Summer 1995

Dynamic Unstructured Method for Prescribed and Aerodynamically Determined Relative Moving Boundary Problems

Kamakhya Prasad Singh
Old Dominion University

DYNAMIC UNSTRUCTURED METHOD FOR PRESCRIBED AND AERODYNAMICALLY DETERMINED RELATIVE MOVING BOUNDARY PROBLEMS

by

Kamakhya Prasad Singh

B.Sc., Engineering, July 1987, Chemical Engineering Department
Regional Engineering College, Rourkela, Orissa, India.

M.E., December 1991, 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
ENGINEERING MECHANICS
OLD DOMINION UNIVERSITY
August 1995

Approved by

Oktay Baysal (Director)	Colin P. Britcher
Robert L. Ash	Neal T. Frink

ABSTRACT

Kamakhya Prasad Singh
Old Dominion University, 1995
Director: Dr. Oktay Baysal

A new methodology is developed to simulate unsteady flows about prescribed and aerodynamically determined moving boundary problems. The method couples the fluid dynamics and rigid-body dynamics equations to capture the time-dependent interference between stationary and moving boundaries. The unsteady, compressible, inviscid (Euler) equations are solved on dynamic, unstructured grids by an explicit, finite-volume, upwind method. For efficiency, the grid adaptation is performed within a window around the moving object. The Eulerian equations of the rigid-body dynamics are solved by a Runge-Kutta method in a non-inertial frame of reference. The two-dimensional flow solver is validated by computing the flow past a sinusoidally-pitching airfoil and comparing these results with the experimental data. The overall methodology is used for two two-dimensional examples: the flow past an airfoil which is performing a three-degrees-of-freedom motion in a transonic freestream, and the free-fall of a store after separation from a wing-section. Then the unstructured mesh methodology is extended to three-dimensions to simulate unsteady flow past bodies in relative motion, where the trajectory is determined from the instantaneous aerodynamics. The flow solver and the adaptation scheme in three-dimensions are validated by simulating the transonic, unsteady flow around a wing undergoing a forced, periodic, pitching motion, and comparing the results with the experimental data. To validate the trajectory code, the six-degrees-of-freedom motion of a store separating from a wing was computed using the experimentally determined force and moment fields, then comparing with an independently generated trajectory. Finally, the overall methodology was demonstrated by simulating the unsteady flowfield and the trajectory of a store dropped from a wing. The methodology, its computational cost notwithstanding, has proven to be accurate, automated, easy for dynamic gridding, and relatively efficient for the required man-hours.

TABLE OF CONTENTS

DEDICATION

Dedicated to my parents
Dibya Kishor Singh and Vijay Lakshmi Singh

ACKNOWLEDGMENT

The author wishes to express his sincere appreciation to his advisor Dr. Oktay Baysal for his invaluable guidance and support during the entire course of studies. I would like to express my sincere gratitude to my committee members Dr. N. T. Frink, Dr. R. L. Ash, and Dr. C. P. Britcher for providing their useful comments and suggestions during the course of the study.

I would like to thank Mr. A. M. Thomas, Dr. S. Pirzadeh, and Dr. P. Parikh for providing essential support in generating the computational grids. I would like to extend my heart-felt thanks to my graduate student colleagues for providing me moral and technical support through out the course of my study. In particular, I would like to acknowledge Mr. James C. Newman III, Dr. G. W. Burgreen, and Dr. G. W. Yen for their invaluable comments, suggestions, and discussions.

Special thanks are due to Dr. E. Oktay, Mr. A. H. Ibrahim, Mr. M. J. Pandya, Mr. C. C. Item, Ms. Florence Vanel, and Mr. D. K. Kaushik for their generous support and encouragement through out the endeavor. I would like to express my sincere appreciation to Mr. R. K. Mohanty, Dr. V. M. Korivi, Mr. S. J. Shah, Dr. G. Vittal, Dr. V. Maroju, Mr. P. D. Patwa, Mr. G. Koganti, Mr. N. Jaunky, Mr. A. Gulati, and all my other friends for their support and encouragement.

I would like to extend my sincere gratitude to my cousins Mr. Alok K. Verma, Dr. Arun K. Verma, and sisters-in-law Mrs. Rashmi Verma and Mrs. Renuka Verma for their love and support through out my course of study.

Finally, I would like to express my deepest appreciation, love, and gratitude to my parents, sister, brother-in-law, parents-in-law, wife, and other members of my family for their constant support and encouragement.

The work was supported by NASA Langley Research Center under grants NAG-1-1150 and NAG-1-1499. Technical monitors were David S. Miller, James M. Luckring, and James L. Thomas.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

a	:	speed of sound
a_t	:	contravariant face speed
A_{ij}	:	cell face area
A_i^x, A_i^y, A_i^z	:	projection of cell i onto x, y, and z directions respectively
ALE	:	arbitrary-Lagrangian-Eulerian description
AS	:	airfoil store configuration
CFD	:	computational fluid dynamics
CFL	:	Courant number (from Courant-Friedrichs-Lewy condition)
C_L	:	lift coefficient
C_N	:	normal force coefficient
C_p	:	pressure coefficient
CPU	:	central processing unit
d	:	store diameter
DOF	:	degree-of-freedom
e	:	total energy
FDS	:	flux difference splitting
FVS	:	flux vector splitting
$\bar{F}$	:	flux vector in cartesian coordinates
GCL	:	geometric conservation law

$\bar{I}$	:	moment of inertia matrix
k_{ij}	:	spring stiffness
k	:	reduced frequency
M	:	moment
M_∞	:	freestream Mach number
m	:	mass
$\hat{n}$	:	outward facing unit normal
p	:	pressure
		parameter to control spring stiffness
$\vec{Q}$	:	vector of conserved variables
$\vec{q}$	:	vector of primitive variables
R_i	:	residual
$R^{\pm}$	:	Riemann invariants
$\vec{r},\vec{R}$	:	position vectors
$\Delta\vec{r}$	:	radius vector
Δr	:	magnitude of radius vector
SOA	:	sinusoidally oscillating airfoil
$\tilde{T},\tilde{T}^{-1}$	:	right and left eigen-vectors
t	:	time
u, v, w	:	Cartesian velocities
U, V, W	:	contravariant velocities
$\vec{V}$	:	velocity vector
V_i	:	volume
WS	:	wing-store configuration

$\bar{x}$	:	position vector
x,y,z	:	non-inertial coordinate system
		grid coordinates
X,Y,Z	:	inertial coordinate system

Greek symbols

α	:	angle-of-attack in degrees
ψ,θ,ϕ	:	Euler angles (yaw, pitch, roll)
γ	:	specific heat ratio
ρ	:	non-dimensional density
$\bar{\omega}$	:	angular velocity vector
↑↓	:	up and down motion

Chapter 1

INTRODUCTION

1.1 Motivation

Due to the increased pressures on cost and time incurred to generate and test existing as well as conceptual configurations, computational fluid dynamics (CFD) has emerged as a crucial technology for the development of advanced aerospace vehicles. Furthermore, for unsteady flow fields involving moving bodies, experimental facilities are scarce and experimental methods are very limited in their capabilities. Thus, increased importance is being placed on unsteady CFD methods as may be the only approaches available for predicting transient phenomena. These simulations also require the analysis of non-trivial and geometrically complex configurations. Some typical examples, where the prediction of dynamic loads, moments, and trajectories are essential, include: flight maneuvers, store (or missile) separation sequences, escape-pod ejections, detachment of multistage-rocket components, and separation of booster tanks from the space shuttle. To answer this demand, new numerical algorithms and grid generation methods need to be developed, and existing ones made more efficient and robust.

Unstructured grid methods have the potential to handle these complex geometries somewhat easier than their structured grid counterparts. This can be attributed to the fact that triangles and tetrahedra are the simplest geometrical shapes possessing area and volume, respectively. As such, they are capable of discretizing irregularly shaped domains more efficiently and with less effort. Hence, an unstructured approach has been the starting point for the present study. For the problems which involve moving bodies, some method of tracking the body dynamics must also be implemented [1].

Computational fluid dynamics has become reasonably mature for steady flows. However, there is a strong need for advancements to compute unsteady flows and, in turn, the flows involving moving boundaries. In simulating a flowfield involving a multicomponent configuration, with one or more components engaged in a relative motion, there are at least four levels of assumptions that can be made for the incident-flow and solid-surface interaction [2]. From the least to the most accurate, they are: 1) All the moving components are assumed to be instantaneously frozen, and at each instant, either a steady-state or unsteady computations are performed; 2) All the moving components are assumed to be engaged in the same rigid-body motion, and the complete computational grid is assigned this motion during the unsteady flow analyses [3]; 3) Each moving component is assigned its own rigid-body motion, but it is assumed to be known, so it can be prescribed as input to the unsteady flow computations [4-7]; 4) Beyond and above level 3, the trajectory is determined from the instantaneous flowfield using the principles of rigid-body dynamics [1, 2, 8-10], i.e. aerodynamically determined.

The proper modeling of complex unsteady moving boundary problems pose a great challenge to the computational fluid dynamicist. This has been the main impetus for the present study. Numerous approaches have been suggested and implemented to simulate the unsteady phenomena associated with these problems. There are two major approaches to tackle this kind of problems. One approach is based on the frame of reference (Eulerian or Lagrangian or Mixed Eulerian and Lagarangian approach), and the other is based on the type of grid employed (structured or unstructured grids). The present study employs an Eulerian approach.

1.2 Literature Survey

1.2.1 Unstructured Grid Methods

Unsteady applications of structured-grids methods to complex configurations require sophisticated strategies, such as, blocked, patched, overlapped or hybrid type grids, that may complicate the solution algorithm. Thus, unstructured grids provide a viable alternative to the structured grid approach.

Unstructured grid methods have gained a lot of popularity in the recent years [11] to solve the equations of fluid flow. This is because of the advantages these methods offer in comparison to the traditional structured grid approach. The primary advantages of these methods are: (i) They can model very complex geometries such as a complete aircraft configuration more easily than their structured counterparts. (ii) These methods lend themselves naturally to adaptation and mesh refinement, thereby predicting the physics of the flow more accurately. Grid points can be added to the high gradient regions of the flow with considerable ease resulting in greater spatial accuracy at an affordable computational cost.

The generation of unstructured grids can be broadly classified into three categories: (i) triangulation of structured grids; (ii) Delaunay triangulation [12]; (iii) Advancing front technique [13, 14]. The triangulation of an existing structured grid is simpler and quicker but does not exploit the inherent advantages of the unstructured grids effectively. Delaunay triangulation requires an initial distribution of the nodes and subsequently connects them to form unstructured meshes. The advancing front technique does not require any initial segment or point distribution, but generates the points as the unstructured mesh is being generated. Hence the method of Delaunay triangulation is more efficient but lacks the self sufficiency and grid quality of the advancing front method. In the present study, the computational mesh was created using an unstructured grid generation package, VGRID3D, which is based on the advancing front technique.

1.2.2 Methods for Moving Boundary Problems

A great deal of work has been done to solve the unsteady moving boundary problems using structured grids. The most promising approach has been the domain decomposition technique [4, 5, 8-10, 15]. The three basic types of domain decomposition techniques are: multiblock, zonal, and overlapped methods. The dynamic overlapped method has proven to have the potential to simulate complex moving boundary problems that include bodies in relative motion. This procedure permits each sub-domain to be meshed independently, thereby reducing the grid generation task especially for complicated flow regions. The main drawbacks of this methodology are that it is not conservative, at least as it has been implemented so far, and that it requires an overlapped region between subdomains which is not possible at all times. It is important to note that the geometric conservation law (GCL) [16, 17] is not needed for the dynamic overlapped method since the computational grid moves like a rigid body with the moving body.

The two basic approaches which have been used to discretize the fluid region by finite volume or finite difference methods are the Eulerian and Lagrangian methods. In the following section, a discussion about both methods are given and their advantages and disadvantages are weighed with respect to each other. Also, some of the recent work using these approaches to model moving boundary problems has been reported.

Lagrangian methods have been used successfully to simulate unsteady moving boundary problems [18, 19]. In this approach, the grid nodes are fixed to the fluid particles and they move with the fluid. Thus, each computational cell is associated with the same fluid element. The method has at least three major advantages : (i) Tracking of material interface is easier; (ii) Implementation of the interfacial boundary conditions is much easier; (iii) Absence of numerical diffusion reduces the numerical error associated with it. The main limitation of this approach is its inability to cope easily with strong distortions. In other words, these methods are limited to cases where mesh tangling does not occur. One of the major problems is the numerical inaccuracies generated due to the highly irregular meshes. There are two solutions to alleviate this problem: one is rezoning where the distorted mesh is mapped into a more regular mesh, and the second is a reconnection in which the topology of the mesh points are modified such that the mesh points acquire new neighbors.

The Eulerian approach [18] treats the computational mesh as a fixed reference frame through which the fluid moves. In this approach, the coordinate system is stationary in the fixed reference frame or moves in a prescribed manner to account for the continously changing shape of the solution domain. Thus, the grid motion is independent of the motion of the fluid. The main advantages of this method over the Lagrangian methods are: 1) Its ability to handle fluid motion undergoing large distortions, and 2) Easy formulation of the method. Owing to the above factors, this method has been used widely [3, 4, 5, 8] to simulate unsteady moving boundary problems. The main limitations of this approach is the difficulty of tracking the interface accurately. Other issues which have not been settled for the Eulerian approach are the finding of an appropriate formulation of the field equations and also the best suited numerical method (finite volume or finite element). Since the fluid particles are free to cross the grid lines in an Eulerian approach, numerical mixing and diffusion are convected across the cell interface which may sometimes affect the solution accuracy severely. The error associated with approximating the convective terms translates into the numerical diffusion. One of the other numerical limitations of this approach is that the contact/shear layer is smeared with the elapse of time and distance. Despite the above factors, the Eulerian approach has enjoyed the confidence of CFD practitioners as it offers convenience and simplicity both conceptually and geometrically.

The shortcomings of the purely Lagrangian and purely Eulerian methodology have prompted researchers to come up with an integrated approach [18, 20]. The integrated approach, better known as Arbitrary Lagrangian Eulerian approach (ALE) has been implemented successfully for two dimensional moving boundary problems [20]. This method embraces the positive features of the Eulerian and Lagrangian method. This approach has no basic dependence on particles and the computational mesh is treated as a reference frame that traverses with an arbitrary velocity which is different from the grid velocity.

In recent years considerable amount of attention has been given towards solving unsteady moving boundary problems using the structured grid approach. The approach which has shown promise, especially for problems with bodies in relative motion, is the dynamic overlapped/embedded scheme or the dynamic Chimera scheme.

Meakin and Suhs [21] have successfully extended the Chimera scheme for unsteady problems involving multiple bodies in relative motion. This method incorporated the unsteady Chimera technique with an implicit, approximately factored, finite difference scheme for the unsteady thin-layer Navier-Stokes equations. Meakin [15] has applied the unsteady Chimera scheme to simulate the complex unsteady flow about the wing/pylon/finned store configuration. Here, unsteady computations have been carried out over a finned store separating from a wing and pylon where the trajectory of the store was determined by aerodynamic forces and moments.

Yen and Baysal [4, 5, 9, 10, 22] have successfully carried out the unsteady calculations using a very effective method called the dynamic domain decomposition technique. An Eulerian approach was also incorporated. The dynamic domain decomposition method has the advantage of treating each subdomain differently, and different solution methods can be incorporated for different subdomains. An interpolation scheme is necessary for the communication between subdomains. Also this method lends itself to parallel processing. One of the other advantages of this dynamic overlapped scheme is its potential to handle large amplitude motions without much need for remeshing or regeneration. This is because the overlapped grid is moved like a rigid body. The strength of this method has been demonstrated by successfully performing the complex unsteady flow simulation around a wing-store (WS) configuration [9]. Here, the complex flowfield around a store separating from a delta wing has been captured by employing the dynamic overlapped scheme. The trajectory of the store was determined aerodynamically. This method has proved to be an effective and robust method for moving body problems involving multiple bodies where at least one body is in relative motion.

Arabshahi et al. [23] have developed a multiblock approach to solve complex unsteady moving boundary problems. They have solved the three-dimensional unsteady Euler equations about a wing/pylon/store configuration and demonstrated the limitation of a static domain decomposition technique for moving boundary problems, where the grid lost its integrity very quickly with the continued motion.

The methods which have been discussed so far are based on an Eulerian approach. Another very promising method which is based on a Lagrangian approach is by Kandil and Chuang [24, 25]. In this approach, the governing equations for fluid flow were derived in the moving frame of reference. During the time accurate calculations, the Navier-Displacement equations were solved sequentially for the grid deformation. For rigid body motion, this method eliminates the need to compute the grid motion at every time step. The method has been implemented successfully in modeling the unsteady flowfield around an oscillating delta wing. Since the grid is not moved as a whole but rather adapted to the motion, this method is limited to small-amplitude motions, such as, those encountered in aeroelasticity problems.

Until now, the discussion has been focused on dynamic structured grid methods for solving unsteady moving boundary problems. Lately, some inroads have been made by solving this class of problems via the method of unstructured grids. The most notable amongst them have been the works by Löhner [26, 20], Batina [3], Probert et al. [27, 28] and Peraire et al. [29, 30].

Löhner et al.[26] have demonstrated a method where they have combined adaptive remeshing techniques, unsteady flow solver, and rigid-body dynamics to simulate a fully coupled fluid-rigid body interaction problem. The method uses the finite element method based on the Arbitrary-Lagrangian-Eulerian (ALE) formulation. This method has been applied successfully in two-dimensions. Some of the application problems where this method has been used are: simulation of the unsteady flowfield around an object falling into a supersonic freestream and the motion of a ramp in a supersonic freestream. Löhner has extended this approach further in three dimensions to simulate the store separation problem [31].

Probert et al. [27, 28, 30] have developed an adaptive finite element technique for time dependent compressible flows in two-dimensions. This approach is similar to the approach by Löhner except for the fact that a partial remeshing technique has been employed, as opposed to the full remeshing and H-refinement used by Löhner. An Eulerian approach is used for most of the flowfield but a Lagrangian view-point is used in the vicinity of the moving boundaries. The methodology has been tested in two-dimensions by performing unsteady calculation around a space shuttle separating from a rocket booster.

Batina et al. [3, 32] have successfully implemented an explicit and an implicit solution algorithm for unsteady moving boundary problems. A dynamic mesh algorithm has been employed to simulate the unsteady flowfield around oscillating airfoils. This method has been extended successfuly in three dimensions and the dynamic mesh algorithm and flow solver tested by performing an aeroelastic analysis on a complex aircraft [33]. The capabilities include mesh enrichment and coarsening. However, a robust remeshing technique is needed to handle bodies in relative motion.

1.3 Objectives of Present Work

The primary objective of this research work was the development of a dynamic unstructured grid method capable of determining aerodynamically the motion and the related unsteady flowfield of released bodies. A list of the objectives are given below:

(i) Development of the static version of the cell-centered explicit finite volume, Euler flow solver, USM3D (version 1.0) into an unsteady flow solver.

(ii) Development of a robust and efficient dynamic mesh algorithm that can handle complex moving boundary problems.

(iii) Development of a trajectory code based on the rigid-body dynamics equations to predict the trajectory of a body under the influence of gravity and aerodynamic forces and moments.

(iv) Coupling the unsteady flow solver, dynamic mesh algorithm, and the trajectory code to obtain a dynamic unstructured methodology which is capable of simulating prescribed and aerodynamically determined relative-moving boundary problems.

(v) Validating the dynamic, unstructured code in two and three dimensions by performing unsteady calculations on configurations for which experimental data are available.

(vi) Demonstrate the present methodology via representative cases.

1.4 Outline of Dissertation

The sequence of the dissertation is as follows. Chapter 2 presents the basic equations governing the fluid flow and the rigid-body dynamics. Here, the governing equations are described and the appropriate assumptions to arrive at these equations are stated. Chapter 3 describes the solution algorithms needed to solve these equations. Here the spatial and temporal discretization methods for the fluid flow equations along with the initial and boundary conditions, are highlighted in detail. Also, the dynamic mesh algorithm which includes the grid adaptation and adaptive window procedures, are described in this chapter. Finally, the functional methodology that couples the fluid flow equations and the dynamic mesh algorithm is explained in chapter 3. Chapter 4 demonstrates the application of the grid adaptation technique in three-dimensions by prescribing a motion for an ONERA M6 wing. Also, the validation of the six degrees-of-freedom trajectory code is presented here.

The unstructured dynamic flow solver has been validated in two and three-dimensions. The two-dimensional dynamic flow solver has been validated by performing an unsteady numerical simulation about a sinusoidally oscillating NACA 0012 airfoil and comparing it with experimental results. The three-dimensional dynamic flow solver has been validated by simulating the unsteady flowfield about an oscillating rectangular wing for which experimental data is given. These validation results are presented in Chapter 5.

The strength of the dynamic unstructured method was tested further by performing unsteady flow simulations about bodies in relative motion. The unsteady flowfield about a store separating from an airfoil, under the influence of aerodynamic forces and moments in two-dimensions, is carried out and are presented in Chapter 6. Also, the results showing the applicability of this method in three-dimensions is demonstrated by simulating the complex six degrees-of-freedom motion of a store separating from a wing under the influence of gravitational and aerodynamic forces and moments, is presented in this chapter. Some conclusions based on this study are presented in Chapter 7. Also, some recommendations for extending the applicability of this method is presented in chapter 7.

Chapter 2

GOVERNING EQUATIONS

2.1 Governing Equations for Fluid Flow

The governing equations of fluid flow that ensure the conservation of mass, momentum and energy form a set of coupled partial differential equations. When the viscosity, heat transfer, and body forces are neglected for simplicity they reduce to the well known Euler equations. In the present study the Euler equations are expressed in their conservative form using the integral formulation. The unsteady Euler equations are capable of modeling moving shock waves, entropy rise across shock waves, and entropy gradient and vorticity generation and advection behind shocks. This is evident from the Crocco's theorem and the inviscid vorticity transport equation.

The three-dimensional time dependent Euler equations for moving boundary problems, which require dynamic grids, can be expressed in integral form for a bounded domain Ω with a boundary $\partial\Omega$ as

$$\frac{\partial}{\partial t}\iiint_{\Omega}\vec{Q}\,dV + \iint_{\partial\Omega}\vec{F}(Q)\bullet\hat{n}\,dS = 0 \tag{2.1}$$

where

$$\vec{Q} = [\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho e_0]^T \tag{2.2}$$

and

$$\vec{F}(Q)\cdot\hat{n} = \left(\vec{V}\cdot\hat{n}\right)\begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 + p \end{Bmatrix} + p\begin{Bmatrix} 0 \\ n_x \\ n_y \\ n_z \\ a_t \end{Bmatrix} \tag{2.3}$$

The velocity vector, $\vec{V}$, of a fluid particle is written relative to the motion of the dynamic grids,

$$\vec{V} = \left\{ (u - x_t), (v - y_t), (w - z_t)\right\} \tag{2.4}$$

and the contravariant face speed is computed by averaging the node speeds and is given by

$$a_t = x_t n_x + y_t n_y + z_t n_z \tag{2.5}$$

In these equations, n_x, n_y and n_z denote the components of the unit vector which points normal to the cell face, and x_t, y_t and z_t are the grid speed terms in the x, y, and z directions, respectively. The equations are normalized with a freestream density ρ_∞ and a speed of sound a_∞. The term e_0 is the total energy per unit volume. The pressure and total enthalpy are related via the ideal gas assumption as

$$p = (\gamma - 1)\left[e_0 - 0.5\,\rho\left(u^2 + v^2 + w^2\right)\right] \tag{2.6}$$

and

$$h_0 = \frac{\gamma}{(\gamma - 1)}\frac{p}{\rho} + 0.5\left(u^2 + v^2 + w^2\right) \tag{2.7}$$

2.2 Governing Equations for Rigid Body Dynamics

To extend the methodology to handle the class of problems for which the motion of the body is not known *a priori*, an interfacing between the force and moment fields computed from the fluid dynamics equations and the rigid body kinematics of the motions involved needs to be established. To arrive at the equations of rigid body dynamics, consider a fluid particle P (Fig. 2.1) which has a position vector $\bar{r}$ with respect to the inertial frame of reference XYZ [34, 35]. The non-inertial frame of reference xyz has its origin at O' with $\bar{R}$ being its position vector with respect to the inertial frame of reference. The non-inertial frame is fixed to the body and translates and rotates with the body relative to the inertial frame. The position vector of P with respect to non-inertial frame of reference is given by $\bar{x}$. Hence the position vectors are related by

$$\bar{r} = \bar{R} + \bar{x} \tag{2.8}$$

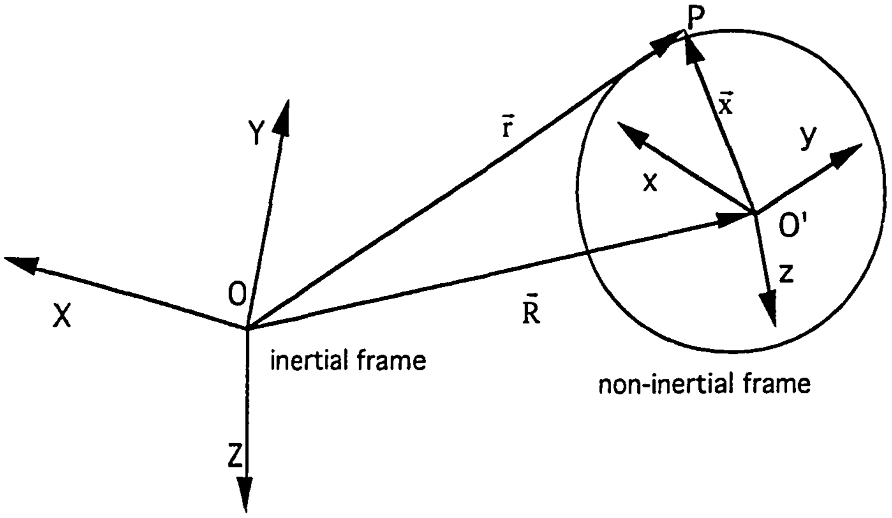


Fig. 2.1 Coordinate systems showing their position vectors.

