are coupled together during the solution process. The cell-center variables on a coarser grid cell that underlies a finer embedded grid cell are replaced with a volume-weighted restriction of variables from the four (2-D) or eight (3-D) finer grid cells, similar to the restriction operators used in a global multigrid scheme.

For the embedded finer grid, the computation boundaries occur either at a physical boundary or along an interior line of a coarser grid. Along such a boundary, two additional lines of data corresponding to an analytical continuation of the finer grid cell centers are constructed from interpolation (or prolongation) of the coarser grid primitive variables (Fig. 3.2). Since for most of the applications subdomain grids are nonuniform, the interpolation formula is most commonly used for the prolongation, an inverse averaging formula is more effective when more than two points are used for the interpolation. First, the flow properties at coarse grid cell centers are interpolated to "pseudo-fine" cell centers in each ξ-constant plane of a three-dimensional grid by the following inverse averaging on a four-point stencil (Fig. 3.2a),

$$q_p(i) = \frac{\sum\limits_{i=A}^{A,B,C,D} \left[q_c(i) \right] \Big/ r_i}{\sum\limits_{i=A}^{A,B,C,D} \left[1 \big/ r_i \right]} \qquad (3.5)$$

Then, the flow properties of a fine grid cell center is obtained by interpolating the values of three contiguous pseudo-fine centers in the ξ direction (Fig. 3.2b),

$$q_f = \frac{\sum_{i=1}^{3} [\, q_p(i) \,] \big/ r_i}{\sum_{i=1}^{3} \left[1 \big/ r_i \right]} \tag{3.6}$$

In general, none of the *d* or *s* values are equal to each other. A 2-D problem requires only Eq. (3.5) and the pseudo-fine cell centers become the actual cell centers of the fine grid.

During the restriction procedure, the primitive variables of a coarse grid are replaced from the nearest eight (3-D) finer grid cells using the following inverse averaging formula based on the instantaneous relative distance (Fig. 3.3),

$$q_c = \frac{\sum_{i=1}^{8} [\, q_f(i) \,] \big/ r_i}{\sum_{i=1}^{8} \left[1 \big/ r_i \right]} \tag{3.7}$$

When degenerate zonal grids are in relative motion, even every *n-th* grid line normal to the interface can no longer be contiguous. If the grid motion is modified at every time step to ensure this contiguity, a fully conservative transfer of fluxes can be maintained as in the static case. However, the resulting grid motion is described by a step-function. An alternative method is developed in this study. This method does not restrict the grid displacement to integer multiples of one spatial step. That is, it allows the displacement to be any fraction of one spatial step. During the restriction procedure, primitive variables of a coarse grid are replaced from the nearest eight (three-dimensional) finer grid cells with an inverse averaging formula using the instantaneous relative distances. This formula is used at every time step and it accounts for the nonuniform grids with non-contiguous normal lines. The basic concept of information transfer during the prolongation process, that is, from the coarse grid to the boundary of the fine grid, is unchanged from its static counterpart explained earlier.

3.4 Overlapped Grids Method

The details of static overlapped grids and the interaction of overlapped grids method (MaGGiE) with the flow solver (CFL3D) are given in [1]. A brief summary, however, is provided in Appendix B. The implementation of the overlapped grids method for complex but static geometry problems are explained in [22]. This section is dedicated to extend the overlapped grids method to moving boundary problems.

The computer programs of this study performing the D^3T operations and solving the governing equations are called MaGGiE and CFL3D, respectively. Three different methods are possible for the interaction of MaGGiE and CFL3D. If the boundary motions are forced and/or prescribed, the grid motions and their relative positions are known prior to solving the flow equations. As the first method, MaGGiE is used as a preprocessor to generate and store out-of-core a database of D^3T information as a function of time with appropriate discrete time intervals. CFL3D is marched in time with the corresponding D^3T information as input for a given time step. The second method is using CFL3D and MaGGiE as different modules driven by the same main program. Therefore, all the programs are on-line and the main driver calls upon the appropriate functions as necessary. The first method, when applicable, requires significantly less computer run-time memory, because it does not require the memory to run both of the programs. However, if the boundary motions are determined from the instantaneous force and moment field as integrated from the flowfield, only the second method can be used.

A third method is an extension of the second method in order to reduce the required computer time. Note that the most computer time (CPU) consuming part of MaGGiE for

D^{3}T is where the cells on each side of an intergrid boundary are searched for appropriate communications between the subdomains involved [12, 16]. Using the fact that the grid motions are small per time step, a smarter searching logic can be build into MaGGiE to reduce the time. The new search can be restricted to a specified proximity of the cells, to or from which the information transfers are made in the previous time step [32]. Specifying of the proximities is a compromise between a successful search and the computer time required for each search. Further computer savings can be obtained by freezing the D^{3}T information for more than one time step. This, however, compromises the accuracy.

3.4 Flowchart for Moving Boundary Simulation

The computational method for the D^{3}T consists of three components: (1) The solver of the governing equations of the flowfield - CFL3D [45, 54]. (2) Dynamic domain decomposition for subdomain grid connections and communications as they move relative to each other - MaGGiE and/or degenerate zonal method [54-55]. (3) Force and moment integrator or prescribed motion function to determine the paths or trajectories of rigid-body components in 6-DOF [36]. The flowchart outlining the interaction of these computational components is given in Fig. 3.4.

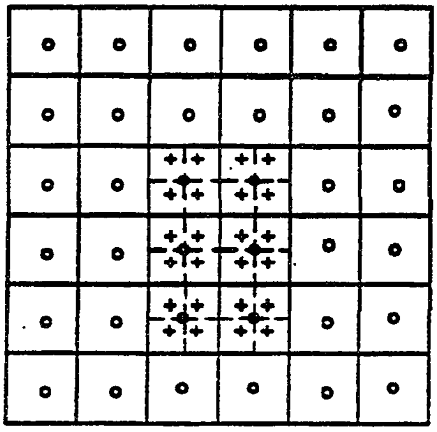


Fig. 3.1: A typical stencil for coarse-fine grid communication in degenerate zonal grids.

○ }
● } coarse grid cell center
\+ finer grid cell center.

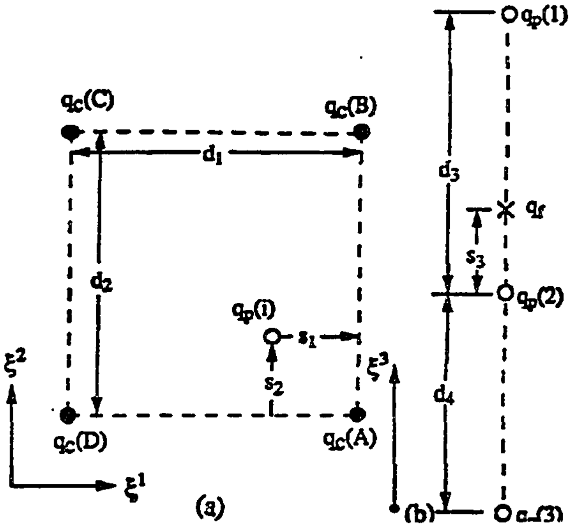


Fig. 3.2: A typical stencil for dynamic degenerate zonal grids during communication from the coarse grid to the fine grid boundary (prolongation).
(a) two dimensional plane,
(b) one dimensional plane.

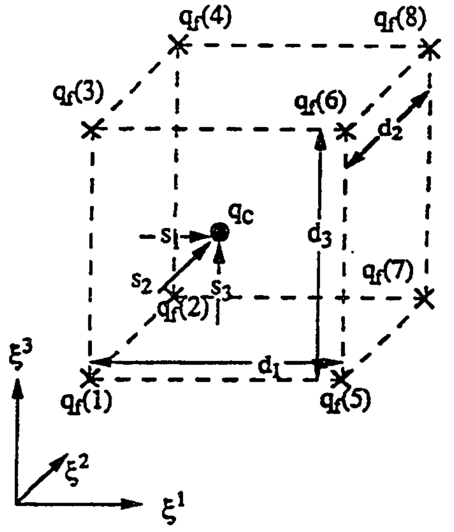


Fig. 3.3: A typical stencil for dynamic degenerate zonal grids during communication from the fine grid to the coarse grid (restriction).

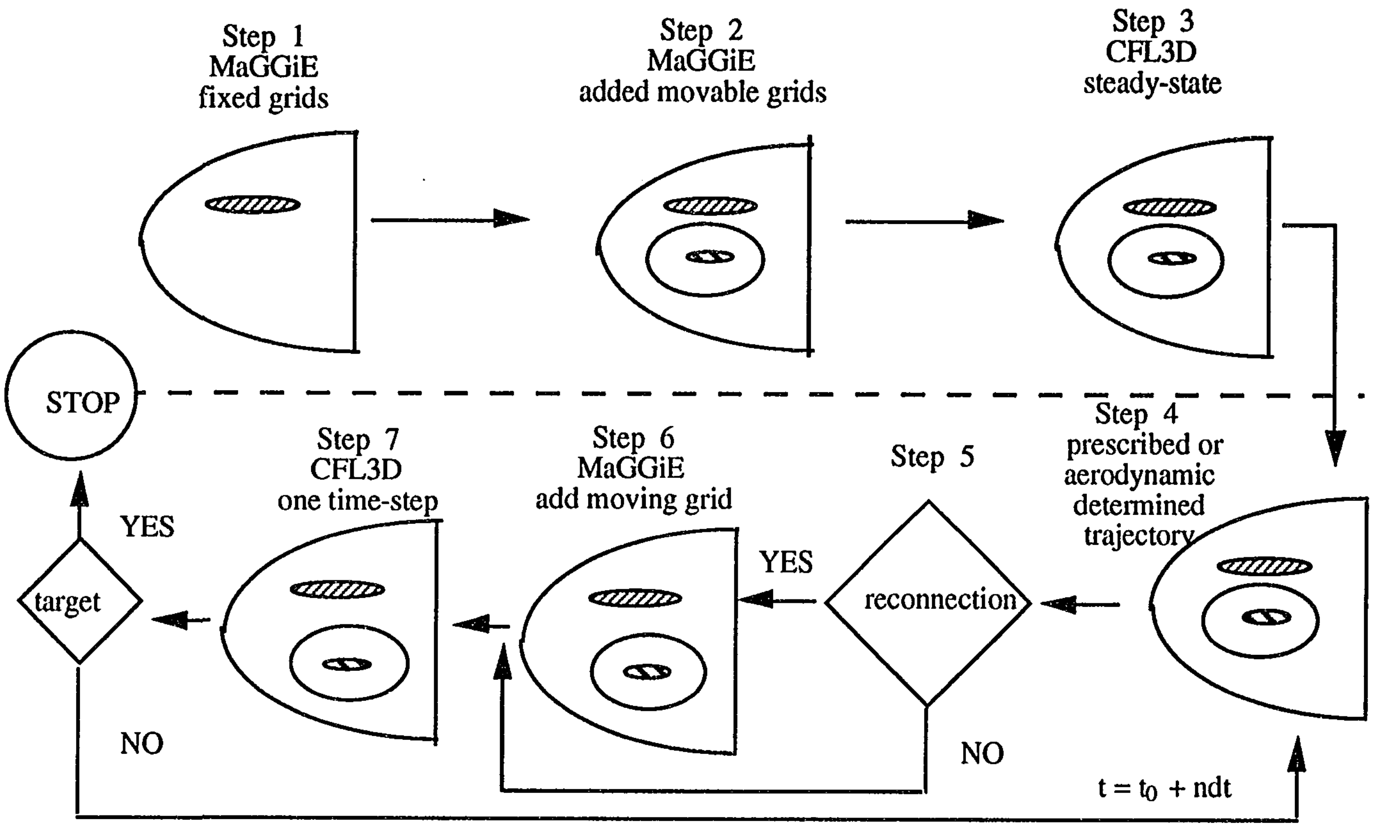


Fig. 3.4: Computational flowchart.

Chapter 4

ACCURACY AND ITS DEMONSTRATION

This chapter focuses on the numerical error studies and the methods of reducing these errors. These numerical errors, which are caused by the CFD algorithm and rigid-body dynamics, are discussed in Section 4.1. Then, a moving normal shock case inside a tube problem is demonstrated and discussed in Section 4.2. Further study of numerical error associated with moving overlapped grids are given in Section 4.3. Finally, the 6-DOF trajectory prediction method is tested and compared with experimental data in Section 4.4.

4.1 Numerical Errors

The error of the computational algorithm to solve the fluid dynamics equations consists of: (a) dissipation and dispersion components of the discretization (or truncation) error, (b) approximate factorization error, (c) interpolation error of the domain decomposition, (d) time-linearization error, and (e) round-off error. Some remarks are made in this section on the first three sources of error as they impact the present solution algorithm.

4.1.1 Dissipation and Dispersion Errors

Conventionally, the accuracy of a numerical scheme is assessed based on the exact solution of the steady-state equation. Hence, only the dissipative error can be correctly identified. However, for unsteady equations, due to the effect of the numerical error, the flow may propagate essentially in a dispersive medium. Since there is no converged solution achieved by a time-accurate marching procedure of an unsteady equation, the phase error due to the dispersion continues to accumulate. Therefore, the group velocity of the motion may deviate significantly from its phase velocity [59-60]. Also, since there is no accurate method to control its direction, the phase error may accumulate in a certain direction. As reported in [58-60], the dispersion error is as important as the dissipation error for unsteady flows. Therefore, the higher order accurate discretization should be applied not only spatially but also temporally [61]. However, more than second-order accuracy may be computationally expensive for three-dimensional and complex geometry cases.

4.1.2 Factorization Error

Implicit numerical methods have been widely used because of their numerical stability with larger time step sizes, but storing the coefficient matrix formed on the left-hand side requires large computer memory. The most common way of reducing the memory requirement is the approximate factorization of the coefficient matrix (Section 2.2), which also has the advantage of rendering the efficiently solvable tridiagonal form. However, this introduces an extra numerical error, which is of the same order as the discretization error, into the discrete system, hence smearing the numerical solutions further. For example, the order of magnitude of ΔQ^n in Eq. (2.19) is assumed to be $\vartheta(\Delta t)$ [49]. However, if the higher flow gradient has been met, then the order of ΔQ^n may rise to $\vartheta(1)$ [48]. Consequently, the approximate factorization method may carry the second-order temporal error. To minimize the factorization error for the discretized equations, controlling the CFL number of each cell is suggested in several reports [58-60].

For unsteady flows, it is suggested that the maximum CFL number should be restricted to unity.

4.1.3 Interpolation Error

An interpolation error is often introduced by most of the domain decomposition techniques. Both of the methods employed herein, namely, degenerate zonal and overlapped methods, rely on such interpolations between subdomains. Considering the overlapped grids, for example, the type of interpolation between the subdomains may be chosen such that the overall accuracy is about the same order as the discretization error [9-11]. To show this mathematically, consider an n-th order, one-dimensional, differential equation,

$$\frac{\partial^n E}{\partial \xi^n} = R(\xi) \tag{4.1}$$

To solve this problem on a composite mesh that consists of two overlapped subdomains *k* and *k'*, for which the indices are *i* and *j*, respectively, the N_1-*th* order accurate discrete system of equations can be written as follows:

$$\left(D_+D_-\right)^{n/2} q_i^k = R_i + \vartheta\left(\Delta\xi^{N_1+1}\right) \tag{4.2}$$

where $(q_i)^k$ is the numerical solution in subdomain *k*. The N_2-*th* order interpolation to the points in subdomain *k* from the points in subdomain *k'* can be represented in the following form:

$$q_i^k = \sum_j \alpha_{i,j}^k q_j^{k'} + \vartheta\left(\Delta\xi_{k'}^{N_2+1}\right), \tag{4.3}$$

where $(q_j)^{k'}$ is the numerical solution in subdomain k' and α denotes the interpolation. The goal, therefore, is to make N_2 as close to N_1 as possible.

It can be shown mathematically [9] that an interpolation formula with an overlap width given by $(0.25mN_1+1)$ is N_1-*th* order accurate, provided that the spatial step sizes are constant in the overlap region and that they are small compared to the width of the overlap region. Therefore, two sets of interpolation points and fringe points [62] are needed to maintain the spatial second-order accuracy in solving a second-order differential equation.

The temporal interpolation error for moving subdomain grids has not been fully studied. Therefore, only a brief preliminary study is shown here. For most cases, a second-order accurate method is deemed adequate enough to resolve the unsteady flow. However, the temporal interpolation error is often accentuated by the high flow gradients.

This point may be illustrated through the one-dimensional example given below. Two fringe points which belong to two different meshes (G_2 and G_3) coincide and receive data from a third mesh (G_1) simultaneously. Even if there were perfect interpolations of primitive variables to these two fringe points, the flow properties of these points may still propagate in different directions in the generalized Eulerian coordinate system. What also needs to be satisfied is the correct transfer of grid speeds from the subdomains. This can be shown from Eqs. (2.6) and (A.14); if the grids are moving, then, the eigenvalues (characteristic wave directions) of these points are rewritten as

$$\begin{aligned} \lambda_1 &= U &&= \partial_k \xi^m u_k + \partial_t \xi^m \\ \lambda_2 &= U + a_\infty &&= \partial_k \xi^m u_k + \partial_t \xi^m + a_\infty \\ \lambda_3 &= U - a_\infty &&= \partial_k \xi^m u_k + \partial_t \xi^m - a_\infty \end{aligned} \tag{4.4}$$

If the meshes G_2 and G_3 have different grid speeds, then, let the eigenvalues in G_2 and G_3 be checked in here. The first eigenvalue of Eq. (4.4) is given as

$$\lambda_1 = U = u_1\xi^1_{x^1} + \xi^1_t \tag{4.5}$$

and from Eq. (3.4) it can be applied to the special case

$$\xi^1_t = -x^1_\tau \xi^1_{x^1} \tag{4.6}$$

Then, substituting Eq. (4.6) into Eq. (4.5) yields

$$\lambda_1 = U = u_1\xi^1_{x^1} - x^1_\tau \xi^1_{x^1} = (u_1 - x^1_\tau)\xi^1_{x^1} \tag{4.7}$$

Assuming this interpolation point has the same primitive variable u_1 and Jacobian matrix $\xi^1_{x^1}=1$, then, it can be concluded that the $\lambda_1(G_2) \neq \lambda_1(G_3)$ because of $x^1_\tau(G_2) \neq x^1_\tau(G_3)$. Hence, the flow properties of this interpolation point which belongs to G_2 and G_3 as well, will propagate in two different directions, and this results in phase shift phenomena.

Another reason which may cause the small discrepancy is attributed to the deficiency of the numerical scheme. Most Eulerian frame, upwind methods [50] based on the Gudonov schemes (such as the Roe scheme used herein) are derived to solve the locally for one-dimensional Riemann problem for the hyperbolic Euler equations. Hence, the directions of numerical fluxes are assumed to be normal to the present cell surfaces. However, if there are two non-similar grid topologies overlapped with each other, then the flow properties in different meshes are most likely computed to propagate in different characteristic directions. Although some researchers [63-64] are currently studying generic multi-dimensional upwind methods, which may greatly reduce the numerical errors, a mature generic multi-dimensional upwind method is not available at this time.

4.2 Moving Shock Problem

A numerical experiment is performed to isolate and identify some of the consequences of each one of these error components, then, to demonstrate a few suggested measures to control them. The round-off error cannot be isolated, but it is responsible for almost the same amount of error in all the computations performed on a given computer. Hence, it can be considered as a non-variable with respect to the numerical cases. The classical problem of a moving normal shock inside a tube is studied for this purpose, since there is a time-dependent exact solution for the shock amplitude (to check the dissipation error) and shock speed (to check the dispersion error). Secondly, this problem has a large gradient region in the unsteady flowfield which reveals the error very easily and clearly [58, 65].

The upstream and downstream conditions of the normal shock are given in Fig. 4.1. Some of the computational details for the numerical cases are tabulated in Table 4.1. Cases 4.A.1-4 are computed on single grids, whereas, Cases 4.A.5-7 are computed on overlapped grids (Fig. 4.2). In Case 4.A.7, the smaller grid moves with the shock wave and it is overlapped on a larger static Cartesian grid. All the computations are started at the time when the shock reaches the axial position of x=0.4. The instantaneous pressure distributions (snap shots) along the centerline of the tube after 0.6 nondimensional time units are plotted in Fig. 4.3. At this moment, the exact values (obtained using the Rankine-Hugoniot relations) of the upstream pressure, downstream pressure, and shock location are 3.214, 0.714, and 1.600, respectively.

Case 4.A.2 is run on a grid that is about 3 times finer than the grid of Case 4.A.1 with all other features being identical (grid refinement). Since their pressure distributions are almost identical (Fig. 4.3a), it is concluded that the grid size of Case 4.A.1 (10,455 cells) is adequate to solve the current moving shock problems. Therefore, this grid is used for the rest of the cases. The difference between Cases 4.A.3 and 4.A.1 is their temporal accuracy; Case 4.A.3 is run first-order accurate in time by switching $\mu=1$ and $\nu=0$ in Eqs. (2.18). There is almost no amplitude error for both cases, but the shock location is underpredicted in Case 4.A.3. Hence, second-order accuracy in time is deemed necessary for an acceptable dispersion error. In order to demonstrate the effect of using a CFL number greater than unity when running Case 4.A.4, the global time step is increased such that the maximum CFL number is about 6.0. From Fig. 4.3a, it can be seen that there is almost no difference for the shock wave location. However, pressure oscillations prevent an accurate prediction of the amplitude. This dissipation error continues to deteriorate the solution as it progresses in time.

Cases 4.A.5 and 4.A.6 are computed on the composite grid shown in Fig. 4.2. Their difference is the number of fringe point sets used for the bilinear interpolations between the subdomain grids; in Case 4.A.5, two sets of fringe points are used to obtain nearly second-order accuracy (Eq. 4.3), whereas, only one set of fringe points is used in Case 4.A.6. It can be observed from Fig. 4.3b, that a smoother data transfer across the subdomain interfaces occurs in Case 4.A.5, thus better preserving the pressure amplitude within the overlapped subdomain, G_2. However, this does not appear to have an observable effect on the shock wave speed. It should be noted that a successful search for two sets of fringe points may not always be possible for configurations where the gaps between surfaces are small [12, 15].

A comparison of Cases 4.A.1, 4.A.5 and 4.A.7 are shown in Fig. 4.3c to demonstrate the accuracy of using static overlapped grids (Case 4.A.5 vs. Case 4.A.1) and kinematic overlapped grids (Case 4.A.7 vs. Case 4.A.1 and Case 4.A.5). All cases display a good comparison in shock speed and amplitude to the exact solution, despite some benign fluctuations at the overlap regions and at the immediate upstream of the shock. Hence, it is concluded that the interpolation error with two sets of fringe points, static or kinematic, is about the same order of magnitude as the discretization (dissipation and dispersion) error of the scheme.

The pressure contours and velocity vectors of Case 4.A.3 are shown in Fig. 4.4. In Fig. 4.4a, the computational shock wave behaves very close to the exact solution, which is a straight line wave front from the top of the tube to the bottom. However, it can be observed from Fig. 4.4b that there is a pressure fluctuation area around the shock wave crossing the overlap boundaries. This phenomenon can also be confirmed from the velocity fluctuation in the normal direction (Fig. 4.4c); the maximum normal velocity in the overlap boundary is 0.3 as compared with the freestream Mach number of 1.25. Theoretically, the normal moving shock inside a tube case should be a one-dimensional problem. Therefore, the pressure and velocity fluctuations in the normal direction are due to numerical errors. The source of this numerical error may be attributed to the differences in grid sizes: a finer Cartesian mesh is embedded inside a coarser Cartesian mesh. With the same grid topology but different grid sizes, the numerical slope for the shock wave that jumps across the discontinuity has two different values (Fig. 4.5). Therefore, the higher pressure (coarser mesh) pushes the flow into the lower pressure area (finer mesh), and causes the flow property fluctuations in the normal direction, then, the shock is smeared. However, the fluctuation is small and does not increase in time. Hence, this minor error is deemed acceptable.

4.3 Two Dimensional Store Separation From An Airfoil

To conduct a further study of numerical errors associated with moving overlapped grids, grids with different topologies are overlapped and moved in 3-DOF. The store separation from an airfoil is designed for this purpose, because this is a time-dependent phenomenon which can have large flow gradient regions with higher prescribed store speeds. Most of all, with the non-similar grid topology meshes, type of interpolation error can be detected in a careful designed case. Also, the comparison of second-order (4*4 block inversion) and diagonal inversion methods will be presented in this section.

The geometry, freestream conditions, and velocity vectors of the airfoil and store assembly are given in Fig. 4.6. The geometry of the airfoil is NACA-64A010, and the store is idealized as ogive-cylinder-ogive store which is in the carriage position (0.7 diameter away from the airfoil). This wing and store assembly experience the freestream Mach number 0.95 from the front, and the freestream temperature is 467.5°R. Some of the computational details for the numerical cases are tabulated in Table 4.2. The composite grid of Cases 4.B.1 and 4.B.2 consists of two meshes; a smaller O grid wrapped around the store and overlapped on a C grid generated for the airfoil. The radius of the C grid is 10 chord length (85 diameters), and the radius of the O grid is approximately 3d. The composite grid of Cases 4.B.3 and 4.B.4 also have an added Cartesian intermediate grid (Fig. 4.7) between the C and O grids to enhance the interpolation of the overlapped scheme. The size of the Cartesian intermediate grid is 24.7d*18.7d. The grid size of these meshes are given in Table 4.2. For these four cases, the intentionally exaggerated prescribed motion of the store is defined by the following equations:

$$\begin{aligned} x &= v_x t = 0.1 \times t \\ y &= v_y t = -0.3 \times t \\ \theta &= \omega_z t = 0.1 \times t \end{aligned} \tag{4.8}$$

Starting from initial freestream conditions, the computation is marched with pseudo time steps to obtain a converged solution for a steady state flow. Then, the computation is continued with physical time steps, that is, for a given time step, all the cells are advanced using a constant time step value.

The pressure contours of the steady state solution of Case 4.B.1 are shown in Fig. 4.8. It should be noted that all the contour figures in this dissertation are generated by 'PLOT3D' [66] which is not capable of interpolating data across the overlap boundary. Therefore, the discontinuous contour lines across the overlapped grids can be observed. This problem is more pronounced if the cell centers of different subdomains in the overlap boundary are further away from each other. The steady state C_p distributions on the airfoil and store are shown in Figs. 4.9. The steady state C_p distributions of Case 4.B.2 are identical to Case 4.B.1, hence, the C_p distributions for Case 4.B.2 are not be presented in this report. In other words, there is no difference between Cases 4.B.1 and 2 at steady state, and these results can also be confirmed by Pulliam's [51] mathematical prediction. The instantaneous pressure contours of Cases 4.B.1 and 2 after four time units of prescribed motion are presented in Figs. 4.10a and b. It is observed that the flow structures of these two Cases are quite similar. The more detailed quantitative representation of C_p distributions, are given in Figs. 4.11. It is noticed that these C_p values on the airfoil (Fig. 4.11a) are almost identical, especially the shock's location and amplitude are almost overlaid for these two cases. However, the C_p difference of upper and lower surfaces of the store in Case 4.B.1 is smaller than Case 4.B.2, but the higher flow gradient locations are still overlaid with each other. Consequently, it is demonstrated that two time-accurate methods, which produce almost identical steady state solutions, may have differences for the instants of their unsteady solutions. The reason that the higher flow gradient locations still be the same for Figs. 4.10-4.11 may be attributed to the location and position of the moving store. However, the position of the moving store is decided by prescribed motion equations.

An intermediate Cartesian grid is added in Cases 4.B.3 and 4. Note that the fringe points for meshes G_2 (intermediate) and G_3 (store) are very close to each other in the lower surface of the airfoil (Fig. 4.12). The purpose of this design is to amplify the interpolation error in the high flow gradient region. Shown in Fig. 4.13a are the steady state pressure contours of Case 4.B.3, which are identical for both Cases 4.B.3 and 4. Despite the additional grid, the C_p distributions on the airfoil and store surface are very similar for Cases 4.B.1 and 3 (Fig. 4.9). Also, the C_p distributions of Cases 4.B.3 and 4 are identical for the Cases 4.B.1 and 2. However, in the close up contours (Fig. 4.13b), one may observe a slight mismatch of the shock waves that emit from the tail of the store, pass through the area of fringe points, and merge with the shock that emits from the lower tail of the airfoil. These discrepancies are not observed in Cases 4.B.1 and 2 (Fig. 4.8), and the only difference between Cases 4.B.1 and 3 is the intermediate mesh. Hence, this suggests that these discrepancies are caused by the fringe points from non-similar grid topologies. However, these discrepancies decrease during the pseudo-time marching to the steady state solution. The pressure contours of Cases 4.B.3 and 4 after 4 time units of store motion are shown in Figs. 4.14 and 4.15, respectively. It can be noticed that the pressure contours for Cases 4.B.3 and 4 are still very similar to each other, however, the small mismatch of the shock in Fig. 4.13b does not exist anymore. The reason may be attributed to the rigid-body motion that moves the overlaid fringe points region away from the shock. Consequently, the interpolation error is reduced. The instantaneous C_p comparison of Cases 4.B.3 and 4 after 4 time units are shown in Fig. 4.16. Some minor C_p differences can be seen in Fig. 4.16a. From the comparison of Fig. 4.16b, it can be noticed that the diagonal inversion method still produces more C_p variation on the upper and lower surface than the block inversion method.

4.4 Validation of Trajectory Prediction

The present 6-DOF trajectory method [36] is tested as follows. A three-dimensional wing-store configuration (Fig. 4.6) reported in [67] has been wind-tunnel tested during the separation of the store from the wing in the quasi-steady mode; that is, as a sequence of steady-state measurements of the positions on a trajectory, which was computed based on the steady measurements of the forces and moments. The experimental configuration consists of a clipped delta wing with 45 degrees of leading edge sweep and a NACA-64A010 airfoil section. Connected to this wing is an ogive-flat plate-ogive pylon, which is located 0.7 diameter above an ogive-cylinder-ogive store when in carriage position. Some of the parameters for the store dynamic are given in Table 4.3. Also, the freestream conditions are M_∞=1.2, ρ_∞=0.5564, and T_∞=430.6°R, respectively. The experimentally measured forces and moments were also used by the present method as the input and the trajectory were computed. Then, the present trajectory was compared with the trajectory reported in [66] for the wind-tunnel tests (Fig. 4.17). The axis system used in the above study is defined with the x-axis pointing forward, y-axis pointing upward, and z-axis pointing outboard. Three of the translational components and two of the rotational components compared very well. The trend of the third rotational component was also in agreement; however, the magnitudes differed by a maximum of 3°. The discrepancies in the results can be attributed to the fact that the ejector characteristic were not computationally simulated. In the present case the ejector force was applied until the store dropped to a certain specified distance, after which it was allowed to perform a 6-DOF motion.

Table 4.1: Computational details for moving normal shock cases.