The absolute velocity of P with respect to the inertial coordinate system is obtained by differentiating Eq. (2.8) with respect to time as

$$\bar{v} = \dot{\bar{r}} = \dot{\bar{R}} + (\dot{\bar{x}})_r + \bar{\omega} \times \bar{x} \tag{2.9}$$

where, $\dot{\bar{R}}$ is the velocity of the origin O', $(\dot{\bar{x}})_r$ is the velocity of the particle relative to non-inertial frame, $\bar{\omega}$ is the angular velocity of the body, and $\bar{\omega} \times \bar{x}$ is the velocity due to rotational motion of non-inertial frame. It is important to note that the velocity of P relative to non-inertial frame, $(\dot{\bar{x}})_r$, is zero for non-deforming bodies.

Differentiating Eq. (2.9) with respect to time, the expression for the absolute acceleration of P is obtained as

$$\bar{a} = \ddot{\bar{r}} = \ddot{\bar{R}} + \dot{\bar{\omega}} \times \bar{x} + \bar{\omega} \times (\bar{\omega} \times \bar{x}) + (\ddot{\bar{x}})_r + 2\bar{\omega} \times (\dot{\bar{x}})_r \tag{2.10}$$

In the above equation, $\ddot{\bar{R}}$ is the absolute acceleration of origin O', $(\ddot{\bar{x}})_r$ is the acceleration of particle P relative to the non-inertial frame, $2\bar{\omega} \times (\dot{\bar{x}})_r$ is the Coriolis acceleration which represents the difference in acceleration between the inertial and non-inertial axes, $\dot{\bar{\omega}} \times \bar{x}$ is the tangential acceleration and can be also termed as effect of angular acceleration caused by the rotation of the non-inertial axes, and $\bar{\omega} \times (\bar{\omega} \times \bar{x})$ is the centripetal acceleration that represents the angular acceleration component introduced due to the angular velocity of the non-inertial axes. In the above equations, $(\ )_r$ terms denote relative terms.

An unconstrained motion of a rigid body has three translational and three rotational degrees of freedom, that is, six degrees-of-freedom (DOF). The six equations of motion can be derived by a direct application of Newton's second law of motion to relate the 6-

DOF motion and the force and the moment fields. The general expression to describe the rotational motion of a rigid body is given by

$$\bar{M} = \bar{I}\dot{\bar{\omega}} + \bar{\omega} \times \left(\bar{I}\bar{\omega}\right) \tag{2.11}$$

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

$$\bar{M} \equiv \left(M_x, M_y, M_z\right) \tag{2.12}$$

$$\bar{I} \equiv \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{2.13}$$

$$\bar{\omega} \equiv \left(\omega_x, \omega_y, \omega_z\right) \tag{2.14}$$

The translational equations of motion for a rigid body are derived with respect to an axis system fixed to the body. The equations are given by

$$\bar{F} = m\,(\dot{\bar{v}}_r) + m\left(\bar{\omega} \times \bar{v}\right) \tag{2.15}$$

The assumptions made to derive the rigid-body dynamic equations are: i) the origin of the coordinate system is at a point fixed in inertial space, ii) the non-inertial coordinate system, xyz, has the same angular velocity as that of the body.

To simplify the rotational equations of motion, the non-inertial coordinate axes are assumed to coincide with the principal axes. With this choice of axis system, all crossproducts of inertia vanish and moments of inertia become time independent. Thus, Eq. (2.11) reduces to the Euler's equation of rigid-body motion:

$$
\begin{aligned}
M_x &= I_{xx}\dot{\omega}_x + \left(I_{zz} - I_{yy}\right) \omega_y \omega_z \\
M_y &= I_{yy}\dot{\omega}_y + \left(I_{xx} - I_{zz}\right) \omega_x \omega_z \\
M_z &= I_{zz}\dot{\omega}_z + \left(I_{yy} - I_{xx}\right) \omega_y \omega_x
\end{aligned}
\qquad (2.16)
$$

Since a matrix multiplication is non-commutative, a finite angular displacement is not a vector but a directed line segment. Consequently, angular velocities, ω_x, ω_y and ω_z about the body axes cannot be integrated to obtain the angular displacements. Therefore, a set of generalized coordinates is needed to describe the orientation of a rigid body. These coordinates are known as the Euler angles. It is important to note that the order of rotation is very important , and in the present study the three rotations are given as follows:

(i) A positive yaw rotation, ψ about the Z axis resulting in the primed system.

(ii) A positive pitch rotation, θ about the y' axis, resulting in the double primed system.

(iii) A positive roll rotation, ϕ about the x" axis, resulting in the final unprimed system.

The sequence of rotations is shown in Fig. 2.2. The Euler angles through the body axis angular velocities are governed by the Euler rate equations:

$$\dot{\psi} = (\omega_y \sin\phi + \omega_z \cos\phi) / \cos\theta$$

$$\dot{\theta} = \omega_y \cos\phi - \omega_z \sin\phi \qquad (2.17)$$

$$\dot{\phi} = \omega_x + \omega_y \sin\phi \tan\theta + \omega_z \cos\phi \tan\theta$$

These equations Eqs. (2.17), along with the translational and rotational equations of motion (2.15) and (2.11) respectively, determine the trajectory of the rigid body.

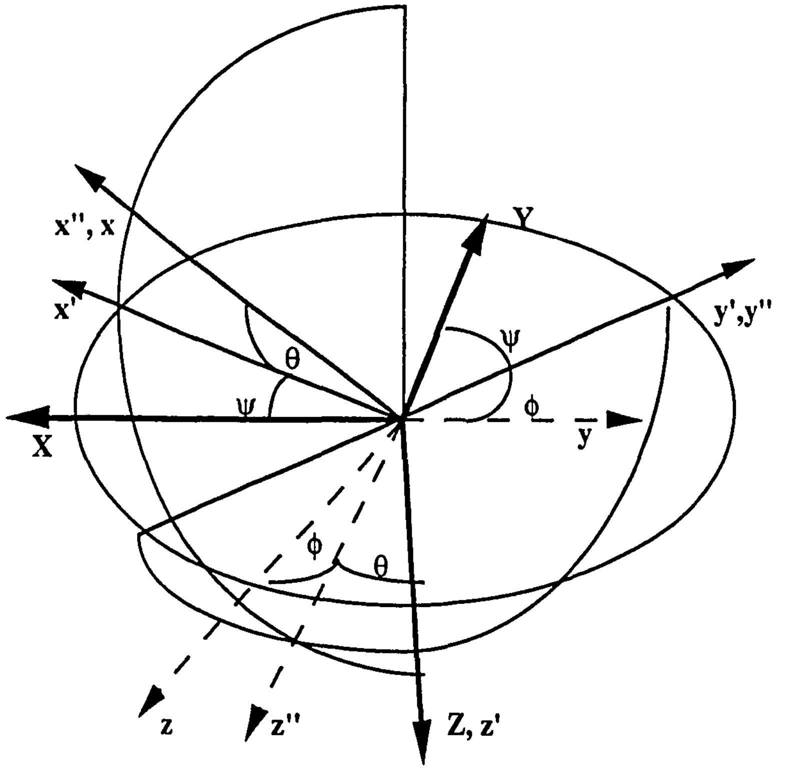


Fig. 2.2 Euler angles (yaw-pitch-roll sequence)

Chapter 3

SOLUTION ALGORITHMS

3.1 Solution Algorithm for Fluid Flow

3.1.1 Finite Volume Discretization

The finite volume formulation is based on the physical conservation laws in integral form for small volumes placed around every mesh point. The finite volume formulation has more flexibility than the finite difference method because it can handle relatively arbitrary point distributions. Also, the direct discretization of the integral form of conservation laws ensures the conservation of mass, momentum, and energy at the discrete level. One of the other advantages of this technique is, in the absence of source terms, the evaluation of fluxes is done on two dimensional surfaces rather than in three-dimensional spaces.

A semi-discrete approximation to the governing equations (2.1) is given by

$$V_i \frac{\partial Q_i}{\partial t} + \sum_{j=\kappa(i)} F_{i,j} A_{i,j} = 0 \tag{3.1}$$

Put to words, Eq. (3.1) states that the time rate of change of volume-averaged conserved variables in the i^{th} control volume is equal to the sum of the area-averaged fluxes $F_{i,j}$ over all the faces of a cell.

The random placement of cells in an unstructured mesh requires the employment of generalized indexing scheme. The present methodology uses the cell-centered finite volume formulation where each cell is considered a control volume Ω, consisting of four triangular surfaces, $\partial\Omega_i$. Equation 3.1 is applied over the four triangular faces of each cell [36, 37]. It is to be noted that a tetrahedral mesh contains about five to six times more cells than nodes. Thus, an unstructured cell-centered scheme may be five to six times more costly in terms of CPU time and memory than an unstructured cell-vertex scheme. On the other hand, an unstructured cell-centered scheme produces a higher spatial resolution due to the higher number of control volumes used in comparison to an unstructured cell-vertex scheme. In the present study, the cell-centered formulation was selected for its superior accuracy compared to the cell-vertex formulation, despite the relative storage increase. Tetrahedral cells were selected over hexahedral cells for easier discretization of irregular volumes; again, despite the relative storage increase.

3.1.2 Upwind Discretizations

The two methods which are used widely for the construction of the interface fluxes are the central differenced discretization and the upwind discretization. In a central differenced scheme, the numerical interface flux $F(Q_L,Q_R)$ is determined by averaging the fluxes corresponding to Q values of the left and right states, (Q_L,Q_R). The advantages of a central-difference type discretization are that they are easier to code and take less memory than the upwind discretization. The drawbacks of these schemes are that they lack dissipation, are inherently unstable, and decouple the adjacent cells. In order to counter these drawbacks, some artificial dissipation must be added. Currently, the most popular dissipation formula is by Jameson et al. [38], which is a blend of second and fourth-differences of the conserved variables. This dissipation formula requires user specified second- fourth-order dissipation coefficients and also is case dependent.

Upwind methods apply a discretization based on the locally one-dimensional propagation direction of waves. In other words, the interface fluxes are evaluated based on the characteristic theory for hyperbolic systems of equations. This approach makes the scheme naturally dissipative. It is important to note that upwinding is actually equivalent to a central differencing plus an artificial dissipation term. Upwind methods are classified into two categories: flux-vector splitting (FVS) and flux-difference splitting (FDS). A review of these schemes is given in [39].

In the present work, FDS of Roe [40] and FVS of van Leer [41] have both been used to compute the inviscid fluxes. A discussion of these schemes for computation on dynamic meshes is given below.

3.1.2.1 Flux Difference Splitting

The basic philosophy behind the concept of flux difference splitting is to construct the cell interface fluxes through the solution of a set of Riemann problems. Unlike the Godunov method [42] where an exact nonlinear solution to the Riemann problem is sought, Roe's method [40] seeks an approximate solution to a locally one-dimensional Riemann problem without sacrificing the nonlinear behavior of the interacting waves. Roe's method is based on extension of the linear wave decomposition, which is the exact linear solution to Riemann's problem, to the non-linear equations.

For the Roe's scheme, the flux across each cell face κ is evaluated using the numerical flux formula

$$F_\kappa = \frac{1}{2}\left[F(Q_L) + F(Q_R) - |\tilde{A}|\,(Q_R - Q_L)\right]_\kappa \tag{3.2}$$

where Q_L and Q_R are the conserved variables to the left and right of the interface, respectively and $\tilde{A}$ is the Roe-averaged flux Jacobian matrix.

The implementation of Roe's scheme to solve the non-linear Euler equations requires a linearization of the Euler equations because the Roe's scheme is based on linear concepts. Linearization of the Euler equations is accomplished by evaluating the Jacobian matrix A, with the averaged quantities (denoted by ~):

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{3.3 a}$$

$$\tilde{u} = \left(u_L + u_R \sqrt{\frac{\rho_R}{\rho_L}} \right) \Bigg/ \left(1 + \sqrt{\frac{\rho_R}{\rho_L}} \right) \tag{3.3 b}$$

$$\tilde{v} = \left(v_L + v_R \sqrt{\frac{\rho_R}{\rho_L}} \right) \Bigg/ \left(1 + \sqrt{\frac{\rho_R}{\rho_L}} \right) \tag{3.3 c}$$

$$\tilde{w} = \left(w_L + w_R \sqrt{\frac{\rho_R}{\rho_L}} \right) \Bigg/ \left(1 + \sqrt{\frac{\rho_R}{\rho_L}} \right) \tag{3.3 d}$$

$$\tilde{h}_0 = \left(h_{0L} + h_{0R} \sqrt{\frac{\rho_R}{\rho_L}} \right) \Bigg/ \left(1 + \sqrt{\frac{\rho_R}{\rho_L}} \right) \tag{3.3 e}$$

$$\tilde{a}^2 = (\gamma - 1) \left(\tilde{h}_0 - 0.5\ (\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) \right) \tag{3.3 f}$$

The Roe-averaged Jacobian matrix, $\tilde{A}$, is the mean value of the Jacobian matrix A and has the following properties:

(i) the Roe-averaged Jacobian matrix $\tilde{A}(Q_L,Q_R)$ approaches true Jacobian matrix A as Q_L and Q_R approach Q,

(ii) the flux difference between left and right states can be written as

$$\tilde{A}(Q_L,Q_R)(Q_R - Q_L) = F(Q_R) - F(Q_L) \tag{3.4}$$

(iii) $\tilde{A}$ has a complete set of real eigenvalues and vectors.

Property (i) ensures the consistency of the governing differential equations. The satisfaction of Rankine-Hugoniot shock jump conditions is ensured via property (ii). This is also responsible for the improved resolution of shocks and contact discontinuities. Property (iii) allows the matrix $\tilde{A}$ to be expressed in the canonical form

$$\tilde{A} = \tilde{T}\tilde{\Lambda}\tilde{T}^{-1} \tag{3.5}$$

where the columns of $\tilde{T}$ are the right eigenvectors of $\tilde{A}$ and the rows of $\tilde{T}^{-1}$ are the left eigen-vectors of $\tilde{A}$. $\tilde{\Lambda}$ is a diagonal matrix comprising of eigenvalues of $\tilde{A}$. By virtue of all the properties stated above, the flux difference can be expressed as

$$F(Q_R) - F(Q_L) = \tilde{T}\tilde{\Lambda}\tilde{T}^{-1}\Delta Q = |\Delta F_1| + |\Delta F_2| + |\Delta F_3| \tag{3.6}$$

where

$$|\Delta F_1| = |\tilde{U}| \left\{ \left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2}\right) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \dfrac{\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2}{2} \end{bmatrix} + \tilde{\rho} \begin{bmatrix} 0 \\ \Delta u - \Delta U\, n_x \\ \Delta v - \Delta U\, n_y \\ \Delta w - \Delta U\, n_z \\ \tilde{u}\,\Delta u + \tilde{v}\,\Delta v + \tilde{w}\,\Delta w - \tilde{U}\,\Delta U \end{bmatrix} \right\} \tag{3.7 a}$$

$$|\Delta F_{2,3}| = |\tilde{U} \pm \tilde{a}|\left(\frac{\Delta p \pm \tilde{\rho}\tilde{a}\Delta U}{2\tilde{a}^2}\right)\begin{bmatrix} 1 \\ \tilde{u} \pm \tilde{a} n_x \\ \tilde{v} \pm \tilde{a} n_y \\ \tilde{w} \pm \tilde{a} n_z \\ \tilde{h}_0 \pm \tilde{a}\tilde{U} \end{bmatrix} \qquad (3.7\ \text{b})$$

with $\Delta U = \Delta u\, n_x + \Delta v\, n_y + \Delta w\, n_z$ and the Roe-averaged contravariant velocity is defined by

$$\bar{U} = \bar{u} n_x + \bar{v} n_y + \bar{w} n_z \qquad (3.8)$$

In the present investigation, the Roe scheme used for stationary boundary problems is modified for moving meshes by redefining the Roe-averaged contravariant velocity as (Eq. 2.5)

$$\tilde{U} = \tilde{u}\, n_x + \tilde{v}\, n_y + \tilde{w}\, n_z - a_t \qquad (3.9)$$

Figure 3.1 gives a better insight as to how the fluxes are calculated. The common face between the two adjacent cells, cell 1 and cell 2, consisting of the nodes N1, N2, N3, from one side and coincident nodes M1, M2, M3 from the other side.

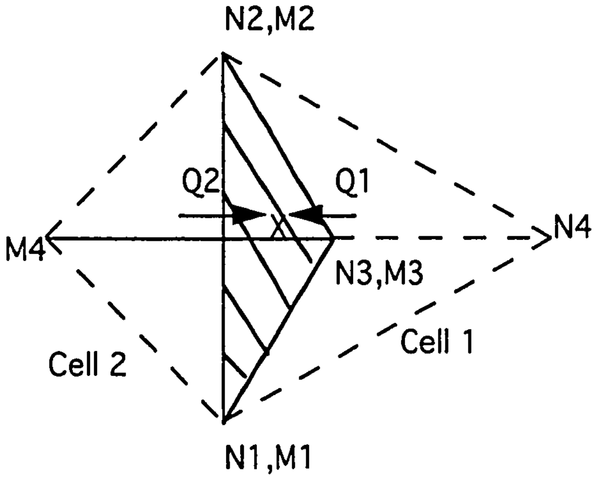


Fig. 3.1 Tetrahedral cells showing common face and nodes.

3.1.2.2 Flux Vector Splitting

Flux vector splitting is based on the construction of a stable upwind differencing method to solve a set of hyperbolic conservation laws comprising of positive and negative eigenvalues.

For the van Leer scheme [41], the flux vectors are given in terms of the Mach number normal to each face. This results in the possibility of the flow being supersonic or subsonic through a face. The conditions which need to be satisfied for the scheme are:

(i) $F(w) = \hat{F}^{+}(w) + \hat{F}^{-}(w)$.

(ii) $\frac{d\hat{F}^{+}}{dw}$ has all eigenvalues ≥ 0 and $\frac{d\hat{F}^{-}}{dw}$ has all eigenvalues ≤ 0. Also, $\frac{d\hat{F}^{\pm}}{dw}$ is continuous, with one eigenvalue vanishing for $|M| < 1$.

(iii) $F^{\pm}(w)$ is continuous, with $\hat{F}^{+} \equiv F$ for $M \geq 1$ and $\hat{F}^{-} \equiv \hat{F}$ for $M \leq -1$.

(iv) $\hat{F}^{\pm}(M)$ must be a polynomial in M with the lowest possible degree.

With the above conditions, supersonic fluxes are evaluated as

$$\hat{F}^{+} = \left(\vec{F}(Q) \bullet \hat{n}\ \right)^{+}, \quad \hat{F}^{-} = \left(\vec{F}(Q) \bullet \hat{n}\right)^{-} = 0 \text{ for } M_n \geq 1 \qquad (3.10)$$

$$\hat{F}^{-} = \left(\vec{F}(Q) \bullet \hat{n}\right)^{-}, \quad \hat{F}^{+} = \left(\vec{F}(Q) \bullet \hat{n}\right)^{+} = 0 \text{ for } M_n \leq -1 \qquad (3.11)$$

For subsonic flow, the fluxes are split into the following contributions

$$F_k = \hat{F}^{+}\left(Q^{-}\right) + \hat{F}^{-}\left(Q^{+}\right) \qquad (3.12)$$

where

$$\hat{F}^{\pm} = \begin{Bmatrix} f^{\pm}_{\text{mass}} \\ f^{\pm}_{\text{mass}}\left\{u + n_x\left(-\overline{U} \pm 2a\right)/\gamma\right\} \\ f^{\pm}_{\text{mass}}\left\{v + n_y\left(-\overline{U} \pm 2a\right)/\gamma\right\} \\ f^{\pm}_{\text{mass}}\left\{w + n_z\left(-\overline{U} \pm 2a\right)/\gamma\right\} \\ f^{\pm}_{\text{energy}} \end{Bmatrix} \tag{3.13}$$

with

$$f^{\pm}_{\text{mass}} = \pm\frac{\rho a}{4}\left(M_n \pm 1\right)^2 \tag{3.14}$$

$$f^{\pm}_{\text{energy}} = f^{\pm}_{\text{mass}}\left[\frac{(1-\gamma)\overline{U}^2 \pm 2(\gamma-1)\overline{U}a + 2a^2}{\left(\gamma^2 - 1\right)} + \frac{u^2 + v^2 + w^2}{2} + \frac{a_t\left(-\overline{U} \pm 2a\right)}{\gamma}\right] \tag{3.15}$$

In Eqs. (3.10)-(3.15), $\overline{U}$ is the adjusted contravariant velocity, which is the scalar product of the modified velocity in Eq. (2.4) with the unit normal vector to the face.

In the present investigation, the van Leer flux vector splitting has been extended for the three dimensional Euler equations to handle problems involving dynamic meshes while ensuring that the properties of the original splittings are preserved [43, 44, 45]. It is to be noted that for unsteady moving boundary problems, the requirements for split fluxes are the same as for stationary grids except that the fluxes are now functions of grid speeds through the contravariant face speed a_t and the grid-speed adjusted contravariant velocity $\overline{U}$ (as used in Eq. (2.3))

$$\overline{U} = \vec{V} \bullet \hat{n} = \left(u - x_t\right) n_x + \left(v - y_t\right) n_y + \left(w - z_t\right) n_z \tag{3.16}$$

3.1.2.3 Higher Order Spatial Differencing

In a first-order scheme, the state of the primitive variables at each cell face is set equal to the cell-centered averages on both sides of the face. For a higher-order scheme, the evaluation of fluxes requires a correct estimate of the left and right state at the cell faces. One method [46] is to expand the cell-centered solution to each cell face through a Taylor series expansion as

$$q(x,y,z) = q(x_c, y_c, z_c) + \nabla q_c \bullet \Delta\bar{r} + \vartheta(\Delta r^2) \qquad (3.17)$$

There are various approaches [45] to calculate the solution gradient efficiently at the cell center. In the present study, the solution gradient at the center of the cell is constructed by exploiting the invariant characteristics of triangles and tetrahedra and by the application of the Green's theorem [37]:

$$\nabla q_c \bullet \Delta\bar{r} = \left[\frac{\frac{1}{3}(q_{n1} + q_{n2} + q_{n3}) - q_{n4}}{4\,\Delta r} \right] \Delta r \qquad (3.18)$$

where $q_{n1}, q_{n2,}\ q_{n3}$ denote the primitive variables at the three nodes that constitute the face through which the flux passes, Δr is the distance from the centroid of a tetrahedron to the center of that face, and q_{n4} are the primitive variables at the fourth node of the tetrahedron.

The nodal values of q in Eq. (3.18) above are determined by inverse distance weighting of the surrounding cell-centered solution quantities. This is achieved by the expression:

$$q_n = \left(\sum_{i=1}^{N} w_{c,i} \, q_{c,i} \right) \Bigg/ \left(\sum_{i=1}^{N} w_{c,i} \right) \qquad (3.19)$$

where the subscript c,i refers to surrounding cell-centered values. After trying several weighting factors, Frink reported [37] that the inverse-distance weighting factor ($w_{c,i} = \left(x_{c,i} - x_n\right)^{-1}$) produces the least error involved in computation of the gradient from four heuristically determined weighting factors. It turns out, however, the accuracy of this weighting is less than second order. A fully second order accurate averaging procedure has been developed for two-dimensions by Rausch et al. [47] and extended to three-dimensions by Frink [46]. The weights were derived on the basis of the property that the Laplacian of a linear function is zero, where:

$$L(x_n) = \sum_{i=1}^{N} w_{c,i} \left(x_{c,i} - x_n\right) = 0 \qquad (3.20a)$$

$$L(y_n) = \sum_{i=1}^{N} w_{c,i} \left(y_{c,i} - y_n\right) = 0 \qquad (3.20b)$$

$$L(z_n) = \sum_{i=1}^{N} w_{c,i} \left(z_{c,i} - z_n\right) = 0 \qquad (3.20c)$$

The weights are evaluated by defining

$$w_{c,i} = 1 + \Delta w_{c,i} \qquad (3.21)$$

with the cost function,

$$C = \sum_{i=1}^{N} \left(\Delta w_{c,i}\right)^2 \qquad (3.22)$$

minimized by soving the minimization problem utilizing the method of Lagrange-multipliers, subject to the constraints given by Eqs. (3.20). The expressions for the Lagrangian-multipliers are given elsewhere [46, 47].

3.1.3 Geometric Conservation Law

The conservation laws for the discretized fluid dynamic equations for moving grids may be violated if the geometric conservation law (GCL) is not incorporated [10, 15]. For a moving mesh, the preservation of a uniform flow can only be maintained when the cell volumes are computed using the discrete form of the geometric conservation law.

For a time interval t_2-t_1, the integral form of conservation law for the Euler equations (2.1) can be recast in the following form:

$$\int_{\Omega(t_2)} Q \, dV - \int_{\Omega(t_1)} Q \, dV + \int_{t_1}^{t_2} \oint_{\partial\Omega(t)} \vec{F} \bullet \vec{n} \, dS \, dt = 0 \qquad (3.23)$$

where V(t) is the cell volume at time t and $\vec{n}dS$ is the unit normal on a surface pointing outward. The flux $\vec{F}$ can be expressed as

$$\vec{F} = \left(u_p - \vec{v}\right) Q = \vec{F}_{static} - \vec{v} \, Q \qquad (3.24)$$

where u_p and $\vec{v}$ are the fluid particle velocity and the local velocity of a cell face, respectively. For the freestream values of Q and $\vec{F}_{static}$, the derivation of geometric identities can be accomplished by combining Eqs. (3.23) and (3.24) as shown below:

$$\left[V(t_2)-V(t_1)\right]Q_\infty + \vec{F}_{static_\infty} \cdot \int_{t_1}^{t_2} \oint_{\partial\Omega(t)} \vec{n}\,dS\,dt \;-\; Q_\infty \int_{t_1}^{t_2} \oint_{\partial\Omega(t)} \vec{n}\bullet\vec{v}\,dS\,dt = 0 \qquad (3.25)$$

The second term $\oint\vec{n}dS$ on the left hand side of Eq. (3.25) vanishes for a closed cell and the resulting equation is simplified to the the geometric conservation law:

$$\left[V(t_2)-V(t_1)\right] = \int_{t_1}^{t_2} \oint_{\partial\Omega(t)} \vec{n}\bullet\vec{v}\,dS\,dt \qquad (3.26)$$

To avoid grid-motion induced errors when dynamic meshes are involved, Eq. (3.26) must be satisfied concurrently with the conservation of mass, momentum, and energy [1, 14, 15]. The integral statement of GCL may be written as

$$\frac{d}{dt}\iiint_{\Omega} dV = \iint_{\partial\Omega} \vec{v}.\vec{n}\,dS \qquad (3.27)$$

Furthermore, to provide a self consistent solution for the local cell volumes, the GCL should be integrated using the same scheme that is used for the fluid equations. A discretization of Eq. (3.27) has been expressed in [1] which is consistent with the above solution algorithm and is given by

$$V_i^{n+1} = V_i^n + \Delta t \sum_{j=\kappa(i)} \left[a_t\Delta S\right]_{ij}^{n+1} \qquad (3.28)$$

Thus, this equation is used to update the local cell volumes for the current time level.

3.1.4 Time Integration

Time can be advanced either explicitly or implicitly. In an explicit method, the matrix of unknown variables formed at the new time level is a diagonal matrix, and the right hand side of the system depends only on the flow variables from the previous time level. The advantages of this method are: that it requires fewer arithmetic operations per time step, it is simple to code, and it can be vectorized with ease. The primary drawback of this method stems from the severe restrictions imposed on the maximum allowable time step due to the stability and convergence conditions.

In an implicit method, there are more than one set of unknown variables at the same time level and hence the matrix to be inverted is not a diagonal one. The number of arithmetic operations required per time step will be higher than for the explicit scheme, but this drawback is counterbalanced by the fact that implicit schemes pose no theoretical limitation on the time step.

It is important to note that, for unsteady flow problems, time accuracy of the numerical solution is required and the temporal conservation error diminishes with decreasing time step. Chaderjian et. al. [48] have shown that the implicit algorithm of Pulliam and Chaussee [49] approaches the Euler explicit scheme as the time step is reduced. Hence, an explicit scheme may become the appropriate choice for certain unsteady problems.

In the present study the spatially discretized form of the governing equations are integrated in time using the explicit fourth-order Runge-Kutta method [38]. This method has second-order temporal accuracy for the non-linear equations and it may be written as follows:

$$\left.\begin{aligned} Q_i^{(0)} &= Q_i^n \\ Q_i^{(1)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{4}\frac{\Delta t}{V_i^{n+1}} R_i^{(0)} \\ Q_i^{(2)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{3}\frac{\Delta t}{V_i^{n+1}} R_i^{(1)} \\ Q_i^{(3)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{2}\frac{\Delta t}{V_i^{n+1}} R_i^{(2)} \\ Q_i^{(4)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{\Delta t}{V_i^{n+1}} R_i^{(3)} \\ Q_i^{n+1} &= Q_i^{(4)} \end{aligned}\right\} \qquad (3.29)$$

where V_i is the cell volume and R_i is the residual given by

$$R_i = \sum_{j=\kappa(i)} F_{ij} A_{ij} \qquad (3.30)$$

where A is the cell face area and the summation of fluxes is taken over the four faces '*k*' of tetrahedral cell 'i'.

3.1.5 Convergence Acceleration Techniques

The explicit Runge-Kutta time integration scheme has a step size restriction based on the Courant-Friedrichs-Lewy condition which corresponds to a Courant number of $2\sqrt{2}$. A fully converged steady state solution is provided as an initial condition for the unsteady calculations. To accelerate the convergence rate to the steady-state, the Courant number can be increased by employing convergence acceleration techniques. In the present study, the method of local time stepping and the method of implicit residual smoothing have been used. A brief discussion of these methods is given in the following section.

3.1.5.1 Local time stepping

The convergence of a solution to the steady state can be accelerated by using local time stepping. This method advances the solution of each cell (denoted by i) in pseudo-time using the maximum possible time step based on the local stability limit:

$$\Delta t_i = \frac{(CFL)V_i}{(|u_i|+a_i)A_i^x + (|v_i|+a_i)A_i^y + (|w_i|+a_i)A_i^z} \tag{3.31}$$

where V and a represent the cell volume and local speed of sound, respectively. A_i^x, A_i^y, and A_i^z are the projected areas in the x, y, and z directions, respectively.

3.1.5.2 Implicit Residual Smoothing

The time step can be enhanced further by an implicit residual smoothing. This method is performed by implicit averaging of residuals between a certain number of neighboring cells. Effectively, this process filters the residuals through a smoothing operator and solves the resulting set of equations by the Jacobi method. The present implementation of this method, is adapted from Jameson and further discussed in [38]. It should be noted that the above discussed acceleration methods are used only for the steady-state solutions, and are by-passed when unsteady simulations are performed.

3.1.6 Initial and Boundary Conditions

In order to have a well-posed problem, initial and boundary conditions need to be imposed. That is, the solution to any partial differential equation is governed by the choice of its initial and boundary conditions. In the following section, initial conditions will be discussed, then physical boundary conditions for steady and unsteady flows are developed. Modifications required to handle unsteady moving boundary problems are presented.

For the steady calculations, uniform conditions are chosen as the initial condition. For the time accurate calculations needed for unsteady flows, a fully converged steady state solution is used as the initial condition.

The far-field boundary conditions are incorporated using the locally one-dimensional characteristic boundary conditions. The velocity normal to the boundary and the sound speed for each cell is calculated from the locally one-dimensional Riemann invariants given by

$$R^{\pm} = U \pm \frac{2}{\gamma - 1} a \tag{3.32}$$

These invariants are used to calculate the local normal velocity and the speed of sound. The local normal velocity at the boundary is calculated by adding the two Riemann invariants and the speed of sound is obtained by subtracting the two Riemann invariants. The density boundary condition is imposed by using the entropy relationship and the pressure boundary condition is applied using the equation of state.

For steady inviscid flows, the velocity components used in the surface boundary conditions are written as

$$\begin{aligned} u_{wall} &= u_{center} - n_x U \\ v_{wall} &= v_{center} - n_y U \\ w_{wall} &= w_{center} - n_z U \end{aligned} \tag{3.33}$$

where U is the contravariant velocity given by

$$\vec{V} \bullet \hat{n} = U = u\, n_x + v\, n_y + w\, n_z \tag{3.34}$$

The density and pressure boundary conditions are got by zeroth-order extrapolation as

$$\begin{aligned} \rho_{wall} &= \rho_{center} \\ p_{wall} &= p_{center} \end{aligned} \tag{3.35}$$

For unsteady moving boundary problems, however, the above conditions must be adjusted since the boundary faces now possess a discernible velocity. The expressions for the unsteady corrected velocity components remain the same as in Eq. (3.34) except for the velocity vector $\vec{V}$, which now has to take into account the grid speed term (Eq. (3.15)). The expression for the velocity vector is given by Eq. (2.4) and it is obtained by subtracting the grid speed from the respective components of the velocity vector $\vec{V}$. For a moving boundary problem, the pressure gradient in the normal direction is non-zero and can be derived from the normal momentum equation [50] as

$$\frac{\partial p}{\partial n} = -\rho \hat{n} \cdot \vec{a} \tag{3.36}$$

where $\hat{n}$ is the unit normal to the boundary face, $\vec{a}$ is the acceleration of the body given by Eq. (2.10).

3.2 Solution Algorithm for Rigid Body Dynamics

The aerodynamic forces and moments are formed by integrating the instantaneous pressure field and their moments. The expression for the forces and moments are given by

$$m\dot{\vec{v}}_{cg} = \sum \vec{F} = mg - \int p\vec{n}d\Omega \tag{3.37}$$

$$I\dot{\vec{\omega}} + \int_{\Omega} (\vec{\omega} \bullet \vec{x}) \bullet (\vec{x} \times \vec{\omega}) d\Omega = \sum \vec{M} \tag{3.38}$$

These forces and moments obtained from the fluid dynamic equations are relative to the inertial frame of reference XYZ. In the present study, the equations governing the rigid body dynamics are solved in the non-inertial frame of reference, xyz, and hence the inertial forces and moments need to be transformed to the non-inertial frame of reference. The non-inertial frame of reference used here is the axis system which is fixed to the body. The axis systems are shown in Fig. (2.1). The transformation of the forces and moments relative to the non-inertial frame of reference is accomplished through the directional cosine tensor C_{ij}.

Let r_j and x_j represent the vector quantities in the inertial and non-inertial frames, respectively, then the two vectors are related through the transformation matrix C_{ij} as

$$r_j = \left[C_{ij}\right]^T x_j \text{ and } x_j = \left[C_{ij}\right] r_j \tag{3.39}$$

where the transformation matrix C_{ij} is given by

$$\left[C_{ij}\right] = \begin{bmatrix} C\psi C\theta & C\theta S\psi & -S\theta \\ S\phi S\theta C\psi - C\phi S\psi & C\phi C\psi + S\phi S\psi S\theta & C\theta S\phi \end{bmatrix} \tag{3.40}$$

$$\left[C_{ij}\right] = \begin{bmatrix} C\psi C\theta & C\theta S\psi & -S\theta \\ S\phi S\theta C\psi - C\phi S\psi & C\phi C\psi + S\phi S\psi S\theta & C\theta S\phi \\ S\phi S\psi + C\phi C\psi S\theta & -C\psi S\phi + C\phi S\psi S\theta & C\phi C\theta \end{bmatrix}$$

Here, S and C denote the sine and cosine of the suffix angles, respectively. The angles ψ, θ, and ϕ are the Euler angles and they are also termed as yaw, pitch and roll angles, respectively (Fig. 2.2). The sequence of rotations is very important since the matrix multiplication is not commutative. The sequence of rotations used here are in the order yaw-pitch-roll. The axis system used here is as follows: x axis is forward along the longitudinal axis (e.g.: store longitudinal axis), the y axis is positive to the right looking forward, and the z axis is vertical downward (Fig. 2.2).

After having transformed the forces and moments to the non-inertial frame of reference, the rotational equations of motion given below are integrated in time using a fourth-order Runge-Kutta method to yield the angular velocity field, $(\omega_x,\ \omega_y,\ \omega_z)$.

$$\begin{aligned}\dot{\omega}_x &= \frac{M_x}{I_{xx}} - \frac{\left(I_{zz} - I_{yy}\right)\omega_y\omega_z}{I_{xx}}\\ \dot{\omega}_y &= \frac{M_y}{I_{yy}} - \frac{\left(I_{xx} - I_{zz}\right)\omega_x\omega_z}{I_{yy}}\\ \dot{\omega}_z &= \frac{M_z}{I_{zz}} - \frac{\left(I_{yy} - I_{xx}\right)\omega_y\omega_x}{I_{zz}}\end{aligned} \tag{3.41}$$

Using the angular velocity field obtained thus, the Euler rate equations (2.15) are integrated in time, again using a fourth-order Runge-Kutta method to obtain the Euler angles, ψ , θ, and ϕ. Finally, the Euler angles, together with the angular velocities, are used in the translational equations of motion.

$$\begin{aligned}\dot{v}_x &= \frac{1}{m}\left[\Sigma F_x - mg\,\mathrm{Sin}\theta\right] - \omega_y v_z + \omega_z v_y \\ \dot{v}_y &= \frac{1}{m}\left[\Sigma F_y + mg\,\mathrm{Cos}\theta\,\mathrm{Sin}\phi\right] - \omega_z v_x + \omega_x v_z \\ \dot{v}_z &= \frac{1}{m}\left[\Sigma F_z + mg\,\mathrm{Cos}\theta\,\mathrm{Cos}\phi\right] - \omega_x v_y + \omega_y v_x\end{aligned} \tag{3.42}$$

Equations (3.42) are integrated in time to yield the translational velocity. In Eq. (3.42), m is the mass of the body whose trajectory is to be determined, and g is the gravitational acceleration. Further, the translational velocity is integrated again to obtain the linear displacement field. Since the linear displacements are relative to a non-inertial frame of reference, they are transformed via the transformation matrix $\left[C_{ij}\right]$ as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \left[C_{ij}\right]^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.43}$$

Thus, the linear displacement of the center of gravity of the body together with the Euler angles, determine the trajectory of the body under the action of aerodynamic forces and moments.

3.3 Dynamic Mesh Algorithm

3.3.1 Grid Adaptation Method

The unstructured mesh about the body (or bodies) of interest is considered as a system of interconnected springs [3]. This system is constructed by representing each edge of each triangle by a tension spring. Various attempts at determining the optimum relationship for specifying the spring stiffness have been made by Chakravarthy et al. [51]. In the present work, however, the spring stiffness is assumed inversely proportional to the length of its edge and may be written as

$$k_{ji} = 1.0 / \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{p/2} \tag{3.44}$$

where p (generally taken between one and three) is a parameter used to control the stiffness of the spring. In some cases normalized edge lengths

$$\bar{L} = \left(\frac{L - L_{min}}{L_{max} - L_{min}} \right) \tag{3.45}$$

may be used in place of the actual lengths used in Eq. (3.44). L_{min} and L_{max} are the minimum and maximum values of the edge lengths, respectively, at a given adaptation stage. Then, for each mesh point, the external forces due to the connecting springs are resolved into their Cartesian components and summed (Fig. 3.2). The resulting set of linear systems are solved for the displacements of each node using the point-Jacobi method:

$$\Delta x_j^{n+1} = \frac{\Sigma k_{ji} \, \Delta x_i^n}{\Sigma k_{ji}} \tag{3.46a}$$

$$\Delta y_j^{n+1} = \frac{\Sigma k_{ji} \, \Delta y_i^n}{\Sigma k_{ji}} \tag{3.46b}$$

$$\Delta z_j^{n+1} = \frac{\Sigma k_{ji} \, \Delta z_i^n}{\Sigma k_{ji}} \tag{3.46c}$$

where i is summed over all edges connected to node j. The positions of the interior points are then updated using the determined displacements.

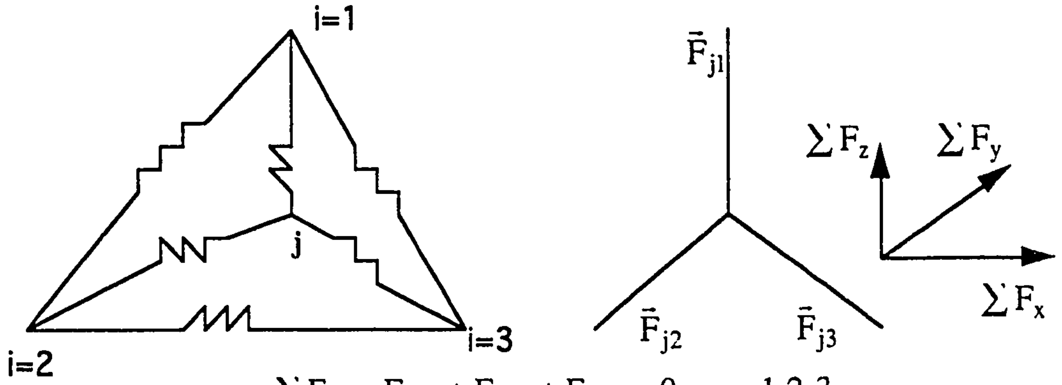
This iterative method has the advantage of not requiring an excessive amount of memory, but it does require an initial guess. For the present system, only the displacements at the current time level are stored, and the initial guesses of the displacements are the displacements at the previous time level. Since the system being solved is diagonally dominant (the diagonal of each row being the sum of the spring stiffness of every node involved in that equilibrium equation), a relaxation factor may be introduced to accelerate convergence. Hence, using this successive over relaxation method, an acceptable mesh movement is achieved in 4 to 6 iterations.

When dealing with cases that involve large amplitude motions, there is every likelihood of having grid skewness and poor grid point distribution. This problem can be alleviated by remeshing, regeneration or smoothing. In the present study, Laplacian type smoothing [53] has been used. The expression for performing such smoothing is given by

$$x_j^{n+1} = x_j^n + \frac{\sigma}{n}\sum_{i=1}^{n}\left(x_i - x_j\right) \tag{3.47a}$$

$$y_j^{n+1} = y_j^n + \frac{\sigma}{n}\sum_{i=1}^{n}\left(y_i - y_j\right) \tag{3.47b}$$

$$z_j^{n+1} = z_j^n + \frac{\sigma}{n}\sum_{i=1}^{n}\left(z_i - z_j\right) \tag{3.47c}$$



$$\sum F_m = F_{j1_m} + F_{j2_m} + F_{j3_m} = 0, \ m = 1,2,3$$

$$F_{ji_{(m)}} = k_{ji}(\Delta x_{i_{(m)}} - \Delta x_{j_{(m)}}), \ m = 1,2,3$$

Fig. 3.2 Schematic of the assumed spring system at node denoted by j.

where σ is the relaxation parameter and i is summed over all the edges connected to node j. The number of smoothing sweeps is user specified and in the present study, the number of smoothing sweeps is taken to be 125.

3.3.2 Adaptive Window Procedure

The computational efficiency of the grid adaptation method above can be improved by limiting the size of the adaptation region, since only a small area of the mesh needs to be stored and adapted. The method used in the present work to restrict the size of the adaptation region is to create a "window" around the physical domain of interest [1, 53]. The nodal points inside this window are considered as the spring network and, thus, allowed to adapt to the body movement. Significant savings in both CPU time and memory are realized using this procedure. In [28], it has been reported that the spatial adaptation procedure can be responsible for anywhere between 3 to 7.5 percent of the total CPU time for an unsteady simulation (the exact percentage depends on whether or not local enrichment or refinement is utilized in the adaptation). Nevertheless, for unsteady flows, where it is understood that a large number of iterations are performed, small savings per iteration will equate to large overall savings. In the present work, the adaptation procedure, regardless of window construction, costs 9.5 microseconds/node/iteration and requires storage of 2+2*m per node, where m is the maximum number of springs connected to any node on a Cray 2. Therefore, CPU time and memory savings may be obtained by reducing the number of nodes being adapted per iteration; for example, by only adapting 30 percent of the total number of nodes, a 70 percent saving is realized.

Creating the window may be carried out by either specifying a normal distance from the body of interest or choosing a basis shape (circle, ellipse, etc.) around it. The entire domain is searched to locate the points which fall within the window and they are flagged as window points. For instance, when a basis shape, such as a circle is used, the points which fall within the circle are the points whose absolute distance from the center of the circle is less than the radius of the circle. The window points are allowed to be adapted from one time step to the next. The next search is for the mesh points which are connected to the outermost window points. These points are flagged as "window frame" points. Mesh points exterior to the window and the window frame points are spatially fixed in time.

For problems in which the body has small or no translational movement, creation of the window takes place only once. However, for problems in which large movements are encountered, the window may need to be constructed on several occasions during the body's trajectory. Thus, window construction must be a quick, reliable, and automated process. In the present work, a basis shape is used to specify the window, and a critical displacement is chosen to determine when a new window is needed.

3.4 Functional Overall Methodology

Until now the solution algorithms for the fluid dynamics equations, rigid body dynamics equations, and the dynamic mesh equations have been stated. In the present section, an outline of the overall solution methodology is explained. The flowchart of the methodology is presented in Fig. 3.3.

The salient steps of the solution methodology can be itemized as follows:

(1) Generate a fixed mesh.

(2) Define a window around the body.

(3) Obtain the steady state solution.

(4) Given an initial value of the grid speed, the governing equations of flow (2.1) are solved to obtain the instantaneous force and moment fields.

(5) These forces and moments are transformed to the non-inertial frame of reference and used in the rigid body dynamics equations Eqs. (2.11 and 2.13) to solve for the translational and rotational velocity fields.

(6) Using the translational velocities and the angular velocities, the velocities of the surface points are found from

$$\vec{v} = \dot{\vec{r}} = \dot{\vec{R}} + \vec{\omega} \times \vec{x} \tag{3.48 }$$

The above equation is obtained using the rigid body assumption, and eliminating the relative velocity term in Eq. (2.9). However, the velocities of the off-surface mesh points are calculated by the full Eq. (2.9) with the relative velocity term being included. This is because the rigid body assumption is no longer valid in this region.

(7) With the velocity field at the current time step, the coordinates of the grid points and the center of mass are updated.

(8) Then, the position of the body is checked to see if a new window is needed. If so, a new window is created about the body in the current position.

(9) Once the window criteria has been evaluated, the mesh is then adapted to the body movement.

(10) The solution is advanced time accurately by one time step.

(11) If the target position has not been reached, steps (4) to (10) are repeated.

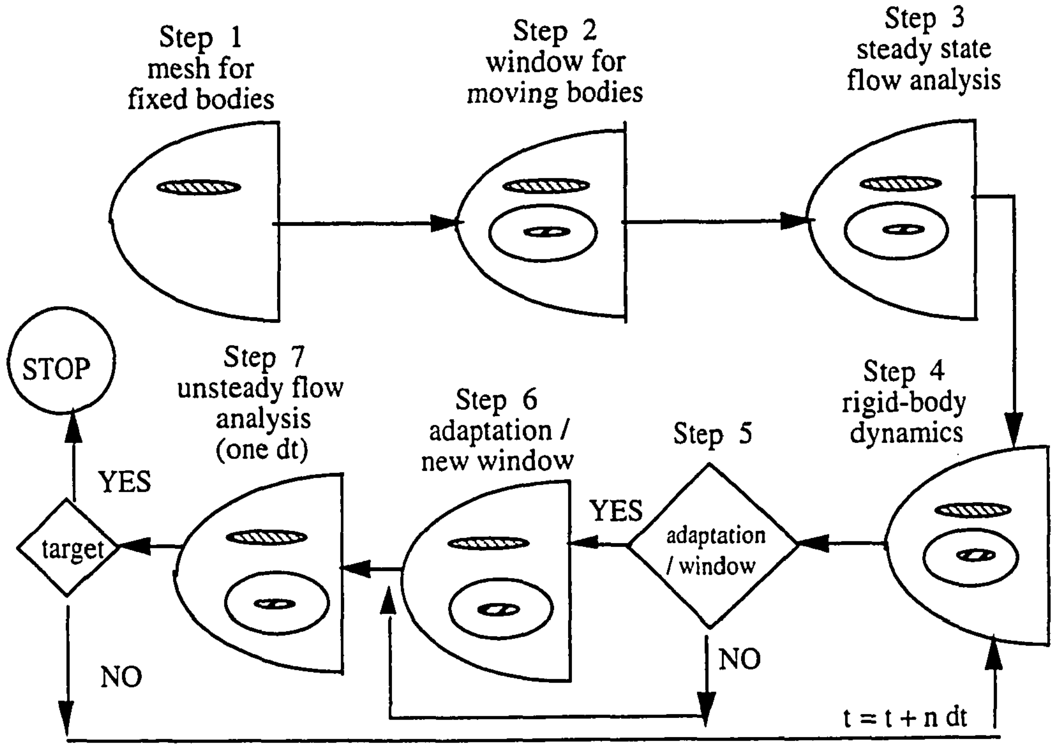


Fig. 3.3 Flow chart of 3D dynamic unstructured method.

Chapter 4

DEMONSTRATION OF ELEMENTS OF METHODOLOGY

Prior to presenting the results of the present study, components of the developed methodology (Fig. 3.3) are tested and demonstrated in this chapter. With this impetus, the baseline flow solver for the steady flow past a static configuration, and the grid adaptation in a window are applied to three problems. Then the trajectory is validated.

4.1 Baseline Flow Solver

The configuration for this computational model consists of a cone-nosed and flare-based cylinder (35 inches long and 6 inches in diameter) as the carrier and an ogive-nosed cylinder (9.5 inches long and 0.4 inches in diameter) as the projectile. The angles of attack were zero degrees for the carrier and ten degrees for the projectile (Fig. 4.1) This configuration is called herein the aft-launched projectile (ALP). The computational grid was generated using an unstructured background grid and consisted of 266,000 cells and 46,000 nodes. A blow-up view of the surface grid for the projectile is shown in Fig. 4.2. The computational domain for this case extended 30d streamwise, 24d normally, and 16d laterally (d denotes the carrier diameter). The steady state soluiton at M_∞ of 1.5 was obtained by using the 3D, static, Euler flow solver called USM3D [38]. The solver served as the baseline to the present investigation which resulted in the 3D, unsteady flow solver for moving boundary problems.