Case	Grid	Grid size	D³T	Set of fringe point	Time accuracy	Δt (ms)	Δx_{min}	$(CFL)_{max}$	CPU time per step per cell (μs)	Computer memory (MW)
4.A.1	Single Cartesian	205*51	No	N/A	$\vartheta(\Delta t^2)$	1.0	.01	0.6	5.3	0.8
4.A.2	Single Cartesian	301*101	No	N/A	$\vartheta(\Delta t^2)$	0.5	.005	0.6	5.2	2.3
4.A.3	Single Cartesian	205*51	No	N/A	$\vartheta(\Delta t)$	1.0	.01	0.6	5.2	0.7
4.A.4	Single Cartesian	205*51	No	N/A	$\vartheta(\Delta t^2)$	1.0	.01	6.0	5.3	0.8
4.A.5	Two Cartesian	205*51 +141*51	static overlapped	2	$\vartheta(\Delta t^2)$	1.0	.007	0.8	5.7	1.4
4.A.6	Two Cartesian	205*51 +141*51	static overlapped	1	$\vartheta(\Delta t^2)$	1.0	.007	0.8	5.6	1.4
4.A.7	Two Cartesian	205*51 +141*51	moving overlapped	2	$\vartheta(\Delta t^2)$	1.0	.007	0.8	11.0	1.5

Table 4.2: Computational details for a 2-D store separation from an airfoil.

Case	Grid size	Matrix inversion method	CPU/per step /per grid (μs)	memory (MW)
4.B.1	7479+3201	4*4 block	43.8	0.6
4.B.2	7479+3201	diagonal	28.2	0.6
4.B.3	7479+3969+3201	4*4 block	53.7	0.8
4.B.4	7479+3969+3201	diagonal	39.3	0.8

Table 4.3: Parameters for store dynamics.

Eject force	Mass	Weight	I_{xx}	I_{yy}	I_{zz}
53416.71 Nt	905.2 Kg	8880.0 Nt	27.13	488.28	488.28

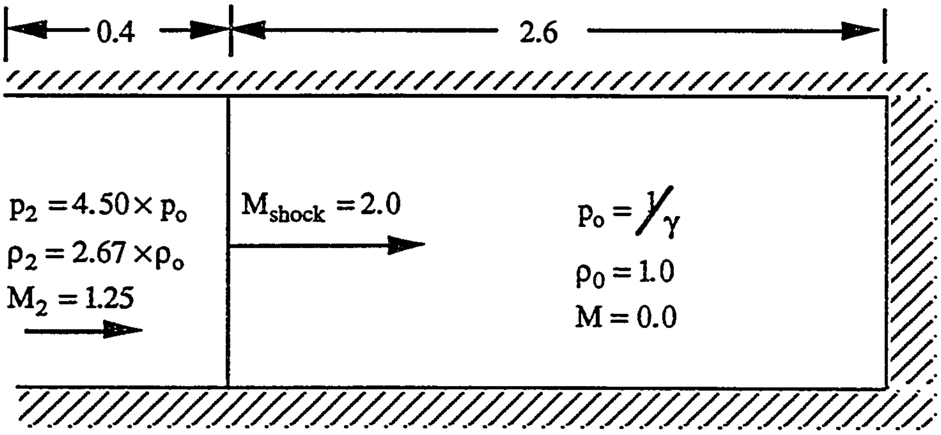


Fig. 4.1: Schematic for moving normal shock.

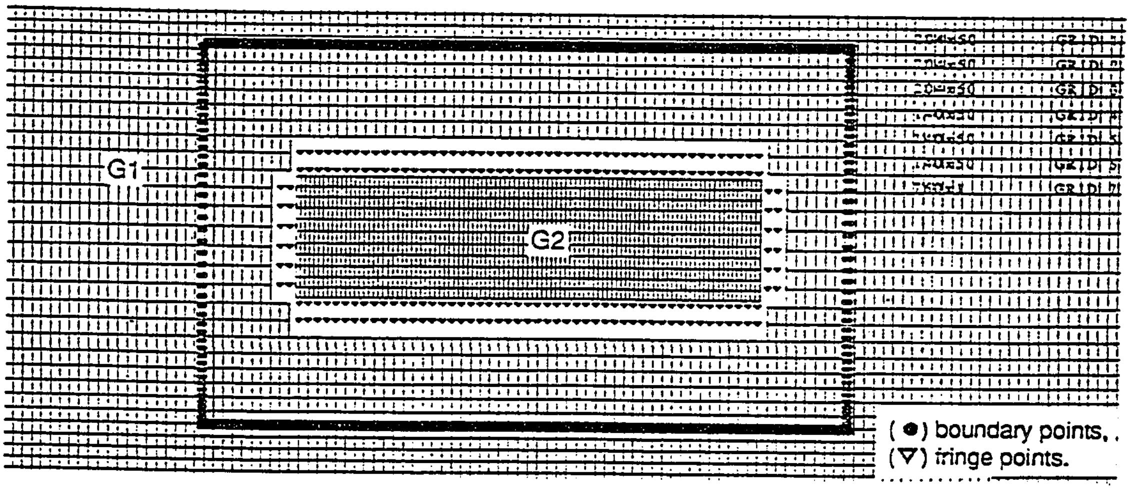


Fig. 4.2: Interpolation points for the normal moving shock.
G_1 : global grid
G_2 : overlapped grid.

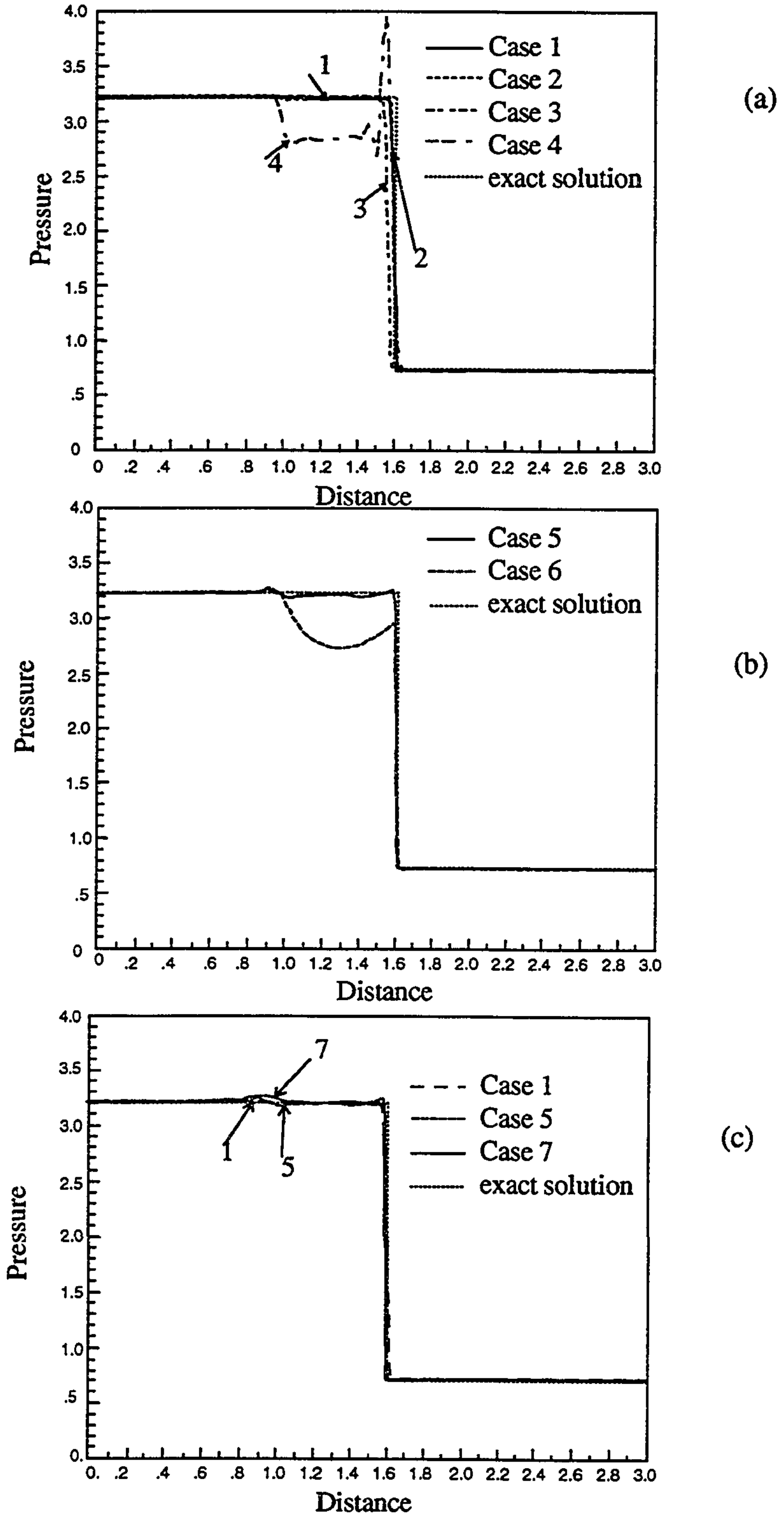


Fig. 4.3 : Comparison of nondimensional pressures for moving normal shock cases. (a) Cases 4.A.1, 4.A.2, 4.A.3 and 4.A.4; (b) Cases 4.A.5 and 4.A.6; (c) Cases 4.A.1, 4.A.5 and 4.A.7.

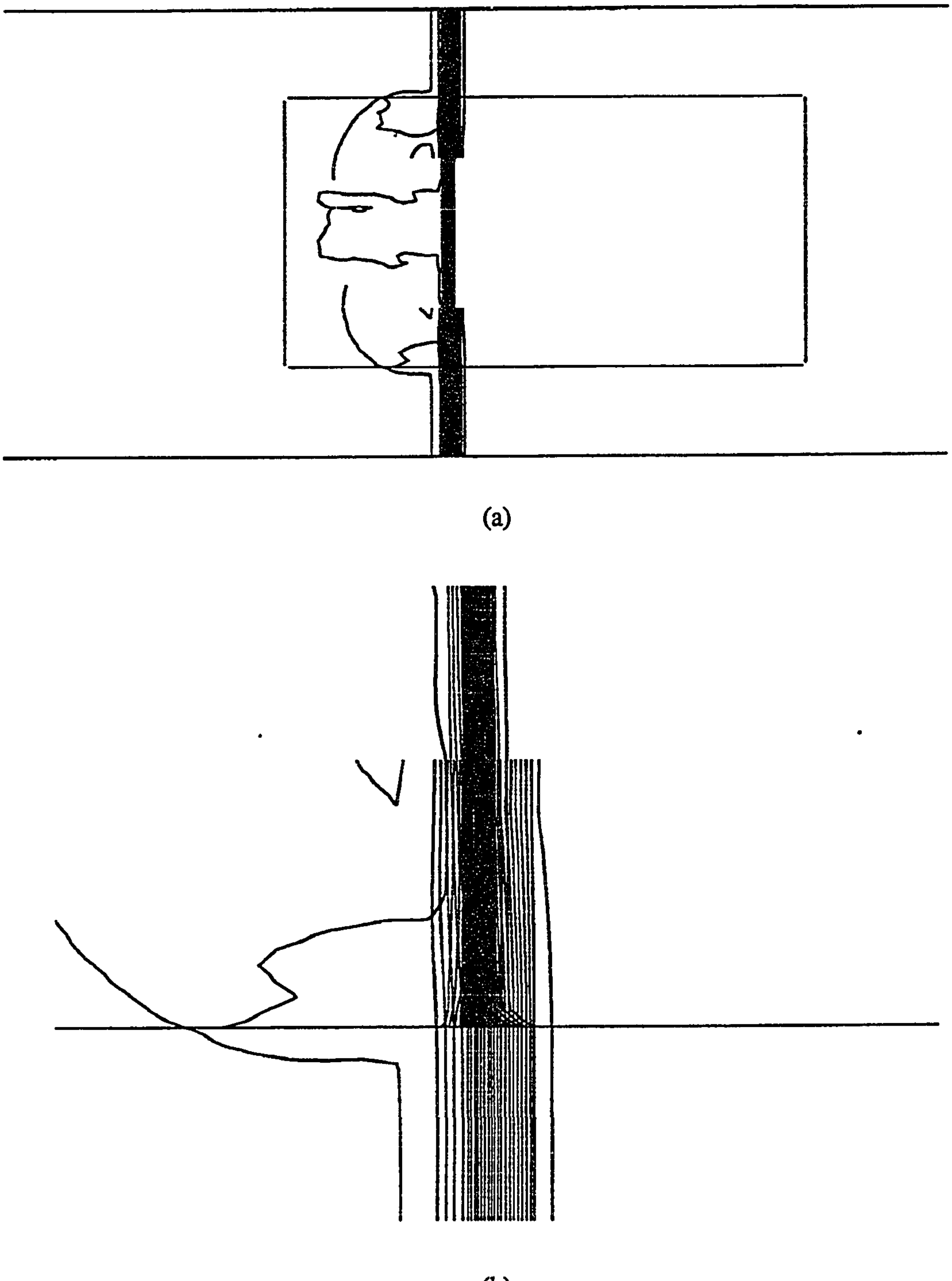
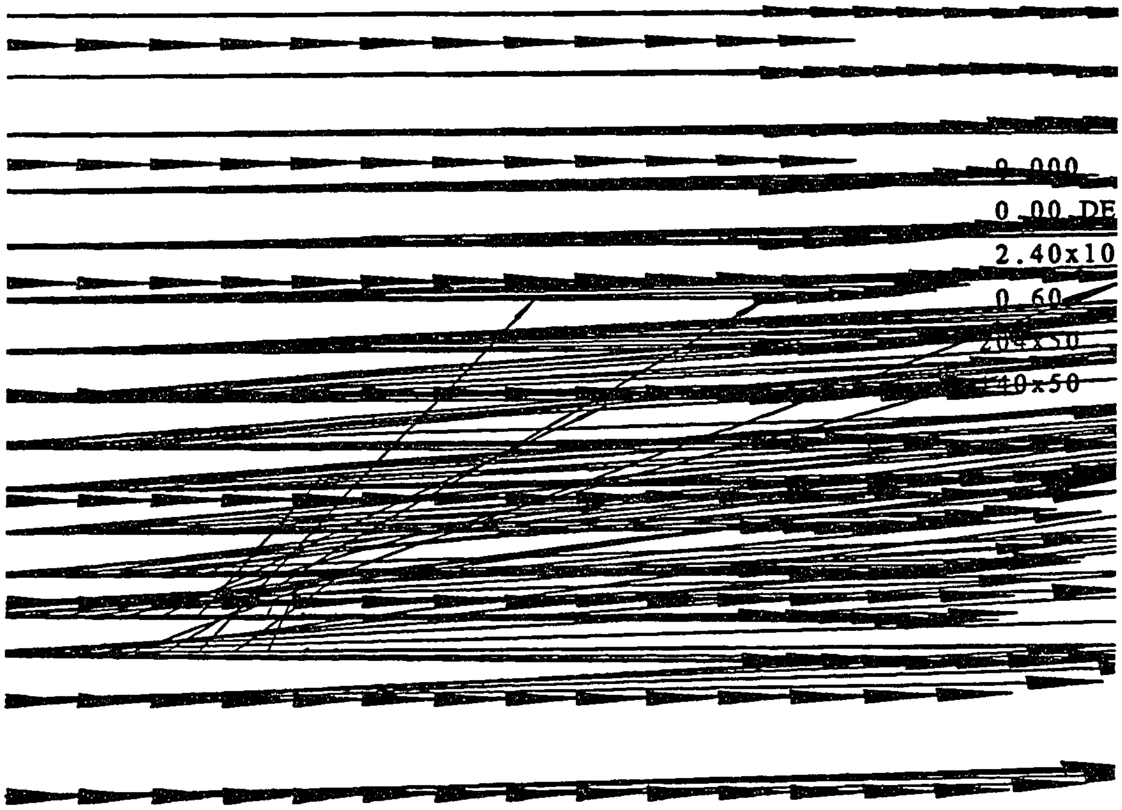


Fig. 4.4: Normal moving shock at t=0.7 for Case 4.A.3,
(a) pressure contours, (b) close up of shock corssing outer overlap boundary,
(c) velocity vectors of shock region crossing outer overlap boundary.



(c)

Fig. 4.4: Concluded.

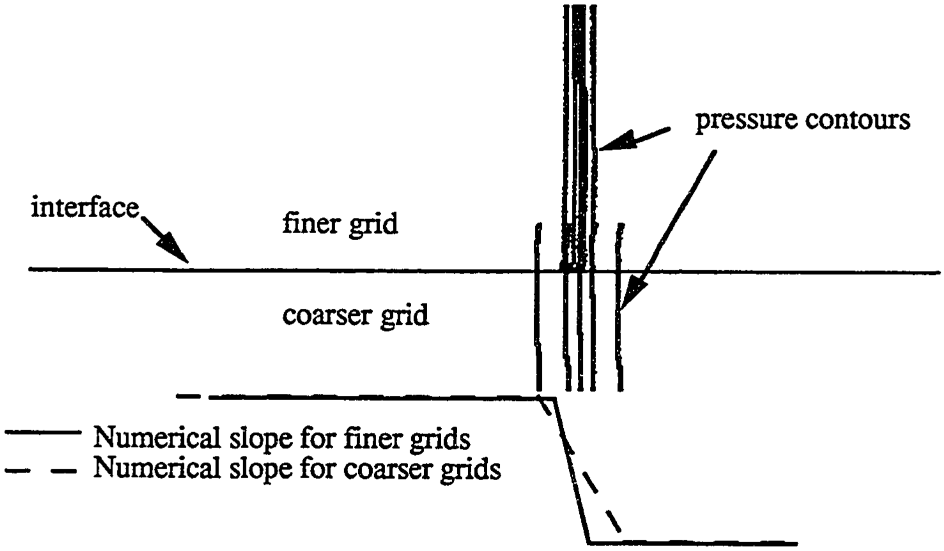


Fig. 4.5: Numerical shock slope at overlap boundary: Cartesian grids with different sizes.

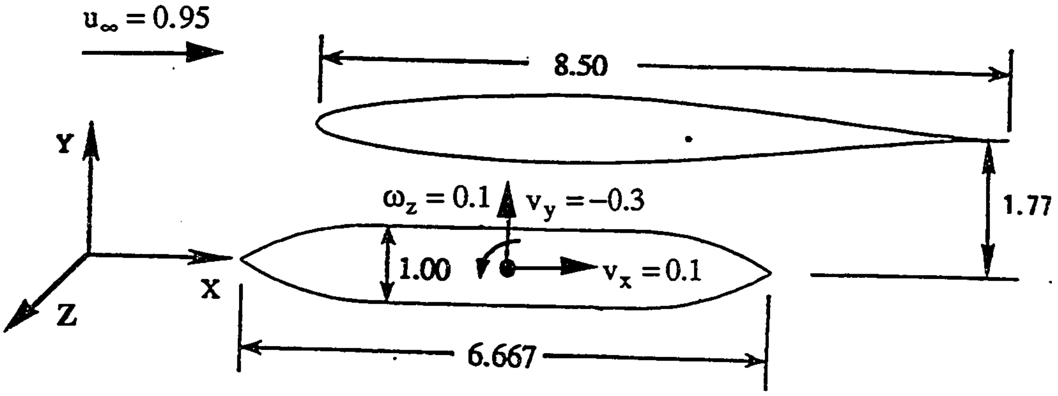


Fig. 4.6: Schematic and geometry of the airfoil and store.

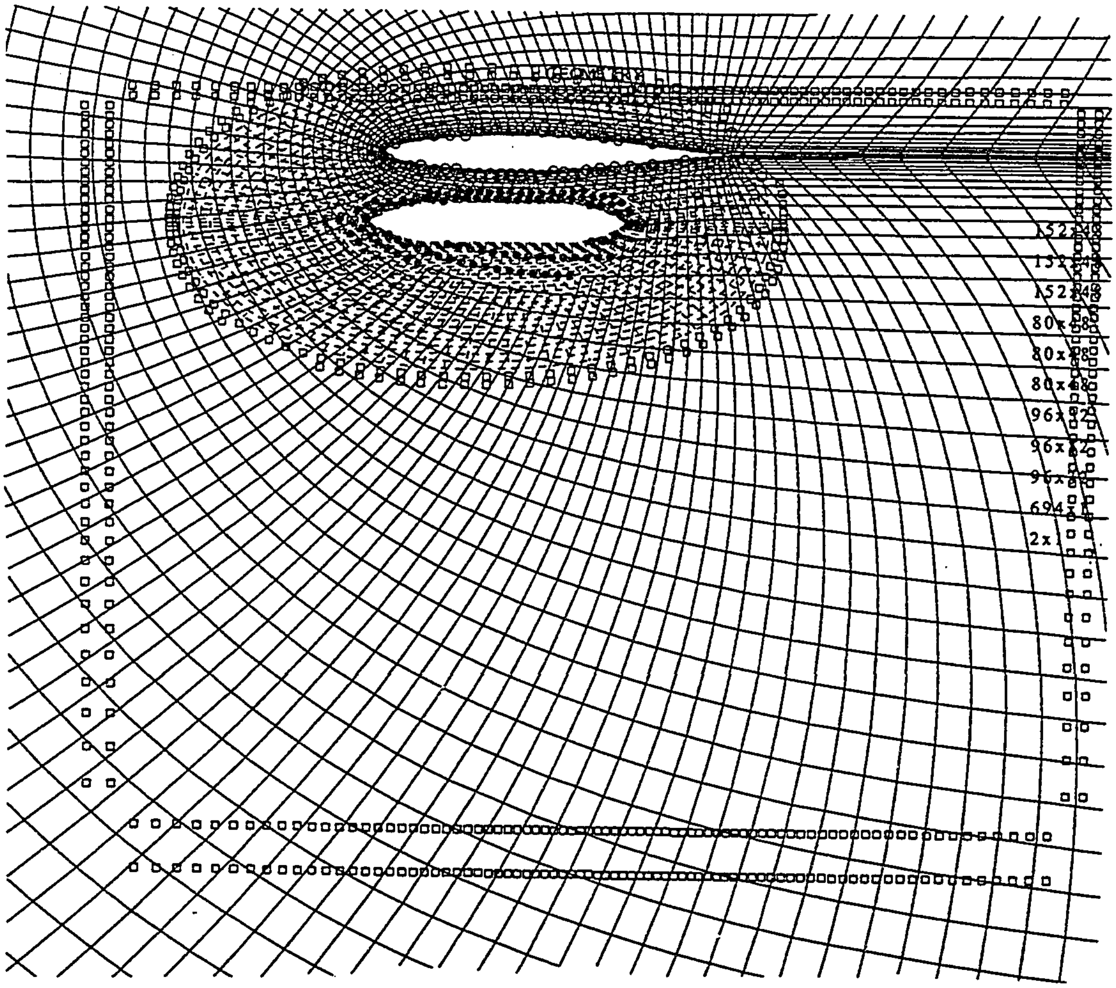


Fig. 4.7: Interpolation points for the airfoil and store grids.

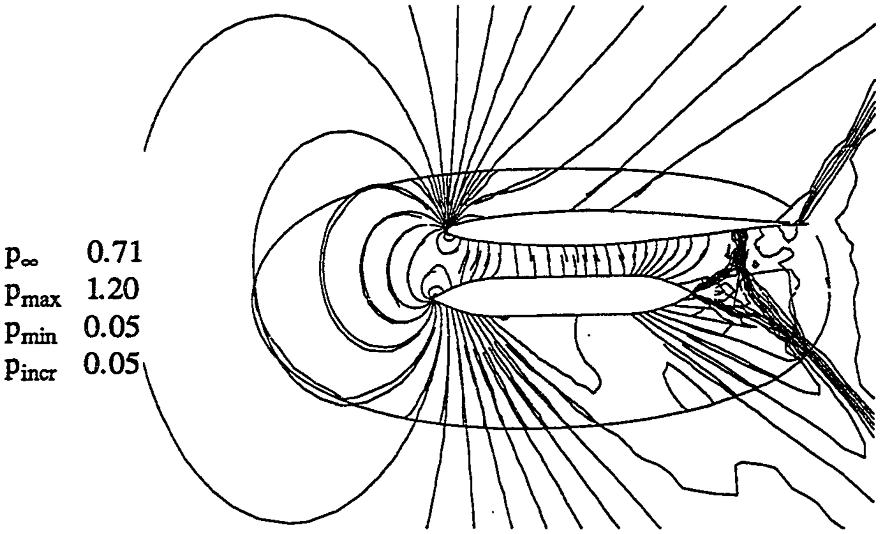


Fig. 4.8: Pressure contours of airfoil and store at steady state for Case 4.B.1.

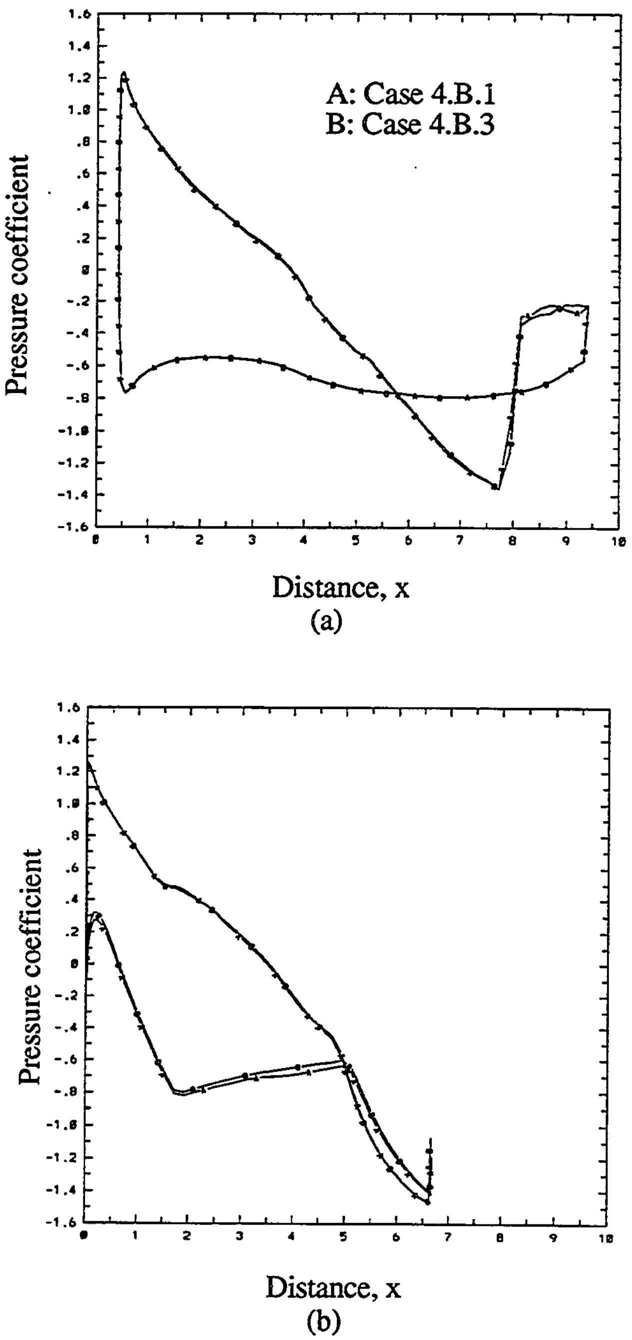
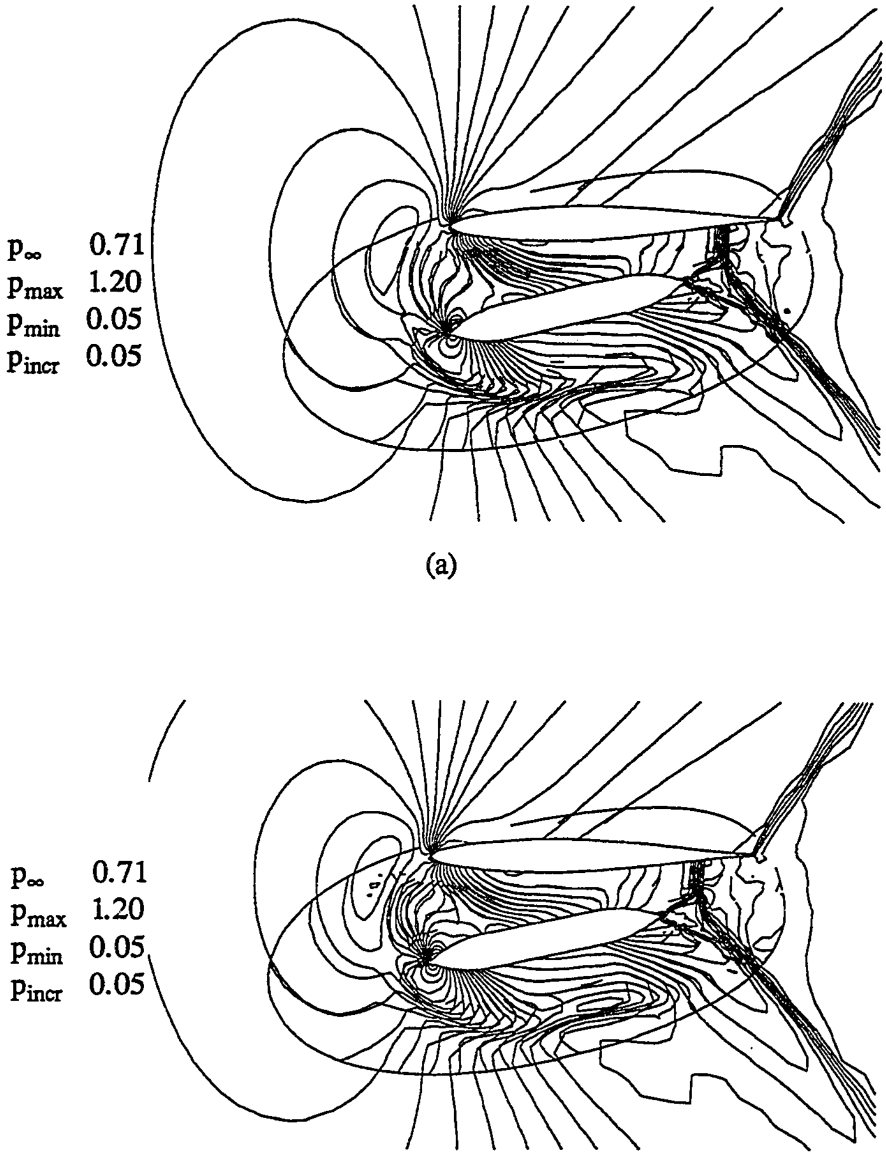


Fig. 4.9: C_p distributions at steady state for Cases 4.B.1 and 4.B.3,
(a) on the airfoil surface,
(b) on the store surface.



(a)

(b)

Fig. 4.10: Pressure contours at 4 time units for
(a) Case 4.B.1, block inversions
(b) Case 4.B.2, diagonal inversions.