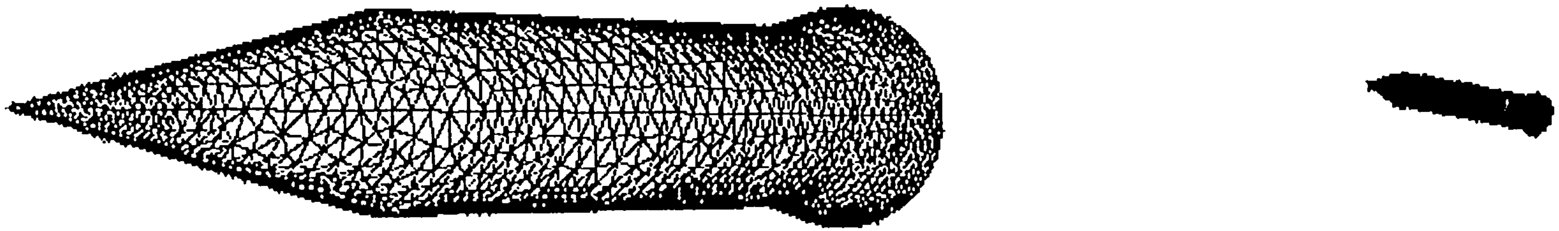


Fig. 4.1 Surface grid on carrier and projectile

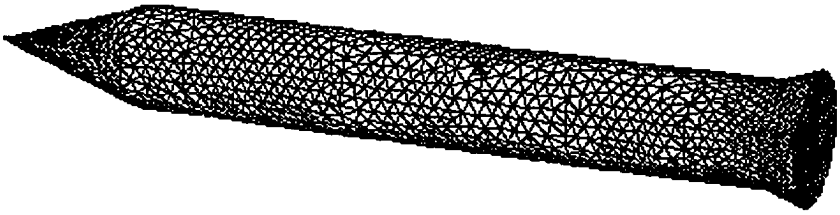


Fig. 4.2 Blow-up view of the surface grid for the projectile

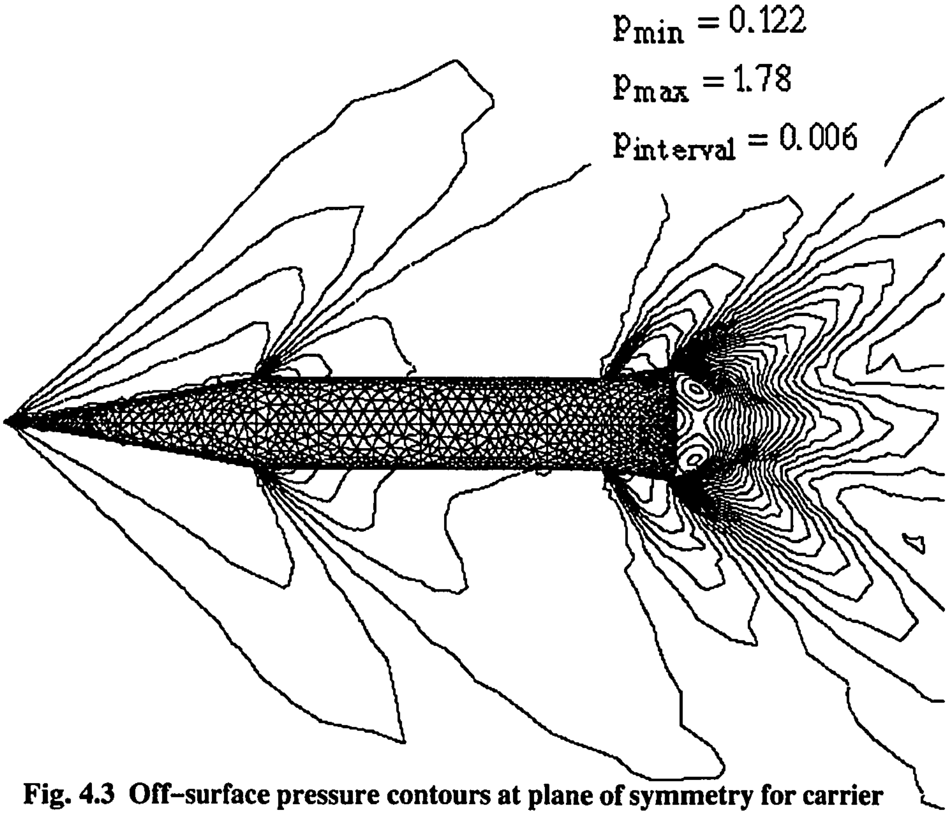


Fig. 4.3 Off-surface pressure contours at plane of symmetry for carrier

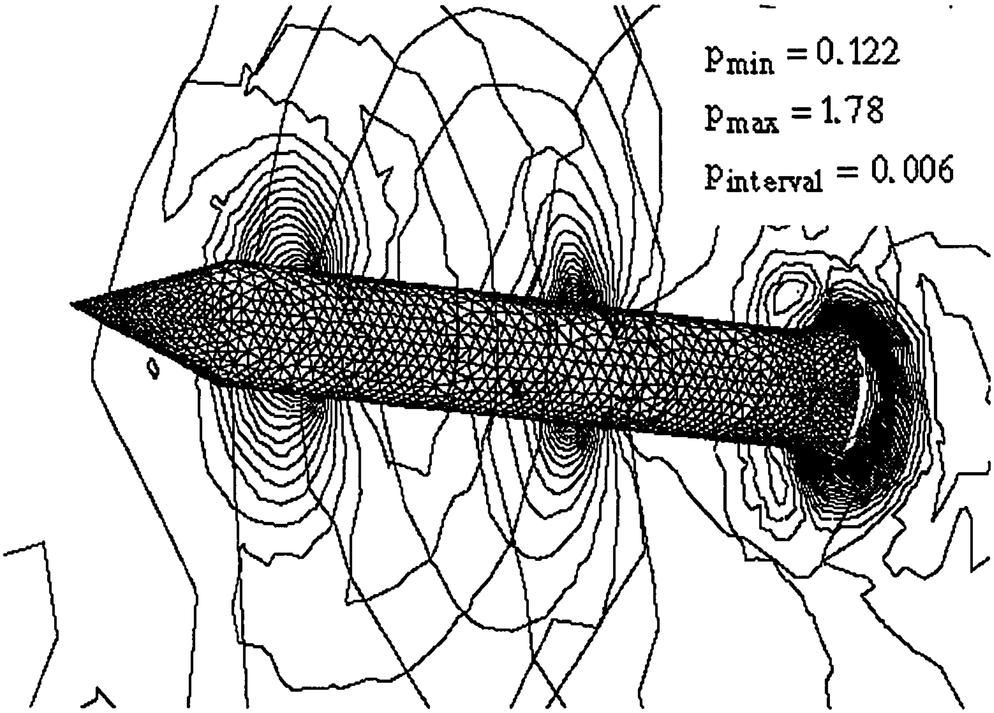


Fig. 4.4 Pressure contours on projectile at 3 longitudinal cross sections (ALP)

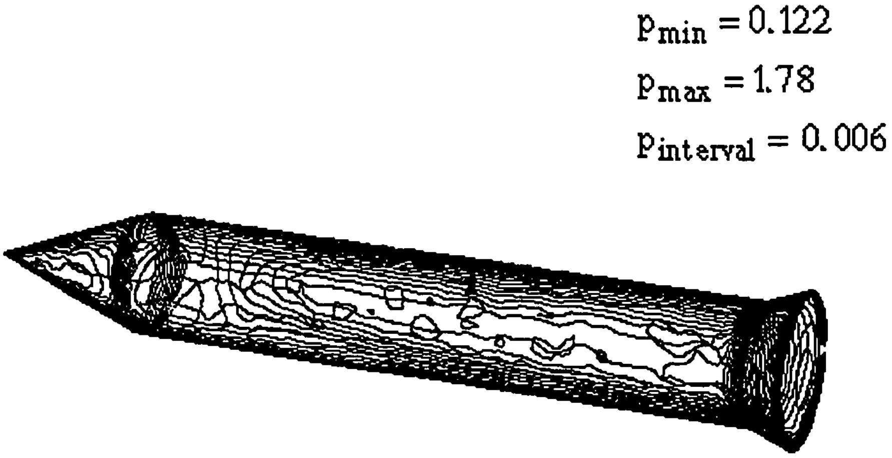


Fig. 4.5 Surface pressure contours over projectile (ALP)

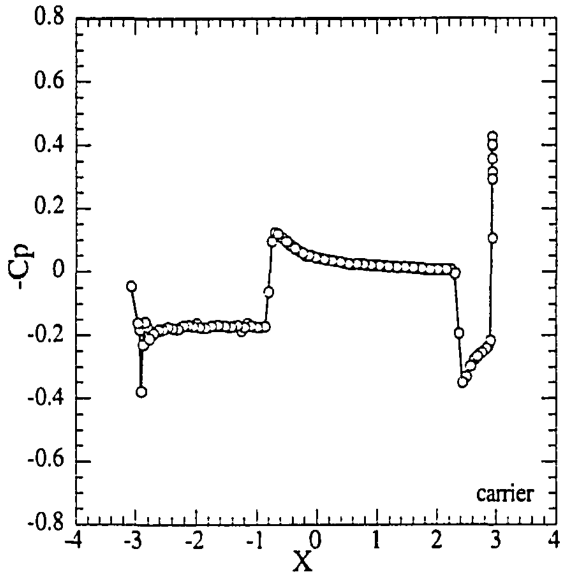


Fig. 4.5 Pressure coefficient distribution for carrier at plane of symmetry, $M_\infty = 1.5$, $\alpha_0 = 0°$.

Presented in Fig 4.3 and 4.4 are the surface grid for the carrier, along with the normalized pressure contours at the longitudinal symmetry plane and three cross-sectional planes, respectively. The nose shock of the carrier is attached. After the expansion at the forebody junction a shock surface is formed at the body-base junction. These are better seen in Fig. 4.5 which displays the surface pressure contours. The pressure coefficient distribution on the carrier at the longitudinal plane of symmetry is shown in Fig. 4.6 and delineates the shock and expansion structure on the carrier. Fig. 4.4 displays the longitudinal cross sectional pressure contours at three locations over the subpack. The wake region is slightly non-axisymmetric, which can be attributed to the presence of the subpack cylinder. The wake starts with a strong base shock which extends to impinge on the subpack cylinder. The base pressure drops to a low non-dimensional value of 0.1.

4.2 2D Adaptation

Two examples illustrating the adaptive window method for moving body problems are presented. The first example is for an NACA 0012 airfoil oscillating sinusoidally about the quarter chord with an amplitude of 35 degrees. The window constructed about this airfoil is shown in Fig. 4.7. This mesh contains 1,577 nodes and 3,042 cells; however, the adaptation window contains 569 nodes and 1,180 cells. Hence, only about 30% of the original mesh is being adapted. Detailed views of the adapted mesh for three different angles of attack are given in Fig. 4.8. To ensure the integrity of the mesh around the airfoil, the stiffness of the springs in this region are increased by increasing p from a value of unity to 1.8 in Eq. (3.42)

A second example demonstrates the applicability of the adaptive window procedure to multiple-body problems. This example illustrates how the adaptive window procedure may

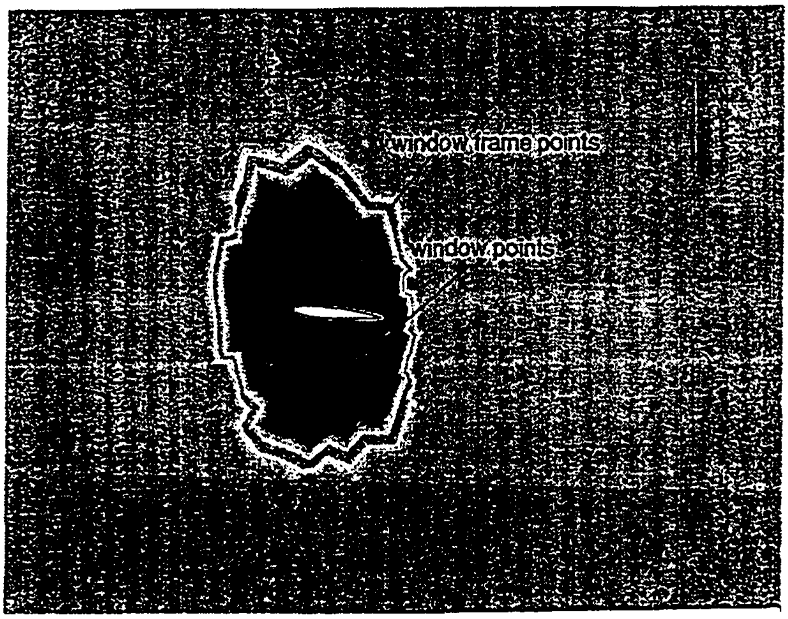


Fig. 4.7 Adaptive window for NACA 0012 airfoil.

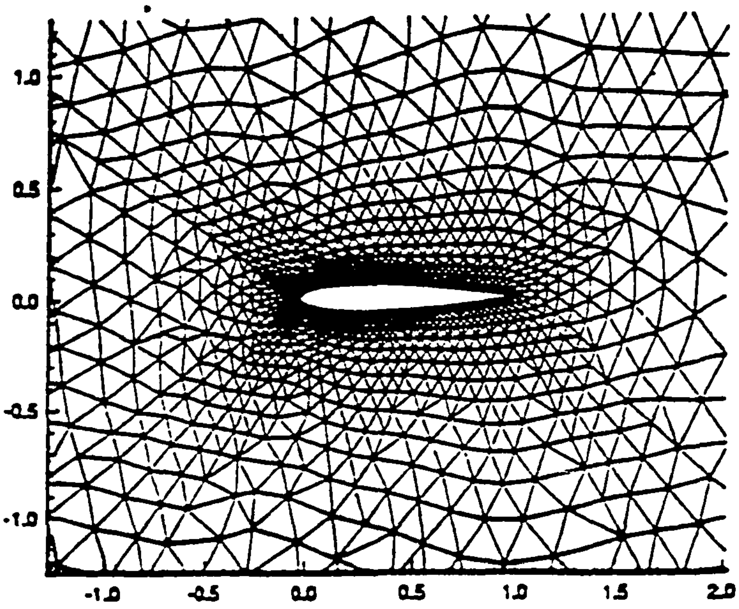


Fig. 4.8 (a) Initial mesh for NACA0012 airfoil.

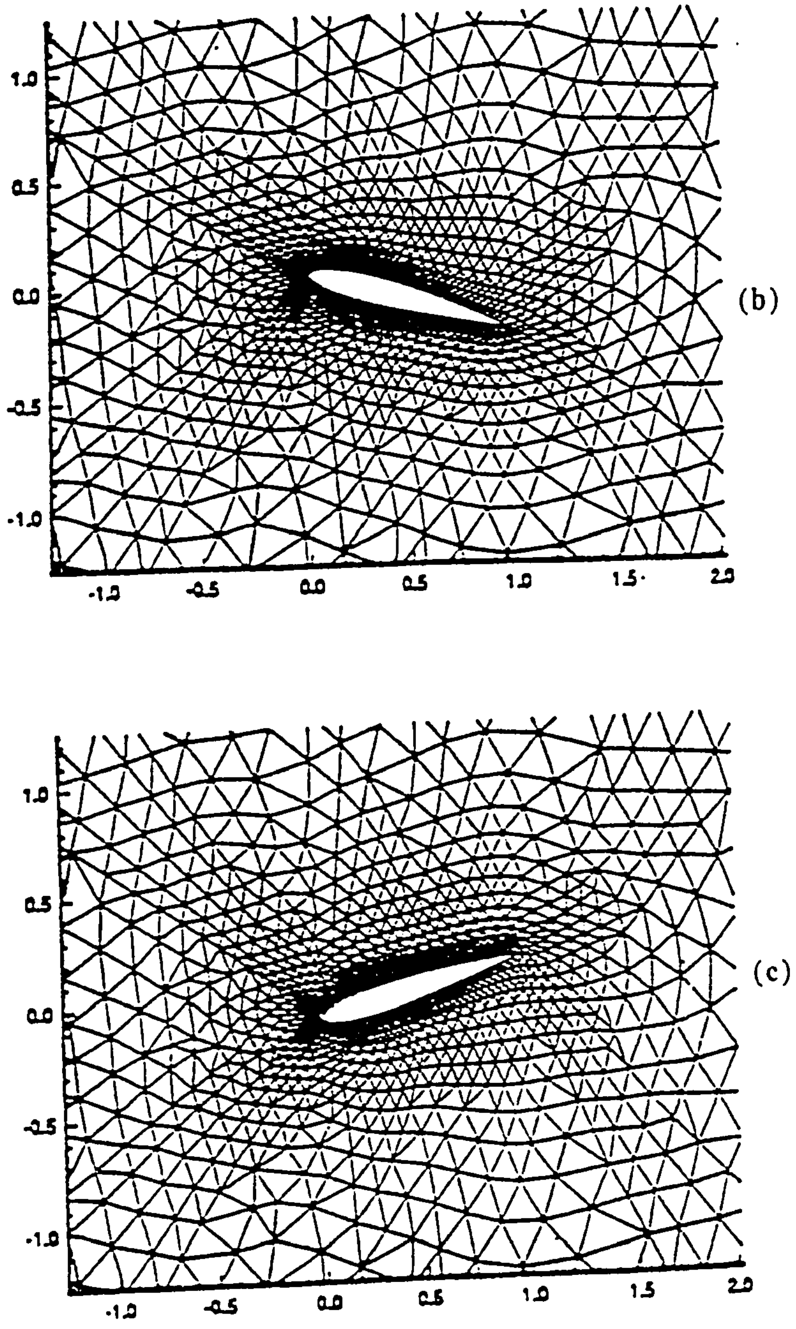


Fig. 4.8 Adaptive mesh for NACA 0012 airfoil (b) +35 degs. (c) -35 degs.

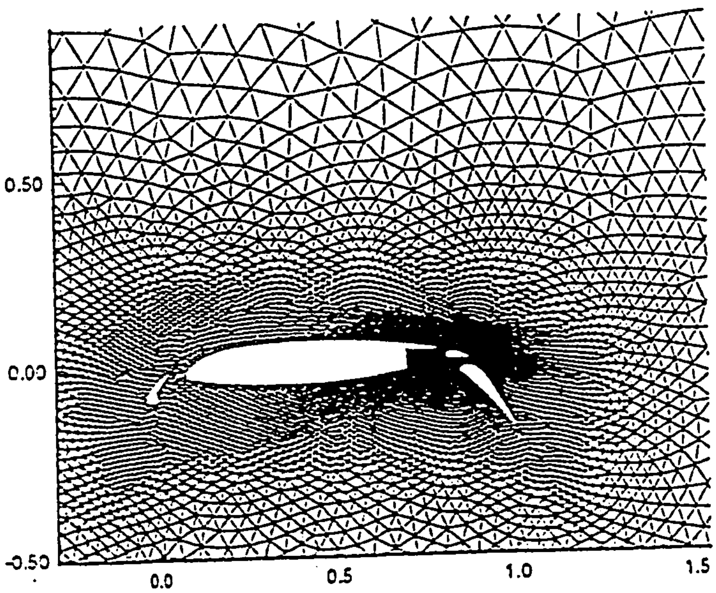


Fig. 4.9 Multielement airfoil mesh.

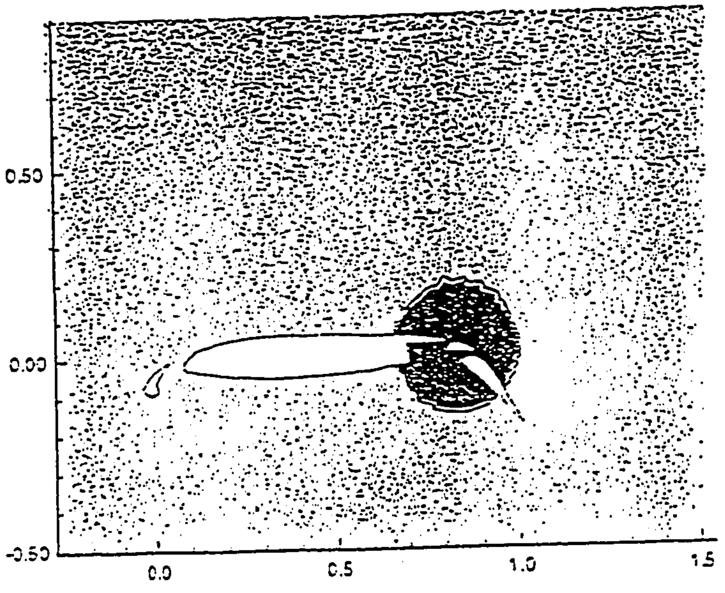


Fig. 4.10 Adaptive window for multielement airfoil.

be used to confine the adaptation region around different or multiple components in a four-element airfoil. This four-element airfoil has a double-slotted flap and a leading edge slat. The mesh and a window about the vane are shown in Figs. 4.9 and 4.10, respectively. Notice that the adaptation window is confined to a circular region around the vane and that it intersects the airfoil and the main flap. Regions of the mesh outside this window, for example, in the vicinity of the leading edge slat, are not affected by the movement of the vane and the subsequent grid adaptation. Conceivably, each element can be given different prescribed motions and the window region for each moving element can be adapted independently.

4.3 3D Adaptation

The primary focus of this work was to develop a reliable three-dimensional dynamic unstructured methodology that can handle unsteady moving boundary problems. One of the challenges posed by this class of problems is the development of a robust three-dimensional adaptation routine, which should ensure the integrity of grid cells and limit the formation of skewed cells. Hence, before embarking on the task of solving unsteady moving boundary problems, the dynamic three-dimensional mesh algorithm needs to be tested. To accomplish this, an ONERA M6 wing was considered; the wing had a leading edge sweep angle of 30 degrees, an aspect ratio of 3.8, and a taper ratio of 0.562. The three-dimensional coarse grid generated for the wing, was comprised of 35,000 cells and 6,000 nodes. The adaptive window used for the adaptation was a half ellipsoid. The spanwise displacement of the grid points in the plane of symmetry was enforced to be zero.

The ONERA M6 wing was given a prescribed one degree-of-freedom sinusoidal motion in angle of attack. Initially, to check the robustness of the adaptive window and dynamic mesh algorithm, the wing was given large amplitude oscillations ranging from +30 degrees to -30 degrees. To ensure the integrity of the grid cells near the body surface, the spring stiffness parameter, p, was increased to the value of 2.2 whereas the value of p=1.8 was used for the adaptations elsewhere. Shown in Figs. 4.11 and 4.12 are a cross-section grid and the surface grid of the wing at the two extreme positions. One of the important factors, while generating a window around the body of interest, was that the window size was selected such that it encompassed the body completely, and the window frame points did not lie extremely close to the surface of the body. For instance, if the span of the wing was one unit, a window size of at least 1.5 units was appropriate. Also, as deciphered from Fig. 4.12, even at large motions, the adaptation has not altered the integrity of the surface grid and, therefore, the surface definition.

4.4 6-DOF Trajectory Validation

An attempt has been made to test the present six-DOF trajectory method (Eqs. 2.11, 2.13 and 2.15) as follows. The separation of the store from the wing in the quasi-steady mode had been wind-tunnel tested and reported by Heim [54]. By quasi-steady mode, it is meant that a sequence of steady-state measurements were taken of the positions on a trajectory, which was computed based on these steady measurements of the forces and moments. The experimental configuration (Fig. 4.13) consisted of a clipped delta wing with leading edge sweep of forty-five degrees and a NACA-64A010 airfoil section. Connected to the wing is an ogive-flat-plate-ogive pylon, which was located 0.7d (store diameter) above an ogive-cylinder-ogive store when in carriage position. The parameters for store dynamics are given in Table 1. The freestream conditions for the experiment were $M_\infty = 1.2$, $\rho_\infty = 0.5564$, and $T_\infty = 430.60°R$. The axis system used in this case was defined with the x-axis pointing forward, y-axis pointing inboard, and z-axis pointing downward.

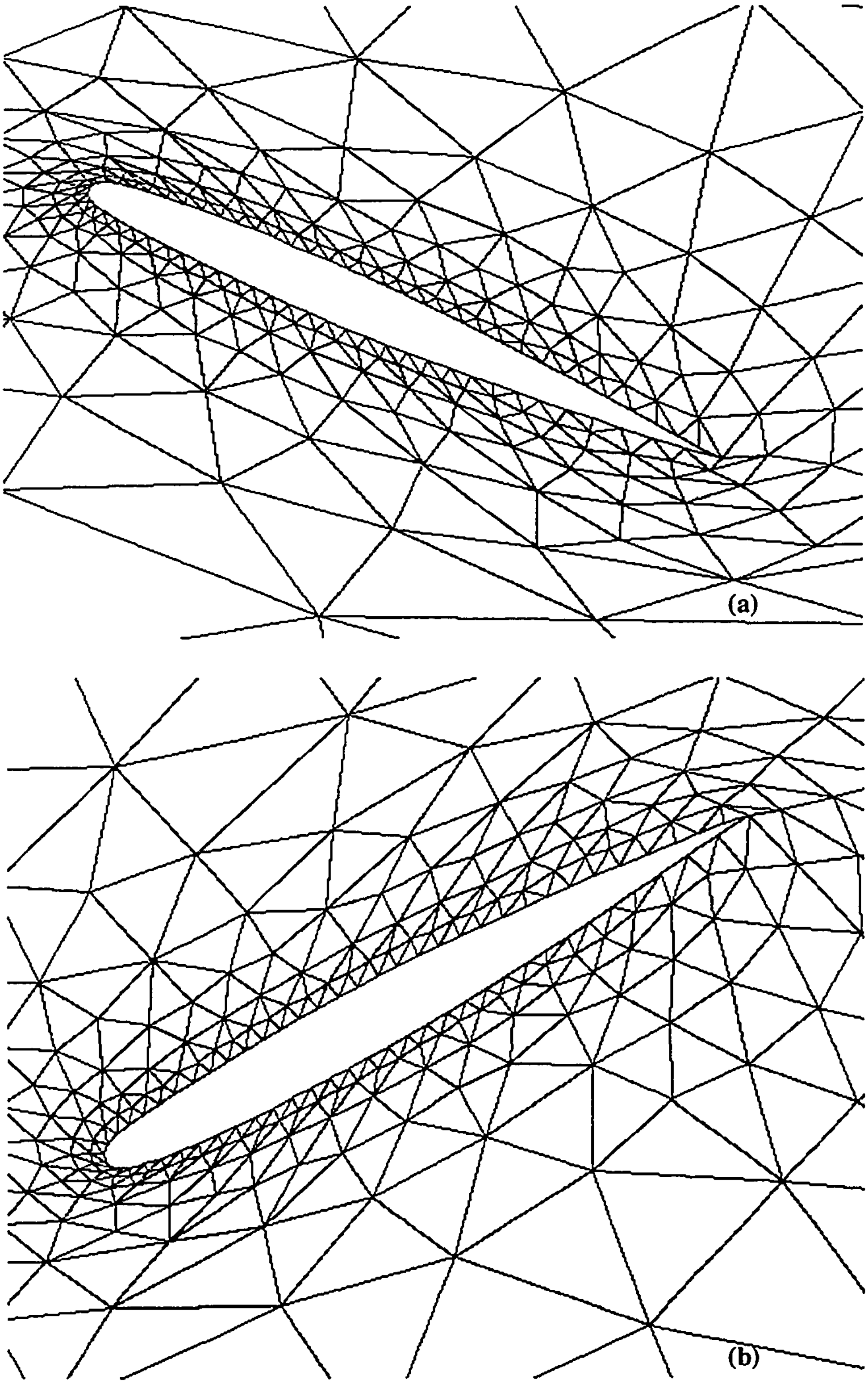


Fig. 4.11 Adapted mesh for plane of symmetry (O–M6 wing) (a) +30 degrees (b) –30 degrees.

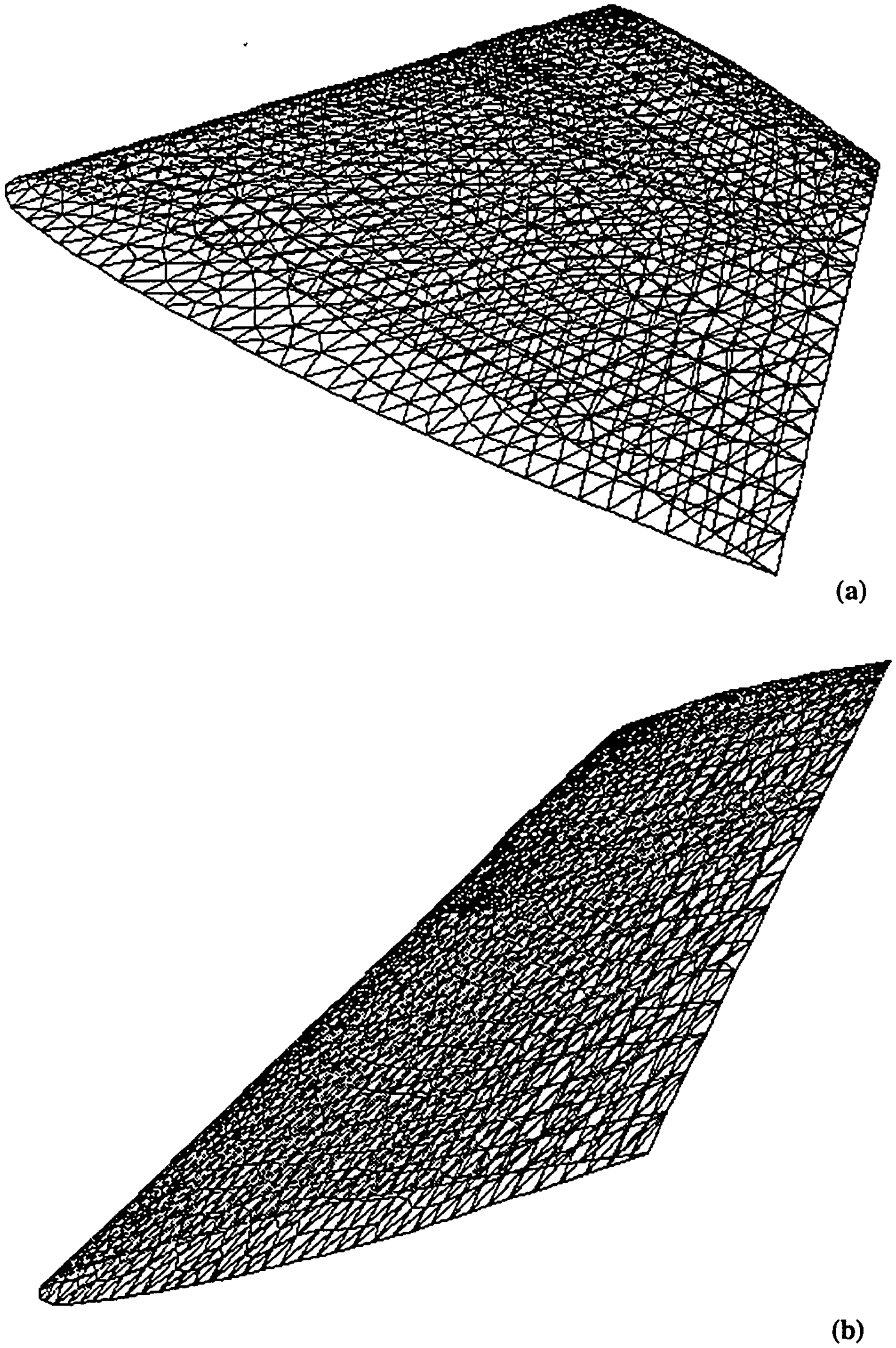


Fig. 4.12 Adapted surface grid for O–M6 wing (a) +30 degrees (b) −30 degrees.

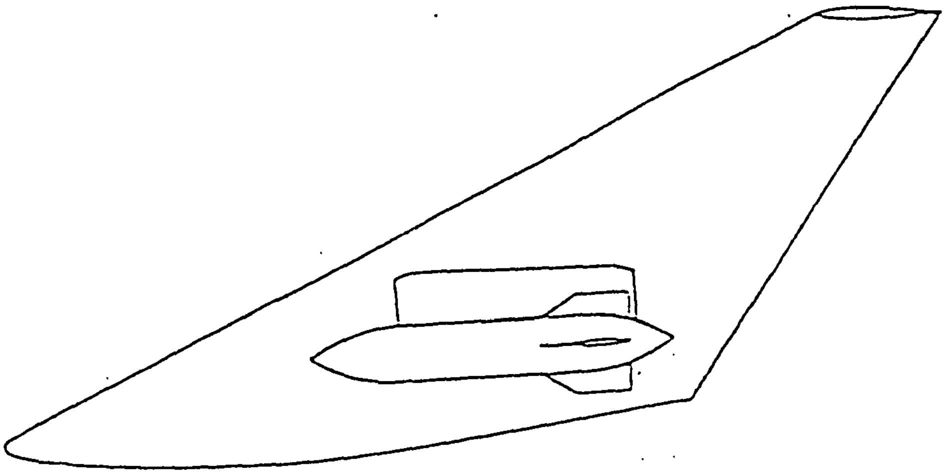


Fig. 4.13 (a) Schematic of the wing/pylon/finned store configuration.

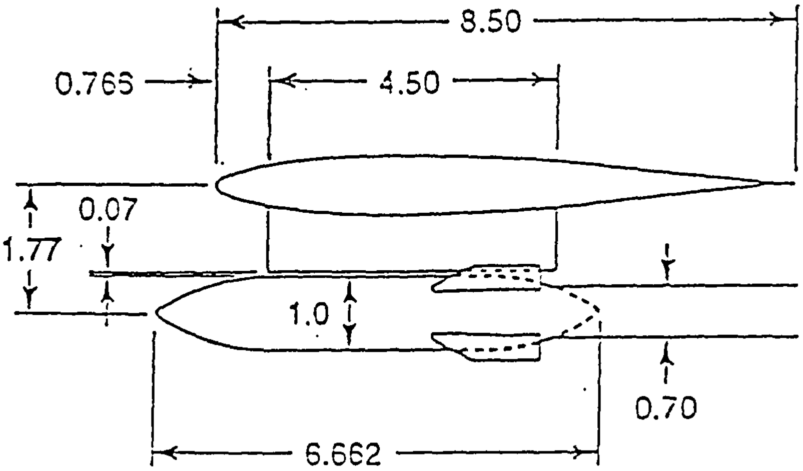


Fig. 4.13 (b) Cross-section of the configuration at wing-pylon junction.
(all dimensions in inches.)

By using the experimentally measured forces and moments as the input, the trajectory was computed. Then, the present trajectory was compared with the trajectory reported in [54] for the wind-tunnel tests as seen in Fig. 4.14. It should be noted, however, that the manner in which the store was ejected from its carrier position was not available, hence, a guessed value was used herein; an estimated ejector force was applied until the store dropped a specified distance, after which it was released to free fall. 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 degrees. The discrepancies in the results can be attributed to the fact that the ejector characteristics were not computationally simulated. In the present case the ejector force was applied until the store dropped a certain specified distance, after which it was released to perform a six degrees-of-freedom free fall.

Ejector force (Newton)	Mass (Kg)	Weight (Newton)	I_{xx} (Kg-m^2)	I_{yy} (Kg-m^2)	I_{zz} (Kg-m^2)
53416.71	905.2	8880.00	27.13	488.28	488.28

Table 4.1 Store parameters for trajectory validation

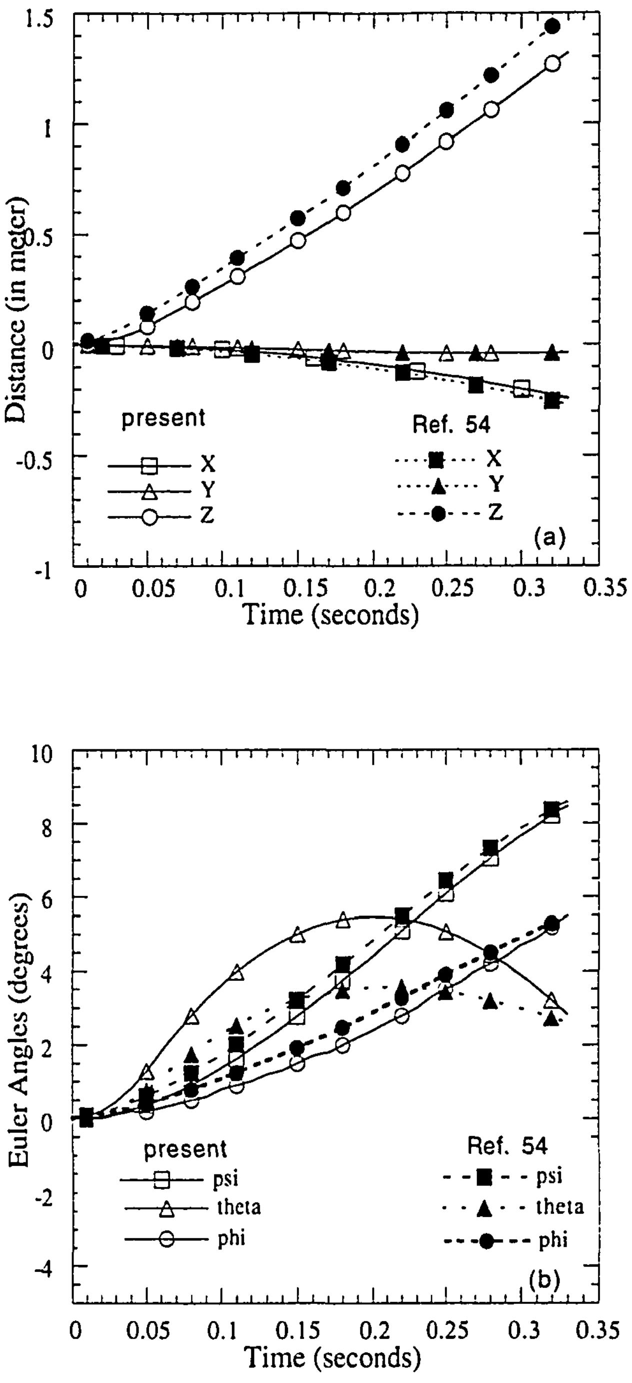


Fig. 4.14 Trajectory comparison (a) Displacements (b) Euler angles

Chapter 5

VALIDATION OF METHODOLOGY FOR MOVING-BOUNDARY PROBLEMS

5.1 Sinusoidally Oscillating Airfoil (SOA)

To validate the present adaptive window method and dynamic solution algorithm, a case where published experimental data [48] was available. This case was comprised of a NACA 0012 airfoil sinusoidally oscillating about its quarter chord at a mean incidence of $\alpha_0 = 4.86^\circ$. The pitching motion in angle of attack was given by

$$\alpha(t) = \alpha_0 + \alpha_1 \sin(M_\infty k\tau) \tag{5.1}$$

The amplitude, reduced frequency, and freestream Mach number of this oscillatory motion are $\alpha_1 = 2.44^\circ$, $k = 0.0810$, and $M_\infty = 0.6$, respectively. A grid dependency study was performed by comparing the solutions obtained on grids with 6,000 cells and 11,000 cells. The initial conditions for the unsteady problem was a fully converged steady state solution. For this case, the Roe's flux difference splitting method was used. The original mesh and the steady state pressure contours are shown in Fig. 5.1. The time step used for the case was 0.002, which allows to complete one cycle in 64,642 iterations and 6 hours of CPU time on a Cray Y-MP. Since the time step used was smaller than those used in other published results [4, 50], a temporal accuracy study was not performed. Illustrated in Fig. 5.2 are the instantaneous grids and the off-surface pressure contours at six positions during the airfoil's motion. In Fig. 5.3 the computed surface pressure coefficient distributions for the same six positions are compared with the experimental data.

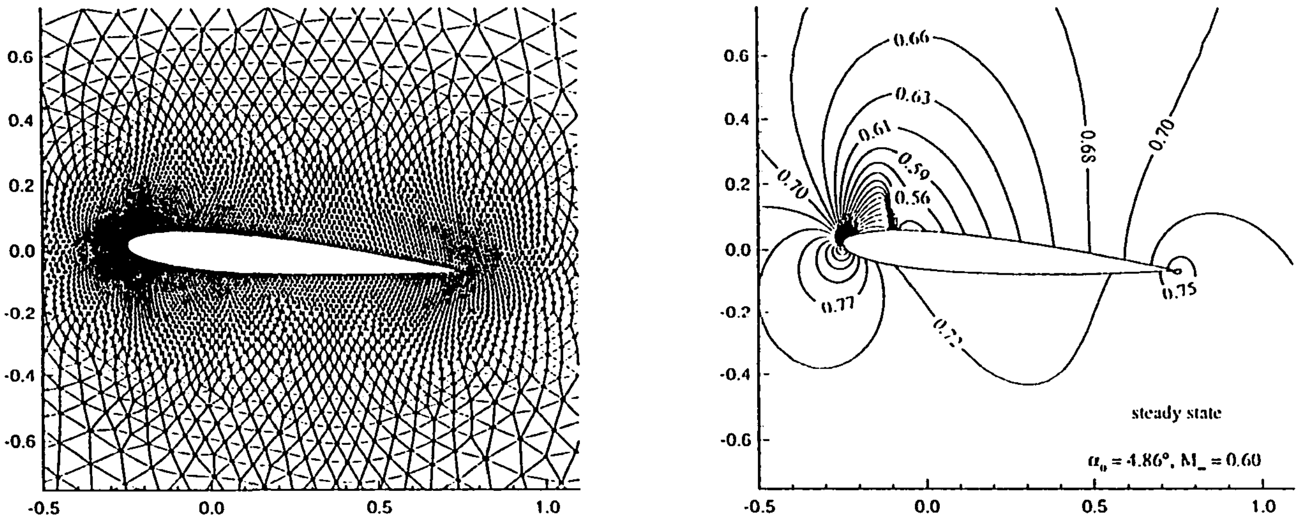


Fig. 5.1 Grid and off-surface pressure contours at steady state (SOA)

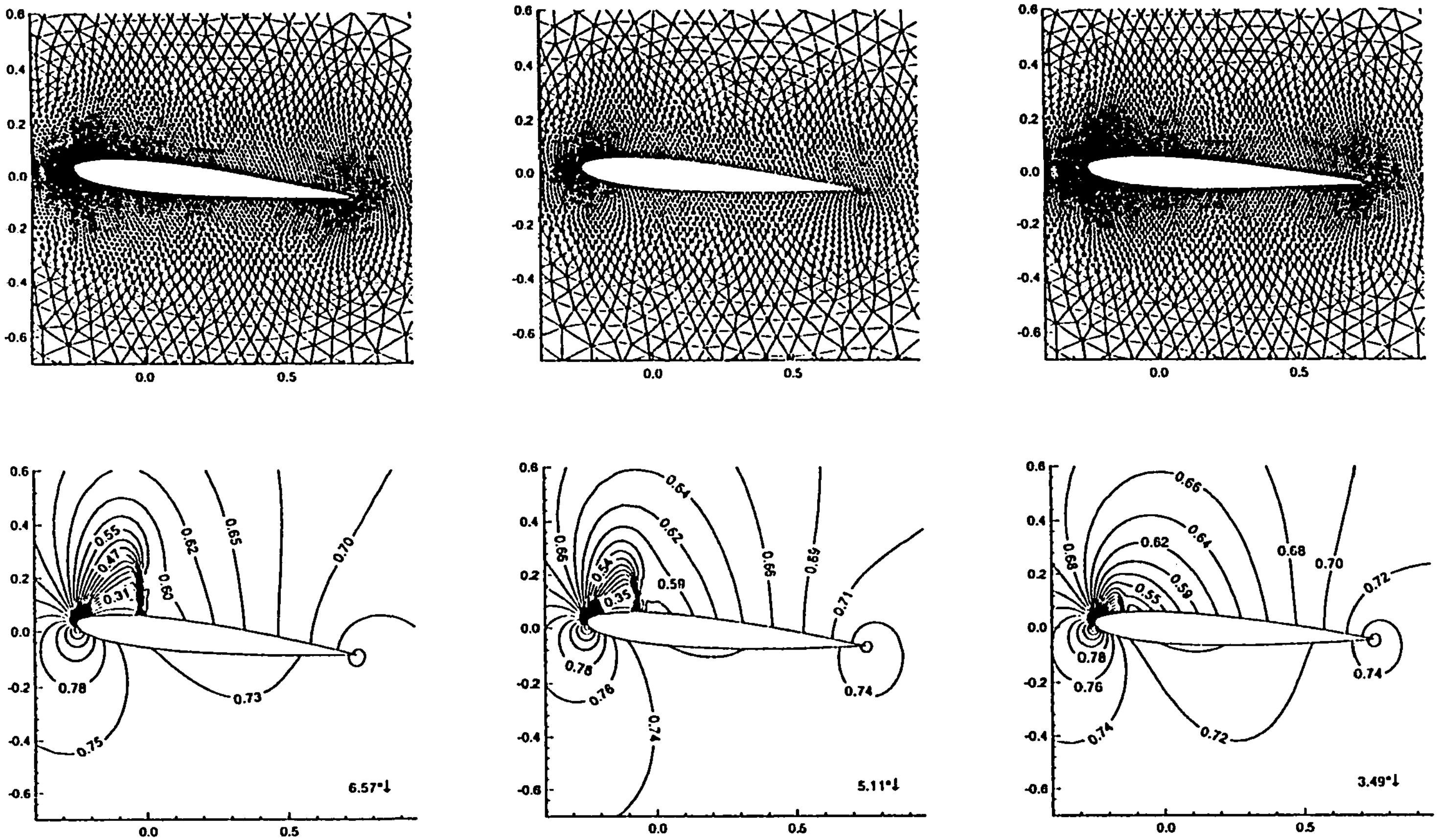


Fig. 5.2 Instantaneous grid and off-surface pressure contours for SOA ($\alpha = 4.86°$, $M_\infty = 0.60$, $k = 0.081$)

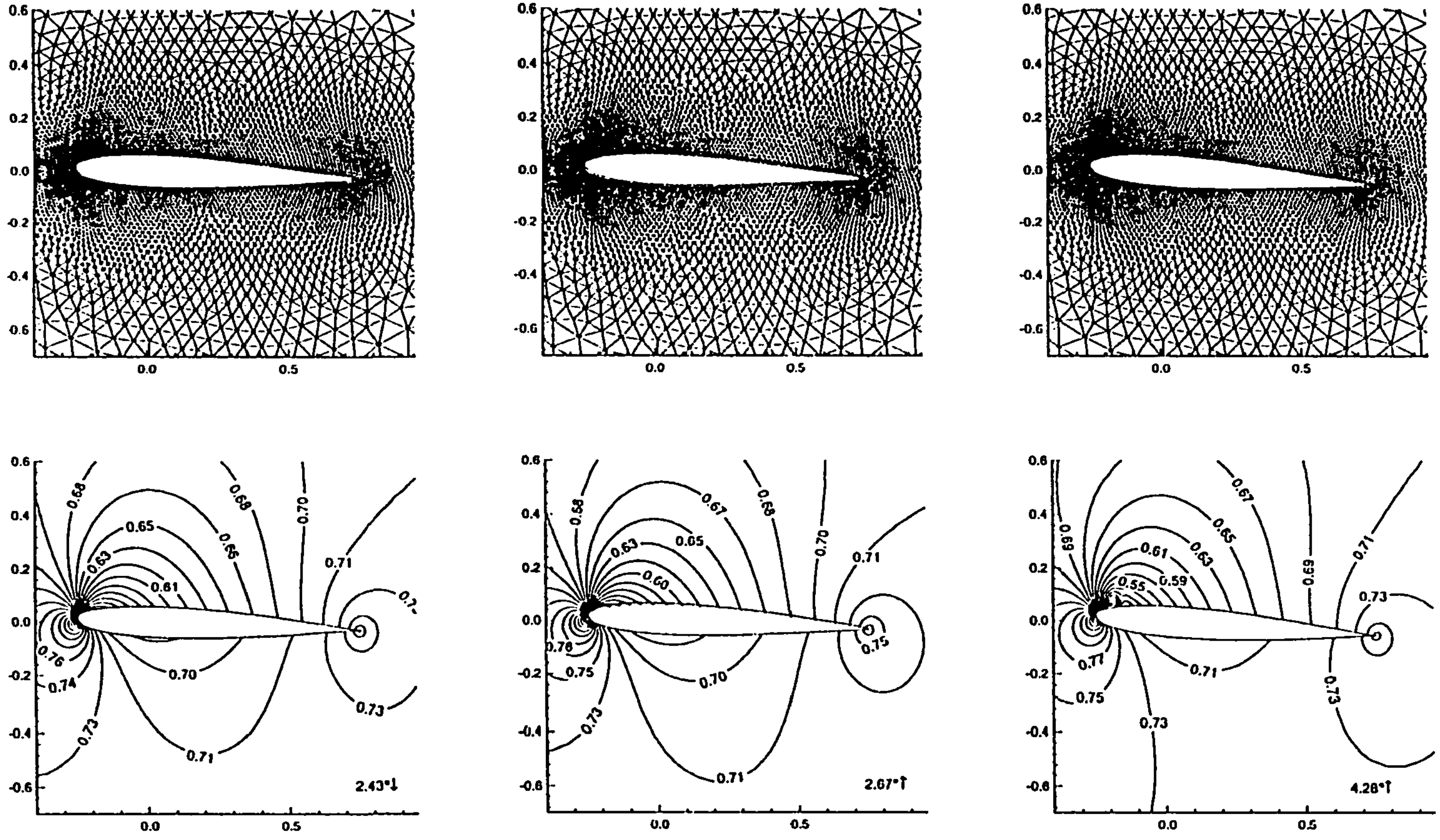


Fig. 5.2 Concluded ($\alpha_0 = 4.86\,°$, $M_\infty = 0.60$, $k = 0.081$)