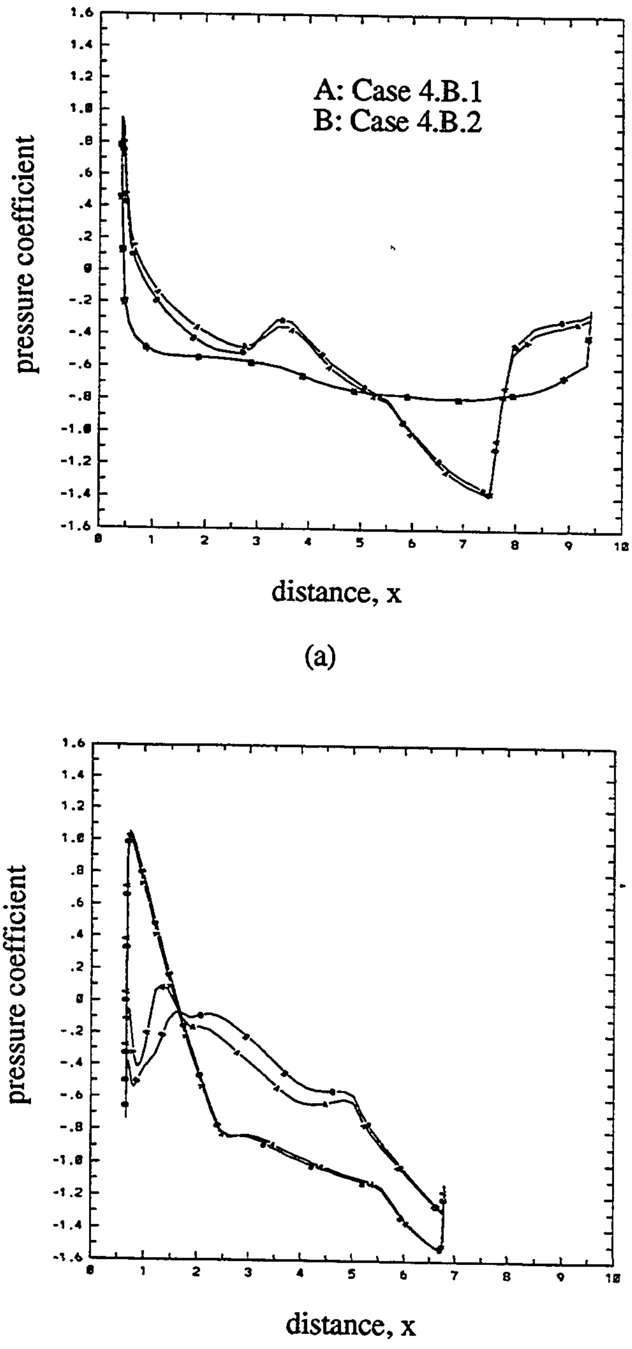


Fig. 4.11: C_p distributions at 4 time units for Cases 4.B.1 and 4.B.2,
(a) on the airfoil surface,
(b) on the store surface.

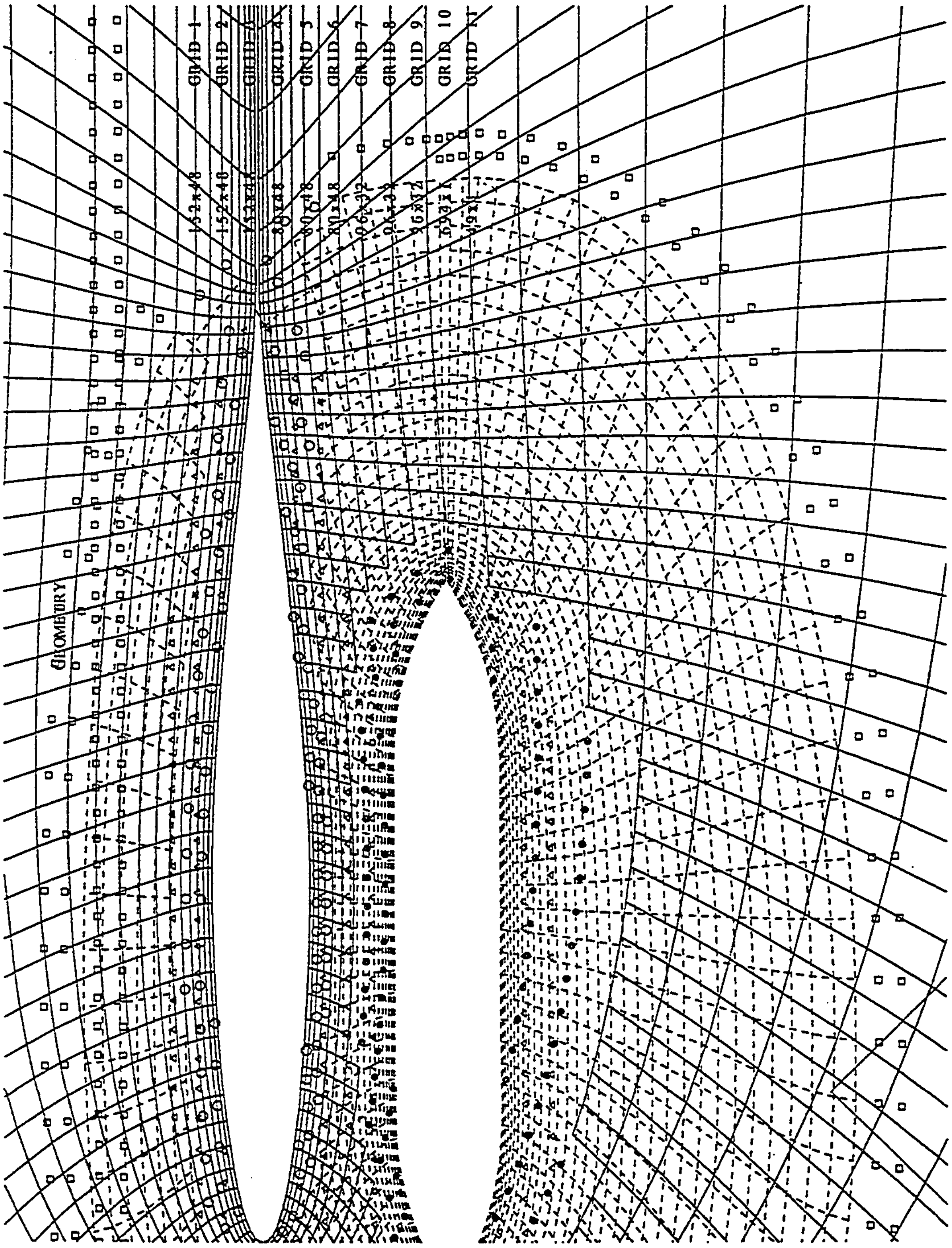
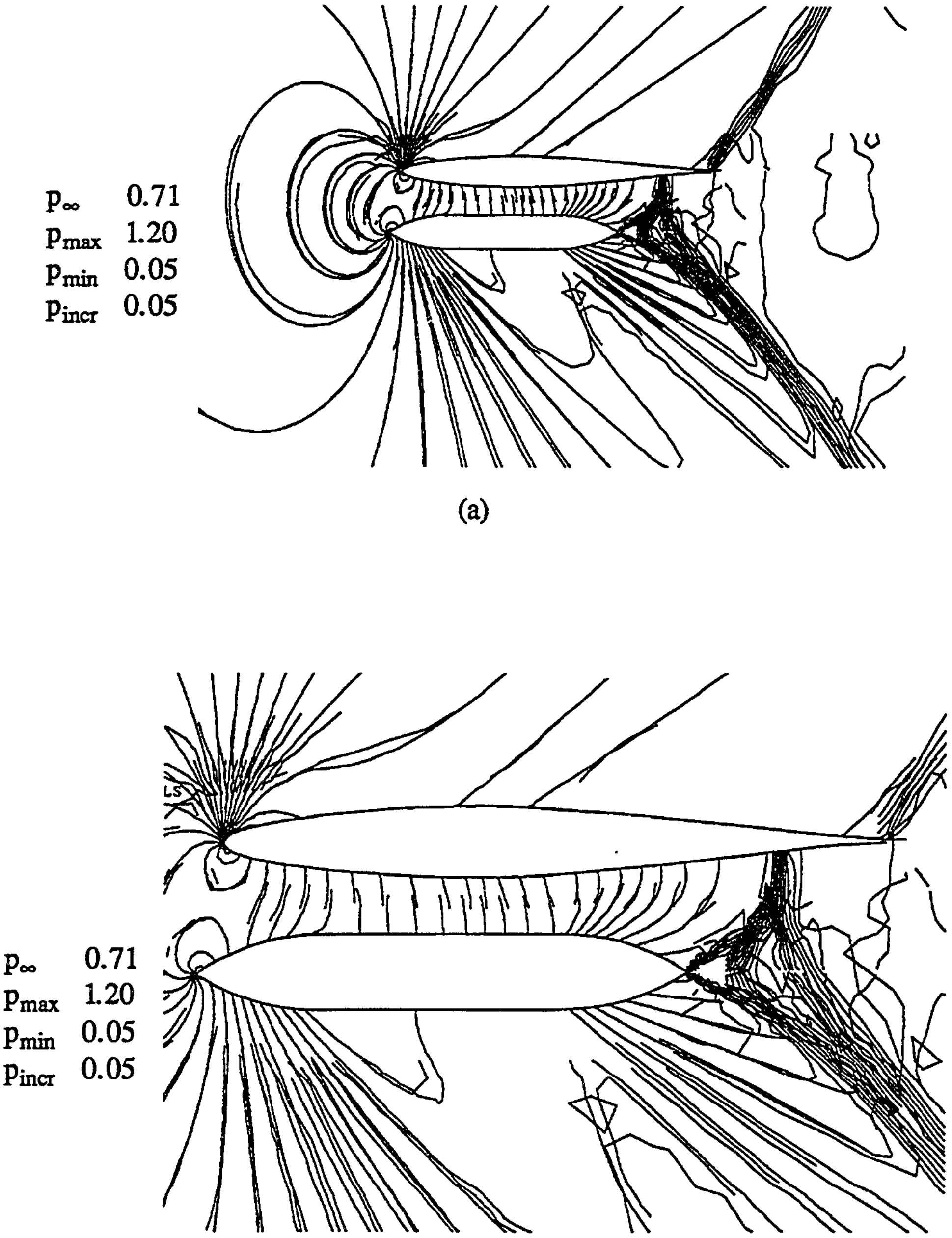
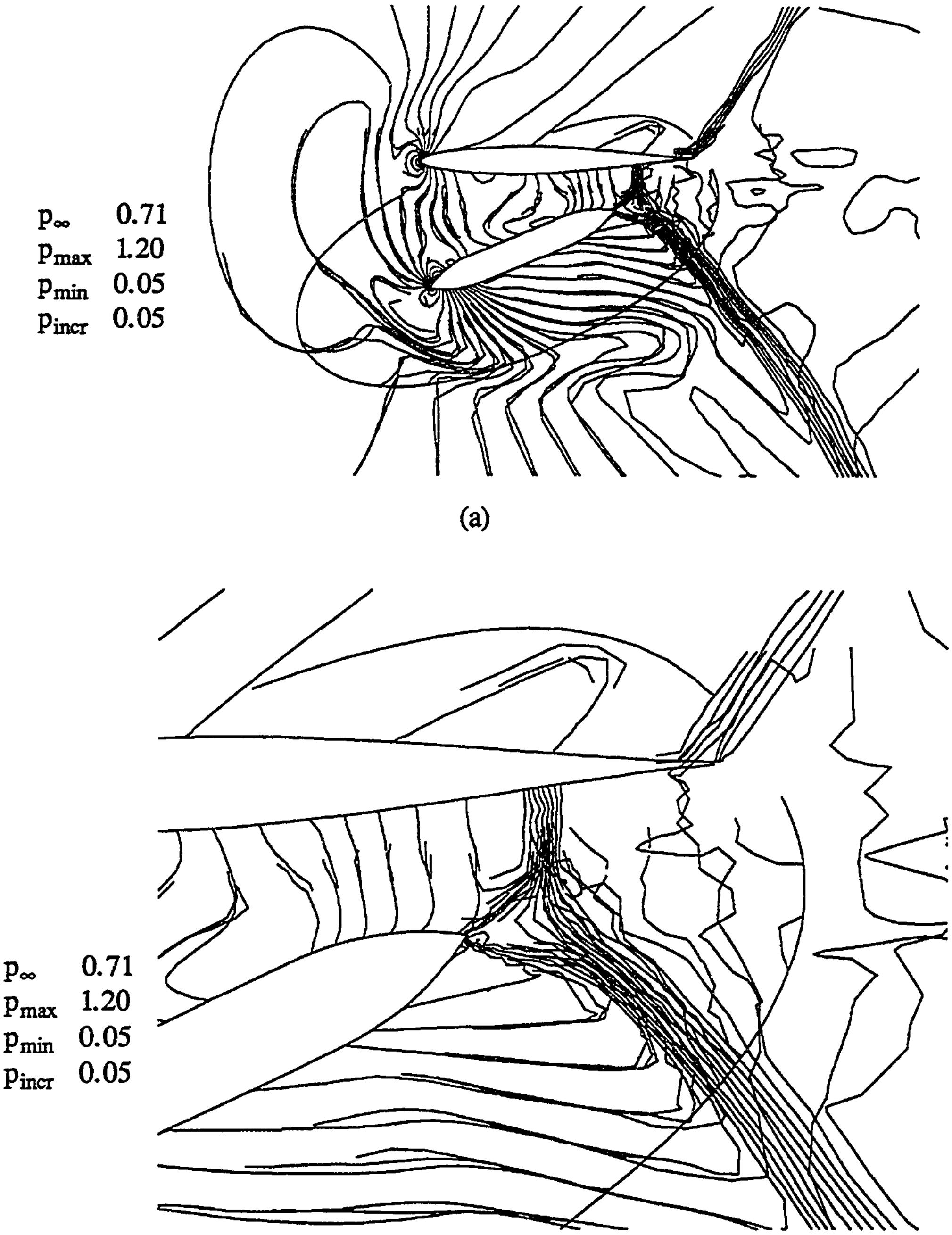


Fig. 4.12: Fringe points for the airfoil, store, and intermediate grids.
(•) symbols indicate fringe points for airfoil grid,
(Δ) symbols indicate fringe points for intermediate grid,
(O) symbols indicate fringe points for store grid.



(b)

Fig. 4.13: (a) Pressure contours at steady state for Case 4.B.3, (b) close up of shock-interaction region.



(a)

(b)

Fig. 4.14: (a) Pressure contours at 4 time units for Case 4.B.3, (b) close up of shock interaction region.

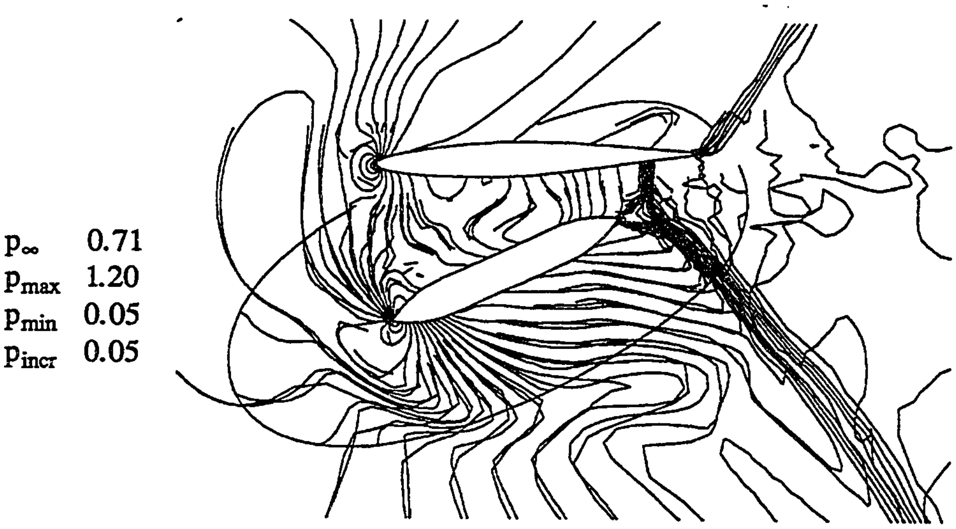
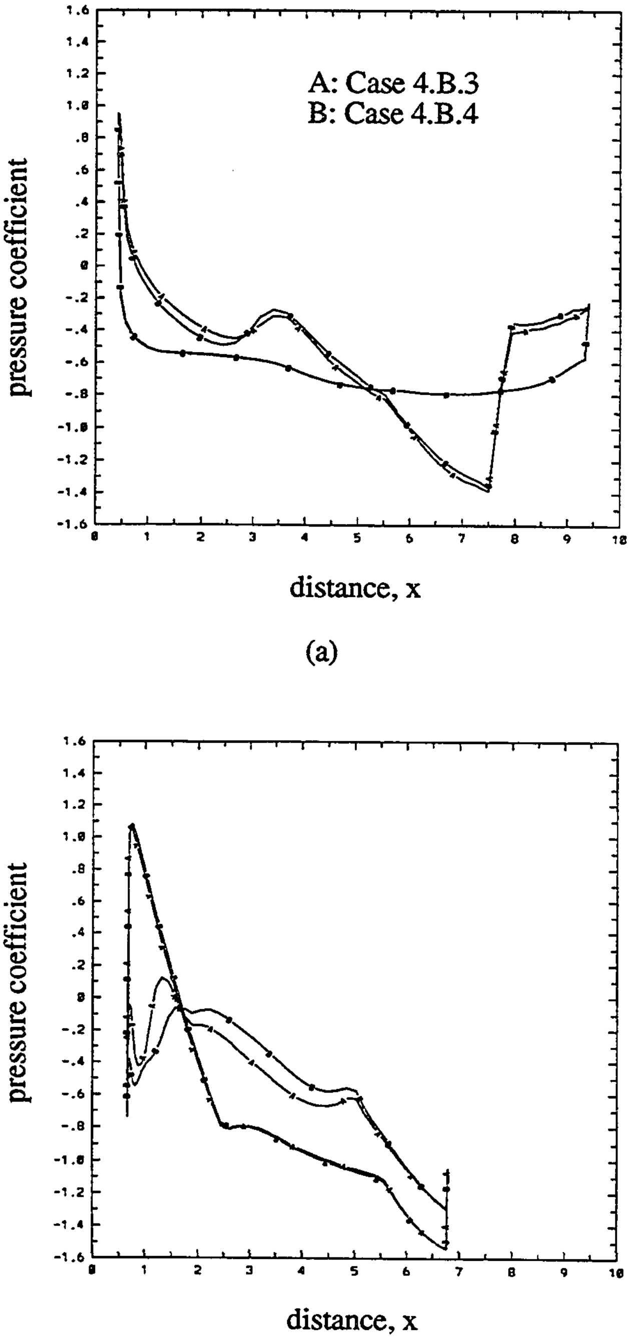


Fig. 4.15: Pressure contours at 4 time units for Case 4.B.4.



(a)

(b)

Fig. 4.16: C_p distributions at 4 time units for Cases 4.B.3 and 4.B.4,
(a) on the airfoil surface,
(b) on the store surface.

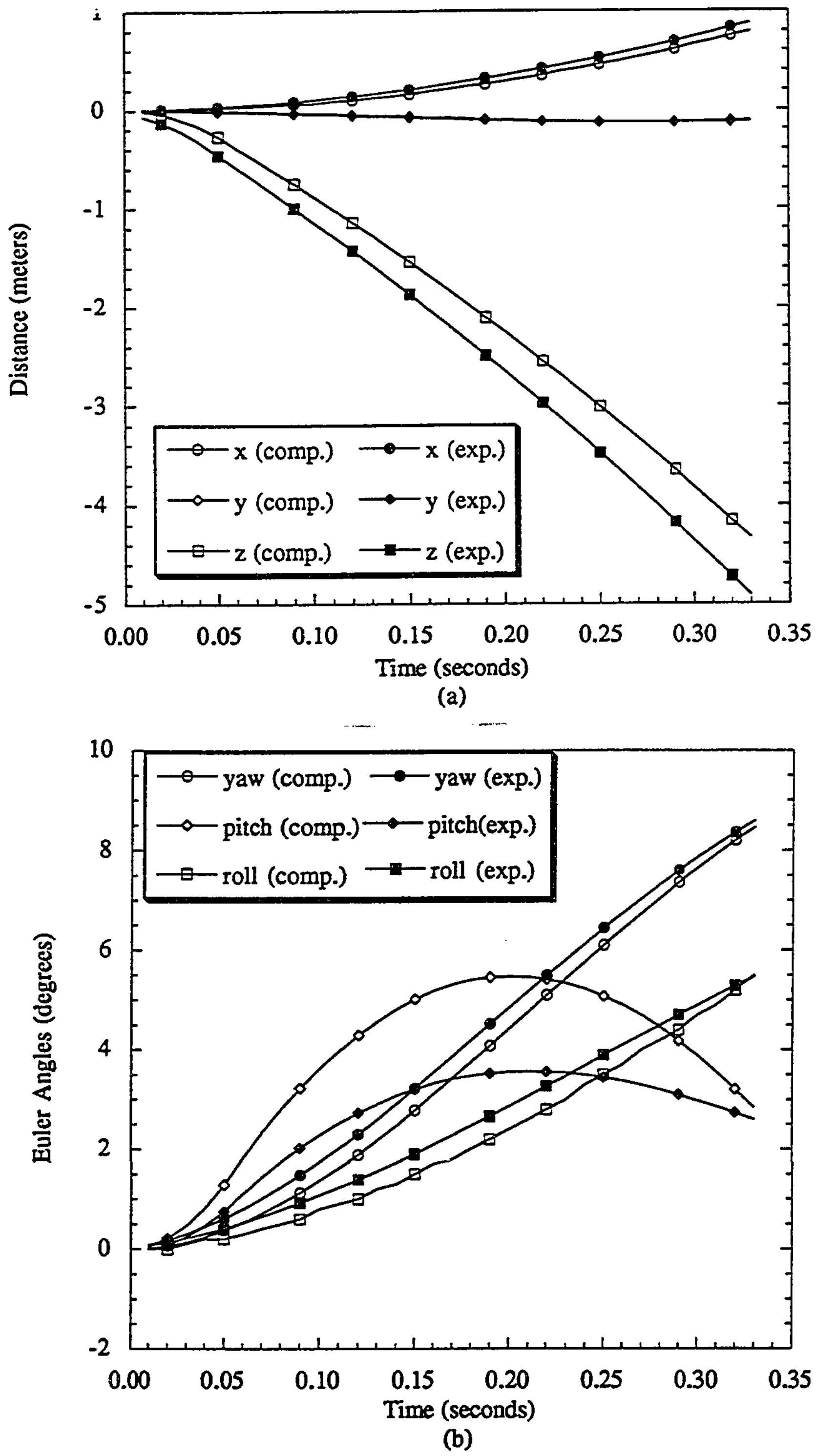


Fig. 4.17: Trajectory comparison, (a) displacements; (b) Euler angles [36].

Chapter 5

UNSTEADY AIRFOIL FLOWS

5.1 Presentation of the Cases

In this chapter, D^3T is demonstrated through a transonic flow past an airfoil, which experiences a combined motion of pitching and plunging [55-56].

The demonstrative cases presented in this chapter are for an airfoil engaged in a prescribed motion of pitching and plunging. The composite grid generated for these cases is shown in Fig. 5.1. A stretched O-grid (117*29) is wrapped around the airfoil and overlapped on a fine Cartesian grid (41*41). Then, the fine grid is embedded in a coarse Cartesian grid (49*49), which is 24C by 24C. For convenience, the subdomain grids are denoted as G_1, G_2, G_3. The absolute frame of coordinates are defined in G_1 which is also space fixed. G_2 and G_3 are free to move relative to each other and G_1. Grid G_1 is space-fixed in order to accommodate a conceivably stationary component of a configuration; for example, the aircraft in the case of store separation, or the ground in the case of low-altitude flight. Also, G_1 is relatively coarser to cover the farfield with fewer cells. Topologies of grids G_2 and G_3 are nonsimilar. Grid G_2 is needed for two reasons: first, it is a relatively finer grid with cell sizes comparable to those of G_3 as needed by the overlapping algorithm, hence a fine grid is used only where needed; secondly, grid G_2 experiences only the plunge motion and the conservative information transfer between the degenerate-zonally embedded G_2 and G_1 is much easier than the overlapping algorithm.

D^3T is demonstrated through a transonic flow past an airfoil, which experiences a combined motion of pitching and plunging. In order to establish the accuracy of the method, however, three building block cases precede. The description of the computational cases are given in Table 5.1.

5.2 Results and Discussion

The first case is the transonic flow past a sinusoidal pitching airfoil. This prescribed motion is described by the following time function for the angle of attack,

$$\alpha(t) = \alpha_0 + \alpha_1 \sin(M_\infty kt), \qquad (5.1)$$

where the amplitudes are α_0=4.86° and α_1=2.44°. The reduced frequency is k=0.162 radians. Since an object with a simple geometry is engaged in a simple motion, this is a good benchmark case to consider in developing and testing all the time-dependent terms of the governing equations for pitching. The flow domain is discretized by a single O-grid which has a 24C diameter, where C is the airfoil chord. Shown in Table 5.2 are the number of cells in this grid, computational time step size, number of computational steps needed for one cycle of the motion, and microseconds of computer CPU time used to simulate one cycle of the motion. Similar information for the other cases are also listed in Table 5.2.

The same flow is simulated using D^3T as the second case (Fig. 5.1). Grid G_3 is fixed to the airfoil, hence it is also pitched sinusoidally. However, grids G_1 and G_2 are fixed in space. The initial solution is obtained for the steady flow at α=4.86° using local time steps and the multigrid convergence acceleration. Then, the computations are performed time accurately for three cycles. The results of the second and third cycles are identical indicating that the periodic flow is captured. The solution for the first cycle is different than the following cycles, which indicates the onset of the boundary-motion-induced flow. As expected, Case 5.2 requires more computer time than Case 5.1. The extra time is mainly needed for the D^3T connection and information transfer operations. A small portion of this extra time is due to the decreased vector processing efficiency since the subdomain grids of D^3T (Case 5.2) have shorter data vector lengths than those of a single global grid (Case 5.1).

There exist experimental surface pressure data [68] for a transonic flow past a NACA-64A010, which goes through a cosine plunge motion described by,

$$X_3 = X_{3,0}\cos(2kt), \tag{5.2}$$

where the amplitude is $X_{3,0}=0.02$, and the reduced frequency is $k=0.2$ radians. The quantity (2kt) is the reference angle. This flow is simulated here as Case 5.3 in order to develop and test the time-dependent terms of the governing equations for plunging. The composite grid for Case 5.3 is similar to that of Case 5.2. Grids G_2 and G_3 are fixed to the airfoil, hence they plunge with the airfoil. Grid G_1 is still space-fixed. The local time steps and the multigrid convergence accelerates, then the solution is obtained time accurately for three cycles for the same reasons given for Case 5.2. The amplitude of this cosine plunging is $\pm 0.02C$. Some computational details for this case are given in Table 5.2.

Case 5.4 is the simulation of a transonic flow past an airfoil, which experiences a combined motion of a constant-rate plunging and sinusoidal pitching, that is Eq. (5.1) with $\alpha_o = 0°$. The upstream flow conditions are identical to those of Cases 5.1 and 5.2. The rate of plunge, M_p, is determined as the vertical component of the freestream velocity approaching at an angle of attack (Fig. 5.2),

$$M_p = M_\infty \sin(\alpha_2), \tag{5.3}$$

where α_2=4.86°, which is equal to the value of α_o (Eq. (5.5)) for Cases 5.1 and 5.2. Therefore, the airfoil is plunged down 3.286 chord lengths during one cycle of the sinusoidal pitching. The effect of constant-rate plunge in Case 5.4 is compensated by the initial amplitude of the angle of attack (4.86°) in Cases 5.1 and 5.2. The flowfield of Case 5.4, therefore, differs from the flowfield of Cases 5.1 and 5.2 by only the plunge-motion-induced flow component. All of the flows include the pitch-motion-induced flow component.

The composite grid of Case 5.4 is identical to that of Case 5.2. Grid G_1 is again space-fixed and grid G_3 pitches sinusoidally. In addition, the assembly of grids G_2 and G_3 plunges. The initial solution is obtained for the steady flow at α=0° using local time steps and the multigrid convergence acceleration. Then the computations are performed time accurately for two cycles ($\alpha = 0° \pm 2.44°$, X_3=0 to 6.56 C). Some computational information for this case is given in Table 5.2. Since this case requires the largest number of D^3T operations, it is computationally the most expensive one. Case 5.1, naturally, does not need any dynamic overlapped grid operations. Case 5.3 has static overlapped grid operations and dynamic zonal grid operations. Case 5.4, however, has dynamic operations for the zonal as well as overlapped grids.

The accuracy of D^3T in simulating a flow past an oscillation airfoil is studied by comparing the results of Cases 5.1 and 5.2, i.e. with and without using D^3T, as well as comparing with the experimental data. The instantaneous Mach number contours at $\alpha = 6.97° \uparrow$ ($\uparrow$ indicates pitching up motion) of Cases 5.1 and 5.2 (Figs. 5.3-5.4) show almost no detectable differences. The flow expands around the leading edge from the stagnation point to form a supersonic pocket with a maximum Mach number of 1.45. The supersonic pocket is terminated by a shock. The wake starts with a small separation region on the upper surface close to the trailing edge and relatively lower Mach numbers are visible in this region.

Instantaneous surface pressure coefficients at angles of attack 6.57° ↓, 5.11° ↓, 3.49° ↓, and 6.97° ↑ for Case 5.1 and 5.2 are given in Fig. 5.5. The comparison is favorable with the experimental data [69]. Also, the instantaneous surface pressure coefficients at angles of attack 6.57° ↓, 5.11° ↓, 3.49° ↓, and 6.97° ↑ for Case 5.4 are given in Fig. 5.6 which can be a reference for Cases 5.1 and 5.2. The maximum and the minimum C_p values of all the instants are captured. Small discrepancies are observed in the shock region. Variations of the lift coefficient with the angle of attack for Cases 5.1 and 5.2 are presented in Fig. 5.7a, and this type of figure for Case 5.4 is given in Fig. 5.7b. This curve can be followed in a counterclockwise sense where the pitch-up and the pitch-down are represented by the lower and the upper portions of the curve, respectively. The C_L values for $\alpha\uparrow$ and $\alpha\downarrow$ are not equal to each other. This is due to the boundary-motion-induced flow, which can only be captured by dynamic-body calculations. The computed results agree well with each other and the experimental data. Some discrepancy, however, is detected at higher angles of attack between the experiment and the computations. This may be due to the boundary-motion-induced flow fluxes and the numerical errors involved in their temporal and spatial representations across the subdomain grid interfaces. The pitching moment coefficient is not given in this dissertation due to the prior computational experience in [70]; it is anticipated that the present method will not yield a good comparison for the pitching moment coefficient of oscillating airfoil cases.

The flowfield of the cosine plunging airfoil at 320.1° reference angle (X_3=0.0153↑) are shown through its Mach number contours in Fig. 5.8. The leading and trailing edge flows are almost symmetric with respect to the chord. However, the flow deviates from symmetry as the mid-chord is approached. The upper surface shock is closer to the leading edge than the one on the lower surface. This is due to the boundary-motion-induced flow which is gusting down on the upper surface as the airfoil moves up at this instant. The lift is positive for the plunge-down and negative for the plunge-up (Fig. 5.9). The time derivative of Eq. 5.6 is a sine function and it is related to the boundary velocity. Hence, the lift versus the time curve has the sine wave shape. A reference angle shift (30.6^o shift near 0^o reference angle and 32.7^o shift near 180^o reference angle) is observed between the forced motion and the lift values. This is due to the time lag between the switching of the forced motion direction and its effect on the boundary-motion-induced flow. This phenomenon is largely instigated by the inertia of the fluid particles resisting the motion changes. Also, a finite amount of time is needed for the farfield to sense the changes and adjust accordingly.