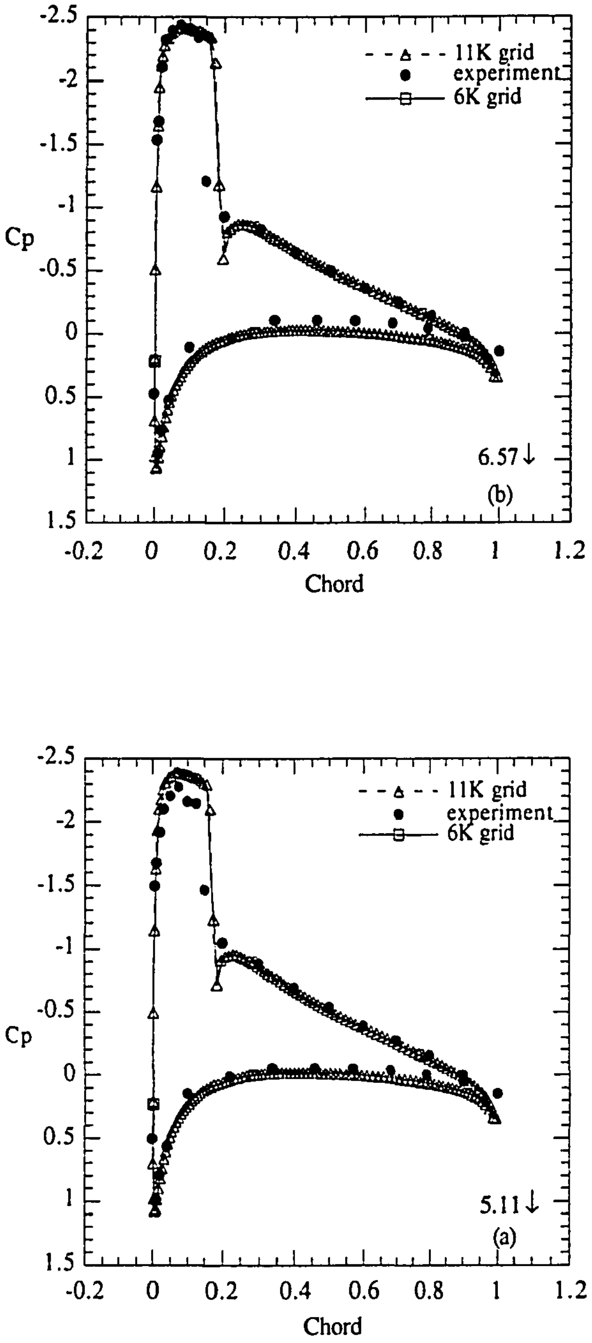
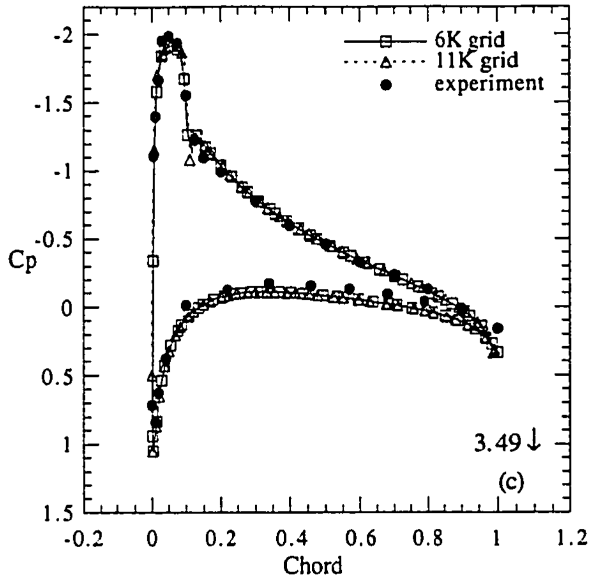


Fig. 5.3 Comparison of pressure coefficient (a) 6.57° ↓(b) 5.11° ↓ ($M_\infty = 0.60$, $\alpha_0 = 4.86°$, k = 0.081)



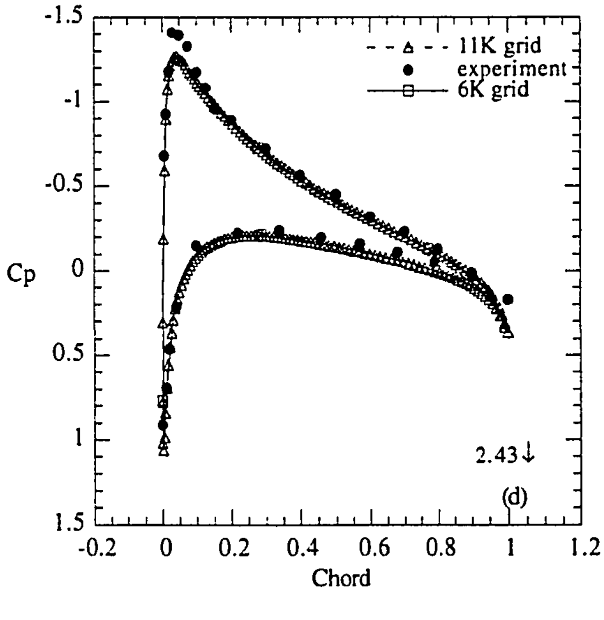
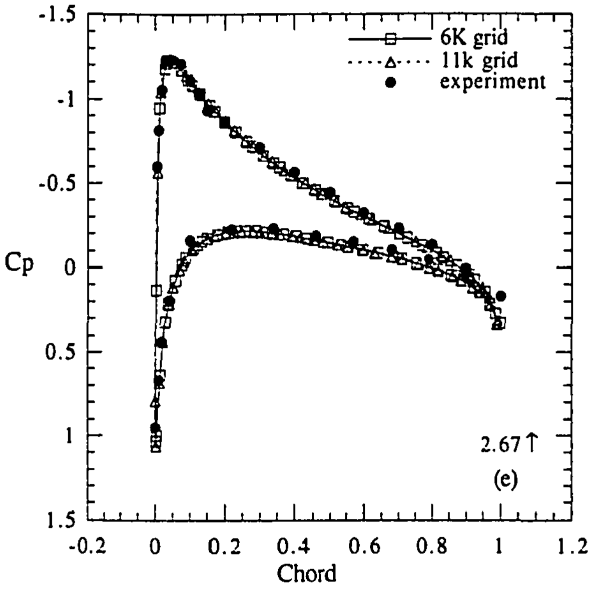


Fig. 5.3 Continued (c) 3.49° ↓ (d) 2.43° ↓
($M_\infty = 0.60$, $\alpha_0 = 4.86°$, k = 0.081)



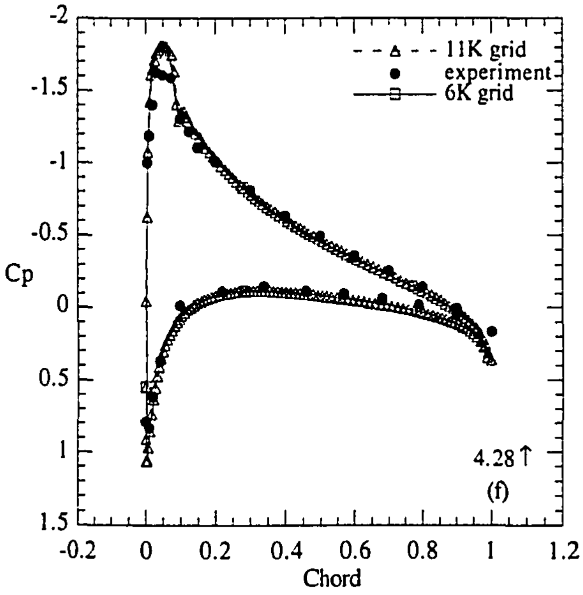


Fig. 5.3 Concluded (e) 2.67°↑ (f) 4.28°↑ ($M_\infty = 0.60$, $\alpha_0 = 4.86°$, $k = 0.081$)

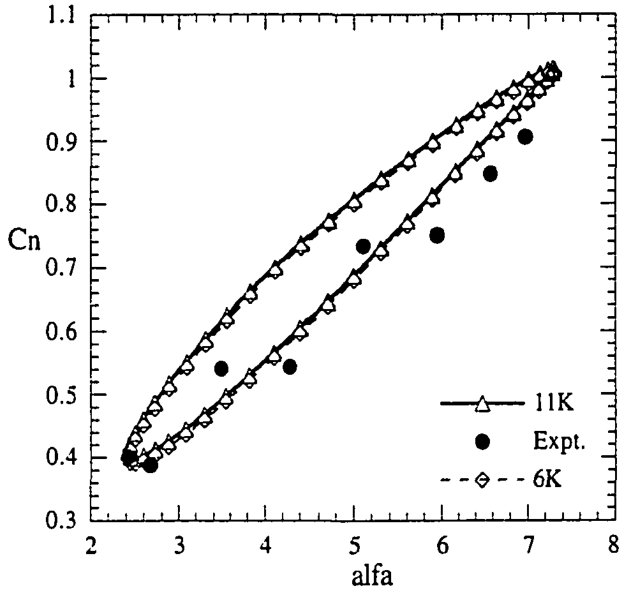


Fig. 5.4 Normal Force coefficient against angle of attack for SOA (alfa taken in counter-clockwise sense)

($M_\infty = 0.60$, $\alpha_0 = 4.86°$, $k = 0.081$)

Note that in Fig. 5.3, the pressure coefficient distribution obtained for both these grids are almost identical and confirms that the solution is practically grid-independent. A shock was formed on the upper-surface, which migrated towards the leading edge as the angle of attack was increased. With the decrease in angle of attack, the shock migrated away from the leading edge, and it disappeared as the angle approached the mean incidence and at all the subsequent angles of attack below mean incidence. Good agreement was observed between the computed and experimental results at all six positions. Small discrepancies over the first 5-10% of the airfoil's upper-surface were believed to be related to the viscous effects neglected with these Euler computations. The maximum disparity occured at angles 3.49↓ and 2.43↓ degrees, respectively. Similar discrepancies between the computed results of [44] and the experimental data of [55] had been reported. The overprediction of pressures at lower airfoil surface for all cases suggested that an angle of attack correction may be needed. Presented in Fig. 5.4 are the variation of the normal force coefficient with the angle-of-attack. As would be expected from an inviscid computation, the agreement improved at lower angles of attack. The plot of normal force coefficient versus angle of attack for both these grids, Fig. 5.4 further confirmed the adequacy of these grids.

5.2 Oscillating Rectangular Wing

The present methodology was developed for a multibody configuration in relative motion; but first, its validation was attempted using a case for which careful experimental data existed: a rectangular half-wing with NACA 64A010 airfoil sections and a complete aspect ratio of 4, in forced sinusoidal pitching motion, placed in Mach 0.8 flow (just supercritical) at zero incidence [56]. The angle of attack varied periodically according to expression given below:

$$\alpha(t) = -\alpha_1 \sin(M_\infty k \tau) \tag{5.2}$$

where

$$k = \frac{f\,c}{U_\infty} \tag{5.3}$$

and the amplitude α_1 and the reduced frequency k of the motion were 1 and 0.27 degrees, respectively. It is important to note that the reduced frequency k is the measure of the unsteadiness of the flow and is given by the Eq. (5.3), where f is the circular frequency, c is the speed of sound, and U_∞ is the freestream velocity of the fluid. The domain boundaries for the unstructured mesh generated for the wing extended 12 chords normally and chordwise and 4 chords spanwise, and had 40,533 cells and 7,775 nodes (Fig. 5.5). After obtaining the initial steady-state solution, time-accurate calculations were performed for three cycles, and the third was deemed as the limiting cycle (Fig. 5.6). The time step used for the time-accurate calculations was 0.0018. Present results required 48 Megabytes, and 0.5 and 7.7 CPU hours on a Cray Y-MP computer for steady and one-cycle unsteady computations, respectively. The corresponding unit processing times were about 21 and 42 microseconds per time step per cell, respectively.

To describe the harmonic variations, two quantities are needed, namely, magnitude and phase shift with respect to the oscillating body. An alternative method of describing this is by the complex number notation, which presumes that the fluid particle's response to a sinusoidal excitation is also sinusoidal. Here, the real part of the pressure perturbation is in phase with the oscillating body, and the imaginary part is out of phase with it. The real part of the pressure perturbation can also be considered as a measure of the actual pressure perturbation at the instant the oscillating body reaches its maximum positive deflection whereas the imaginary part of pressure perturbation is the measure of the pressure perturbation at the instant the oscillating body passes its mid position in the positive direction.

For the harmonic motion considered here for validation, pressure coefficient histories computed in the time-domain were Fourier decomposed into their complex components,

$$\text{Real}\{ \ \bar{C}_p \ \} = \frac{2}{\alpha_1(t_2 - t_1)} \cdot \int_{t_1}^{t_2} \left[C_p(\tau) \cdot \{ \ \sin(M_\infty k\tau) \ \} d\tau \right]$$

$$\text{Imaginary}\{ \ \bar{C}_p \ \} = \frac{2}{\alpha_1(t_2 - t_1)} \cdot \int_{t_1}^{t_2} \left[C_p(\tau) \cdot \{ \ \cos(M_\infty k\tau) \ \} d\tau \right] \qquad (5.4)$$

to distinguish the in-phase from out-of-phase pressures [56]. In Eq. (5.4), α_1, t_2-t_1, M_∞ and k denote, respectively, the pitching magnitude, cycle time, freestream Mach number, and the reduced frequency.

After obtaining the steady-state flow solution, the unsteady computations were performed and the results compared with the experimental data [38]. Fig. 5.6 shows the surface grid over the rectangular wing used for the flowfield computations. Figure 5.7 shows a comparison between the steady Euler pressure coefficients with the experimental values at 50%, 77%, and 94% semispan locations. The results compared very well except the fact that the shock appears smeared. This might be due to the use of a coarse grid. Comparisons of the present unsteady computations with the data (three spanwise stations presented in Figs. 5.8-5.10; two supercritical and one fully-subsonic critical) were by and large successful. At the 50 % spanwise location (Fig. 5.8), there is supersonic flow for a wide extent of the chord.

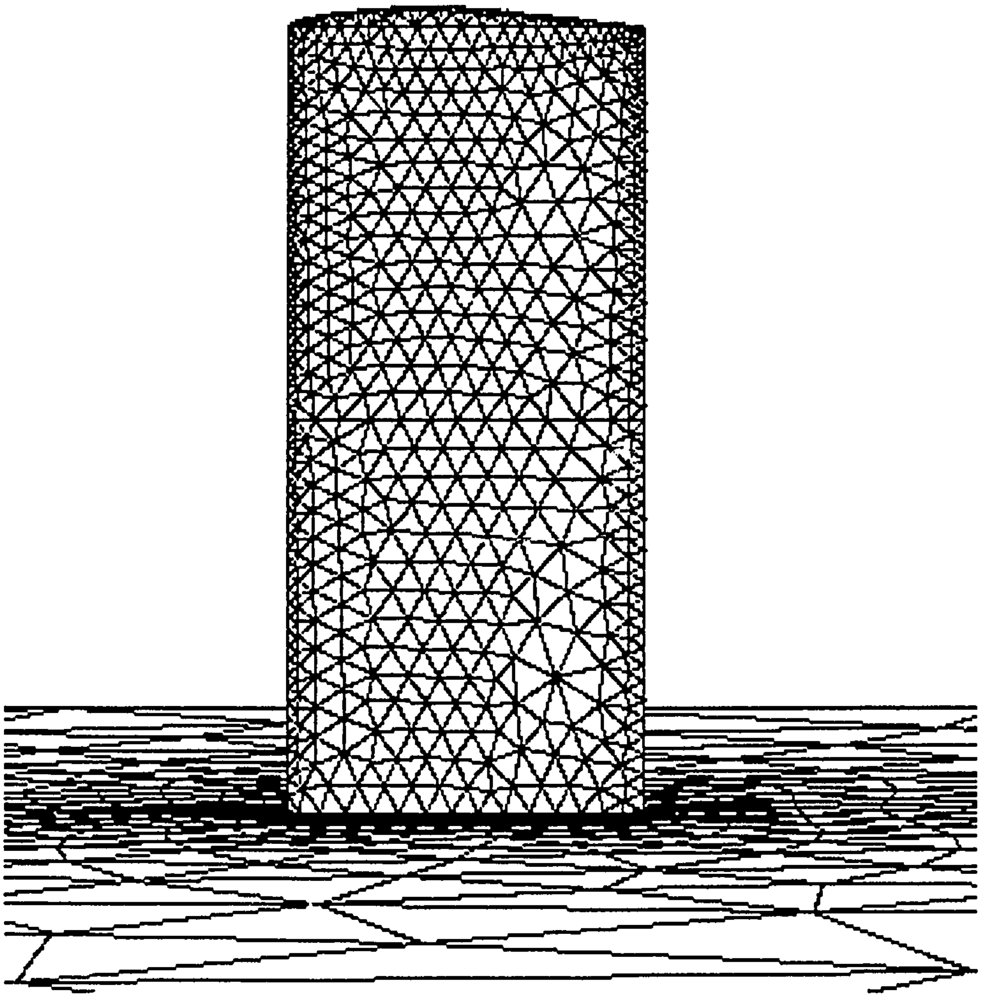


Fig. 5.5 Surface grid of rectangular wing

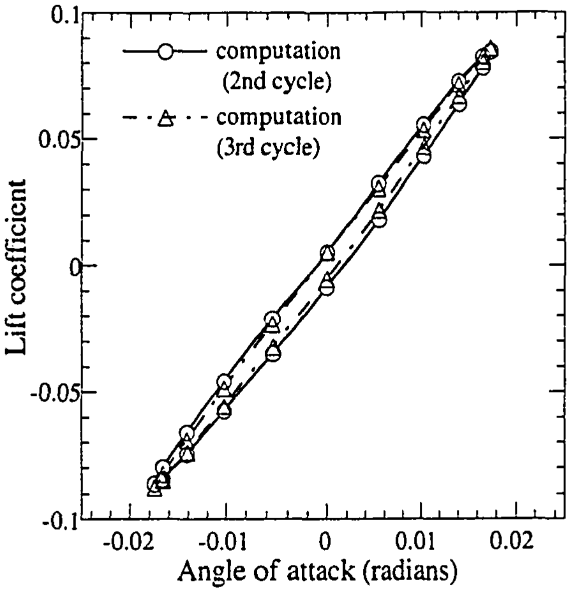


Fig. 5.6 Lift Coefficient against angle of attack for rectangular wing ($M_\infty = 0.80$, $\alpha_0 = 0°$, $k = 0.27$)

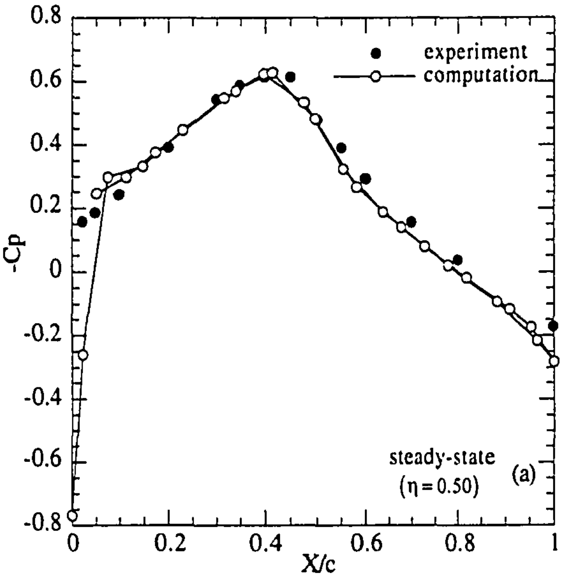


Fig. 5.7 Comparison of steady state pressure coefficient (a) 50% semispan $\left(M_\infty = 0.80, \alpha_0 = 0°\right)$

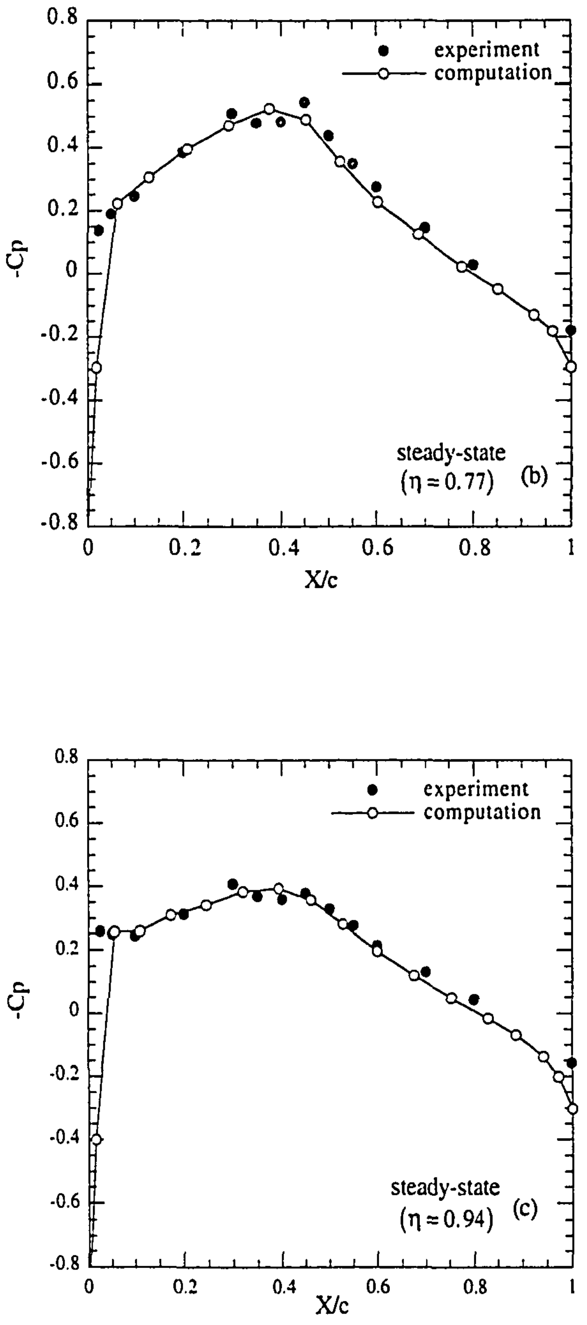
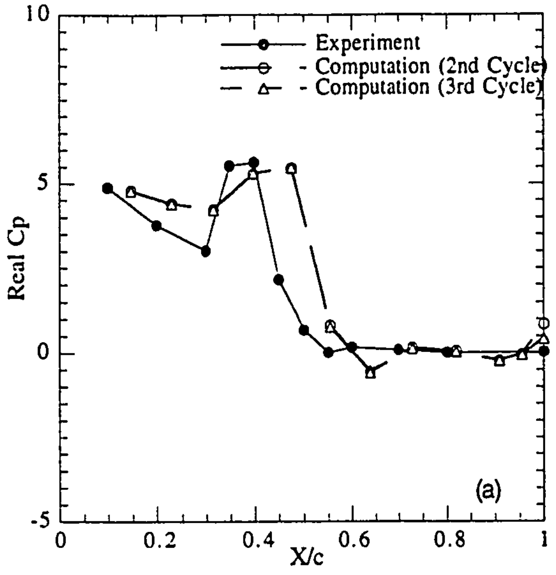


Fig. 5.7 Concluded.