Shown in Fig. 5.10 are the instantaneous surface pressure coefficient distributions for Case 5.3 at reference angles 45.8^o, 116.6^o, 183.3^o, and 263.7^o. Shock locations on the upper and lower surfaces change with the motion. The upper surface shock is further downstream than the lower surface shock for the plunge-down and it is further upstream for the plunge-up. This can be explained with the help of Fig. 5.2. For example, a plunge-down motion creates the effect of a positive angle of attack in the steady flow sense (Figs. 5.10). Conversely, a plunge-up motion creates the effect of a negative angle of attack (Figs. 5.8, 5.10a). This phenomenon is affected at reference angles where the direction of the motion is changed, as a result of the initial reference angle shift (Fig. 5.10c). Computed C_p values agree well with the experimental values. Almost consistently, however, the computed values slightly underpredict ahead of the shock and overpredict downstream of the shocks.

The transonic flow past an oscillating and plunging airfoil is depicted in Fig. 5.11 through the instantaneous Mach number contours at $\alpha = 2.11^o \uparrow$ and X_3=-0.544C. The instantaneous effect of the plunge-down at M_p=0.0508 on the flow is somewhat similar to a steady flow at 4.86^o angle of attack. This can be observed by comparing Fig. 5.11 with Figs. 5.3 or 5.4. Differences are observed on the upper surface half-chord downstream of the leading edge, which also leads to a small difference in the wake shapes. In comparing Figs. 5.11 and 5.8, it can be seen, that the lower surface shock of the plunge motion does not appear in the pitch-plunge motion.

Table 5.1: Description of the unsteady airfoil cases.

Case	Airfoil	M_∞	Re_∞	Motion	Grid
5.1	NACA-0012	0.6	$4.8*10^6$	Sinusoidal Pitch (Eq. 5.5)	G_3 (no DDT)
5.2	NACA-0012	0.6	$4.8*10^6$	Sinusoidal Pitch (Eq. 5.5)	G_1, G_2, G_3
5.3	NACA-64A10	0.8	$1.3*10^7$	Cosine Plunge (Eq. 5.6)	G_1, G_2, G_3
5.4	NACA-0012	0.6	$4.8*10^6$	Sinusoidal Pitch (Eq. 5.5 with $\alpha_o = 0°$) and Constant-Rate Plunge (Eq. 5.7)	G_1, G_2, G_3

Table 5.2: Computational details of the unsteady airfoil cases.

Case	CPU time per step per cell (μS)	CPU time per cycle (min.)	Time step $(\frac{ta_\infty}{c})$	Time steps per cycle	Total number grid cells
5.1	18.0	14.6	0.01	6,464	7,475
5.2	22.2	17.9			
5.4	30.6	24.7			
5.3	26.7	10.5	0.005	3,142	

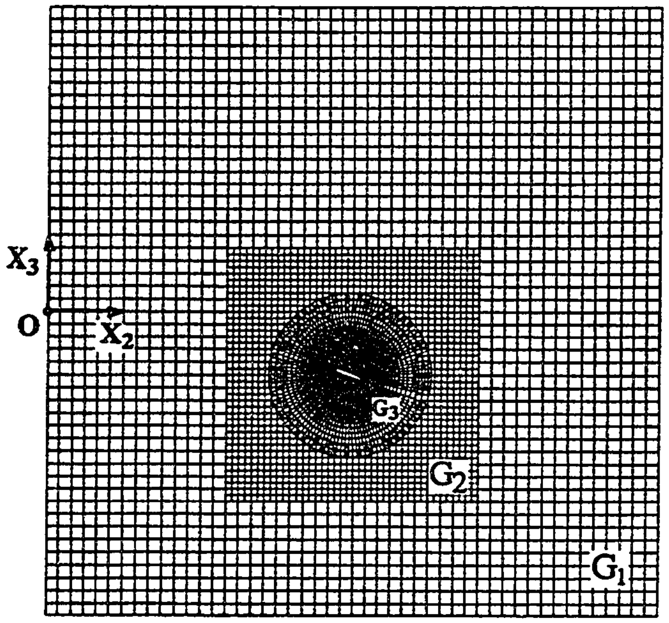


Fig. 5.1: Dynamic domain decomposition grid for an airfoil moving with two degrees-of-freedom.
G_1 : Global Cartesian grid
G_2 : Fine Cartesian grid zonally embedded in G_1 for translational motion
G_3 : O-grid overlapped on G_2 for rotational motion.

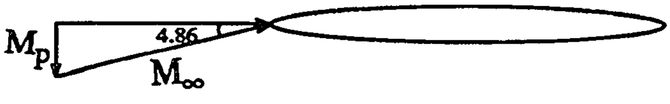


Fig. 5.2 Velocity diagram for an airfoil plunging down at a constant rate given by Eq. 5.7 (Case 5.4).

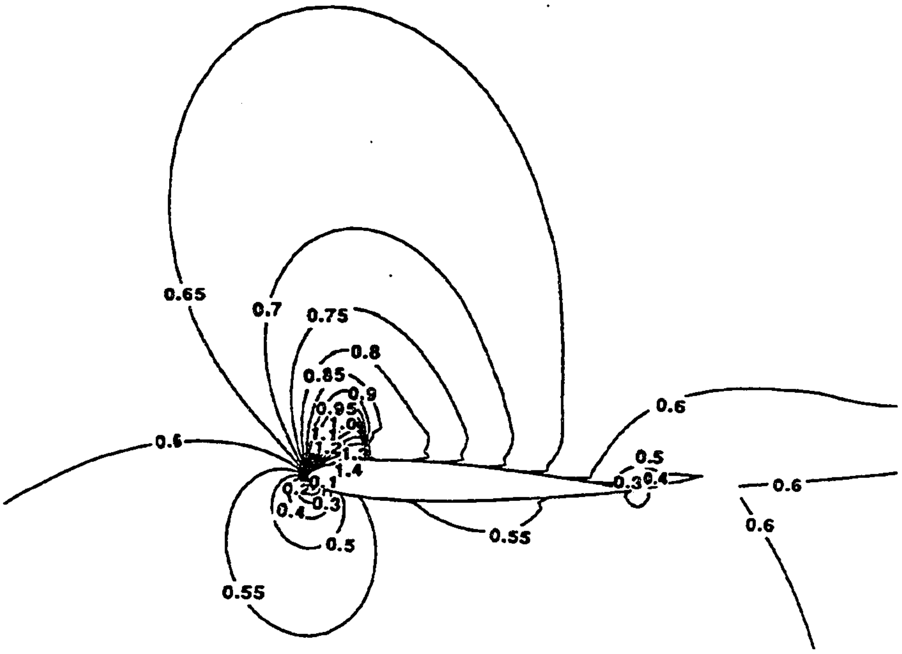


Fig. 5.3: Mach number contours of a sinusoidally pitching NACA-0012 airfoil at $\alpha = 6.97\uparrow$ computed on a single O-grid (Case 5.1).

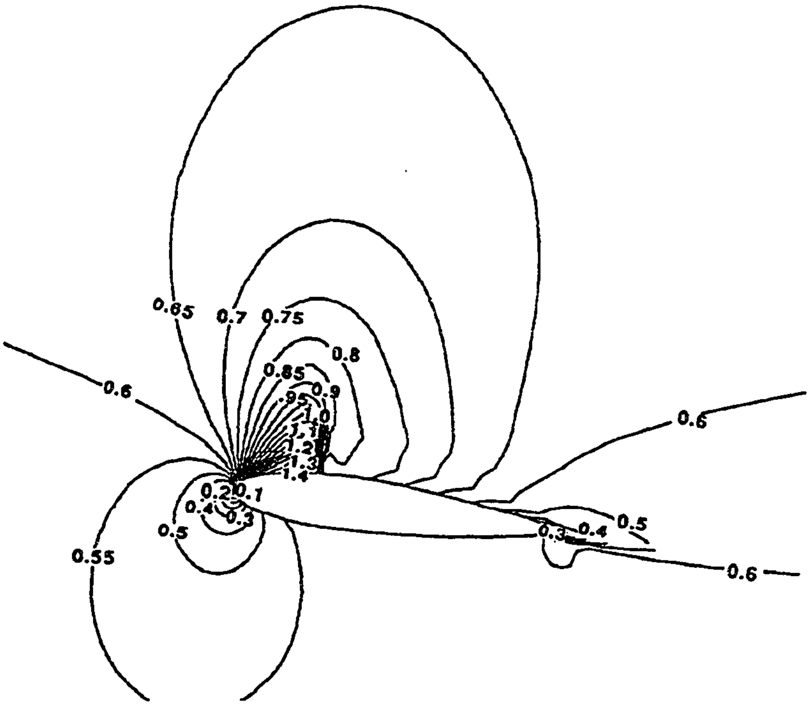


Fig. 5.4: Mach number contours of a sinusoidally pitching NACA-0012 airfoil at $\alpha = 6.97\uparrow$ computed on D^{3}T grid (Case 5.2).

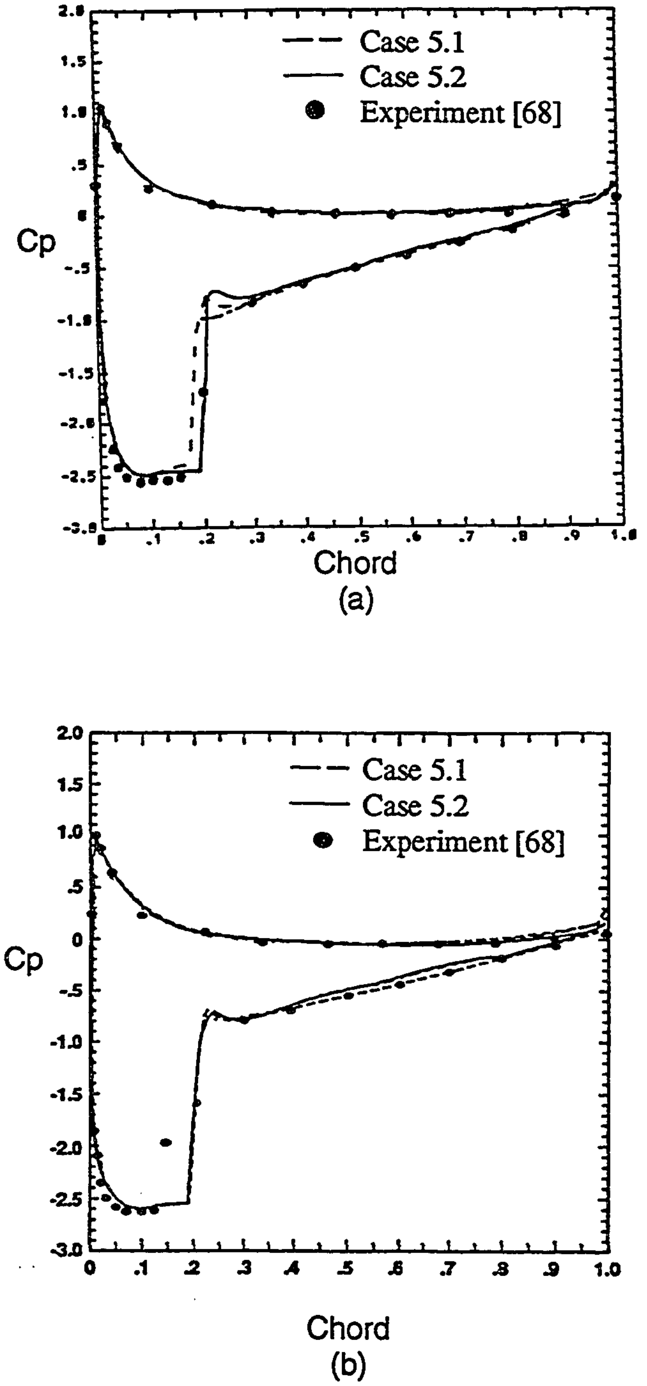
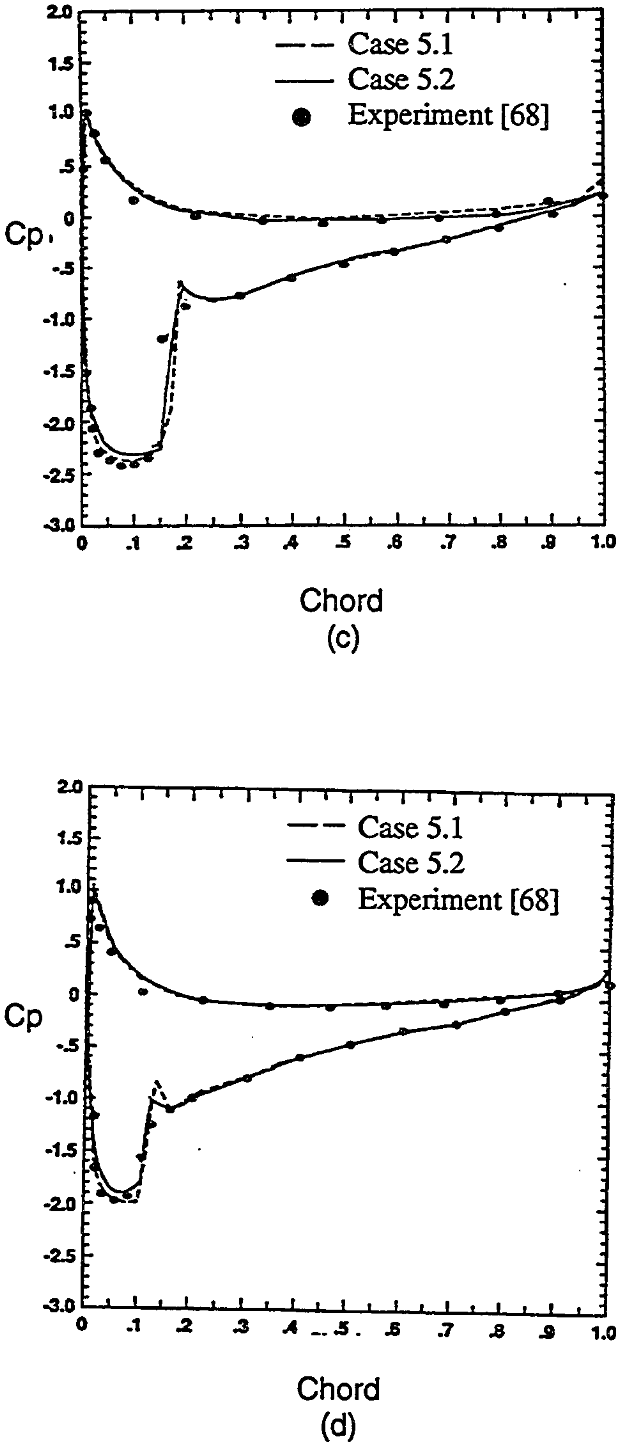


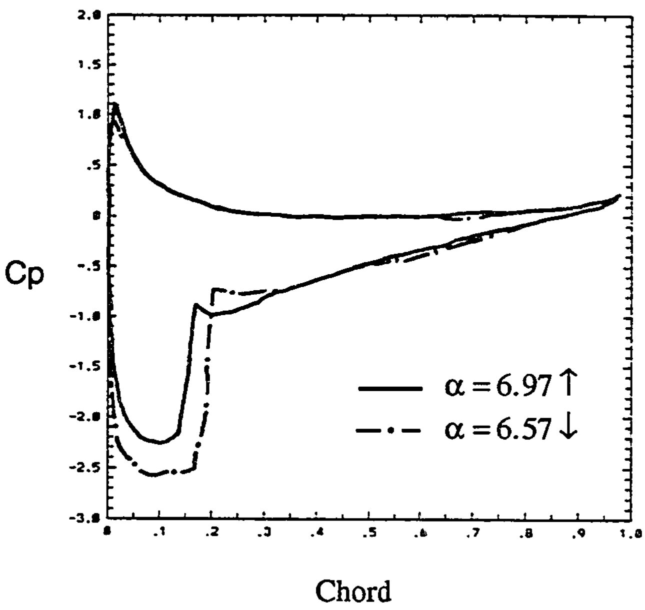
Fig. 5.5: Surface pressure coefficients at various angle-of-attack for a NACA-0012 airfoil.
(a) $\alpha = 6.97\uparrow$, (b) $\alpha = 6.57\downarrow$, (c) $\alpha = 5.11\downarrow$, (d) $\alpha = 3.49\downarrow$.



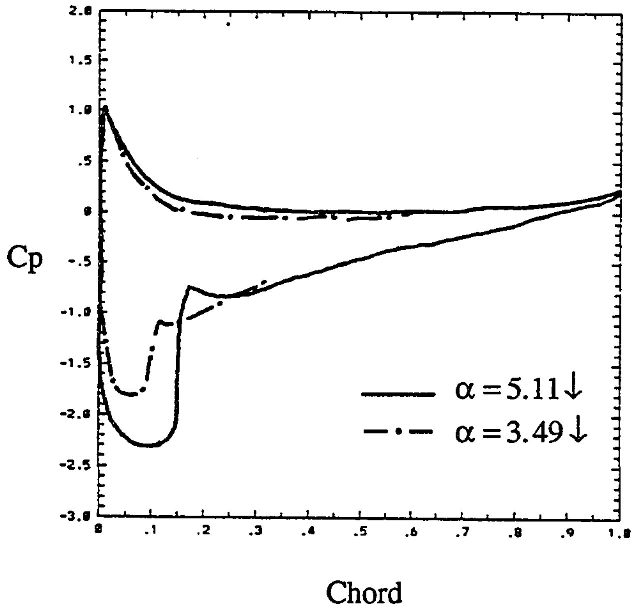
(c)

(d)

Fig. 5.5: Concluded.



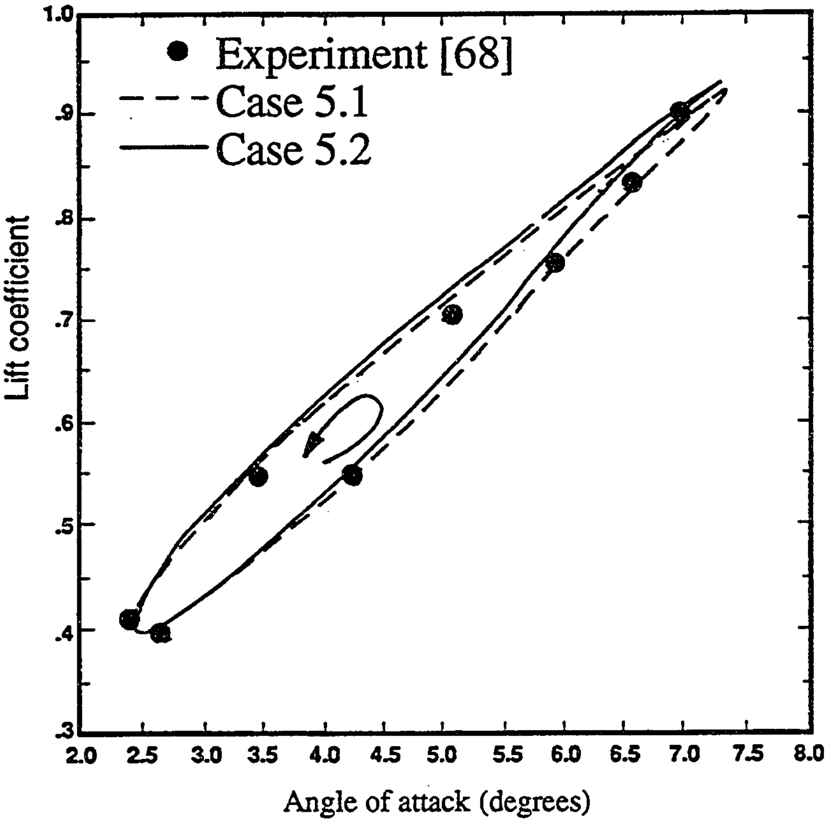
(a)



(b)

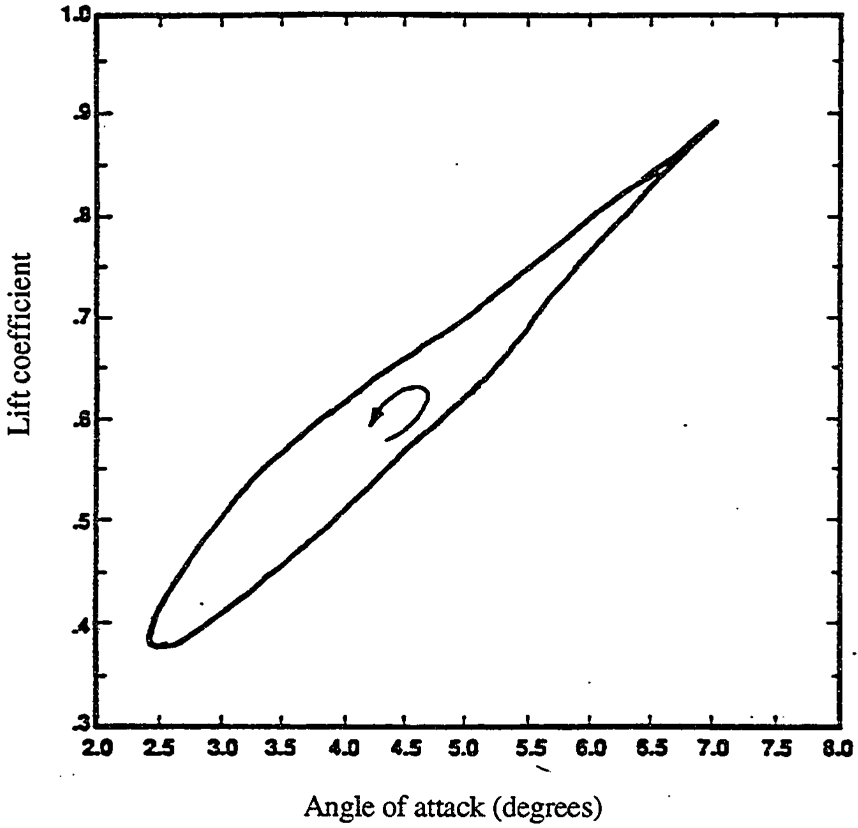
Fig. 5.6: Surface pressure coefficients at various angle-of-attack for Case 5.4,

(a) α = 6.97 ↑ and α = 6.57 ↓, (b) α = 5.11 ↓ and α = 3.49 ↓.



(a)

Fig. 5.7: Lift coefficient versus the angle-of-attack for one cycle of a sinusoidally pitching NACA-0012 airfoil.



(b)

Fig. 5.7: Concluded.

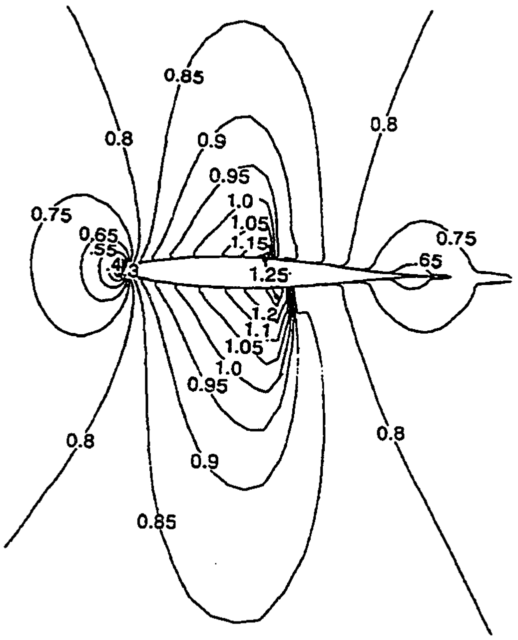


Fig. 5.8: Mach number contours of a cosine plunging NACA-64A010 airfoil at 320.1° reference angle and $x_3 = 0.0153\uparrow$ computed on D^3T grid (Case 5.3).

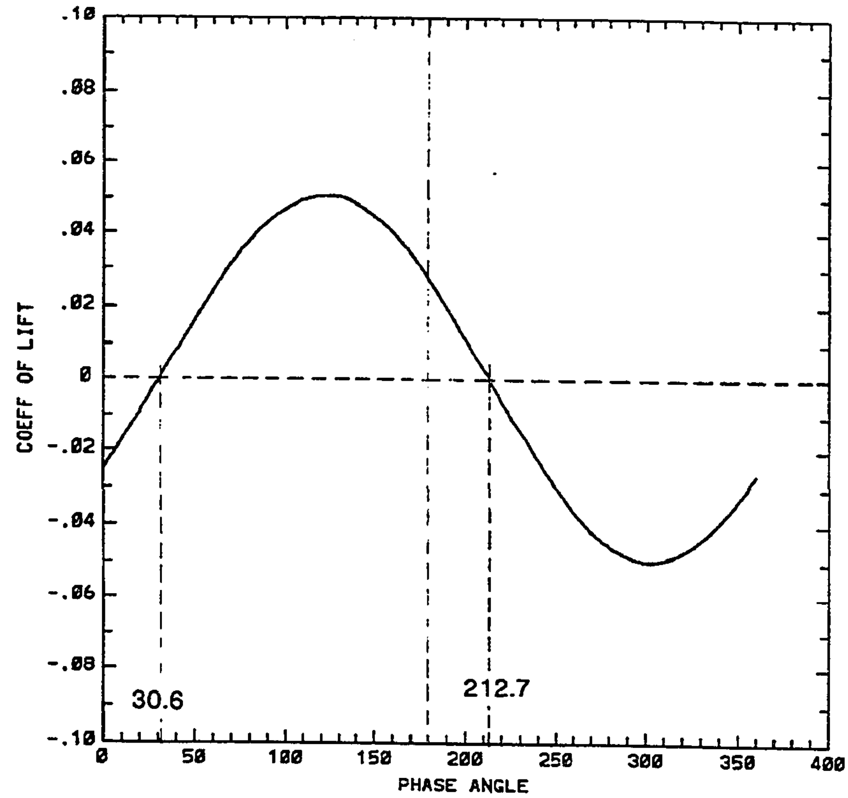
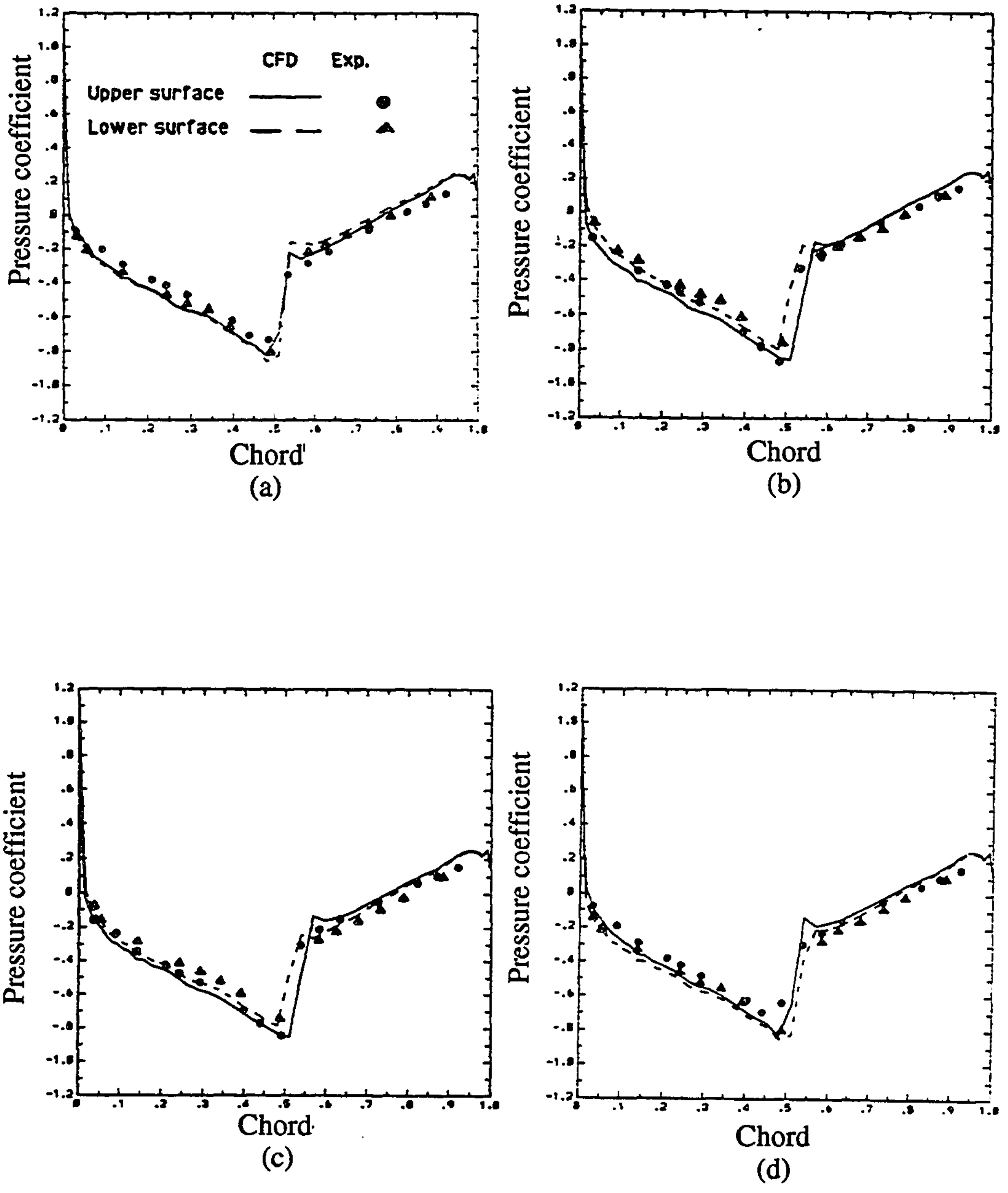


Fig. 5.9: Lift coefficient versus the reference angle for one cycle of a cosine plunging NACA-64A010 airfoil computed on D^3T grid (Case 5.3).



(a) (b) (c) (d)

Fig. 5.10: Surface pressure coefficients at various phase angles for a cosine plunging NACA-64A010 airfoil computed on D^3T grid (Case 5.3).
(a) 45.8°, (b) 116.6°, (c) 183.3°, (d) 263.7°.