(b) 50% (c) 77% semispan $\left(M_\infty = 0.80,\ \alpha_0 = 0°\right)$



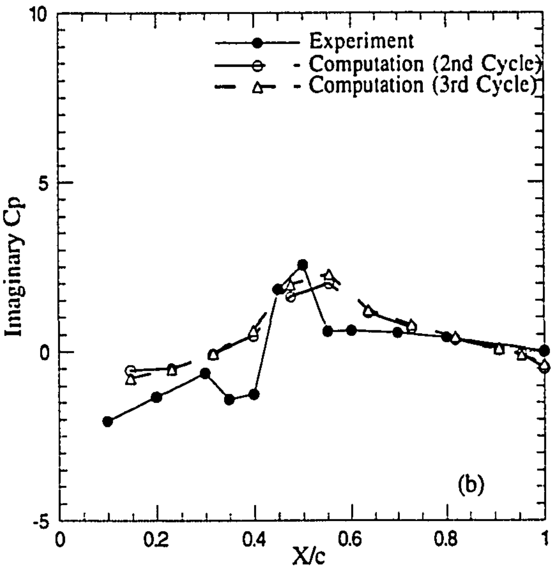


Fig. 5.8 Comparison of pressure coefficient at 50% semispan (a) Real (b) Imaginary ($M_\infty = 0.80$, $\alpha_0 = 0°$, $k = 0.27$)

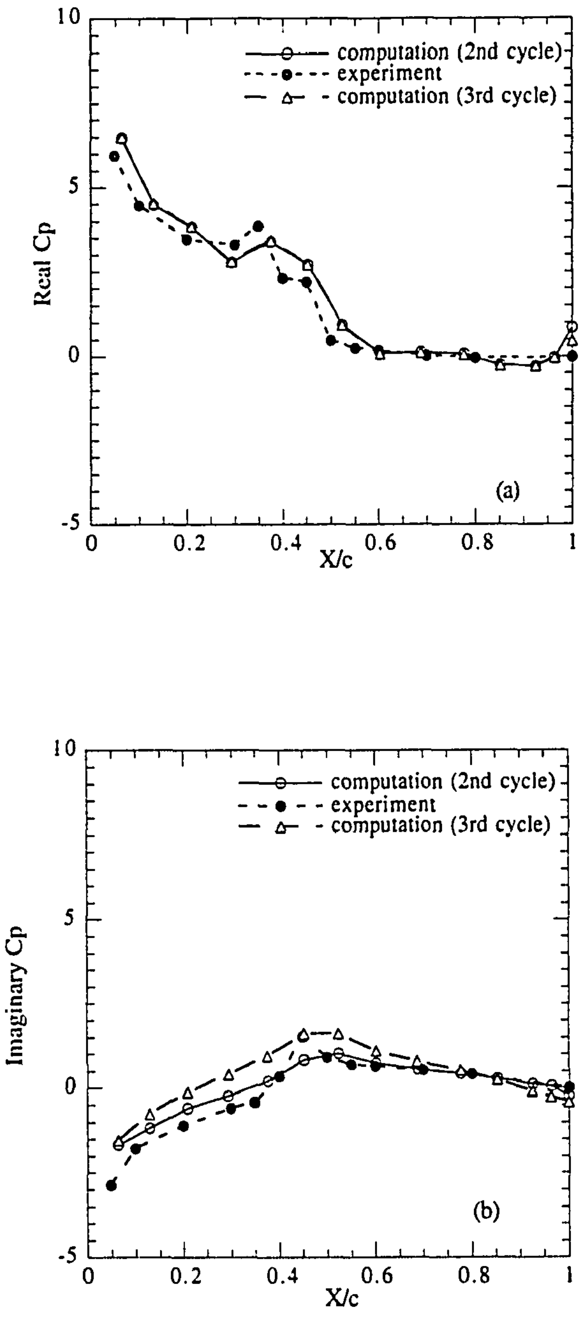
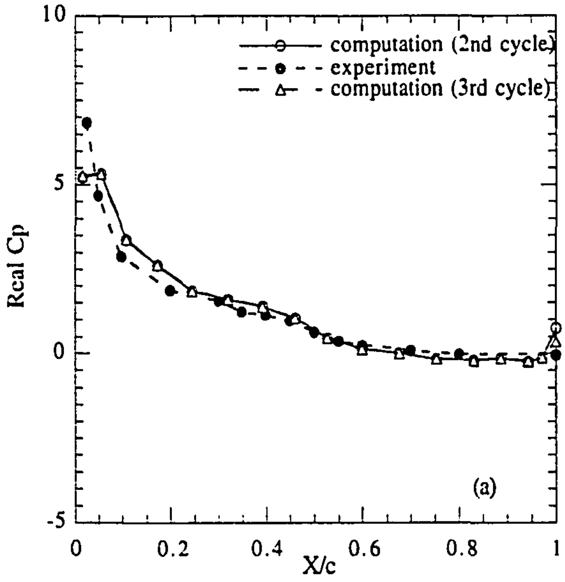


Fig. 5.9 Comparison of pressure coefficient at 77% semispan (a) Real (b) Imaginary ($M_\infty = 0.80$, $\alpha_0 = 0°$, $k = 0.27$)



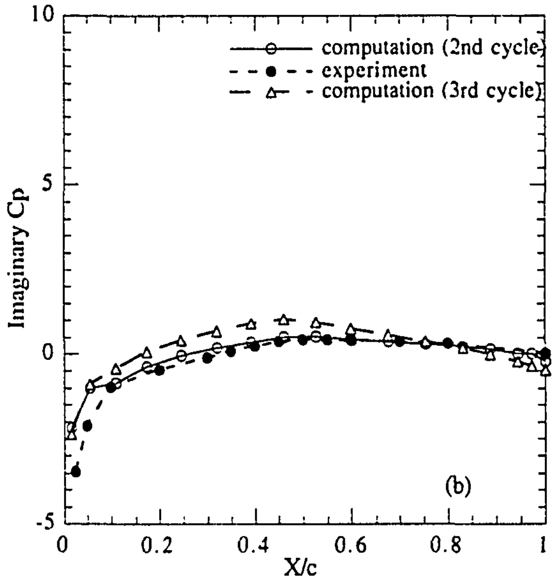


Fig. 5.10 Comparison of pressure coefficient at 94% semispan (a) Real (b) Imaginary ($M_\infty = 0.80$, $\alpha_0 = 0°$, $k = 0.27$)

The curve for imaginary pressure coefficient distribution has a lowered maxima and minima and is attributed to the supercritical flowfield. At the 77% spanwise location (Fig 5.9), the extent of supersonic flow is reduced and the oscillating pressures are less effected compared to that of 50% spanwise location. Finally, subsonic flow is observed everywhere for the outboard section at 94% spanwise location (Fig 5.10). Hence, real and imaginary pressure coefficient distribution are found to be subcritical in characteristics.

The figures clearly indicate that the shock traverses 25% of the chord length in one complete cycle. Some discrepancy in the out-of-phase pressures was observed, even amongst the second and third cycle computations, which might be attributed to a number of factors, such as: pressure oscillations with differing phases on upper and lower surfaces, overprediction by the inviscid equations, relatively coarse grid resolution in shock-excursion region, discrepant representation of the wing tip between computation (round) and experiment (flat), and, certainly, the truncation error.

Chapter 6

DEMONSTRATION OF OVERALL METHODOLOGY

6.1 3-DOF Airfoil Motion

This case also involves a prescribed motion of a NACA 0012 airfoil as in section 5.1, but this time the airfoil traverses a 3-DOF trajectory. This motion assumes that the x-position of the center of gravity (quarter chord) and the angle-of-attack of the body vary linearly with time, whereas, the y-position is given a parabolic relationship. For this case the freestream Mach number and angle-of-attack are 0.8 and 1.25 degrees, respectively. For this case, flux vector splitting of van Leer is used.

Unsteady solutions about this moving body are presented in Fig. 6.1 via the instantaneous grids and off surface pressure contours for the body at three positions along its trajectory. The initial position has the center-of-gravity at the origin with a zero degree angle of attack, whereas position 3 relocates the center of gravity at 1.5 chords below and 1.25 chords aft the origin with a -15 degrees angle of attack. It should be noted that the abscissa and ordinate are at different scales for each position in order to closely view the flow field characteristics near the body. The initial position corresponds to the initial condition for this case, which is a fully converged time accurate solution. As the airfoil begins to plunge down, the strong upper surface shock moves forward and the weak lower surface shock strengthens and moves aft. By the time the airfoil reaches position 3, only a strong lower surface shock exists with a mild expansion on the upper surface. This case

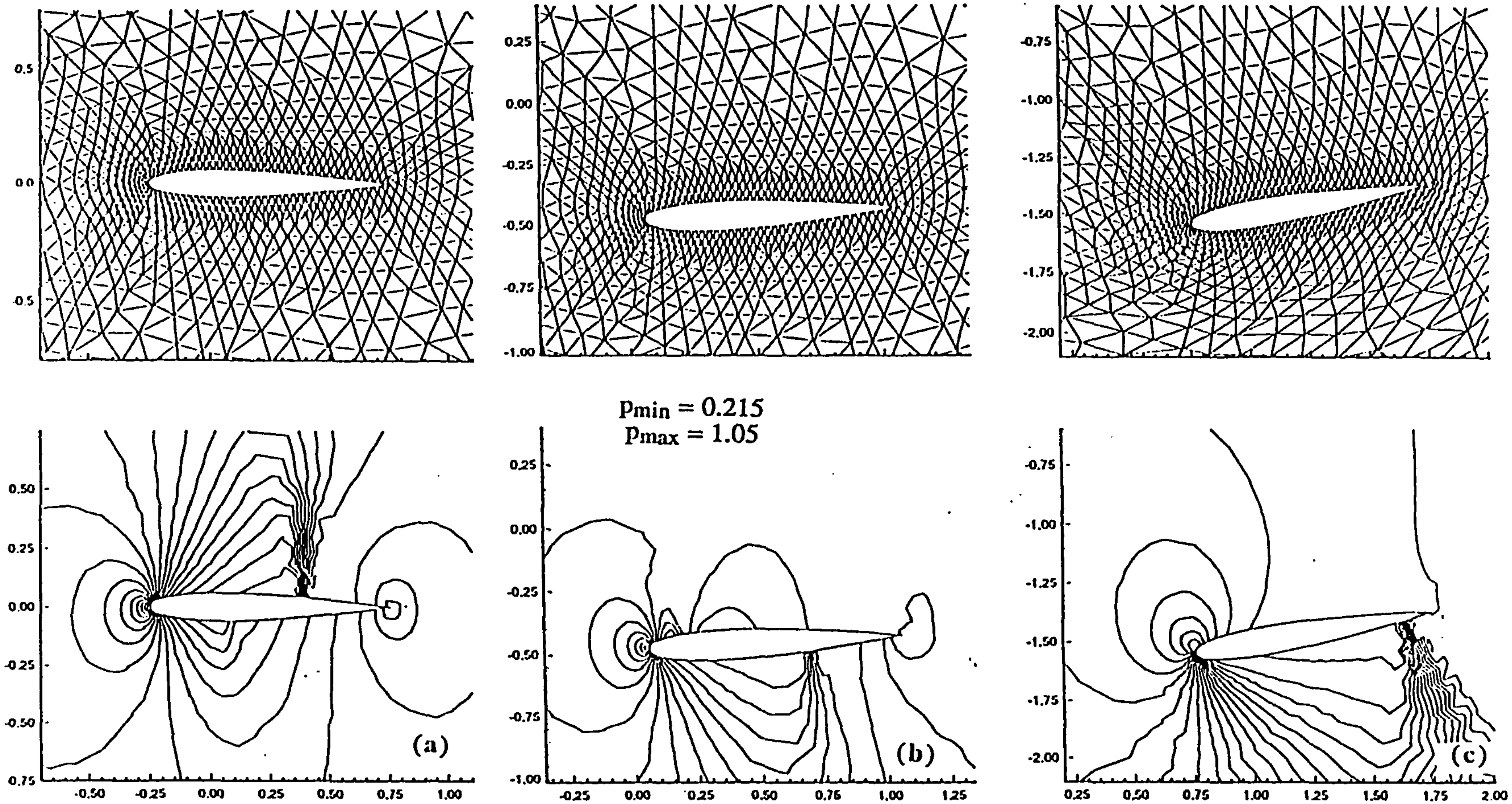


Fig. 6.1 Instantaneous grids and off-surface pressure contours for 3-DOF airfoil. (a) position 1 (b) position 2 (c) position 3 $(M_\infty = 0.80,\ \alpha_0 = 1.25°)$.

serves as a test of the robustness of the adaptive window procedure and the flow solver for large displacements.

6.2 2D Store Separation

The primary focus of this research is the development of a dynamic unstructured solution method which can aerodynamically determine the free-fall motion of released bodies. Even though the methodology has been developed for three-dimensional flows and for six-DOF motion, (as will be demonstrated later) initially a two-dimensional test case that limits the trajectory to a three-DOF motion has been chosen for computational efficiency. This case simulates the unsteady flow about an airfoil/store (AS) configuration where the store has been released and its position is determined by the aerodynamic forces and the force of gravity. This configuration has been adapted from the three-dimensional store separation problem detailed in [54] and [57], which will be discussed in section 6.3. The grid and the initial adaptive window for AS configuration are shown in Fig. 6.2. The grid contains 19,707 cells and 10,073 nodes and the computational domain is 50 store diameters long. The NACA 64A010 airfoil represents a cross-section of the wing reported in [54, 57, 58]. The store has an ogive-cylinder-ogive cross section. All dimensions in this study are normalized by the store diameter. Time has been non-dimensionalized using local speed of sound and store diameter. The time step used for this case is 0.001 and it takes 2.5 Cray Y-MP hours (CPU time) to perform the store separation sequence.

Depicted in Fig. 6.3 are the off-surface pressure contours for the initial condition (steady-state flow of static AS at $M_\infty = 0.3$) and three selected positions from the separation sequence. Position 1 displays the beginning of a compression region near the store's lower surface. This compression is caused by the moving-store-induced force and the subsequent flow. It should be noted that this simulation is two-dimensional, which does not allow the lateral relieving effect of axisymmetric or three-dimensional flows. Hence, a nozzle-like flow behavior is observed between the airfoil and the store. At position 2, a strong compression is developed between the airfoil and the store. Furthermore, near the store's trailing edge, a vortex is formed. By the time the store drops to position 3, the compression region between the store and the wing develops into a strong normal shock, the trailing edge vortex is enlarged and moved downstream, and the store-induced downward flow is strengthened.

Displayed in Fig. 6.4 are the pressure coefficient distributions on the store and the airfoil surfaces. In order to compare the change in the pressure distributions as a result of the separation, initial steady-state distributions are superimposed on the instantaneous distribution at position 3. As expected, there is little change on the upper airfoil surface. However, the lower airfoil surface indicates an expansion terminated by a shock. The upper store surface also displays the effect of the shock and the following compression. The lower store surface experiences a significant distribution change in time; from a predominantly constant distribution, it transforms to a large compression at the leading edge followed by an expansion extending to the trailing edge.

This separation sequence was also simulated using the Roe flux-difference splitting method. It can be observed, through the sample comparison of position 3 in Fig. 6.5a using van Leer's scheme with the same position in Fig. 6.5b using Roe's scheme, that almost identical results are obtained with a slightly better shock resolution of shock with Roe's scheme as expected.

The aerodynamically determined three-DOF trajectory is shown in Fig. 6.6. The translational motion is primarily downward with a small axial displacement. The rotational motion is an initially gradual, but then significantly increasing, pitch-down motion. It should be noted that no unsteady experimental data exists for this configuration.

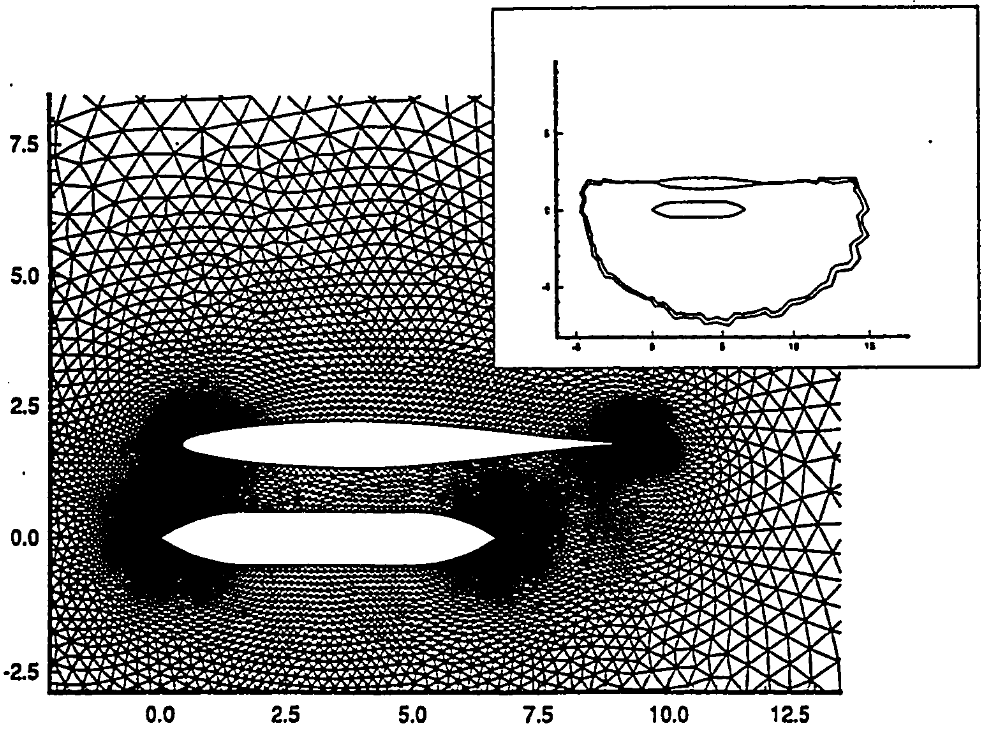


Fig. 6.2 Grid and the initial adaptive window for initial AS configuration.

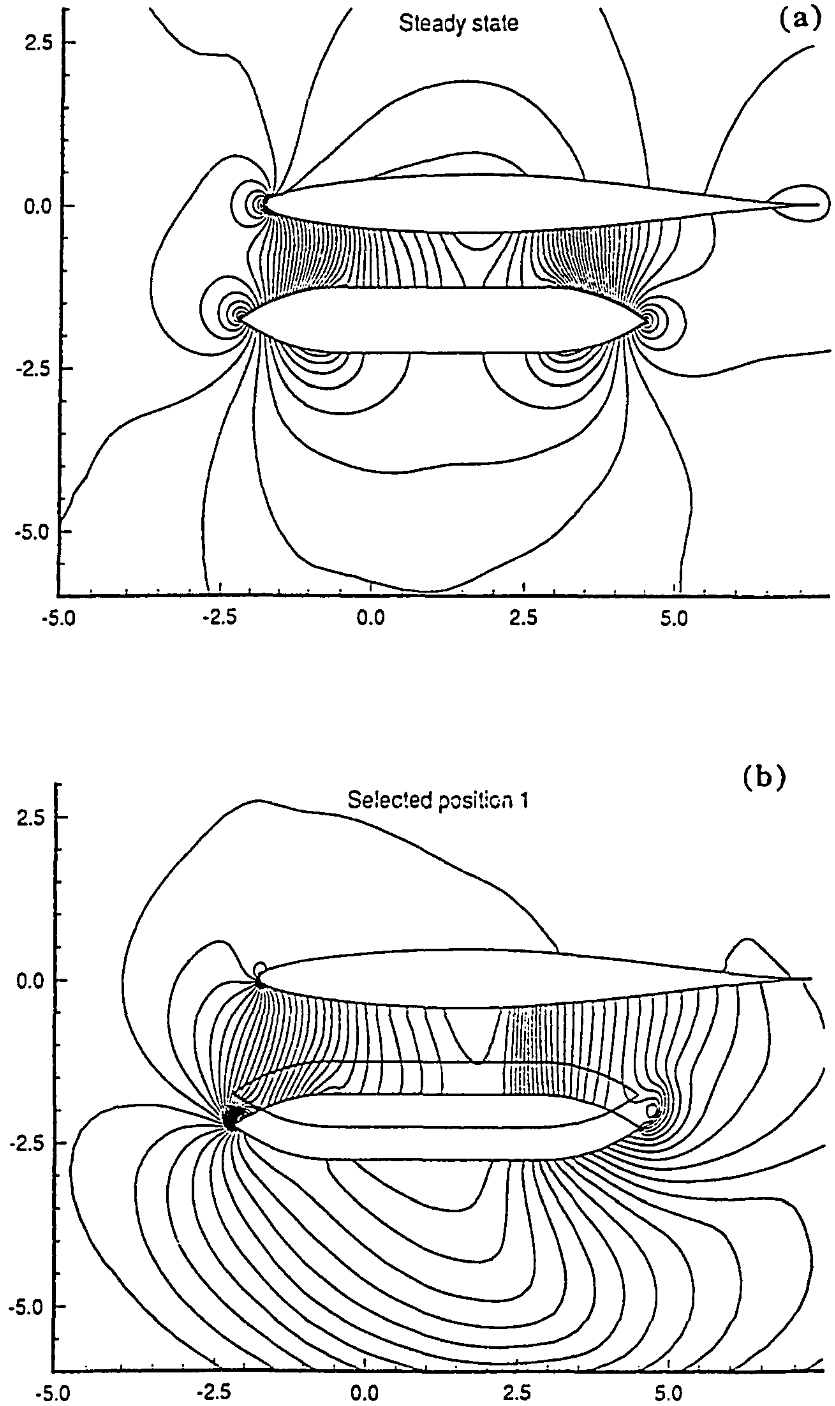


Fig. 6.3 Off-surface pressure contours for AS configuration (a) steady state (b) position 1, (c) position 2, (d) position 3.
($M_\infty = 0.30$, $\alpha_0 = 0^\circ$)

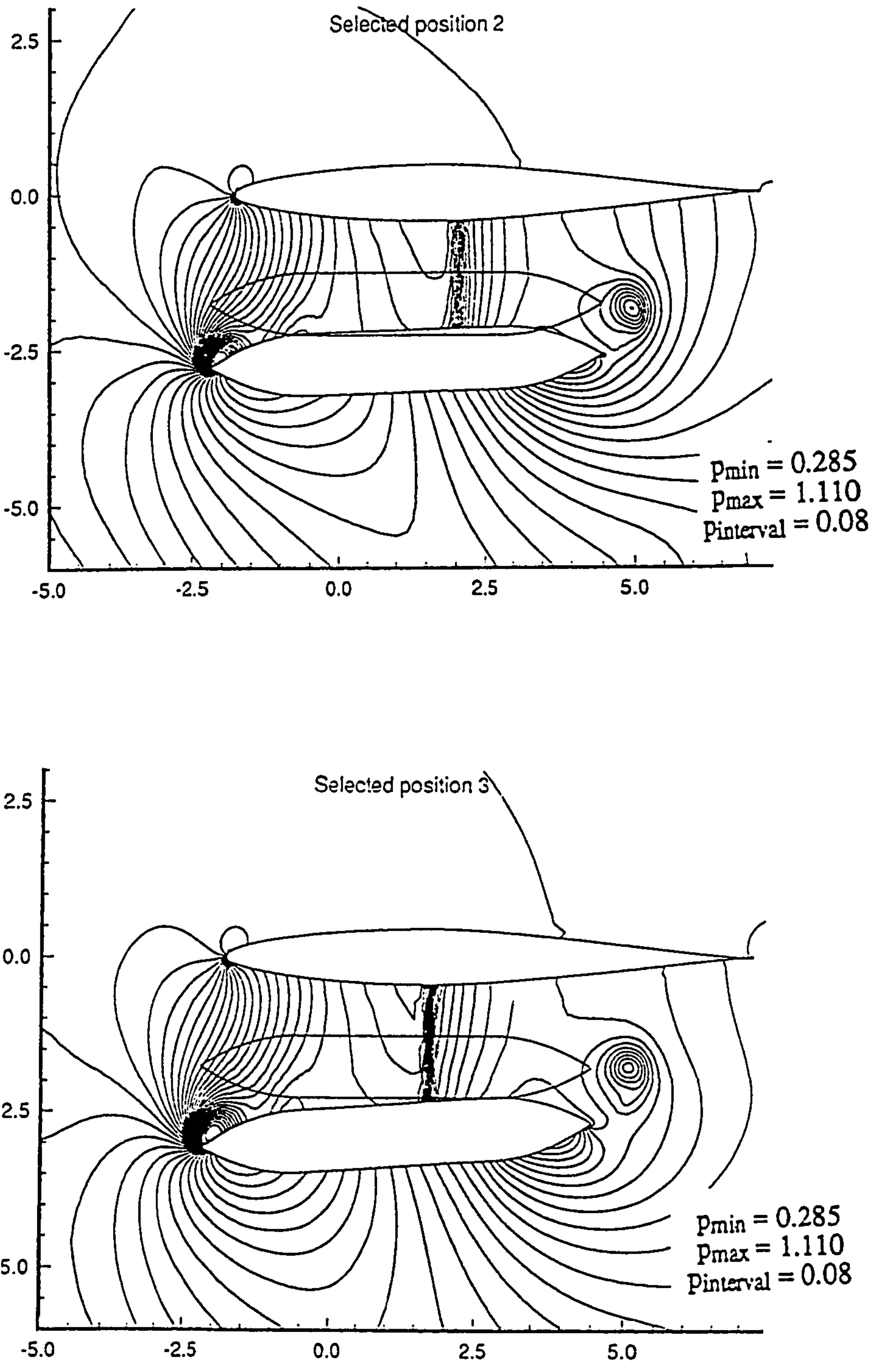
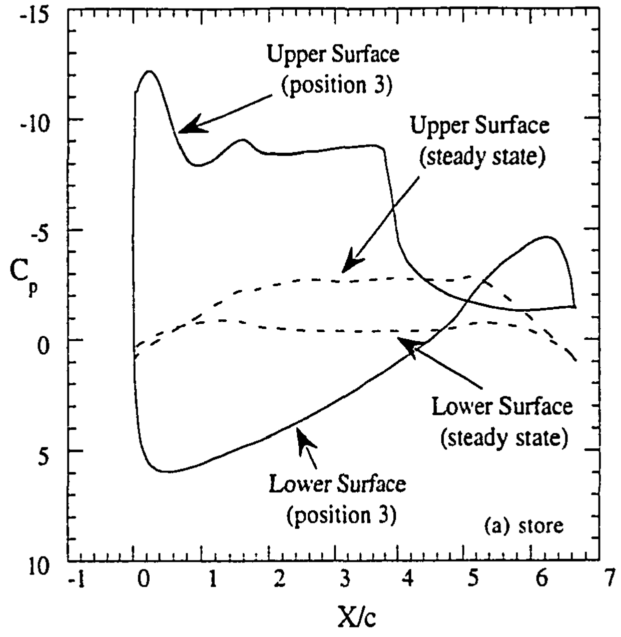


Fig. 6.3 Concluded.