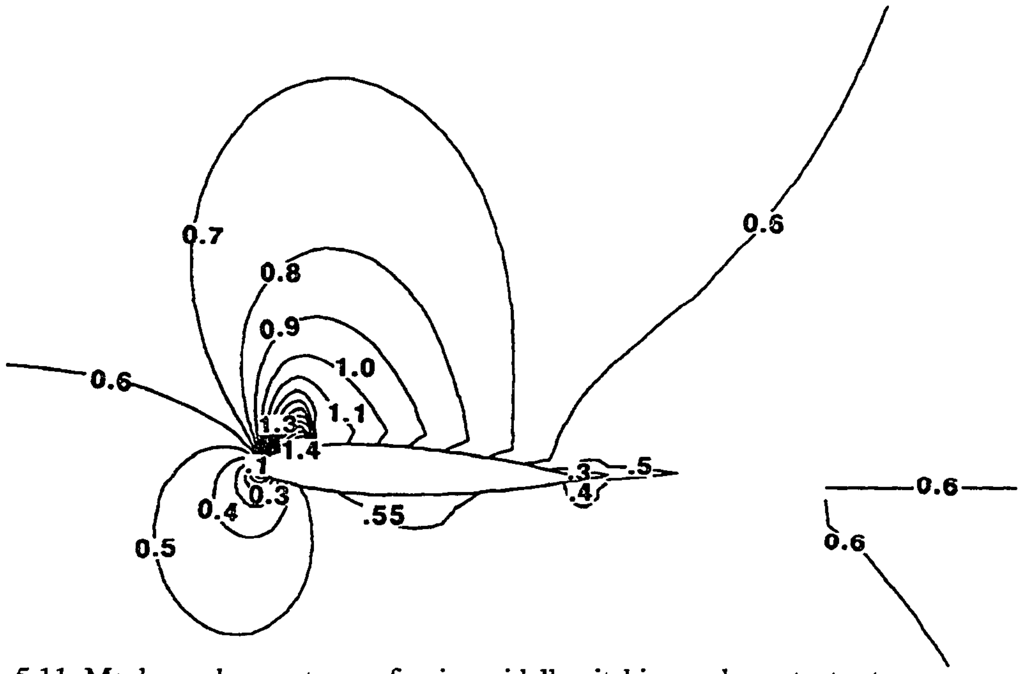


Fig. 5.11: Mach number contours of a sinusoidally pitching and constant-rate plunging NACA-0012 airfoil at α = 2.11° and M_p=0.0508 computed on D^3T grid (Case 5.4).

Chapter 6

FLOW PAST AN OSCILLATING BODY OF REVOLUTION NEAR A VERTICAL WALL

6.1 Presentation of the Cases

Some flowfields involve multiple bodies with at least one of the components in relative motion with respect to the others. Due to the inherent dynamic interference, the generated unsteady flowfield displays features which are significantly different than that around a static body. With this motivation, this chapter presents the extension of the two-dimensional D^3T given in Chapter 5. This method aims at solving such three-dimensional problems with a higher degree of accuracy. Time-dependent Jacobian metrics are used to transform the moving physical domain to the computational domain. As such, the method intends to achieve the following: (i) simulate the flowfield about three-dimensional moving multiple bodies and their aerodynamic interference, (ii) reduce the dispersion error which strongly affects the propagation of gasdynamic waves between the objects, and (iii) reduce the phase error which accumulates with the numerical time advancing procedure.

This time differencing formula (Eq. 2.12), with the appropriate choice of the parameters μ and ν, produces first and second order time accurate implicit schemes [59-60]. The three-point-backward formula, i.e. $\mu=1$ and $\nu=0.5$, is used for the present computations.

For the present applications, three-dimensional, overlapped and moving grids are employed. The computational method is demonstrated by considering a secant-ogive-cylinder (SOC) defined in Fig. 6.1a. The cylinder is placed 4 diameters (d) away from a vertical wall in order to form a multiple body configuration and it is forced to pitch in a plane (X-Y) parallel to the wall (Fig. 6.1b). Prior to presenting the demonstrative cases, however, the adequacy of the grid shown in Fig. 6.1b needs to be established. Unfortunately, accurate and reliable experimental data for the dynamic SOC (Fig. 6.1a) could not be found for the validation.

Three cases are reported in the present thesis (Table 6.1). The cylinder is kept stationary in the first case, so that a means of contrasting the dynamic effects is established. The second and third cases are designed to investigate the unsteady flowfields generated by the cylinder experiencing the prescribed pitching motion. This forced motion is described by the following time function of α:

$$\alpha(t) = \alpha_0 + \alpha_1 \sin(kt) \tag{6.1}$$

To study the effect of the flow regime, the freestream is chosen to be subsonic in the second case and it is supersonic in the third case.

The composite grid consists of two subdomains (Fig. 6.1b). The cylinder subdomain is discretized by a C-O grid, which is 7 diameter , 30d long, and its local origin is 4d away from the vertical wall. The wall subdomain is discretized by a stretched Cartesian grid, which has a length, width, and height of 41d, 30d and 30d, respectively (Fig. 6.1b). Then, the C-O grid is overlapped on the Cartesian grid. There are 70*31*21 points in the C-O grid and 65*40*58 points in the Cartesian grid (Fig. 6.1b). To study the grid dependence of the solution, a much finer grid is also constructed with 97*41*28 points in the C-O grid and 88*54*78 points in the Cartesian grid (Table 6.2). As can be discerned from Fig. 6.2, Case 6.1.b solution is sufficiently close to the one obtained on the finer grid of Case 6.1.a, yet it is obtained at a much lower computational cost. Since this cost difference would be prohibitive for the unsteady, dynamic cases, where the calculations must be performed time accurately, Cases 6.2 and 6.3 are computed on the grid with 196,370 cells.

Shown in Table 6.2 are some of the computational details. The normalized CPU time (CPU per time step per cell) for Case 6.2 is greater than that of Case 6.1. Most of this extra time is needed for the time-dependent D^3T connectivity and information transfer operations, and a small portion of it is for the rigid-body-motion computations.

In Case 6.2, the grid and the freestream flow conditions are identical to those of Case 6.1, but the cylinder as well as its C-O grid are forced to pitch sinusoidally. Therefore, the converged solution of Case 6.1 is used as the initial conditions for Case 6.2, then the computations are performed time accurately for two cycles. The solution for the first cycle is discarded to reduce the effects of the numerical transients on the final solution. In addition, some of this first cycle time is needed for the cyclic boundary-motion-induced flow component to develop. That is, due to the inertia of a fluid particle, its response to the rigid-body-motion requires small amount of lag time.

6.2 Results and Discussion

The converged solution for the steady flow past the static SOC (Case 6.1) is obtained using local time steps with the global Courant number limited to unity. Shown in Figs. 6.3a and 6.3b are the normalized pressure contours on the vertical wall and the symmetry plane of the cylinder, respectively. It can be observed, that only minor aerodynamic interference between the two objects is present with the maximum p' value of -0.006 occurring on the wall. The subsonic flow is initially compressed on the top surface, then it gradually expands toward the tail of the cylinder. An expansion wave can be seen on the bottom surface being terminated by a weak shock.

The normalized pressure contours of Case 6.2 at 7.5°↑ angle-of-attack are shown in Figs. 6.4a and 6.4b. To facilitate some qualitative comparison between Figs. 6.3 and 6.4, they are plotted for the corresponding planes at the identical locations. The maximum p' value of 0.05 can be observed on the wall (Fig. 6.4a), which is an order of magnitude greater than that of the stationary cylinder at the same angle of attack. Also, the gas dynamic compression and expansion regions are switched in their locations as compared with Case 6.1. The shift in the contour locations (Fig. 6.4b) toward the far field indicates the formation of a lower pressure zone between the two objects. This may be attributed to the interference as well as the increased relative speed between the moving cylinder and the flow [56-57].

The aerodynamic coefficients for Cases 6.1 and 6.2 are presented in Figs. 6.5a and 6.5b. Note that the definition of all the aerodynamic coefficients follow the right-hand-rule. C_A and C_N are defined to be along and perpendicular to the cylinder axis, respectively. C_{mp} is in the plane parallel to the vertical wall (X-Y), and C_{my} is in the X-Z plane.

The computed aerodynamic coefficients are very close to zero for the static cylinder in Case 6.1. However, for the dynamic cylinder in Case 6.2, C_A is always greater than zero and displays peaks at the upper and lower extremes of α (Fig. 6.5a). C_N does not appear to be proportional to α, but it rather displays sudden changes near the α value at which the pitching motion transitions to the opposite direction. Note that as the cylinder pitches up from 0^o to 15^o, C_N remains negative, but suddenly changes its sign at 15^o, which corresponds to the peak value of C_A. A similar phenomenon is observed during the pitch-down motion.

C_{mp} corresponds favorably to the cylinder's pitching motion (Fig. 6.5b). For example, when the cylinder pitches up from -2° to 17°, C_{mp} is always negative. This indicates that the aerodynamic forces are assisting in raising the cylinder's nose up with respect to oscillation center. The C_{my} history displays only minor fluctuations throughout the cycle and it only takes negative values. In other words, the cylinder's nose is constantly pushed in towards the vertical wall.

The same subdomain grids and connectivity information used for Case 6.2 are also used for Case 6.3, where the freestream flow is supersonic. Again, computations are performed time accurately for two cycles. Shown in Figs. 6.6a and 6.6b are the instantaneous normalized pressure contours at 17.3°↑ which are plotted at exactly the same planes as those for Cases 6.1 and 6.2. Shown in Fig. 6.6b is the cross section of a three dimensional conical shock attached to the cylinder tip. The shock is weaker along the upper surface and stronger along the lower surface of the cylinder. The flow expands around the cylinder's base. Comparing Figs. 6.6a and 6.6b, it can be seen that the conical shock generated at the nose of cylinder impinges on the vertical wall. The pressure values on the wall are almost proportional to the strength of the shock about the cylinder, that is, weaker above the cylinder and stronger below the cylinder.

The aerodynamic coefficients for Case 6.3 are shown in Fig. 6.7. C_A is always positive with a maximum between 4° and 8° during the pitch-up motion. C_N remains positive during the pitch-up motion, but suddenly becomes negative during the pitch-down motion. The pitching moment corresponds favorably to the cylinder's pitch-up motion. During the pitch-down motion, such a favorable correspondence is observed from 12.5° to 17.5° and from -2.5° to 1.0°. Unlike the subsonic flow of Case 6.2, the yawing moment is zero for the entire motion of Case 6.3. This is predominantly due to the fact that the nose shock reflecting off the wall impinges on the sting and not on the cylinder. Since the force calculations are performed for the cylinder only, and since the upstream is not affected by the downstream in inviscid, supersonic flows, the yawing moment is not influenced by the interference occurring in Case 6.3.

Shown in Figs. 6.8a and 6.8b are the instantaneous normalized pressure contours at 17.3° ↓ which are plotted at the same planes as those of Figs. 6.6a and 6.6b. They are intended to show the differences in the response of the flow to the pitch-up and the pitch-down motions of the cylinder.

Comparing Figs. 6.6b (pitch-up) and 6.8b (pitch-down), it can be observed that the stronger shock located along the bottom surface during the pitch-up motion is switched to the top surface during the pitch-down motion. The higher pressure values along the bottom surface seen in Fig. 6.6b generate the positive normal force seen in Fig. 6.7. Conversely, the pitch-down motion (Fig. 6.8b) causes a negative normal force. The shock interference between the two bodies can be observed from Figs. 6.6a and 6.8a. During the pitch-up motion, the stronger shock is located along the bottom surface, which in turn impinges on the vertical wall. Hence, the higher pressure values can be seen on the lower portion of the wall in Fig. 6.6a. However, during the pitch-down motion, the stronger shock is along the top surface, which causes the higher pressure values on the upper portion of the vertical wall (Fig. 6.8a).

Table 6.1: Description of the computational cases

Case	Mode	M_∞	α_o	α_1	k	ω(Hz)
6.1.a, 6.1.b	Static	0.7	7.5°	0	0	0
6.2	Pitching			10.0°	0.3	36.5
6.3		1.8			0.2	62.6

Table 6.2: Computational details

Case	Grid Size	CPU time per step per cell (μs)	CPU time per cycle (hr.)	time step (ms)	time steps per cycle
6.1.a	482,012	14.0	n/a	0.4	n/a
6.1.b	196,370				
6.2		16.4	46.8		52,360
6.3			70.0		78,540

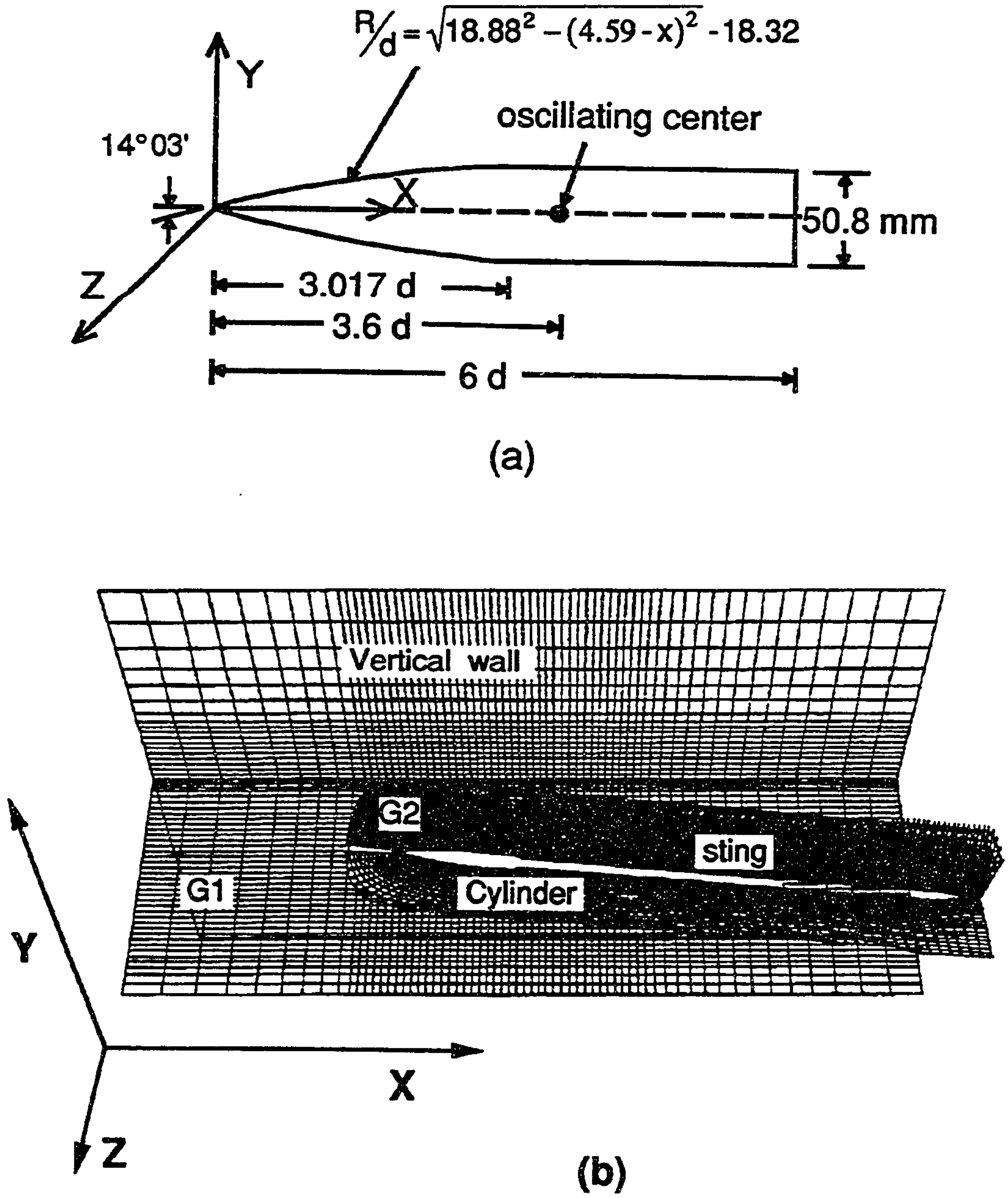


Fig. 6.1: (a) Geometric definition of the secant-ogive-cylinder (SOC).
(b) Composite grid for SOC near a vertical wall.

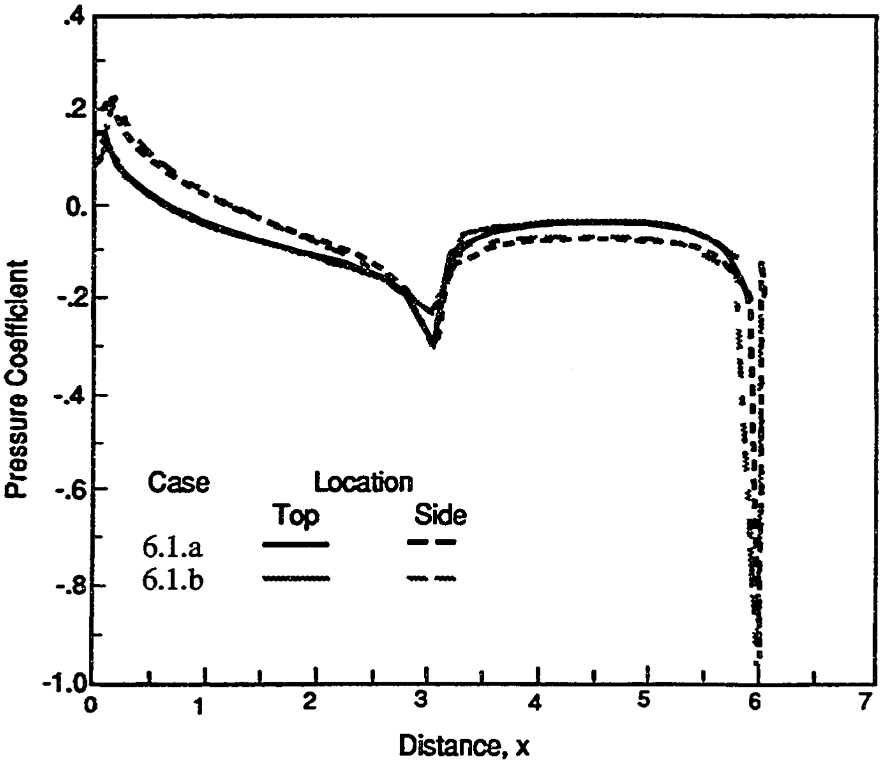
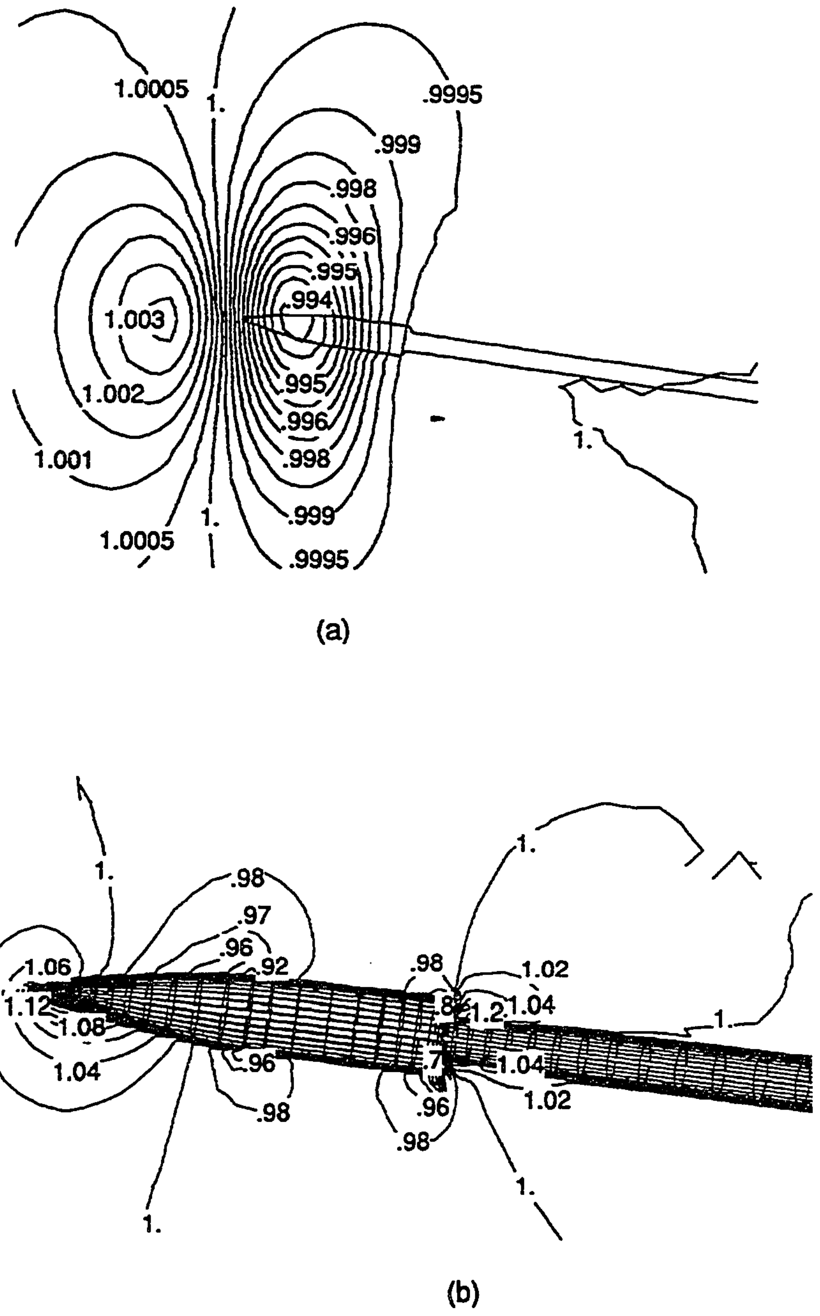


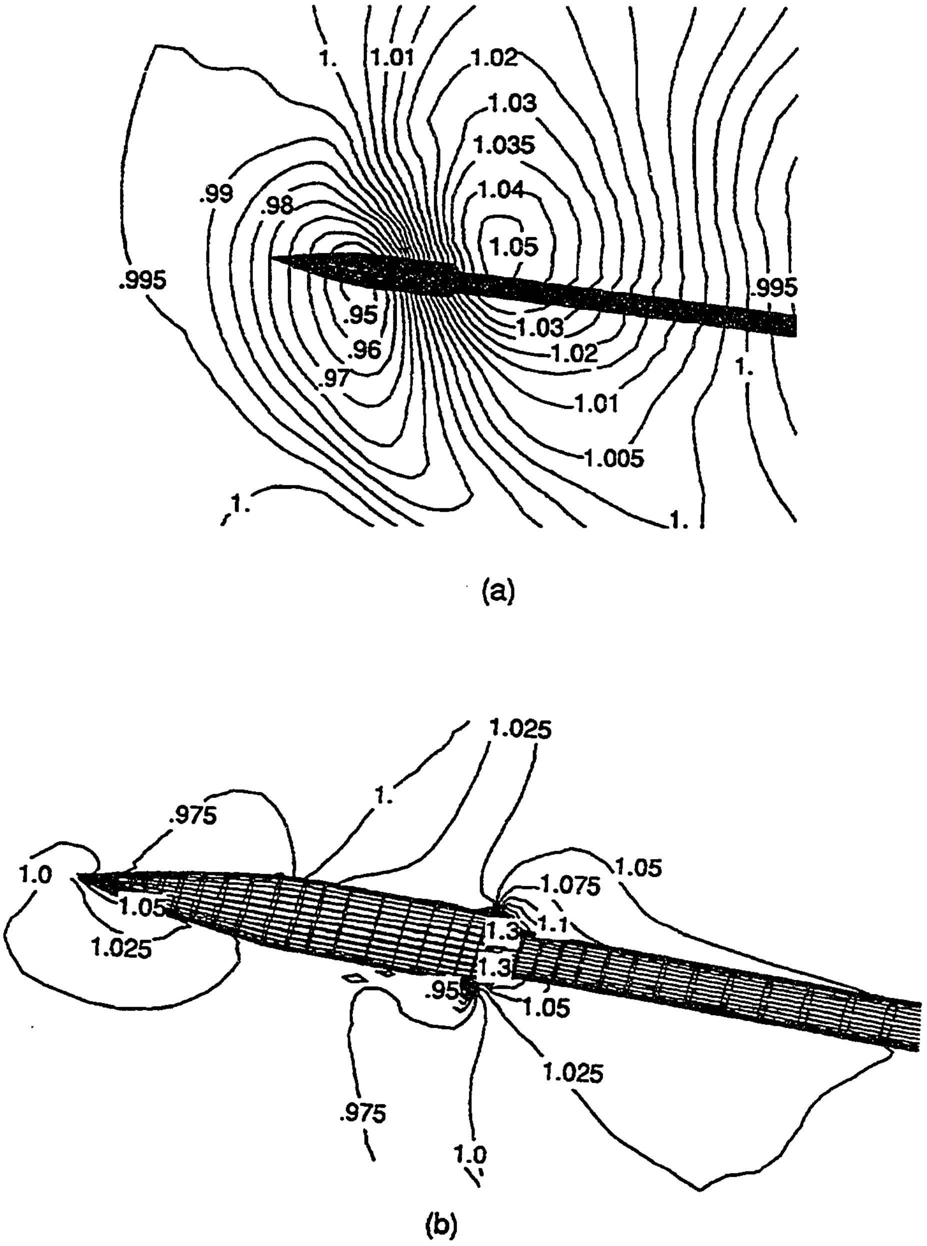
Fig. 6.2: A grid refinement study for Case 6.1. Pressure coefficient distributions along the top and the side of the cylinder using a finer grid (Case 6.1.a) and the regular grid (Case 6.1.b)



(a)

(b)

Fig. 6.3: Normalized pressure contours for Case 6.1.b:
(a) on the wall (footprint),
(b) on the symmetry plane of the static cylinder at $\alpha = 7.5^o$.



(a)

(b)

Fig. 6.4: Normalized pressure contours for Case 6.2 during the pitch-up motion at $\alpha = 7.5^\circ \uparrow$:

(a) on the wall (footprint),

(b) on the symmetry plane of the sinusoidally pitching cylinder.

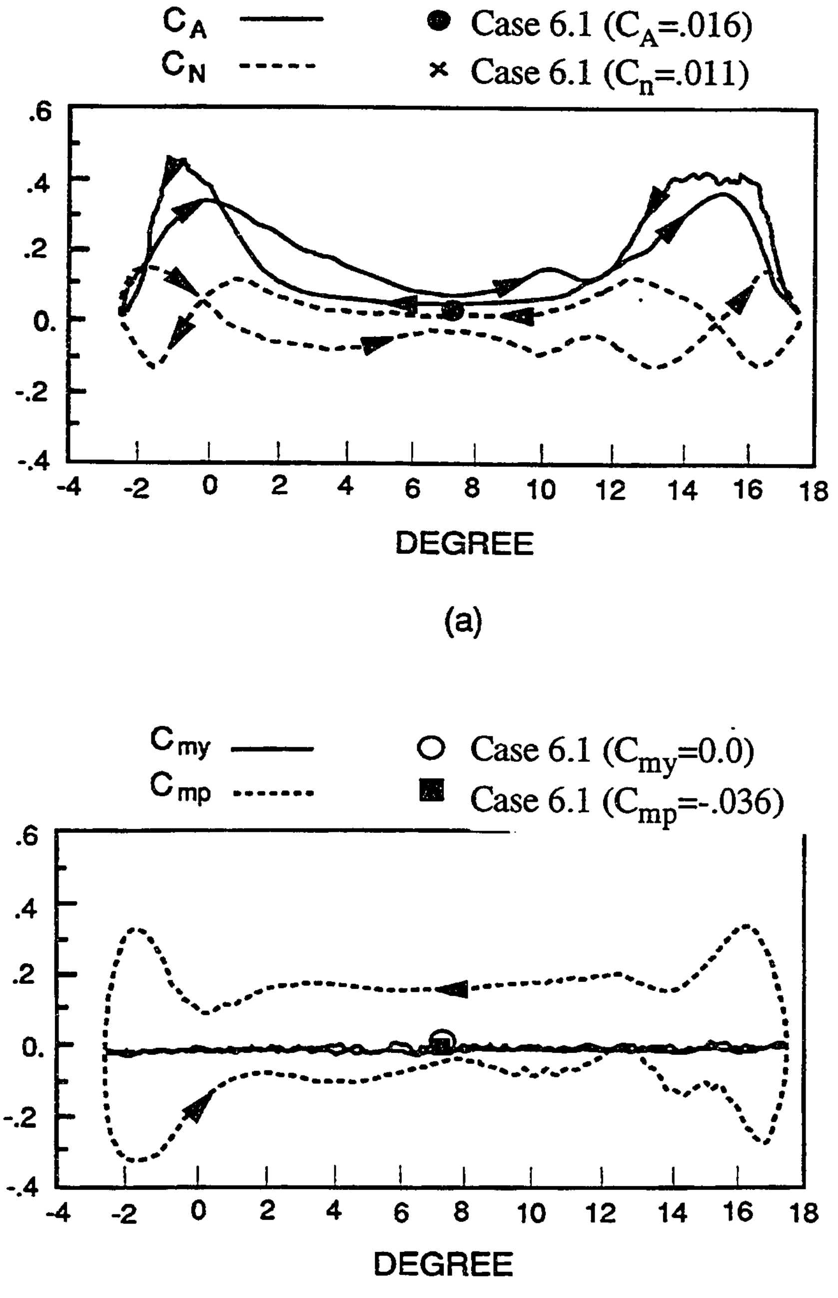


Fig. 6.5: Aerodynamic coefficients versus the angle-of-attack for one cycle of a sinusoidally pitching cylinder for Case 6.2:
(a) force coefficients,
(b) moment coefficients.
Positive C_{mp} and C_{my} push the nose down and away from the wall, respectively.

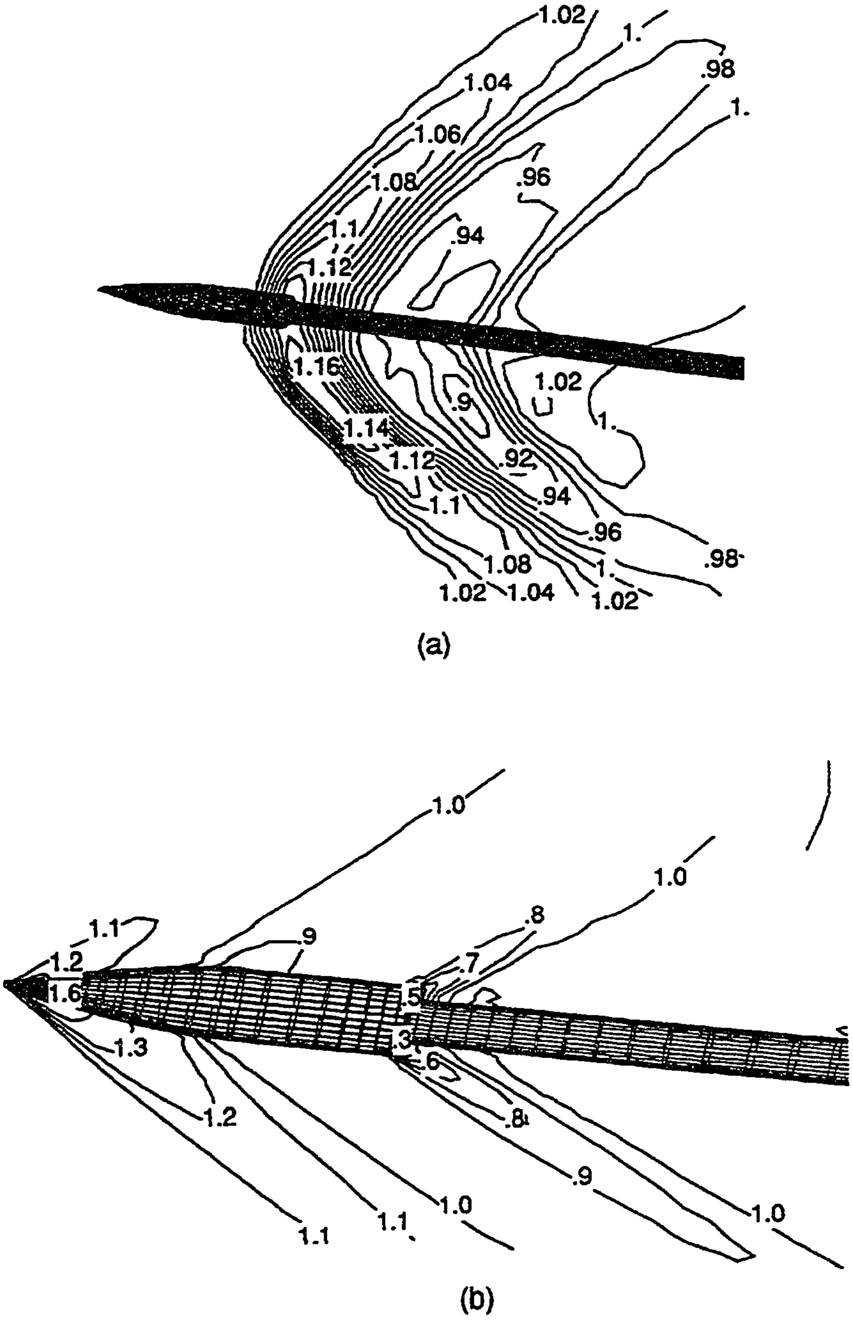


Fig. 6.6: Normalized pressure contours for Case 6.3 during the pitch-up motion at $\alpha = 17.3^\circ \uparrow$:
(a) on the wall (footprint),
(b) on the symmetry plane of the sinusoidally pitching cylinder.

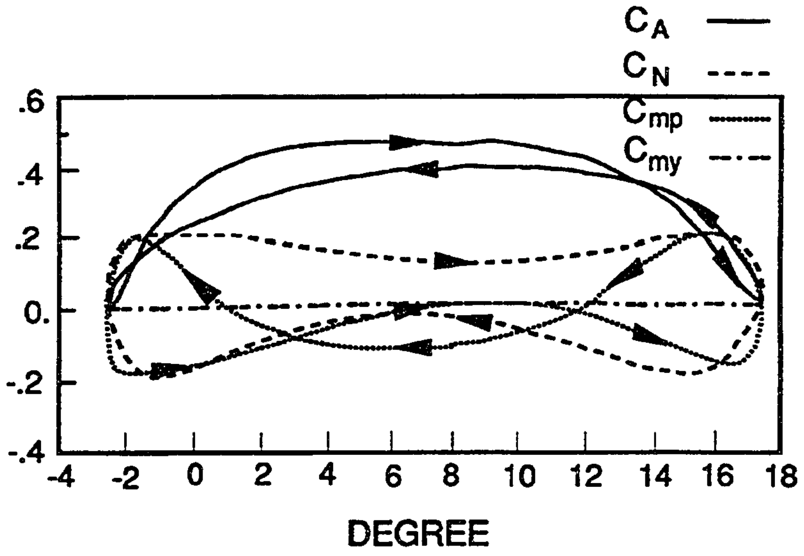


Fig. 6.7: Aerodynamic coefficients versus the angle-of-attack for one cycle of the sinusoidally pitching cylinder for Case 6.3.

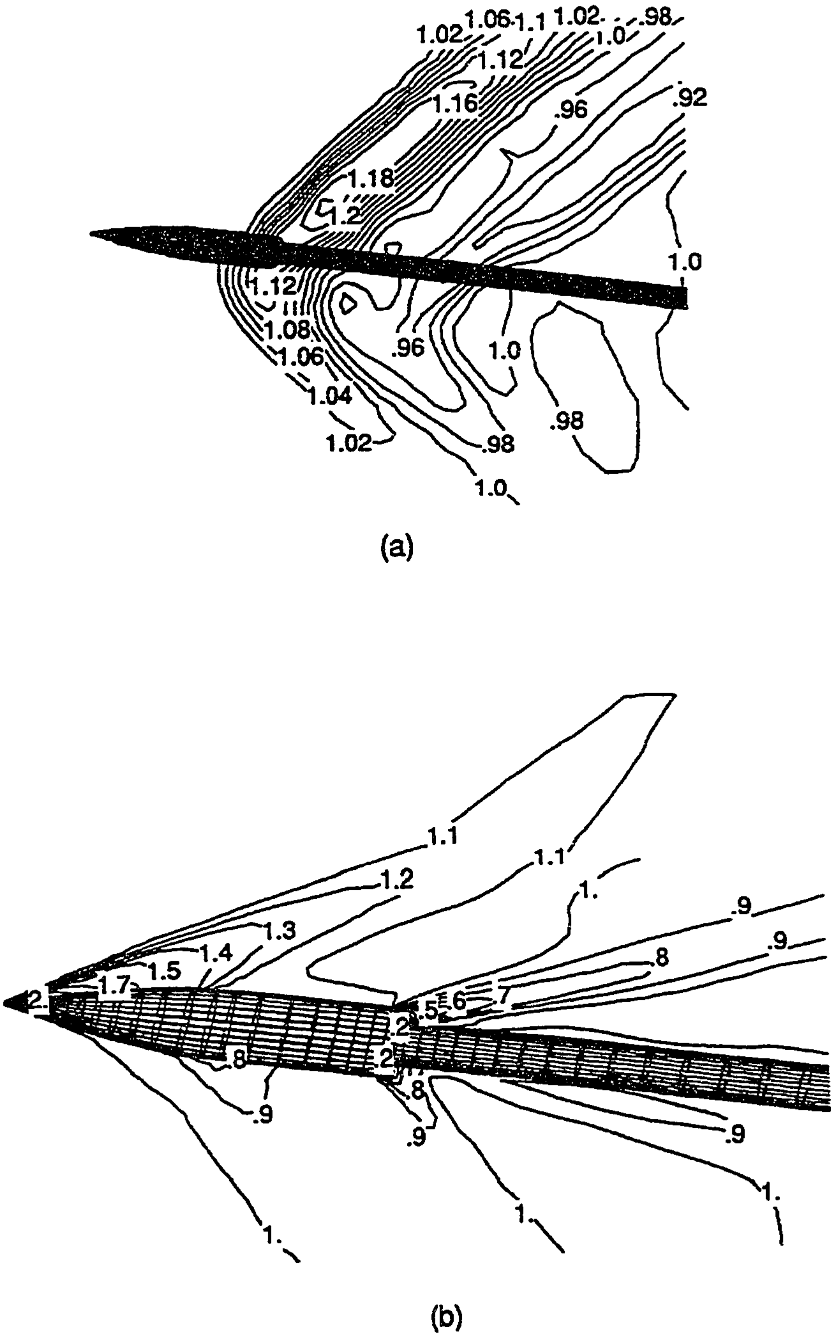


Fig. 6.8: Normalized pressure contours for Case 6.3 during the pitch-down motion at $\alpha = 17.3^{\circ} \downarrow$:

(a) on the wall (footprint),

(b) on the symmetry plane of the sinusoidally pitching cylinder.

Chapter 7

STORE SEPARATION FROM A WING

7.1 Presentation of the Case

The present wing/store (WS) configuration is simplified from the wing/pylon/fin/store configuration reported in [17, 34]. It consists of a clipped delta wing with a 45-degree leading edge sweep and a NACA-64A010 airfoil section and directly below this wing is an ogive-cylinder-ogive store (Fig. 7.1).

The governing equations are simplified from the thin-layer Navier-Stokes equations to the Euler equations in order to save the computational time and computer memory via the inviscid grids. The grid is generated for the transonic (Mach number of 0.95) and inviscid flow. The boundary conditions in the wing root is using symmetry plane. Two cases are demonstrated in this chapter. Case 7.2 consists of two subdomains that contain approximately 0.36 million grid points, and that are interconnected through grid overlapping (Fig. 7.2). The first grid is conformed to the wing and has a C-O topology. This grid serves as the global grid and it extends 95 units (store diameter is used as the unit length) upstream, 104 units downstream, and 45 units outboard of the store nose. The second grid is an O-O grid about the store. Upstream, downstream and radial outer boundaries are located at 4.0, 9.2, and 2.5 units from the nose of the store, respectively. For Case 7.1, an intermediate Cartesian grid is added into the Case 7.2, which increased the total number of grid points to 0.5 million. The purpose of this 3-grid design is to study the effects of multiple sets of interpolation on the temporal error.

The proper lines of communication between these two or three overlapped grids are established using the overlapped scheme [13-22]. This procedure is carried out by the computer code MaGGiE [13-22]. Due to extensive overlapping, care must be taken to ensure that all points falling within body boundaries are removed from the computational domain. The removal of these points creates three-dimensional hole boundaries in the wing and store grids. Furthermore, around holes created for another body and the outer boundaries of the overset grids, i.e., the store and the interface grids, intergrid communication takes place through a trilinear interpolation of the required boundary conditions from the solution of a neighboring grid. The points around hole boundaries that receive this interpolated data of Case 7.2 are depicted in Fig. 7.3 for the wing and store grids when the store is in the initial position. For this position, the domain connectivity takes 4.5 seconds on a Cray Y-MP computer and results in 8,228 points needing interpolation data for Case 7.2. However, it takes 20 seconds and results in 45,000 points needing interpolation data for Case 7.1.

As mentioned in Chapter 4.1.3, the grid speed is a factor which affects the accuracy of overlapped scheme. Case 7.1 substantiates this fact. Although the overlaid fringe points which belong to different meshes are not desirable [1], it is impossible to avoid this difficulty in certain cases. Therefore, Case 7.1 is a demonstration of how the temporal interpolation error affects the overlapped scheme.

7.2 Results and Discussion

Table 7.1 contains some of the parameters related to the dynamics of the store being dropped. An ejection force is added for the current drop case, because the initial aerodynamic lift (C_y) turns out to be greater than the weight.

Both cases are started from the initial freestream conditions and the computations are marched with pseudo time steps to obtain a converged solution for a steady state flow. The pressure contours for Cases 7.1 and 7.2 of the flowfield when the store is at its initial position are presented in Figs. 7.4 and 7.5, respectively The freestream impinges on the nose of the store and the leading edge of the wing, then it gradually expands to supersonic speeds along their surfaces. Shocks are observed on all the surfaces near their trailing edges. The stagnation points on the wing and store are shifted towards each other, and behave like a channel flow. The upper surface shock of the store reflects from the lower surface of the wing and merges with the shock originating at the lower surface of the wing. Note that all the contours pass through the subdomain interfaces smoothly. Also, it can be observed from Figs. 7.4 and 7.5, that there is almost no difference for these two steady state contours.

The computations are continued with physical time steps, and the initial time step (Δt) is 0.0002 with the corresponding maximum CFL number as 0.6. The pressure contours of Case 7.1 after 5 time units are given in Fig. 7.6. It is noticed that the shock emanating from the trailing edge of the wing passes the interpolation boundary, but then it splits into two directions, which is obviously a numerical error in this case. However, Case 7.2 does not have this type of error and can be continued for further simulation. The only difference between Cases 7.1 and 7.2 is the intermediate subdomain grid, and both cases have almost the same steady state solutions. Hence, it is concluded that the split shock phenomenon of Case 7.1 is due to the temporal interpolation error, which is caused by the discrepancy in the eigenvalues (Section 4.1.3 and Eq. (4.4)). It is pertinent to highlight that the number of degrees of freedom in the present cases is greater than those of the cases studied in Chapter 4 and, consequently, the differences of the characteristic wave directions are more pronounced.

The instantaneous pressure coefficient distributions on the store surface for Case 7.2 are shown in Fig. 7.7. They are plotted when the nondimensional time is zero and 46. It can be seen that the pressure difference between the upper and lower surfaces after 46 time units of dropping (Fig. 7.7a) is much smaller than the initial difference (Figs. 7.7b). This is mainly attributed to the pitching down (negative angle-of-attack) of the store that results in elevated lower surface pressures. Also, the increasing clearance between the store and wing produces less of an interference flow effect.

Comparing the four off-surface pressure contours of Case 7.2 (Figs. 7.5a, 7.8a, b, and Fig. 7.9a), it can be observed that the interference shock that emits from the tail of the store has a time-varying effect on the wing. In Fig. 7.8a (33 time units), the upper surface shock of the store impinges on the lower surface of the wing, then reflects down and merges with the shock originating at the lower surface of the wing. However, the reflection and merged positions in Fig. 7.8a are closer than Fig. 7.5a. In Fig. 7.8b (69 time units), the upper surface shock of the store is almost overlaid with the shock originating from the lower surface of the wing at the tail region, and both of them merge immediately. The tail shock of the store in Fig. 7.9a (105 time units) merges with the lower surface shock of the wing and does not impinge on the wing. Consequently, the strongest interference can not be observed in Fig. 7.9.

The 6-DOF trajectory computed in Case 7.2 for the store's center of gravity is shown in Figs. 7.10. The translational motion is mainly in the streamwise and downward directions, but a small displacement also occurs toward the wing root. The store's nose pitches down with a gradual yawing toward the wing root. The sideway motion and rolling are not expected to be significant for this axisymmetric store despite some non-symmetry of the wing flowfield in this direction (wing tip effect).

The force and moment coefficients (global frame of reference) computed for Case 7.2 are shown in Fig. 7.11. It can be noticed that the axial (C_x) and lift (C_y) forces are decreasing in time, hence, the downward and downstream accelerations are decreasing. But, the outboard (C_Z) force is increasing from negative value towards zero. Then, it may be concluded that the inboard acceleration is decreasing. From Fig. 7.11b, the rolling moment (C_{mx}) is constantly zero, the reason can be attributed to the symmetric geometry of the store. The yawing moment (C_{my}) gradually increases in time, and consequently, the store nose tilts towards the outboard. The pitching moment (C_{mz}) slightly decreases first, but then it increases; however, the store is always pitched downward.

The three-dimensional unsteady simulation is very expensive and therefore efficiency is also as important as accuracy. In order to reduce the computational work, the diagonal inversion method is used in this case that can save approximately 35% CPU time compared with the block inversion method. Also, the maximum CFL number of this case has been increased up to 15 despite the associated time inaccuracies. This case requires 46 hours on a Cray Y-MP.

Table 7.1: Parameters for Store Dynamics

F^1_{eject}	W^1	m^2	I^3_{roll}	I^3_{pitch}	I^3_{yaw}
1.456	0.243	5,636.250	653.822	11,770.313	11,770.313

[1] normalized by $\rho_o a_o^2 d^2$. [2] normalized by $\rho_o d^3$. [3] normalized by $\rho_o d^5$.

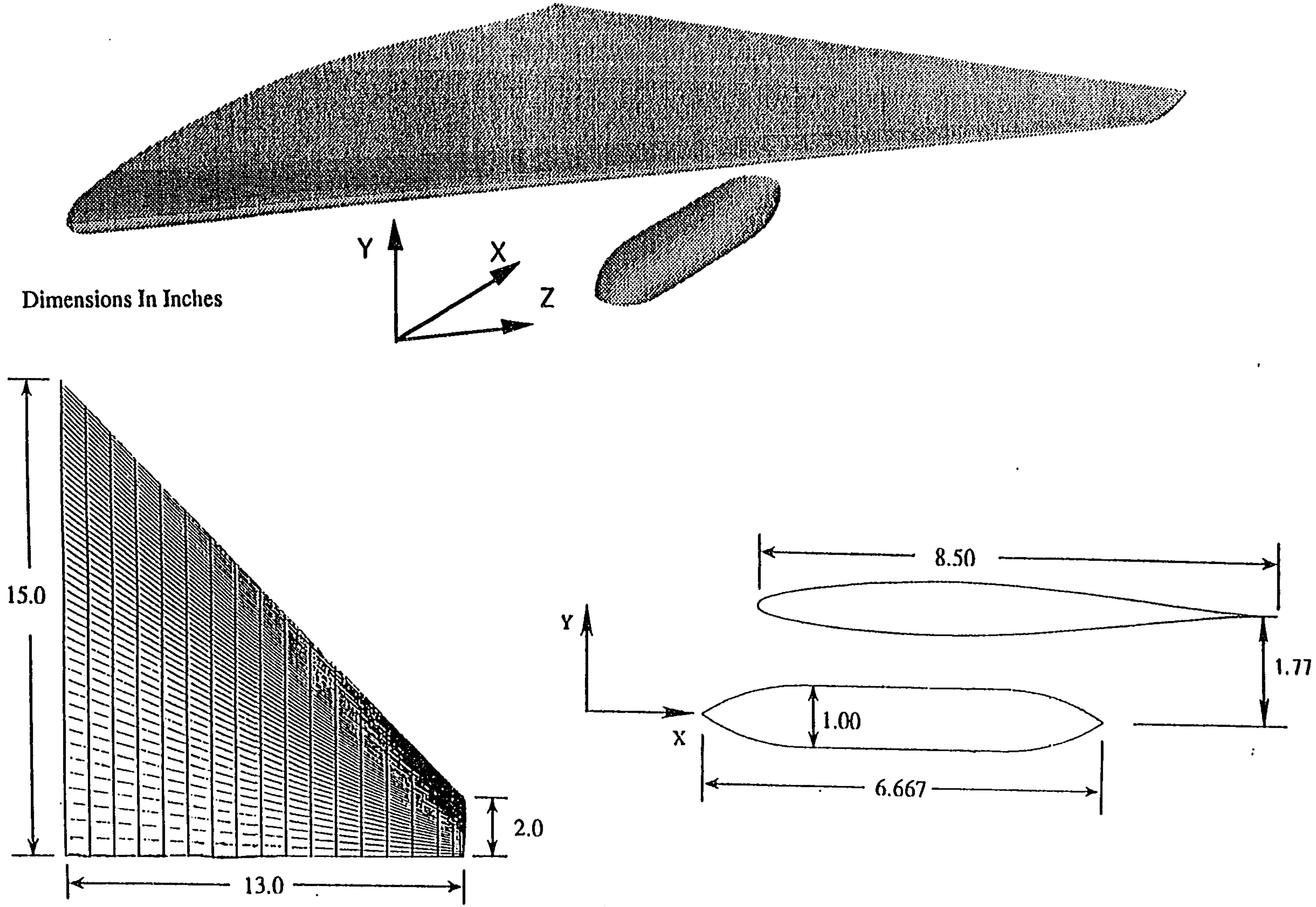
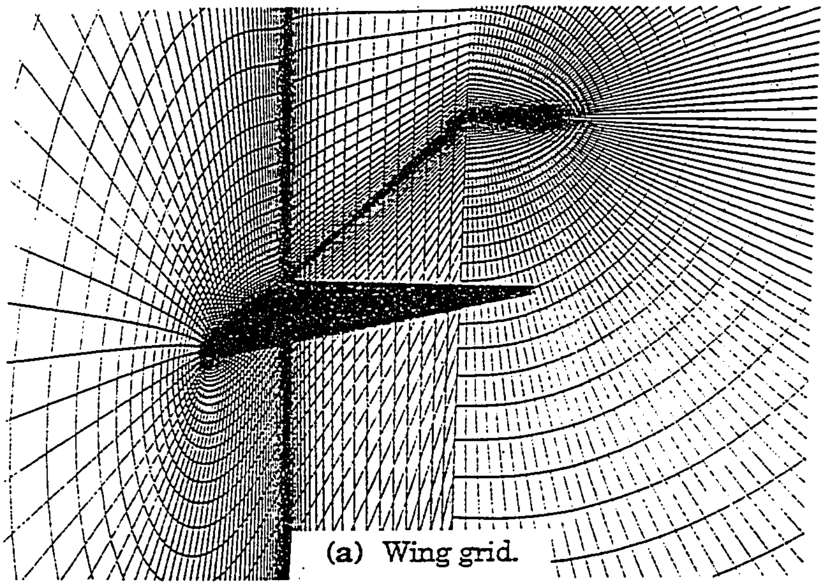
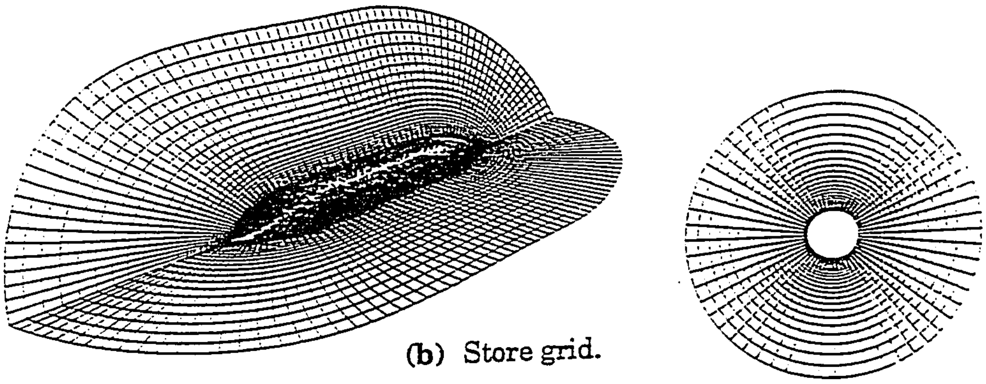


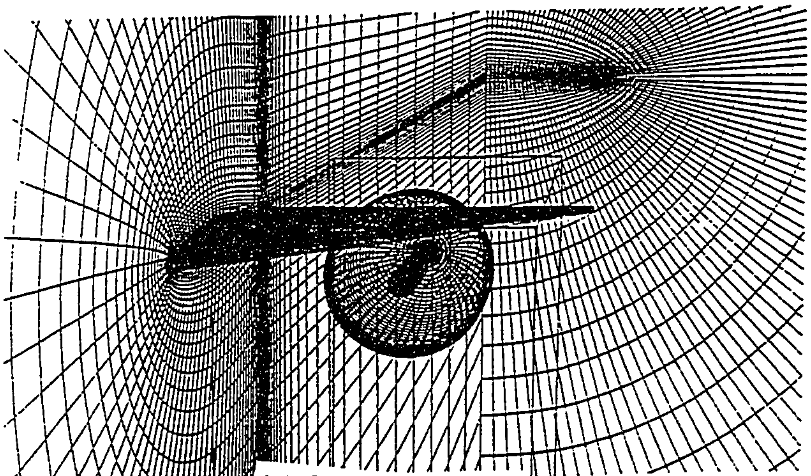
Fig. 7.1: The wing/store geometry.



(a) Wing grid.



(b) Store grid.



(c) Composite mesh

Fig. 7.2 The grids and composite mesh for the wing and store configuration.

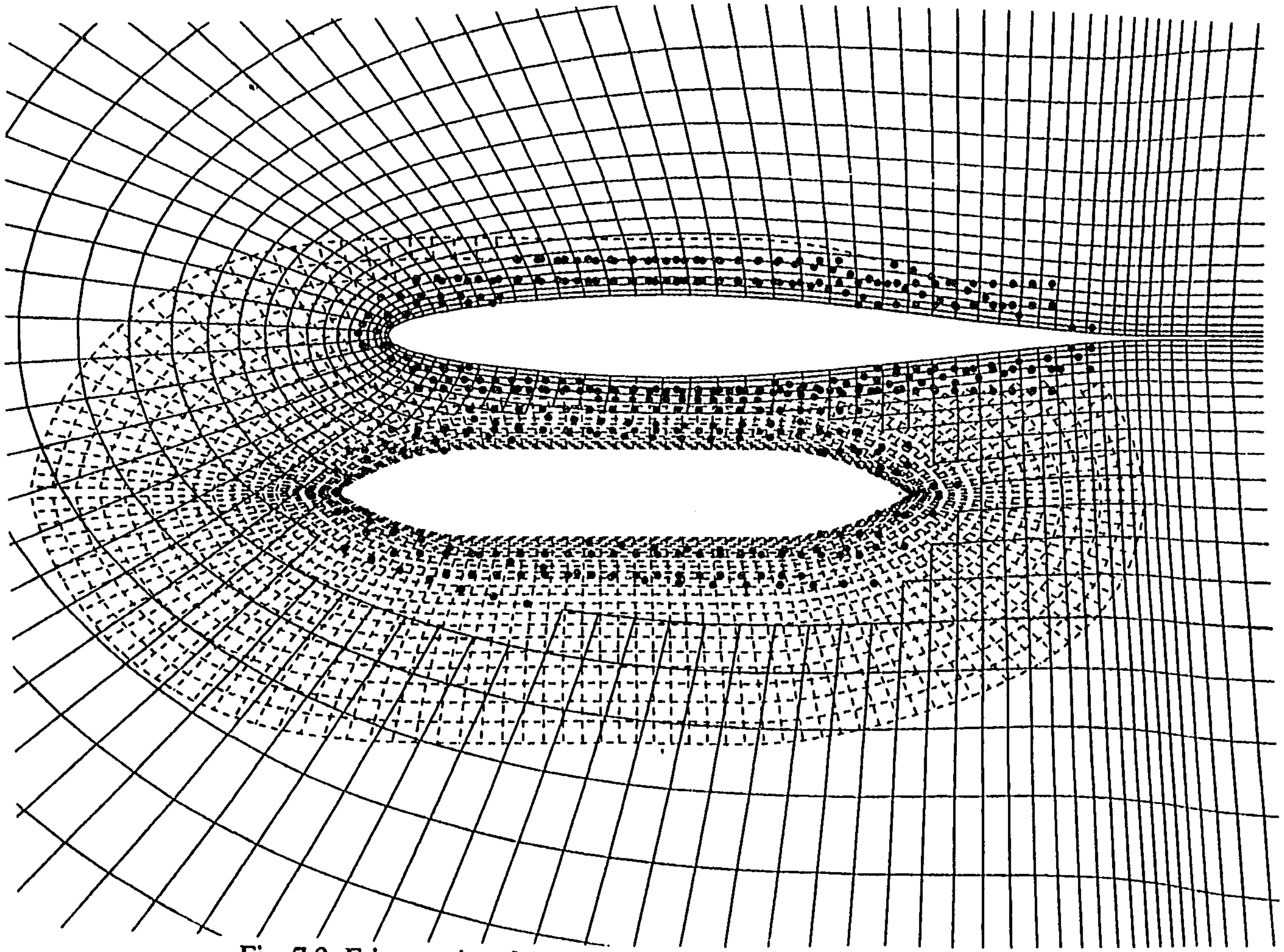


Fig. 7.3: Fringe points for the wing and store grids at the mid span plane.

(•) symbols indicate fringe points

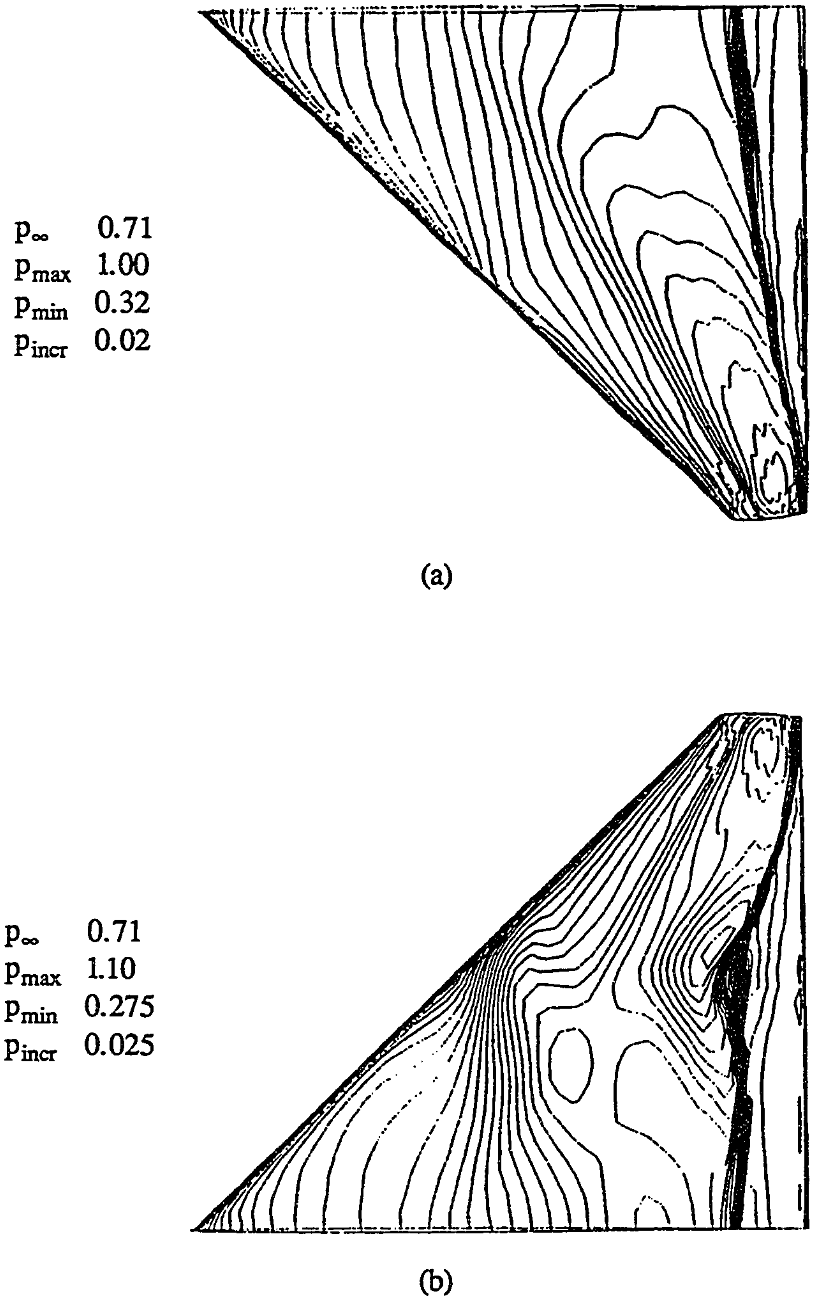
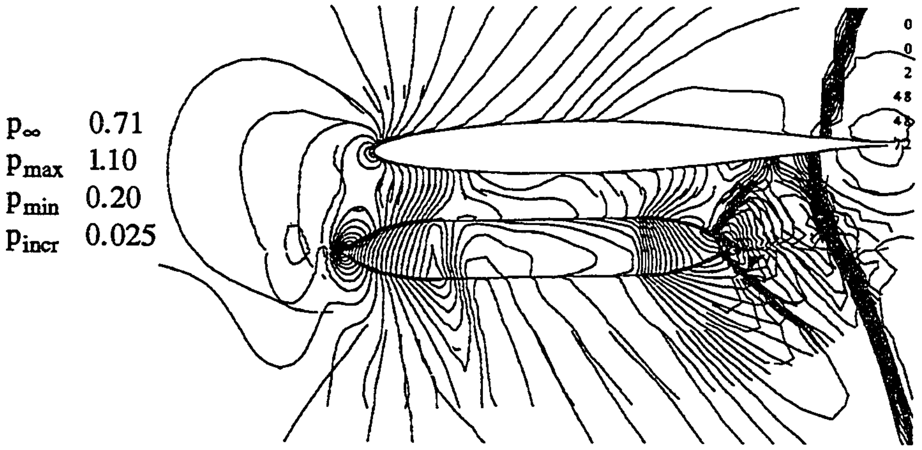
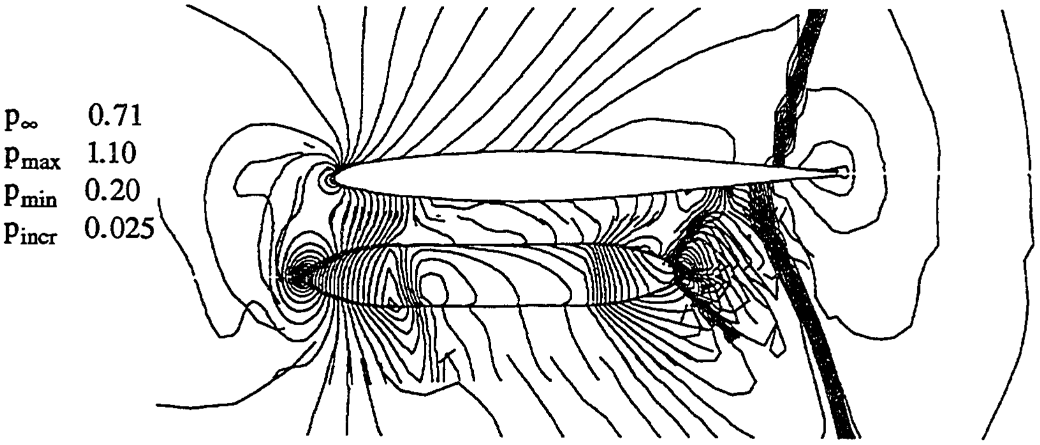


Fig. 7.4: Pressure contours for Case 7.1 at steady state,
(a) on the upper surface, (b) on the lower surface,
(c) off surface for the wing and store through the mid-span plane.



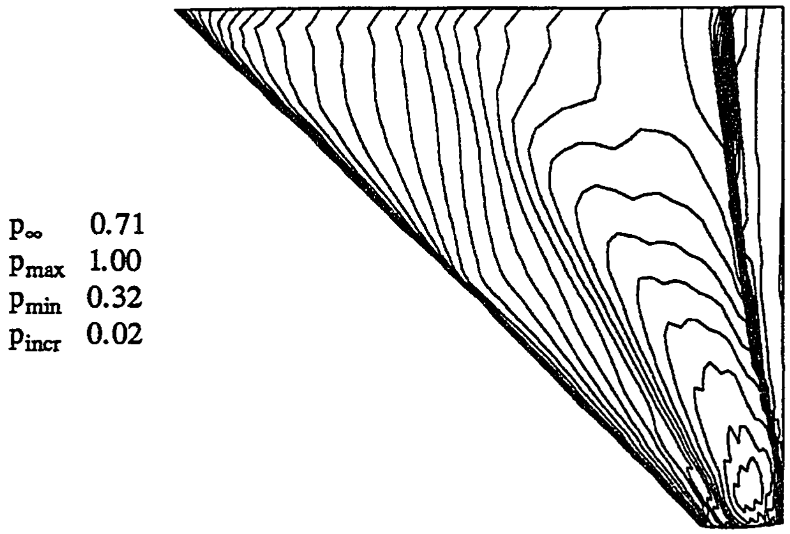
(c)

Fig. 7.4: Concluded.

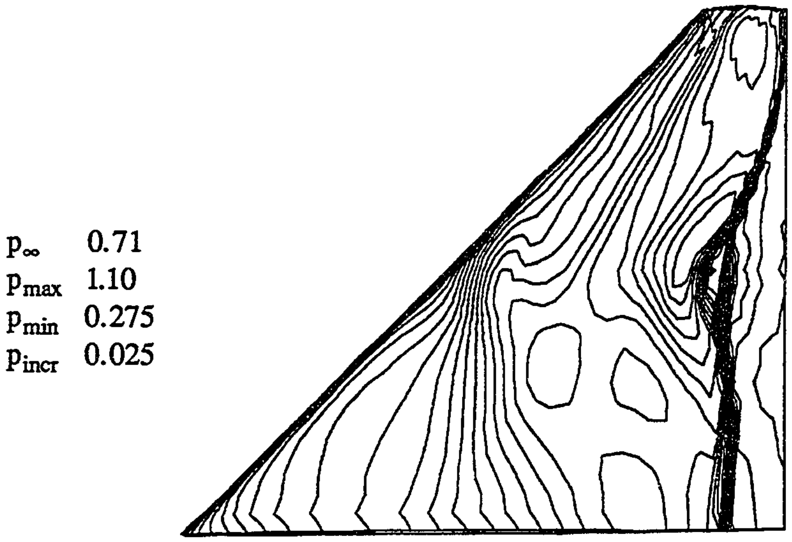


(a)

Fig. 7.5: Pressure contours for Case 7.2 at the steady state.
(a) off surface for the wing and store through the mid-span plane,
(b) on the upper surface of the wing,
(c) on the lower surface of the wing.



(b)



(c)

Fig. 7.5: Concluded.

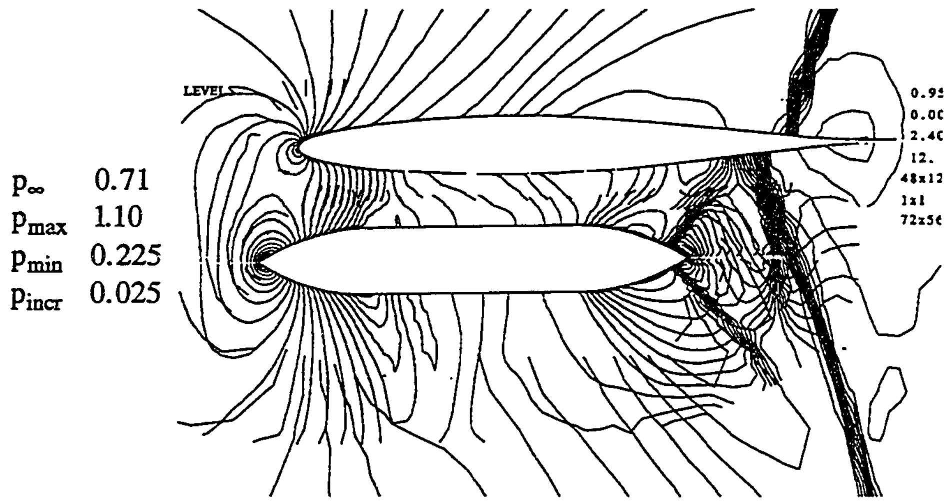
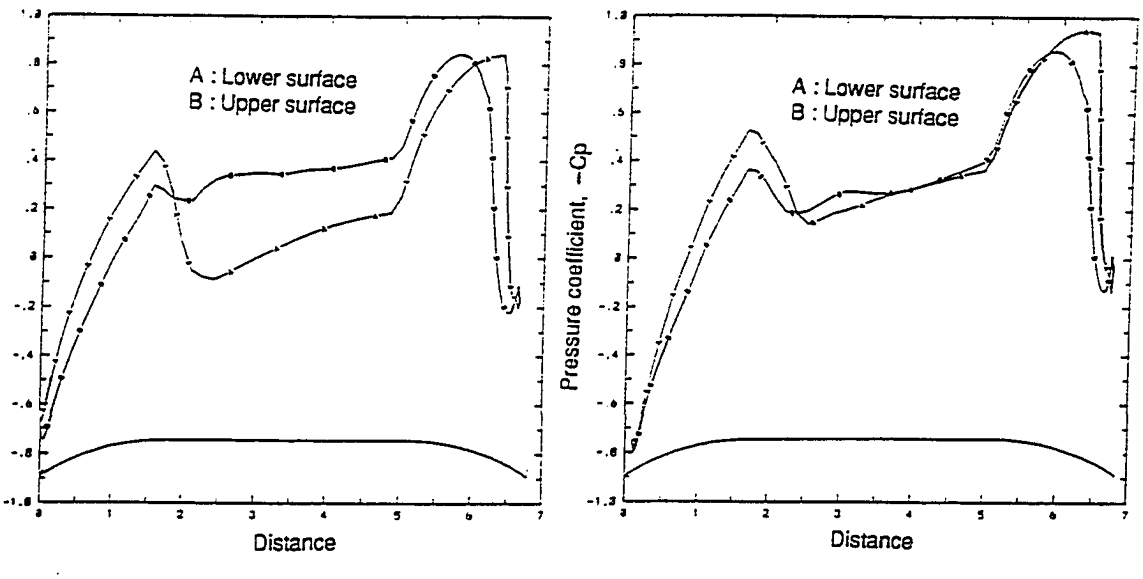


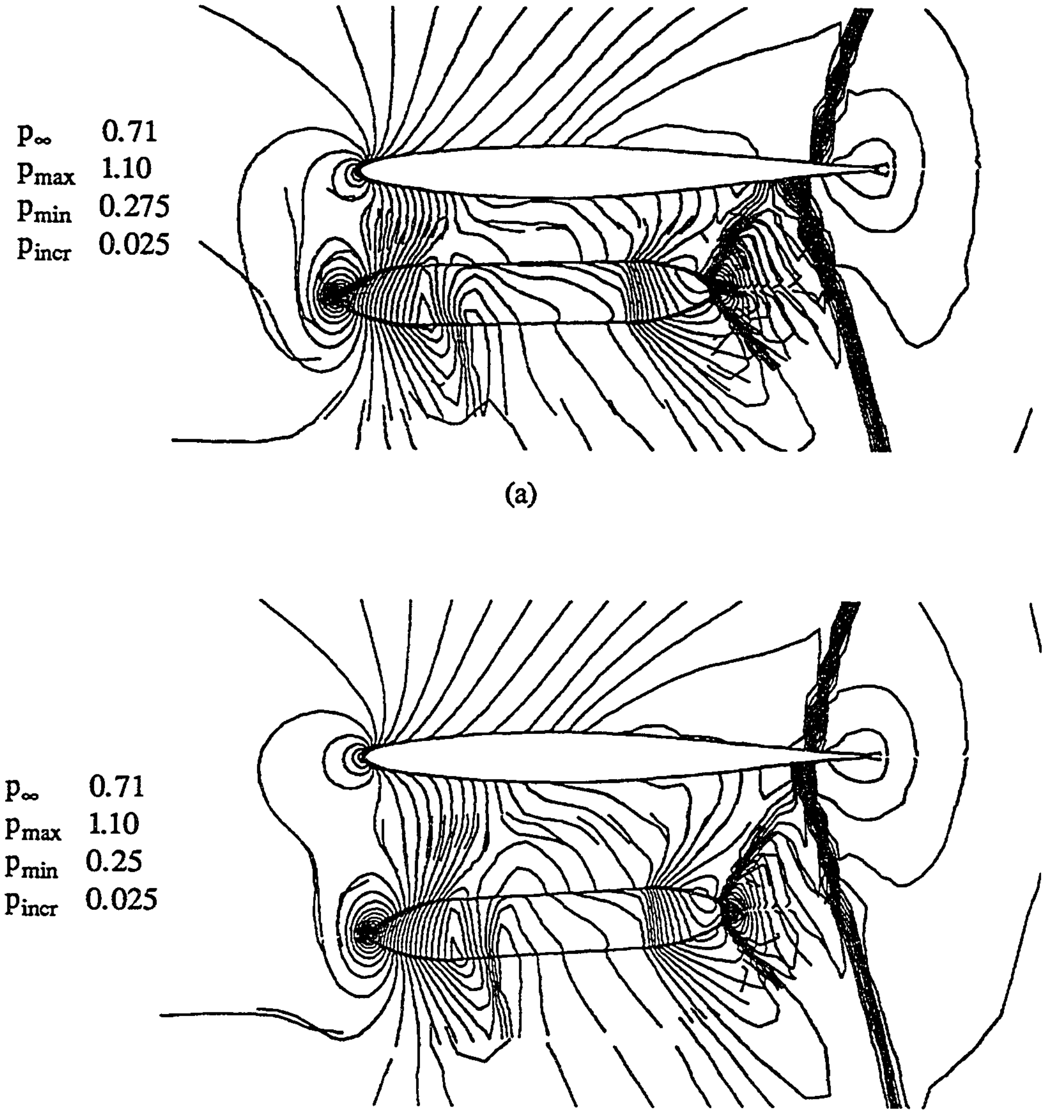
Fig. 7.6: Off surface pressure contours for Case 7.1 through the mid-span plane after 5 time units.



(a)

(b)

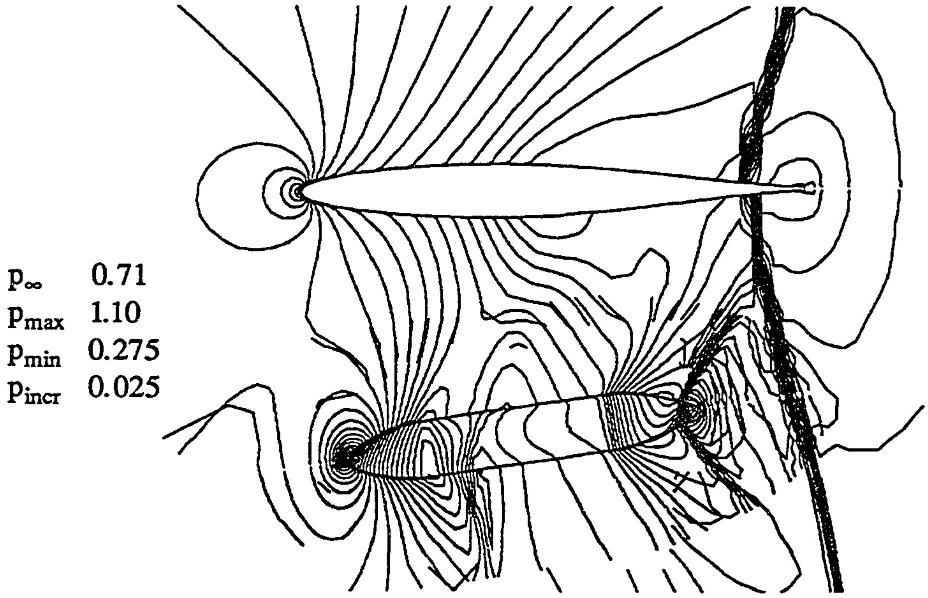
Fig. 7.7: Pressure coefficient distributions on the store, (a) static initial position, (b) after 46 time units.



(a)

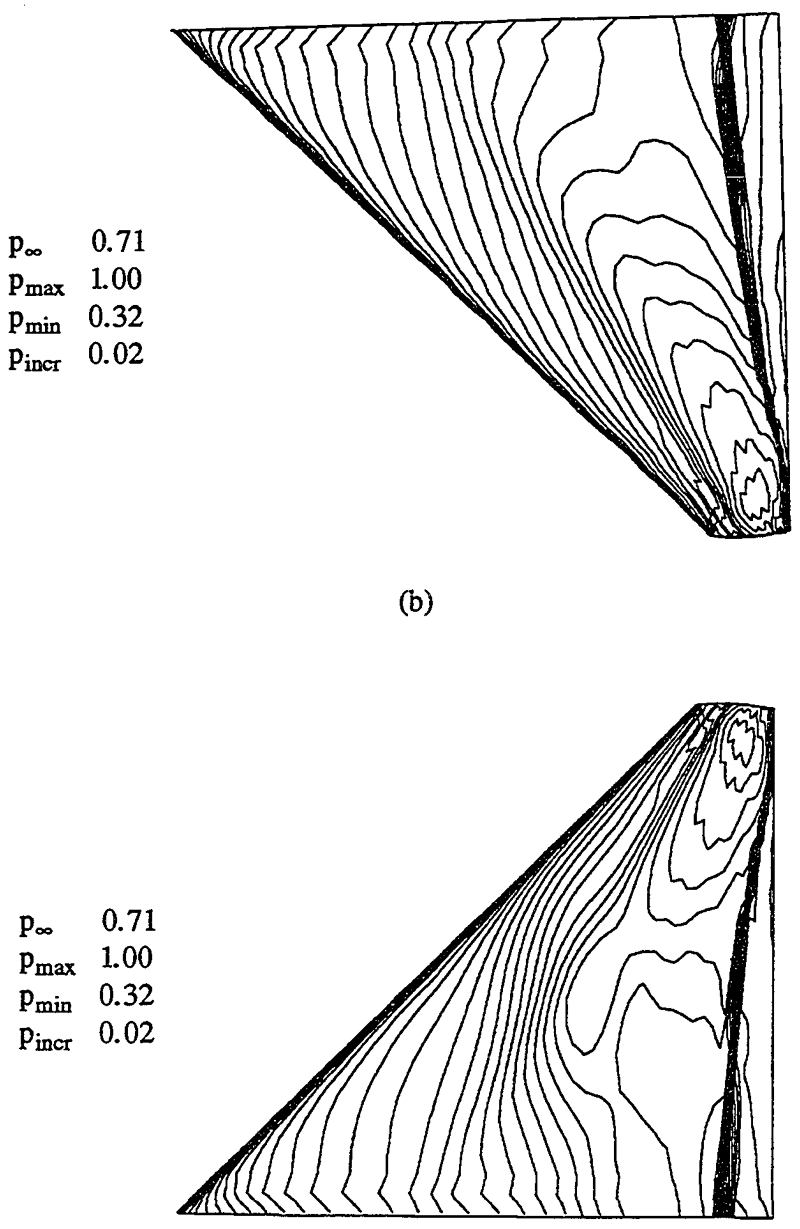
(b)

Fig. 7.8: Off surface pressure contours for Case 7.2 through the mid-span plane at (a) 33 time units, (b) 69 time units.



(a)

Fig. 7.9: Pressure contours for Case 7.2 at 105 time units,
(a) off surface for the wing and store through the mid-span plane.
(b) on the upper surface of the wing,
(c) on the lower surface of the wing.



(b)

(c)

Fig. 7.9: Concluded.

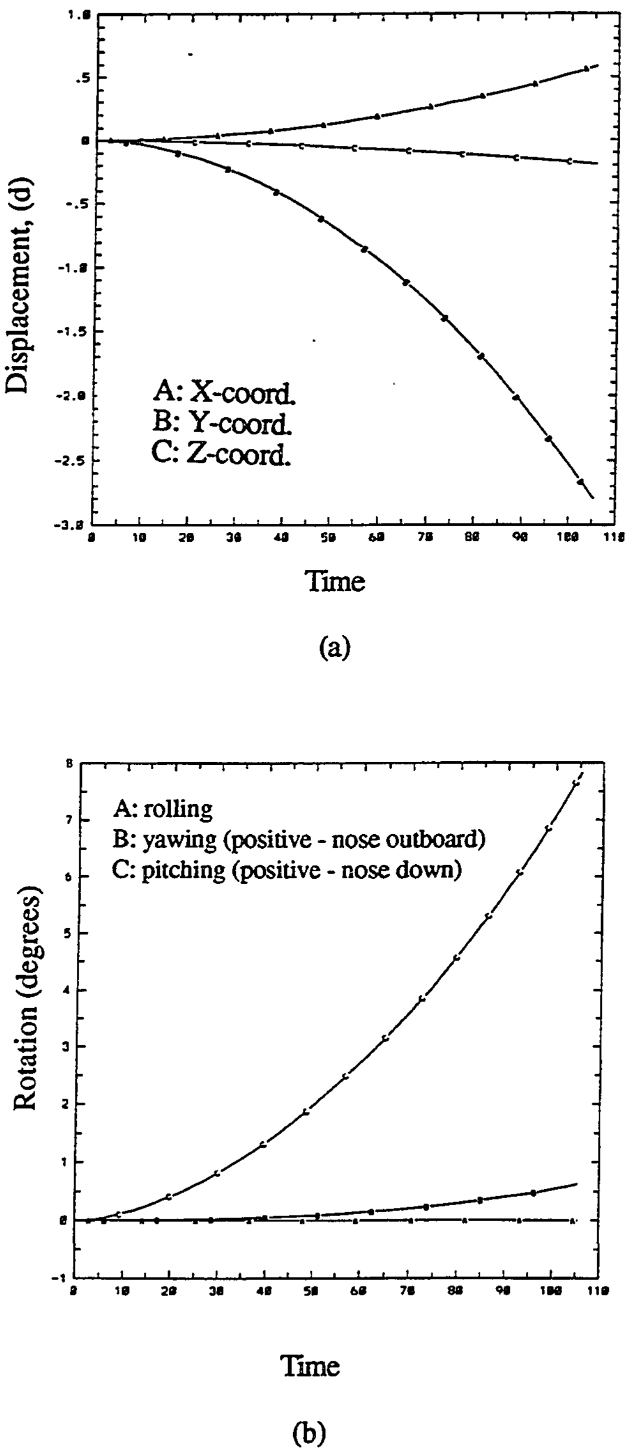


Fig. 7.10: (a) Displacement (x, y, z), (b) rotation (θ_x, θ_y, θ_z), of the store C.G. in time unit for Case 7.2.

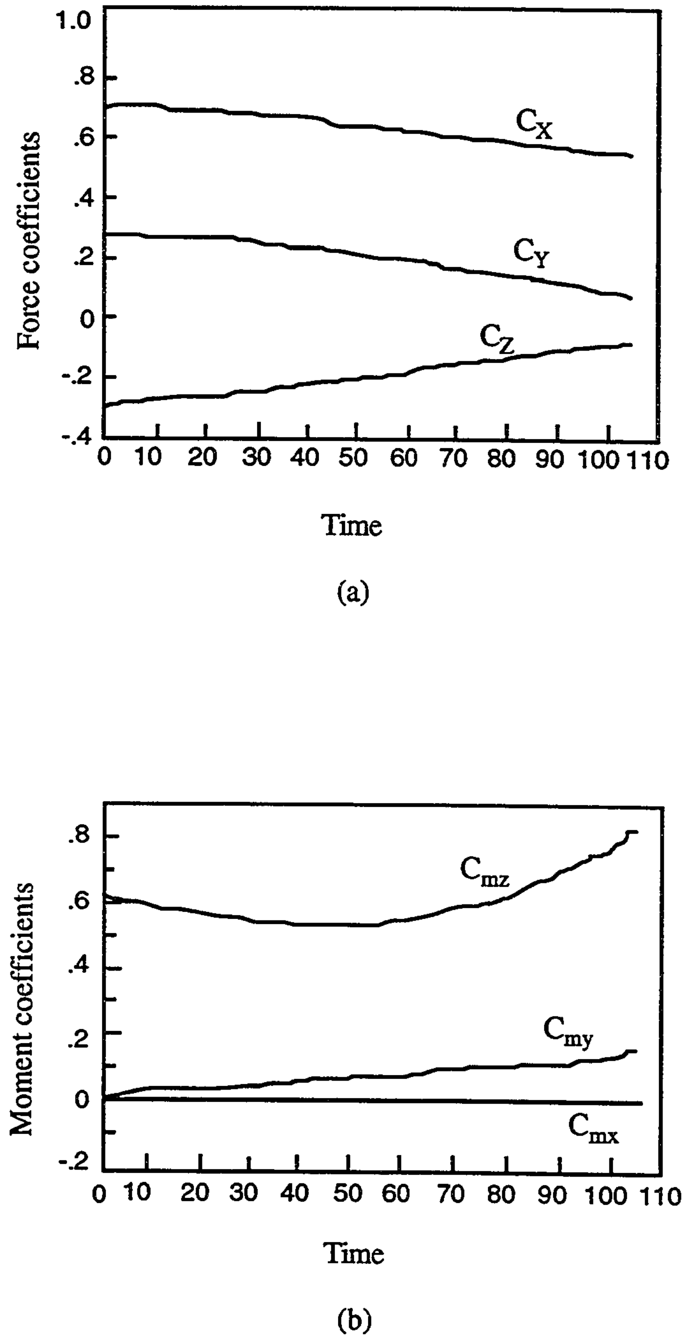


Fig. 7.11: (a) Force coefficients, (b) moment coefficients, of the store in time units for Case 7.2.

Chapter 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions on Accuracy Demonstration Cases

Initially, several error reducing measures were studied and compared with the exact solution of a moving normal shock in a tube. It is concluded that at least a second-order accurate method is needed not only spatially but also temporally. For decomposed computational domains, second-order accuracy is also needed for the intergrid interpolations. This is accomplished, for the present upwind-biased scheme on static and dynamic overlapped grids, using two sets of fringe points. It is also concluded that, to retain the numerical accuracy of unsteady computations, the CFL numbers should be limited to unity, particularly when an alternating-direction-implicit method is used.

The diagonal inversion method is about 1.5 times faster than the block inversion method, and an acceptable solution still can be obtained for unsteady flow with moderate gradients. Hence, the diagonal inversion still has its value for a class of unsteady, moving boundary problems.

The 6-DOF trajectory method is validated through a three-dimensional store separated from a wing case. The computational data show good agreement with the independently obtained results used for the experiments. However, it should be noted that the experimental data provided in [67] is quasi-steady, and this should contribute to the mild discrepancy in the predicated pitching angles.

8.2 Conclusions on Unsteady Airfoil Cases

A method is developed to simulate computationally an unsteady flow past an object engaged in a complex motion. The method employs a composite of subdomain grids, where each subdomain grid can engage in a different type of motion. The composite grid can be a hybrid of degenerate zonal and overlapped grids. The flow equations are solved using the time dependent generalized coordinates in the absolute frame of reference. The absolute values of the primitive variables are transferred across the grid interfaces.

D^3T is demonstrated through three transonic flow simulations. The results compare favorably with the experimental data as well as the computations without using D^3T. The extra work for the information transfer between subdomains and the reduced data vector lengths due to D^3T increase the CPU time by 23 to 70 percent over a single grid approach. Numerical errors can be further reduced by; (1) a judicious hybridization of D^3T, whereby the weaknesses of each method are optimally avoided, (2) decreasing the time steps, (3) refining the grids at the subdomain interfaces, and (4) more frequent updating of the intergrid information.

Simulating the flowfield with the objects in motion, as opposed to a series of solutions where the objects are frozen instantaneously (static), gives more accurate results, captures the boundary-motion-induced flow component, and is more efficient.

8.3 Conclusions on Oscillating Cylinder Near a Vertical Wall Case

A three-dimensional method is developed to computationally simulate an unsteady flow past multiple bodies engaged in a relative motion. The method employs a composite of subdomain grids which are independently generated. The flow equations are solved using the time dependent, generalized coordinates in the absolute frame of reference. The values of the primitive flow variables are directly transferred across the grid interfaces.

The D^3T is demonstrated through the three-dimensional subsonic and supersonic flow simulations of a cylinder pitching sinusoidally in the proximity of a vertical wall. An advantage of D^3T is that each subdomain and its local origin can be defined independently. As such, this method has the capability to resolve the aerodynamics of various unsteady problems involving the relative motion of multiple bodies.

The demonstrative results for a subsonic case indicate a strong interference between a pitching cylinder and a static wall. For a supersonic case, the reflected shock wave does not interact with the cylinder, but it impinges on its sting at the distance it is positioned from the wall. The aerodynamic coefficients of the dynamic cylinder differ significantly from those of the static cylinder. Therefore, it is concluded that performing a number of sequential steady-state computations would not yield adequate results for an unsteady flowfield around an object in relative motion near another object.

8.4 Conclusions on Three-Dimensional Store Separation From a Wing

The method is used to simulate the flowfield history and predict the aerodynamically determined trajectory of a store dropped from its initial position under a wing. The results indicate that the unsteady CFD algorithm, dynamic overlapped grids, and the rigid-body-dynamics algorithm work well in concert to produce the unsteady solution of a complex, three-dimensional problem involving a relative body motion.

The temporal interpolation error has been detected by a carefully designed case. This type of error causes the phase shift which severely affects the numerical solution.

However, there are no effective ways at the present time to reduce this error other than avoiding the overlaying of fringe points of different subdomains.

There is no experimental data to validate these computational results. It should be noted that a fair amount of interference motion (about 2.8 store diameters) is obtained. Therefore, the computational results of the store trajectory should indicate that the unsteady CFD algorithm, dynamic overlapped grids, and the rigid-body-dynamics algorithm work well in concert to produce the unsteady solution of a complex, three-dimensional problem involving relative body motion.

8.5 Recommendations for Future Work

The D^3T has been demonstrated to be a very powerful CFD method. However, there are several options that can further extend the capability of D^3T. They are listed as follows.

1. 'Smart search' (Section 1.1.2) for connectivity can increase the efficiency of the moving overlapped grid connections.

2. The current research is focused on rigid-body problems with non-deforming grids; however, this assumption does not hold for aeroelasticity problems. Therefore, the GCL (Section 2.5) can be incorporated into D^3T for deforming grid problems in order to resolve the aeroelasticity problems.

3. Grid adaptation method [36,71] can further optimize the point distributions for D^3T, hence, it should be incorporated..

4. Extend zonal grid method for moving boundary problems. Zonal grid method is a conservative interpolation algorithm, however, it requires the common interface

between connective subdomains. This method can broaden the options of D^3T by removing the restrictions of the degenerate zonal method.

5. Study conservative interpolation algorithms for overlapped grids. The current overlapped grid method is nonconservative, hence, it allows loss or gain of numerical fluxes across the overlap boundary. Hence, there is a need for a conservative feature.

6. Establish the parallel processing of for D^3T. DDT's have the feature of being easily employed in parallel processing. Hence, the D^3T also can be adopted to the parallel processing with minimum effort and greatly increasing the efficiency.

7. More comprehensive studies of the temporal interpolation error associated with D^3T are needed.

REFERENCES

1. Lessard, V.L., *Domain Decomposition for Multigrid, Finite Volume Flow Solvers*, Master's Thesis, Mechanical Engineering and Mechanics Department, Old Dominion University, Dec. 1989.
2. Atta, E., "Component-Adaptive Grid Interfacing," AIAA-81-0382, Jan. 1981.
3. Atta, E., and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows About Aircraft Configurations," AIAA-82-1017, Jun. 1982.
4. Steger, J.L., Dougherty, F.C., Benek, J.A., "A Chimera Grid Scheme," ASME Symposium on Advances in Grid Generation, FED-Vol. 5, 1983, pp. 59-69.
5. Benek, J.A., Steger, J.L., and Dougherty, F.C., "A Flexible Grid Embedding Technique with Application," AIAA-83-1944, Jul. 1983.
6. Benek, J.A., Donegan, T.L., and Suhs, N.E., "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows," AIAA-87-1126, Jun. 1987.
7. Benek, J.A., Steger, J.L., Dougherty, F.C., and Buning, P.G., "Chimera; A Grid-Embedding Technique," AEDC-TR-85-64, Arnold Engineering Development Center Report, Apr. 1986.
8. Suhs, N.E., "Computations of Three-Dimensional Cavity Flow at Subsonic and Supersonic Mach Numbers," AIAA-87-1208, Jun. 1987.
9. Chesshire, G., Henshaw, W.D., "Composite Overlapping Meshes for the Solution of Partial Differential Equations," IBM Research Report, RC 14355, Jan. 1989.
10. Chesshire, G., and Henshaw, W.D., "Multigrid on Composite Meshes," SIAM Journal Scientific Statistical Computing, Vol. 8, No. 6, Nov. 1987, pp. 914-923.
11. Henshaw, W. D., *Part I: The Numerical Solution of Hyperbolic Systems of Conservation Laws; Part II: Composite Overlapping Grid Techniques*, Ph.D. Dissertation, Department of Applied Mathematics, California Institute of Technology, 1985.
12. Baysal, O., Fouladi, K., Lessard, V. R., "Multigrid and Upwind Viscous Flow Solver on 3-D Overlapped and Embedded Grids," AIAA Journal, Vol. 29, No. 6, June 1991, pp. 903-910.
13. Baysal, O., Lessard, V. R., Fouladi, K., "A Multigrid Domain Decomposition Scheme with Embedded/Overlapped and Zonal Grids," SIAM Conference on Domain Decomposition Methods, Houston, TX, March 20-22, 1989.

14. Baysal, O., and Lessard, V R., "An Overlapped Grid Method for Multigrid Finite Volume/Difference Flow Solvers," NASA-CR-182008, NASA LaRC, Hampton, VA, February 1990.

15. Baysal, O., Fouladi, K., Leung, R. W., Sheftic, J. S., "Interference Flows Past Cylinder-Fin-Sting-Cavity Assemblies," (AIAA) Journal of Aircraft, Vol. 29, No. 2, March/April 1992, pp. 194-202.

16. Fouladi, K. and Baysal, O., "Viscous Simulation Method for Unsteady Flows Past Multicomponent Configurations," (ASME) Journal of Fluids Engineering, Vol. 114, No. 2, 1992, pp. 161-169.

17. Newman, J. C. III, and Baysal, O., "Overlapped-Multiblock Algorithm for Transonic Interference Flow About a Wing-Pylon-Store-Fin Assembly," AIAA Atmospheric Flight Mechanics Conference, AIAA-92-4571-CP, Aug. 1992.

18. Baysal, O., "Navier-Stokes Solutions for Store Separation and Related Problems," Proceedings of NASA Computational Fluid Dynamics Conference, NASA-CP-10038, Vol. 1, pp. 385-410, Moffet Field, CA, March 7-8, 1989.

19. Baysal, O., "Viscous Analyses of Flows Related to Internal Store Carriage and Separation," Proceedings of Third Workshop on Weapons Internal Carriage and Release, Naval Air Development Center, Warminster, PA, April 25-26, 1989.

20. Baysal, O., Fouladi, K., Lessard, V. R., Miller, D. S., "Navier-Stokes Simulations of Internal and External Store Carriage and Separation," Proceedings of 8th JOCG Aircraft/Stores Compatibility Symposium, Ft. Walton Beach, Florida, October 23-25, 1990, pp. 11.1-11.28.

21. Fouladi, K., Baysal, O., Newman, J.C. III, "Hybrid Domain Decomposition for Configurations with Multiple and Nonsimilar Components," Proceedings of 5th International Symposium on Domain Decomposition Methods for Partial Differential Equations, Norfolk, VA, May 6-8, 1991, Society for Industrial and Applied Mathematics, Ch. 39, pp. 476-484.

22. Fouladi, K., *Navier-Stokes Simulations of Flows About Complex Configurations Using Domain Decomposition Techniques*, Ph. D. Dissertation, Mechanical Engineering Department, Old Dominion University, Aug. 1990.

23. Krist, S.L., Thomas, J.L., Sellers, W.L., Kjelgaard, S.O., "An Embedded Grid Formulation Applied to A Delta Wing," AIAA-90-0429, Jan. 1990.

24. Berger, M.J., "On Conservation at Grid Interfaces," SIAM J. of Numerical Analysis, Vol. 24, No. 5, Oct. 1987, pp. 967-984.

25. Thomas, J.L., Walter, R.W., Reu, T., Ghaffari, F., Weston, R.P., Luckring, J.M., "A Patched-Grid Algorithm for Complex Configurations Directed Towards The F/A-18 Aircraft," AIAA 29th Aerospace Sciences Meeting, AIAA-89-0121, Jan. 1989.

26. Batina, J. T., "Unsteady Euler Airfoil Solution Using Unstructured Dynamic Meshes," AIAA-89-0115, Jan. 1989.

27. Schulz, W. D., "Two-Dimensional Lagrangian Hydrodynamics Difference Equations," Methods in Computational Physics, Vol. 3, Academic, New York, 1964, pp. 1-45.

28. Floryan, J. M., and Rasmussen, H., "Numerical Methods for Viscous Flows with Moving Boundaries," Applied Mechanics Review, Vol. 42, No. 12, Dec. 1989, pp. 323-340.

29. Liou, M. S., "An Extended Lagrangian Method," AIAA-93-3305-CP, Jul. 1993.

30. Dougherty, F.C., Benek, J.A., Steger, J.L., "On Applications of Chimera Grid Schemes to Store Separation," NASA-TM-881936, Oct. 1985.

31. Dougherty, F.C., Kuan, J.H., "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme," AIAA 27th Aerospace Sciences Meeting, AIAA-89-0637, Jan. 1989.

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

33. Meakin, R.L., "A New Method for Establishing Intergrid Communication Among Systems of Overset Grids," AIAA-91-1586 CP, 1991.

34. Meakin, R.L., "Computations of the Unsteady Flow About a Generic Wing/Pylon/Finned-Store Configuration," AIAA-92-4568 CP, Aug. 1992.

35. Meakin, R.L., "Moving Body Overset Grid Methods for complete Aircraft Tiltrotor Simulations," AIAA-93-3350 CP, Jul. 1993.

36. Singh, K. P., Newman, J. C. III, and Baysal, O., "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," AIAA 32th Aerospace Science Meeting and Exhibtit, AIAA-94-0058, Jan. 1994.

37. Thomas, P.D., Lombard, C.K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," AIAA Journal, Vol. 17, No. 10, Oct. 1979, pp. 1030-1037.

38. Vinokur, M., "An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws," Journal of Computational Physics, Vol. 81, No. 1, Mar. 1989, pp. 1-52.

39. Obayashi, S., "Freestream Capturing for Moving Coordinates in Three-Dimensions," AIAA Journal, Vol. 30, No. 4, Apr. 1992, pp. 1125-1128.

40. Tamura, Y., and Fujii, K., "Conservation Law for Moving and Transformed Grids," AIAA-93-3365-CP, Jul. 1993.

41. Amsdem, A. A., Ruppel, H. M., and Hirt, C. W., "A Simplified ALE Computer Program for Fluid Flow at All Speeds," Los Alamos National Laboratory Report LA-8095, 1980.

42. Lohner, R., "Adaptive Remeshing for Transient Problems with Moving Bodies," AIAA-88-3736-CP, 1988.

43. Kandil, O. A., and Chuang, H. A., "Unsteady Transonic Airfoil Computation Using Implicit Euler Scheme on a Body-Fixed Grid," AIAA Journal, Vol. 27, No. 8, Aug. 1989, pp. 1031-1037.

44. Kandil, O. A., Chuang, H. A., and Salman, A. A., "Unsteady Flow Computation of Oscillating Flexible Wings," AIAA Structures, Structural Dynamics and Materials Conference, AIAA-90-0937 CP, Apr. 1990.

45. Thomas, J.L., Krist, S.T., and Anderson, W.K., "Navier-Stokes Computations of Vortical Flows Past Low Aspect Ratio Wings," AIAA Journal, Vol. 28, No. 2, Feb. 1990, pp. 205-212.

46. Baysal, O. and Yen, G-W., "Implicit and Explicit Computations of Flows Past Cavities With and Without Yaw," AIAA Paper No. 90-0049, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 8-11, 1990.

47. Baysal, O., Yen, G.W., and Fouladi, K., "Navier-Stokes Computations of Cavity Aeroacoustics with Suppression Devices," Proceedings of DGLR/AIAA 14th Aeroacoustics Conference, Germany, May 1992, pp. 940-948.

48. Baysal, O., Yen, G.W., and Fouladi, K., "Navier-Stokes Computations of Cavity Aeroacoustics with Suppression Devices," Journal of Vibration and Acoustics Vol. 116, No. 1, Jan. 1994, pp. 105-112.

49. Beam, R. M., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, No. 4, Apr. 1978, pp. 393-402.

50. Roe, P.L., "Approximate Riemann Solvers, Parameter Vector, and Difference Schemes," Journal of Computational Physics, Vol. 43, Oct. 1981, pp. 357-372.

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

52. Greenwood, D. T., *Principles of Dynamics*, Prentice-Hall Inc., Englewood Cliff, NJ, 1965, pp. 49-366.

53. Pulliam, T.H., "Characteristic Boundary Conditions for The Euler Equations," Numerical Boundary Condition Procedures, NASA-CP-2201, Oct. 1981, pp. 165-182.

54. Baysal, O., Yen, G-W., "Kinematic Domain Decomposition to Simulate Flows Past Moving Objects," AIAA 29th Aerospace Science Meeting, AIAA-91-0725, Jan. 1991.

55. Baysal, O. and Yen, G-W., "Kinematic Domain Decomposition for Boundary-Motion-Induced Flow Simulations," Proceedings of 5th International Symposium on Domain Decomposition Methods for Partial Differential Equations, Norfolk, VA, May 6-8, 1991, Society for Industrial and Applied Mathematics, Ch. 35, pp. 411-419.

56. Yen, G.W., and Baysal, O., "Computing Unsteady High-Speed Flows Past an Oscillating Cylinder Near a Vertical Wall," AIAA Atmospheric Flight Mechanics Conference, AIAA-92-4653 (revised), Aug. 1992.

57. Yen, G.W., and Baysal, O., "Unsteady Flow Computation Past An Oscillating Cylinder Near a Vertical Wall," (AIAA) Journal of Spacecraft and Rockets Vol. 31, No. 4, July/August 1994.

57. Yen, G.W., and Baysal, O., "Unsteady Flow Computation Past An Oscillating Cylinder Near a Vertical Wall," (AIAA) Journal of Spacecraft and Rockets Vol. 31, No. 4, July/August 1994.

58. Yen, G-W., and Baysal, O., "Dynamic Overlapped-Grid Simulation of Aerodynamically Determined Relative Motion," AIAA 24th Fluid Dynamics Conference, AIAA-93-3018, Jul. 1993.

59. Hsich, K.C., "Assessment of Numerical Techniques for Unsteady Flow Calculations," AIAA-89-1956 CP, June 1989.

60. Ridder, J.P., and Beddini, R.A., "Temporal and Acoustic Accuracy of an Implicit Upwind Method for Ducted Flows," AIAA Journal, Vol. 29, No. 11, Nov. 1992, pp. 1860-1867.

61. Hoffmann, K.A., *Computational Fluid Dynamics for Engineers*, Engineering Education System, Austin, TX, 1989, pp. 493-495.

62. Biedron, R.T., Krist, S.L., Thomas, J.L., and Baysal, O., "Overset Grid Applications at LaRC," Overset Composite Grid Workshop, NASA Ames Research Center, May 4-5, 1992.

63. Van Leer, B., "Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws," NASA CR 172254, 1983.

64. Rumsey, C., *Development of a Grid-Independent Approximate Riemann Solver*, Ph. D. Dissertation, Aerospace Engineering Department, University of Michigan, 1991.

65. Hirsch, C., *Numerical Computation of Internal and External Flows*, Vol. 2, A Wiley-Interscience Publication, p. 204.

66. Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., 'PLOT3D User's Manual,' NASA-TM-101067, Mar. 1990.

67. Heim, E. R., "CFD Wing/Pylon/Finned Store Mutual Interference Wind Tunnel Experiment" AEDC-TSR-91-P4, Arnold Engineering and Development Center, Tullahoma, TN, Jan. 1991.

68. David, S. S., Malcolm, G. N., "Experimental Unsteady Aerodynamics of Conventional and Supercritical Airfoil," NASA-TM-81221, Aug. 1980.

69. Landon, R., "NACA-0012 Oscillatory and Transient Pitching," Compendium of Unsteady Aerodynamic Measurements, AGARD Report No. 702, Aug. 1982, pp. 3.3-3.25.

70. Rumsey, C. L., Anderson, W. K., "Some Numerical and Physical Aspects of Unsteady Navier-Stokes Computations Over Airfoils Using Dynamic Meshes," AIAA-88-0329, Jan. 1988.

71. Baysal, O., Eleshaky, M. E., and Engelund, W. C., 'Computations of Multispecies Mixing Between Scramjet Nozzle Flow and Hypersonic Freestream,' (AIAA) Journal of Propulsion and Power, Vol. 8, No. 2, Mar./Apr. 1992, pp. 500-506.

73. Chamberlain, R., "Influence of Flux Limiting on the Calculation of Supersonic Separated Flows Using Roe's Scheme," 28th Aerospace Science Meeting, AIAA-90-0588, Jan. 1990.

Appendix A

CFD Solution Method

A.1 Finite Volume Formulation

The reason that the equations of fluid flow are solved using finite volume method is that this method begins with the integral equations, the solution obtained using this formulation satisfies the integral conservation law of mas, momentum, and energy. Hence, it retains good accuracy in the presence of flowfields with large flow gradients, such as shocks and contact surfaces.

The governing equations are written in a finite-difference form. However, the equations are solved with the finite-volume scheme the fluxes passing the face of each cell are summed. Any appropriate pressure area terms acting on the faces are added. Then, the result is equated to the time rate of change of the conserved quantity in each respective cell volume. The state variable $Q_{i,j,k}$ is an approximation to the average state in the cell [72] as

$$Q_{i,j,k} = \frac{1}{\Delta V} \int_{(i-\frac{1}{2})\Delta\xi^1,\,(j-\frac{1}{2})\Delta\xi^2,\,(k-\frac{1}{2})\Delta\xi^3}^{(i+\frac{1}{2})\Delta\xi^1,\,(j+\frac{1}{2})\Delta\xi^2,\,(k+\frac{1}{2})\Delta\xi^3} Q\left(\xi^1, \xi^2, \xi^3, t\right) dV \qquad \text{(A.1)}$$

where ΔV is the cell volume.

The finite-volume differencing of the fluid flow equations (Eq. (2.2)) is formulated by integrating them over a control volume

$$\frac{\partial}{\partial t} \int_V Q\, dV + \oint_s \bar{E} \cdot \bar{n}\, ds = 0 \qquad \text{(A.2)}$$

where V(t) is the cell volume and $\bar{n}\,ds$ is a vector element of surface area with outwardly normal $\bar{n}$. The flux $\bar{E}$ can be decomposed into the flux in the stationary frame E_{st} and the contribution due to surface element velocity $\bar{v}$ as

$$\bar{E} = E_{st} - \bar{v}Q \ . \tag{A.3}$$

For Eulerian frames, $\bar{v}$ is equal to zero. Surface integrals are written as the sum of the contributions from six faces of the hexahedron cell. Applying Eq. (2.2) to each cell of the computational domain and the resulting semi-discrete finite-volume form is

$$\left(\frac{\partial Q}{\partial t}\right)_{i,j,k} + \frac{\left[(E_1)_{i+\frac{1}{2}} - (E_1)_{i-\frac{1}{2}}\right]}{\Delta\xi^1} + \frac{\left[(E_2)_{j+\frac{1}{2}} - (E_2)_{j-\frac{1}{2}}\right]}{\Delta\xi^2} + \frac{\left[(E_3)_{k+\frac{1}{2}} - (E_3)_{k-\frac{1}{2}}\right]}{\Delta\xi^3} = 0 \tag{A.4}$$

where $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ for computational domain. Also $Q_{i,j,k}$ represents the conserved variables at the cell center and the fluxes are evaluated at the six cell surfaces.

A.2 Upwind Differencing and Flux Difference Splitting

The inviscid fluxes and the temporal term of the Euler equations form a hyperbolic differential equation system. Upwind space discretization methods model the characteristic nature of these equations in that information at each grid cell is obtained from directions dictated by characteristic theory. The naturally dissipative nature of upwind schemes has made them a prevailing alternative to the central difference scheme, for which artificial dissipation terms are generally needed to overcome oscillations or instabilities arising in regions of high gradients.

In this study, Roe's flux difference splitting is used to construct the upwind differences for the convective and pressure terms. This method accounts for different waves by which neighboring cells interact, including entropy and shear waves. The amplitude and propagation speed of these waves are determined by solving the approximate Riemann variables. However, directions of these waves are assumed normal to the cell surface which makes the scheme grid dependent. The Roe's scheme [50] seeks the exact solution to this approximate equation and distinguishes between the influence of the moving waves. For the initial condition, the one dimensional hyperbolic governing equation can be approximated as

$$\frac{\partial Q}{\partial t}+\tilde{A}\left(Q^L,Q^R\right)\frac{\partial Q}{\partial \xi}=0, \qquad \text{(A.5)}$$

$$Q(\xi,t=0)=\begin{cases} Q^L & \text{for } \xi<0 \\ Q^R & \text{for } \xi>0 \end{cases}$$

During the time marching procedure, the conserved variables are updated subsequently. The interface flux is written as the exact solution to an approximate Riemann problem

$$E_{i+\frac{1}{2}} = \frac{1}{2}\left[E(q_L)+E(q_R)-\left|\tilde{A}\right|(Q_R-Q_L)\right]_{i+\frac{1}{2}} \cdot \qquad \text{(A.6)}$$

Matrix $\tilde{A}$ is formed from matrix A evaluated at Roe-averaged variables. Hence

$$\left|\tilde{A}\right| = \left|A\left(\tilde{Q}\right)\right| \quad \text{and} \quad \tilde{Q} = \tilde{Q}(Q_L,Q_R).$$

A monotone upwind-centered scheme for conservation laws (MUSCL) approach is used to construct the interface flux (Eq. (A.6)). For the primitive variables q, the higher order accurate differencing is defined by

$$\left(q^L\right)_{i+\frac{1}{2}} = q_i+\left\{\frac{1}{4}\left[(1-\kappa)\nabla+(1+\kappa)\Delta\right]\right\}_i \qquad \text{(A.7)}$$

$$\left(q^R\right)_{i+\frac{1}{2}} = q_{i+1}-\left\{\frac{1}{4}\left[(1+\kappa)\nabla+(1-\kappa)\Delta\right]\right\}_{i+1} \qquad \text{(A.8)}$$

where

$$(\Delta)_i = q_{i+1} - q_i \ , \quad (\nabla)_i = q_i - q_{i-1} \tag{A.9}$$

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

The flux limiting algorithm is employed at this scheme to capture the high flow gradients without numerical oscillations. In this study, the minimum-modules limiter [73] is used to maintain monotonicity, where

$$\begin{aligned} \bar{\Delta}_+ &= \max[0, \min(\Delta_+ \mathrm{sgn}\Delta_-, \ \beta\Delta_- \mathrm{sgn}\Delta_+)]\mathrm{sgn}\Delta_+ \\ \bar{\Delta}_- &= \max[0, \min(\Delta_- \mathrm{sgn}\Delta_+, \ \beta\Delta_+ \mathrm{sgn}\Delta_-)]\mathrm{sgn}\Delta_- \\ \beta &= \frac{(3-\kappa)}{(1-\kappa)} \end{aligned} \tag{A.10}$$

A.3 Approximate Diagonalization Inversion

Pulliam et al. [51] proposed a modification for the Roe's flux-difference split scheme that allows spatial factors in each direction to be approximated with a diagonal inversion. This results in a significant saving in the computational time and memory,

For hyperbolic flow governing equations, the flux Jacobian A_i has real eigenvalues and a complete set of eigenvectors. Therefore, the Jacobian matrices can be linearized and diagonalized as

$$A = \frac{\partial E}{\partial Q} = T\Lambda T^{-1} = T\left(\Lambda^+ + \Lambda^-\right)T^{-1} \tag{A.11}$$

where

$$\Lambda^{\pm} = \frac{\Lambda \pm |\Lambda|}{2} \tag{A.12}$$

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

$$\Lambda = \mathrm{diag}(\ \lambda_1\ ,\ \lambda_2\ ,\ \lambda_3\ ,\ \lambda_4\ ,\ \lambda_5\ ) \tag{A.13}$$

where

$$\begin{aligned} \lambda_1\ ,\ \lambda_2\ ,\ \lambda_3 &= \frac{\tilde{U}|\nabla\xi|}{J} \\ \lambda_4 &= \frac{(\tilde{U}+\tilde{a})|\nabla\xi|}{J} \\ \lambda_5 &= \frac{(\tilde{U}-\tilde{a})|\nabla\xi|}{J} \end{aligned} \tag{A.14}$$

where J is the Jacobian transformation matrix, a is the freestream sonic speed, and U is the contravariant velocity normal to the cell interface. The spatial flux, for example in the one-dimensional sweep as Eq. (2.20) can be approximated as

$$\left[I + h_1\delta_{\xi 1} A_1^n\right]\Delta Q^{**} \approx T_{\xi 1}\left[I + h_1\delta_{\xi 1}\Lambda_1\right]T_{\xi 1}^{-1}\Delta Q^{**} + \vartheta(\Delta t^2) \tag{A.15}$$

where $h_1 = \dfrac{\mu\Delta t}{1+\nu}$.

With further decomposition, the sweep then becomes

$$T_{\xi 1}\left[I + h_1(\delta_{\xi 1}^{-}\Lambda^{+} + \delta_{\xi 1}^{+}\Lambda^{-})\right](T_{\xi 1}^{-1}\Delta Q^{**}) = -T^{-1}(\mathrm{R.H.S.}). \tag{A.16}$$

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

$$\begin{aligned} &-\Lambda^{+}\left(M_{i-1/2}, Q_{i-1}\right)\left(T^{-1}\Delta Q^{**}\right)_{i-1} + [I\Lambda^{+}\left(M_{i+1/2}, Q_i\right) - \Lambda^{-}\left(M_{i-1/2}, Q_i\right)]\left(T^{-1}\Delta Q^{**}\right)_i \\ &+\Lambda^{-}\left(M_{i+1/2}, Q_{i+1}\right)\left(T^{-1}\Delta Q^{**}\right)_{i+1} = -T_i^{-1}(\mathrm{R.H.S.}), \end{aligned} \tag{A.17}$$

The matrix terms M and the state variables Q are evaluated at the cell interfaces and cell center, respectively. The diagonalization of Roe's flux-difference splitting does affect the time accuracy (the error term in Eq. (A.15) is $\vartheta(\Delta t^2)$) of the algorithm. It reduces the scheme to, at most, first-order in time and gives time accurate shock calculations a nonconservative feature [51], i.e., error in shock speeds and shock jumps. However, the steady-state solution is fully conservative because the steady state equations (RHS) are unmodified. Also, computational experiments have shown that the convergence and stability limits of the diagonal algorithm are essentially identical to that of the unmodified algorithms.

Appendix B

Overlapped Grid Method

The overlapping scheme involves the automatic connection of multiple, overset grids, and the use of different solution procedures for different subdomain grids. The overlapping scheme is one in which a major grid covers the entire flow region, and minor grids are then overset on the major grid to resolve secondary features of the configuration. The minor grids are fully or partially overlapped without, requiring the mesh boundaries to join in any special way. Communications between the major and minor grids occur within the overlap regions. The overlapping method has been successfully demonstrated on many geometries for inviscid and viscous flows.

To obtain a logical sequence of grid communications between overlapped grids, a form of grid hierarchy is needed. An order of hierarchical form between the grids allows the interaction of appropriate grids, simplifies the development of the data structure required for this interaction, and limits the search to locate points in other grids for the purpose of interpolation. Shown in Fig. B.1 is an example of such a hierarchical grid arrangement. The basic algorithm of overlapping/embedding scheme [1] is highlighted as follows: (1) establishing the proper lines of communication among the grids through appropriate data structure; (2) constructing holes within grids (Fig. B.2); (3) identifying points with holes and illegal zones (Fig. B.3); (4) locating points from which outer and hole boundary values can be interpolated (Fig. B.4); (5) evaluating interpolation parameters.

The key process is the trilinear interpolation used in the intergrid communication of conserved or primitive variables. A simple example is illustrated as follows. Given a fringe cell in G_l, and the corresponding target cell in G_{l+1}, a search is then conducted to locate seven other cells in G_{l+1} near the target cell. The objective is to form a hexahedron that has the seven cell centers and target cell as the vertices, such that the hexahedron includes the fringe cell of G_l. The information is transferred from the eight cells, that define the vertices of the interpolation cell of G_{l+1}, to the fringe cell of G_l using trilinear interpolation. The transformation/interpolation equation is written as

$$Q = a_1 + a_2\xi + a_3\eta + a_4\zeta + a_5\xi\eta + a_6\xi\zeta + a_7\eta\zeta + a_8\xi\eta\zeta \tag{B.1}$$

where a_i, i=1,....8 are coefficients depending on the values of Q_i at the vertices of the unit cube:

$$\begin{aligned}
a_1 &= Q_1 \\
a_2 &= -Q_1 + Q_2 \\
a_3 &= -Q_1 + Q_4 \\
a_4 &= -Q_1 + Q_5 \\
a_5 &= Q_1 - Q_2 + Q_3 - Q_4 \\
a_6 &= Q_1 - Q_1 - Q_5 + Q_6 \\
a_7 &= Q_1 - Q_4 - Q_5 + Q_8 \\
a_8 &= -Q_1 + Q_2 - Q_3 + Q_4 + Q_5 - Q_6 + Q_7 - Q_8 \ .
\end{aligned} \tag{B.2}$$

The trilinear interpolation can only be used on cubes. Hence, each interpolation cell containing a fringe cell at which a function value is to be interpolated is mapped to a unit cube using an isoparametric mapping. The isoparametric equations mapping the interpolation space to the physical space is given by

$$\bar{X} = \Sigma\, a_i\, \bar{\xi}_i \ , \tag{B.3}$$

The values for ξ, η, and ζ corresponding to the fringe cell are determined iteratively by applying the Newton's method of locating roots of a set of algebraic equations. The system of algebraic equations (B.3) can be written in the following form,

$$\begin{aligned} \bar{X} &= \bar{G}(\xi, \eta, \zeta) = \bar{G}(\bar{\xi}) \\ \bar{R} &= \bar{G}(\bar{\xi}) - \bar{X} = 0 \end{aligned} \tag{B.4}$$

Newton's method gives

$$\bar{\xi}^{n+1} = \bar{\xi}^{n} - \left[M_{i,j}^{n}\right]^{-1} \bar{F}\left(X, \bar{\xi}\right)^{n} \tag{B.5}$$

for each iteration, where

$$M_{i,j} = \frac{\partial F_i}{\partial \xi_j} \tag{B.6}$$

Overlapping/embedding method gives the most freedom for grid generation task, however, there are several drawbacks of this method : (i) the technique requires an overlap region between subdomains, which may not always be easily available; (ii) the accuracy of boundary data transfers depends on the interpolation procedure, whether it be conservative or nonconservative; (iii) the accuracy and convergence speed of the solution depend indirectly on the degree of overlap of the grids relative to the size of the subdomains. It should be noted that the overlapping scheme uses the nonconservative trilinear interpolation approach for intergrid communication.

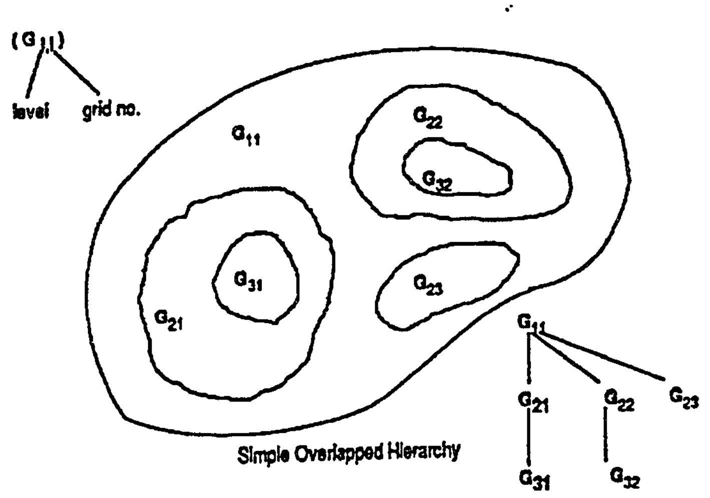


Fig. B.1: Hierarchy ordering for intergrid communications.

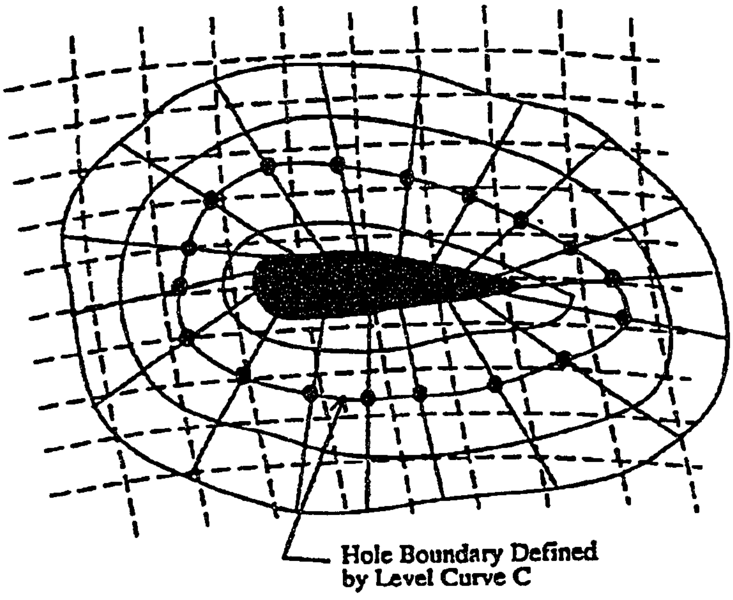


Fig. B.2: Sketch of an initial composite mesh hole boundary surface.

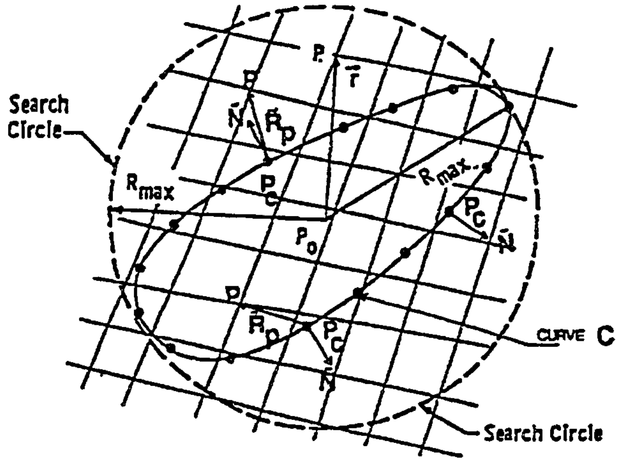


Fig. B.3: Sketch of hole search method.

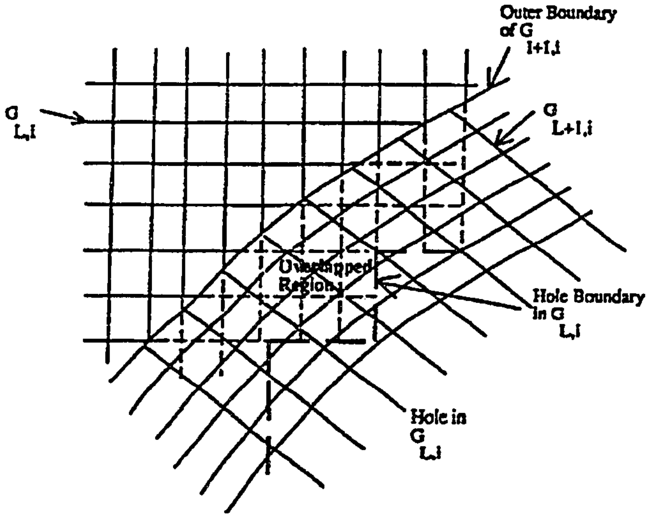


Fig. B.4: Sketch of the hole and outer boundary of a composite mesh.