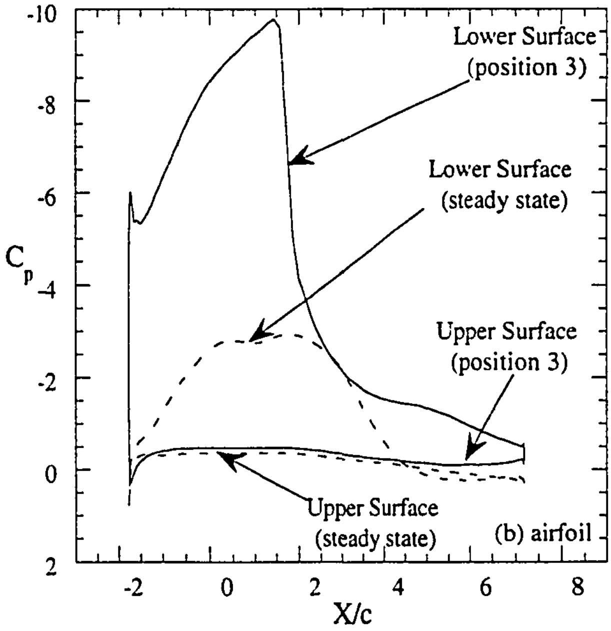
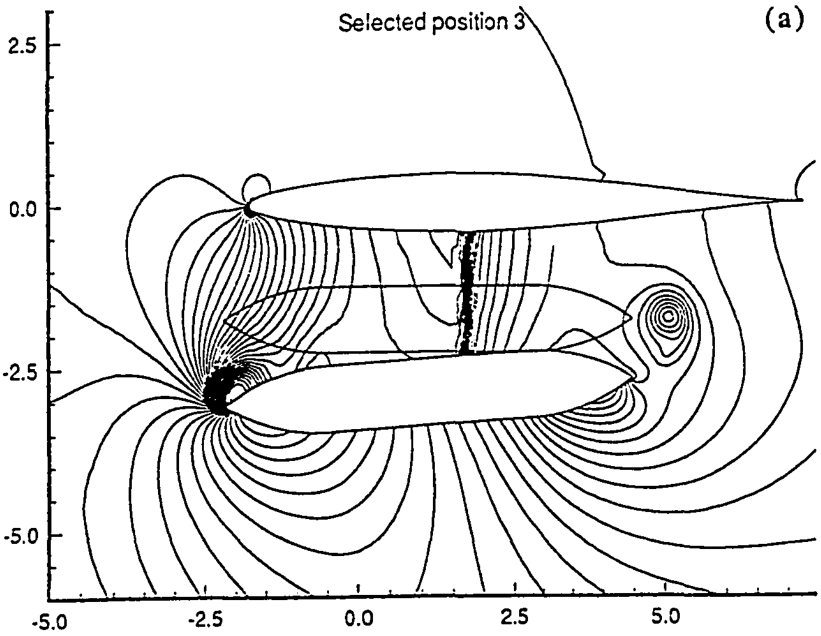


Fig. 6.4 Pressure coefficient distribution (a) store (b) airfoil



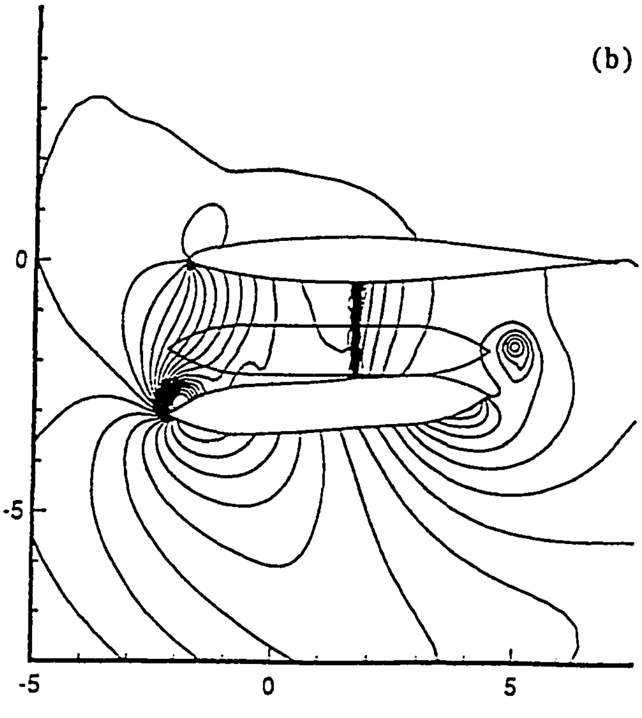


Fig. 6.5 Comparison of off-surface pressure contours (a) van Leer, (b) Roe

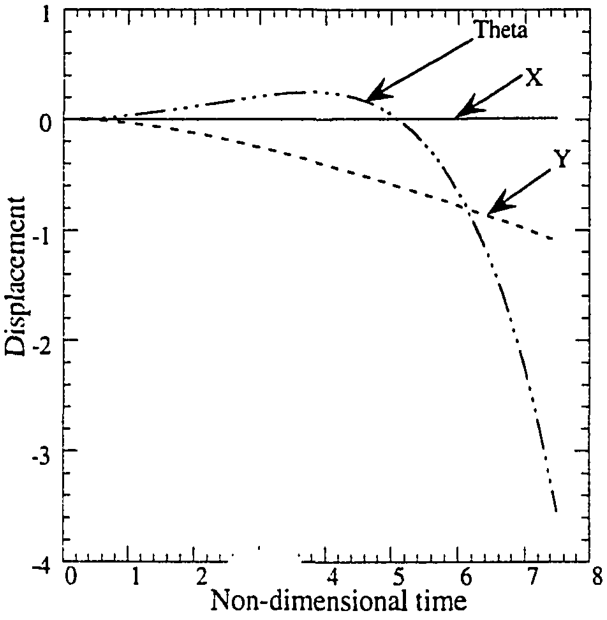


Fig. 6.6 Trajectory of store (3-DOF)

6.4 Three Dimensional Store Separation

Finally, the present method was demonstrated using a 3D wing-and-store configuration which was derived by simplifying the wing/pylon/fin/store configuration given in [57, 58]. It consisted of a clipped delta wing with a 45 deg leading-edge sweep and NACA-64A010 airfoil sections, and directly below this wing, an ogive-cylinder-ogive store. Both the wing and the store were at 0 deg yaw and angle of attack. The oncoming freestream Mach number was 0.95. A computational domain stretching 60d*16.5d*60d (d denotes store diameter) in the three directions, was discretized by a relatively coarse unstructured mesh, with 115,864 cells and 21,515 nodes (Fig. 6.7) due to scarce computational resources. The adaptation window placed around the store included less than 30% of the total cells. For the rigid-body dynamics, the store's mass, weight, and the non-zero elements of its moment of inertia tensor are assumed to take the values in Table 2. Also given in this table is the impulsive ejection force, which was applied until 0.1d drop was achieved. The three dimensional unsteady flow simulation is computationally very expensive and moreover, the time step restriction for an explicit scheme further increases the computational cost. Hence, to reduce the computational cost, the store was imparted a strong ejection force to make the store drop faster (exaggerated by 40 times). The objective of the present case was to demonstrate that the present dynamic unstructured methodology can handle moving boundary problems engaged in relative motion.

First, the steady-state solution was obtained for the carriage position using local time steps. The steady-state offsurface pressure contours at the mid-span plane are shown in Fig. 6.8. The freestream flow impinges on the nose of the store and the leading edge of the wing and subsequently expands to supersonic speeds. Near the trailing edges of the wing and store, shock waves are formed. The shock on the upper surface of the store reflects from the lower surface of the wing and merges with the shock emanating from the lower surface of the wing. The normalized pressure contours for the wing and store surfaces are presented in Fig. 6.9. The computations were continued using time-accurate (global) steps of 0.001 normalized time units ($t = \tilde{t} \cdot a_\infty / d$), with corresponding maximum Courant number of about 3.

Fig. 6.10 shows the offsurface pressure contours at midspan plane for the three selected positions (0.2d, 0.4d, 0.65d). The computations were performed by using the flux difference splitting of Roe. Fig. 6.10a shows the pressure contours at position 1, which is after 1.3t of separation (equivalent to 0.2d store drop). It can be clearly seen from the figure that a compression region below the surface of the store is beginning to form. The upper surface shock of the store impinges on the lower surface of the wing, then reflects down and coalesces with the shock emanating from the lower surface of the wing. Fig. 6.10b displays the same phenomena after the store has dropped 0.4d, however the compression region appears to be more pronounced and the suction pressure near the store trailing edge has strengthened owing to the pitching down motion of the store. The unsteady flow after 2t of separation (store dropped 0.65d) is depicted via its instantaneous pressure contours Fig. 6.10c. As compared to the carriage position Fig. 6.8, there were hardly any differences on the upper surface flow (hence upper surface not shown). However, the dynamic, mutual interference manifested itself in a time-varying footprint of the store on the wing's lower surface. This transient effects were more accentuated on the store, as evidenced by the pressure coefficient distributions on the store surface Fig. 6.11. The pressure difference between the upper and lower surfaces at 2t was much smaller than it was at the carriage position. This was attributed to: first, elevated lower surface pressures due to pitching down of the store; and secondly, the widening gap between the store and wing diminished the interference effect.

The computed 6-DOF trajectory for the store's center of gravity is shown in Fig. 6.12. The translational motion was mainly in the downward direction, but small displacements were also observed aft and toward the wing root. The store's nose pitched down with a gradual yawing toward the wing outboard. The sideslip and rolling were not quite expected for this axisymmetric store. However, apparently due to the wing-tip effect and the impulsive ejection, some non-symmetry was introduced into the flowfield. The unsteady load (pressure integration only) histories resulting from the store motion are shown in Fig. 6.13. The normal force changed direction abruptly as the ejection force was lifted, and kept on increasing. The magnitudes and the temporal changes of the axial and side forces were small. The moments also displayed relatively very small temporal changes, but their magnitudes were appreciable and quite disparate. A finite value of the rolling moment has been observed, which may be largely due to an error in integrating the pressures on the store. The coarseness of a surface grid can easily result in the numerical directions of the pressures being skewed from the center-line. Finally, the present dynamic simulation for 2t and 0.65d required 8.3 CPU hours (including 1.64 h for steady-state solution) and 104 Megabytes of memory on a Cray Y-MP computer. The unit processing time for the method was 17 and 103 ms per time step per cell for steady and moving-boundary computations, respectively.

(d_f =20 d =0.508 m, ρ_∞ =0.776 kg/m^3, a_∞ =322.3 m/s)

F_{eject}*	Weight*	m**	I_{roll}***	I_{pitch}***	I_{yaw}***
102.4	0.43	8,016.2	1,032.8	18,591.9	18,591.9

normalized by: $^{*}\rho_\infty a_\infty^2 d_f^2$, **, $\rho_\infty d_f^2$ *** $\rho_\infty d_f^5$

Table 6.1 Store parameters for 3D store separation

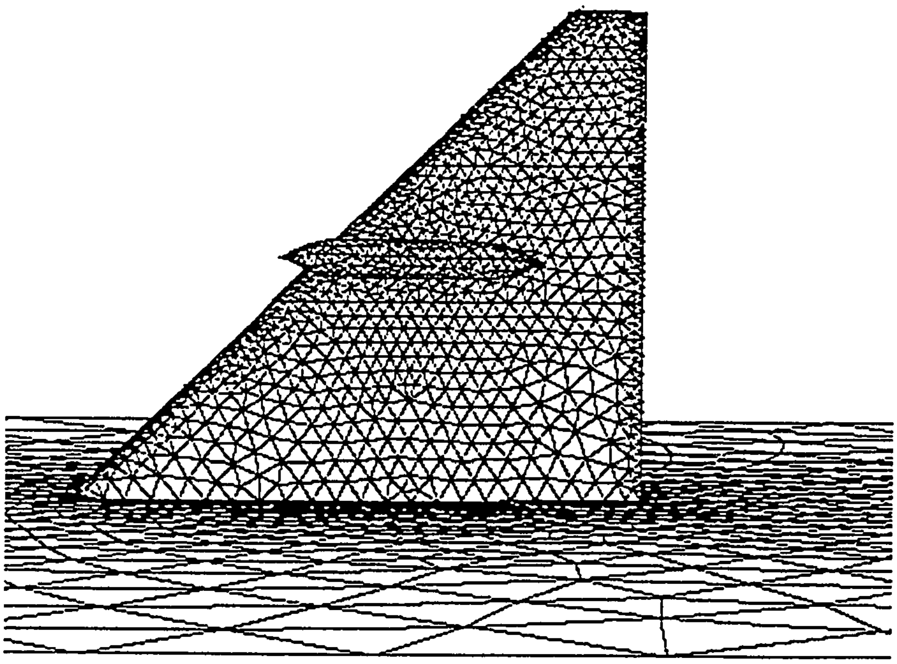


Fig. 6.7 Surface grid on wing-store (WS) configuration

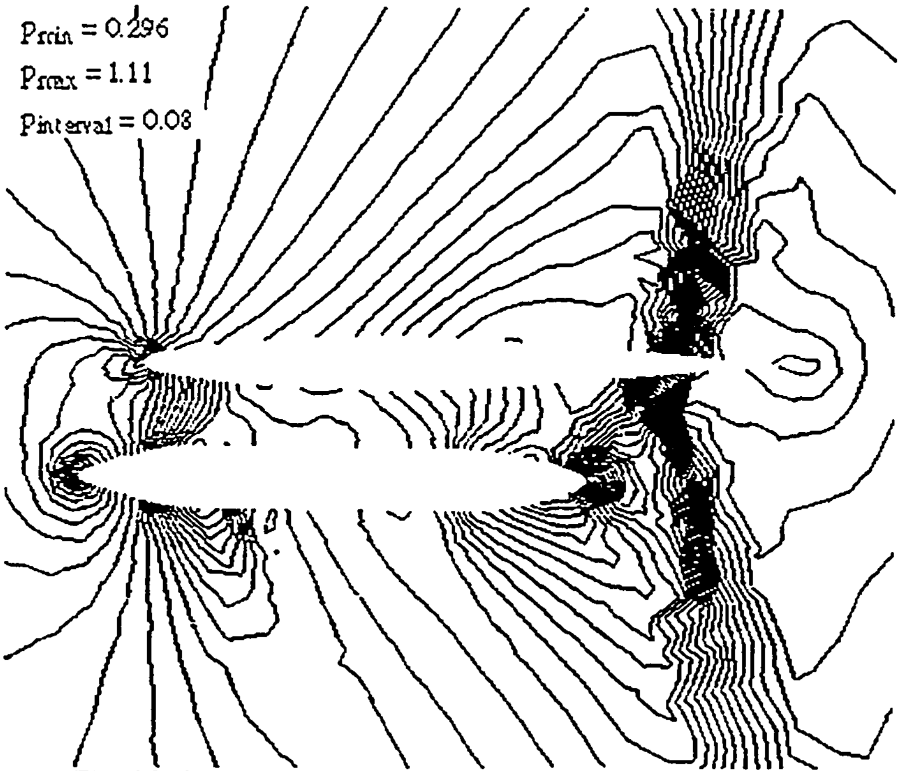


Fig. 6.8 Off-surface pressure contours at steady state (WS)

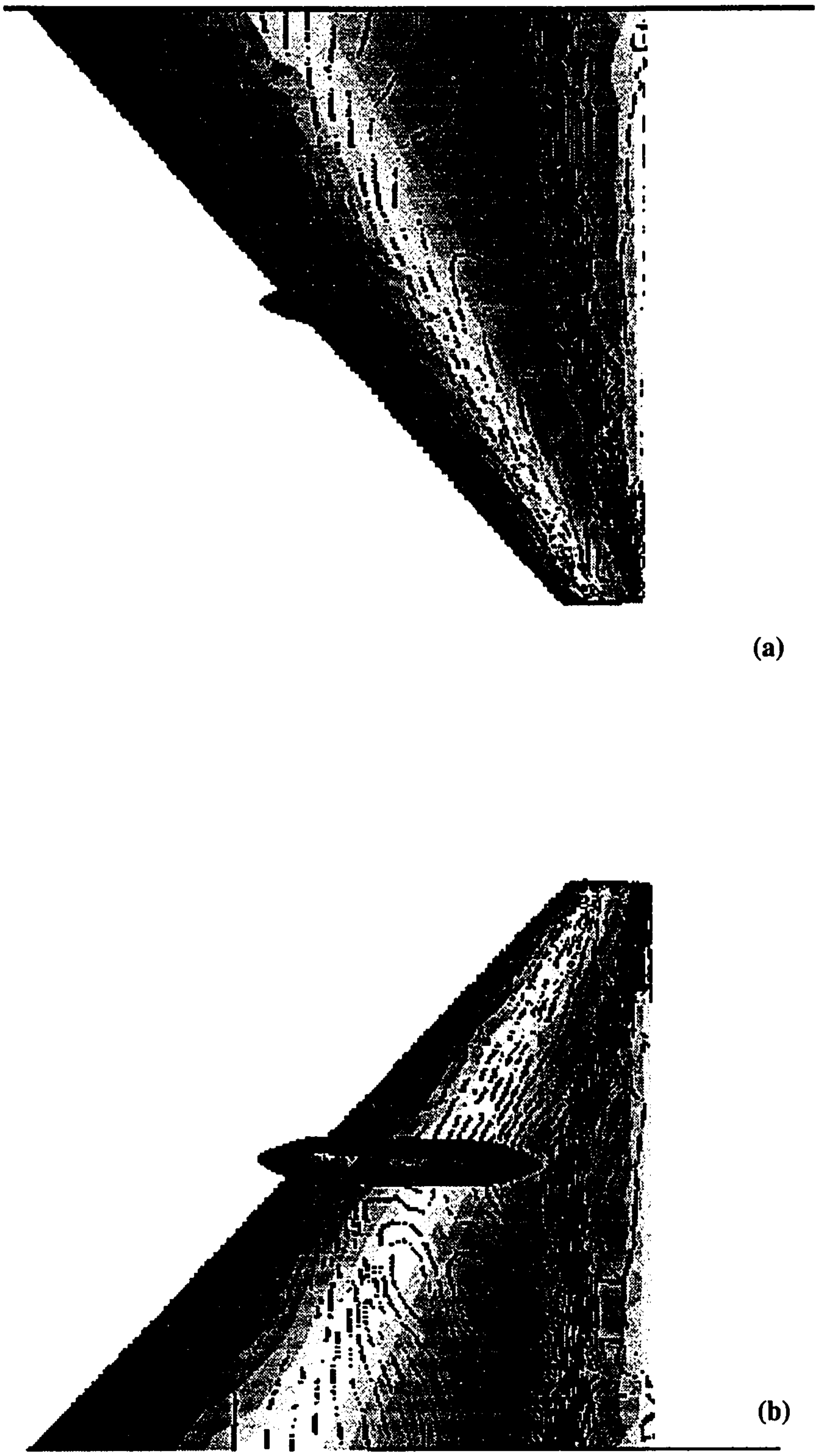


Fig. 6.9 Steady state normalized pressure contours for (WS)
(a) wing upper surface (b) wing lower surface

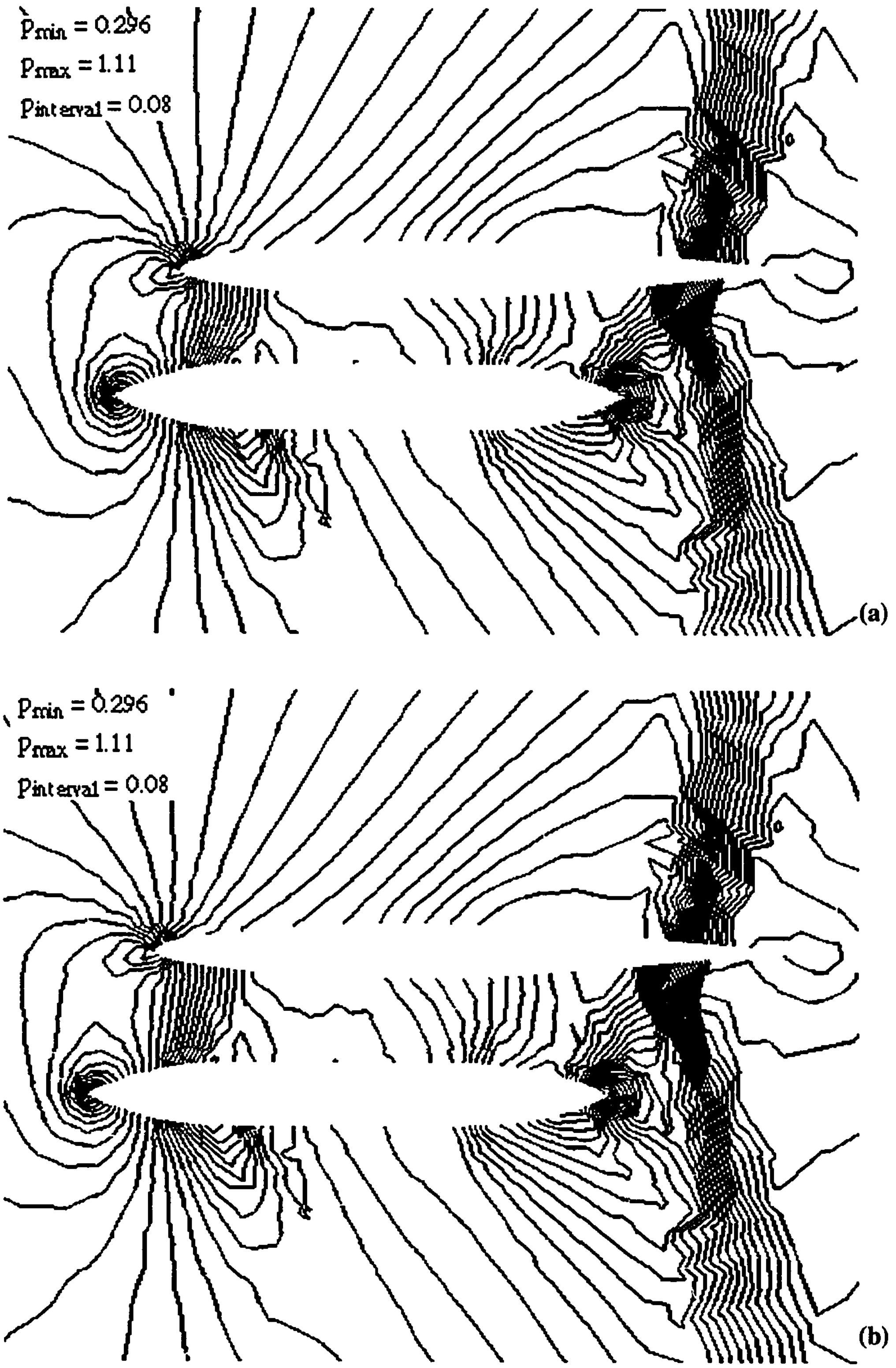


Fig. 6.10 Off-surface pressure contours (WS) (a) 0.20d (b) 0.40d (c) 0.65d store drop.

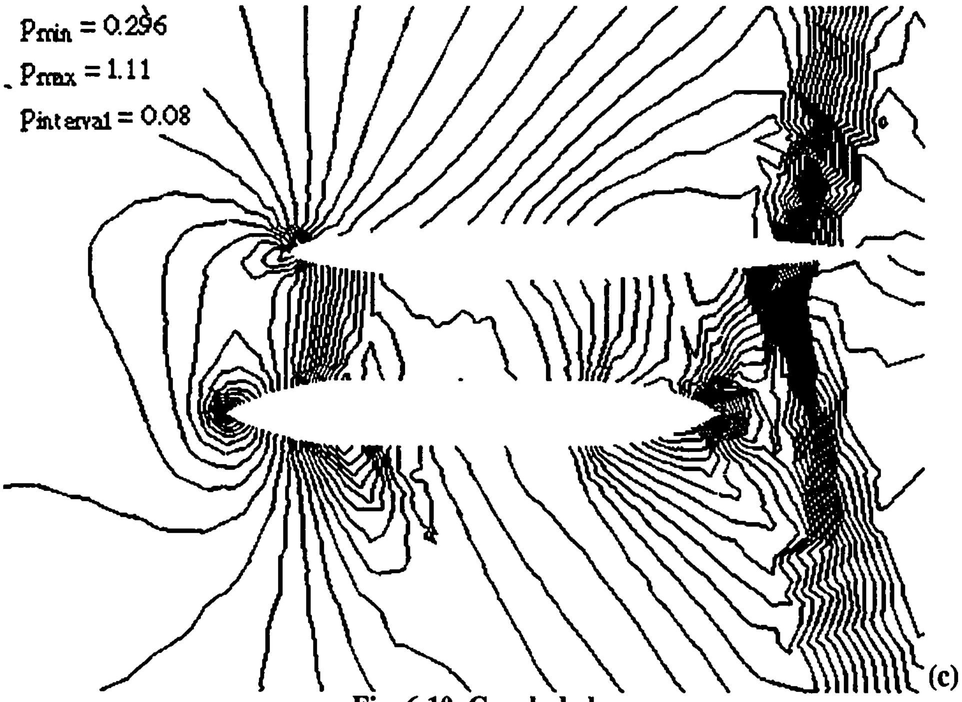


Fig. 6.10 Concluded

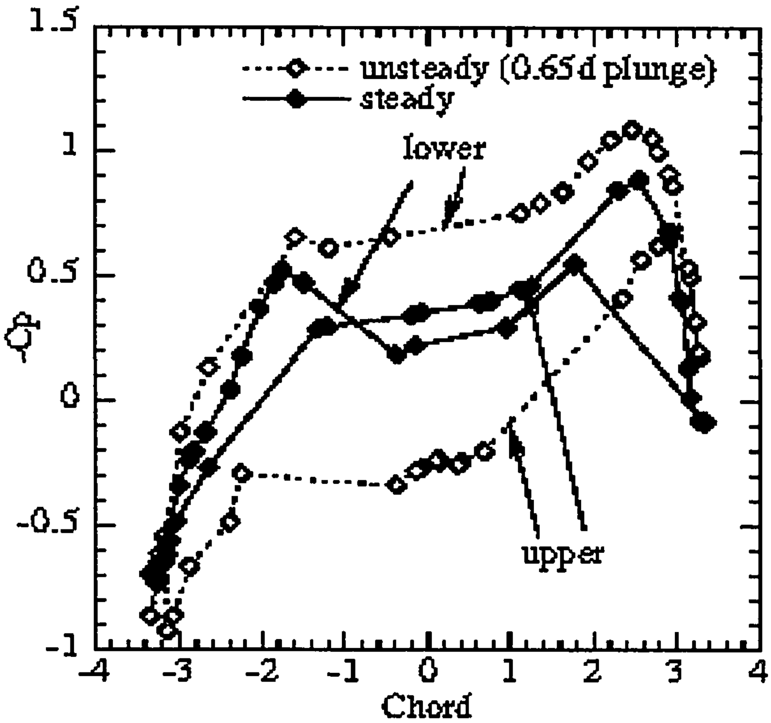


Fig. 6.11 Pressure coefficient distribution on store (steady state vs. unsteady after 2t separation.)

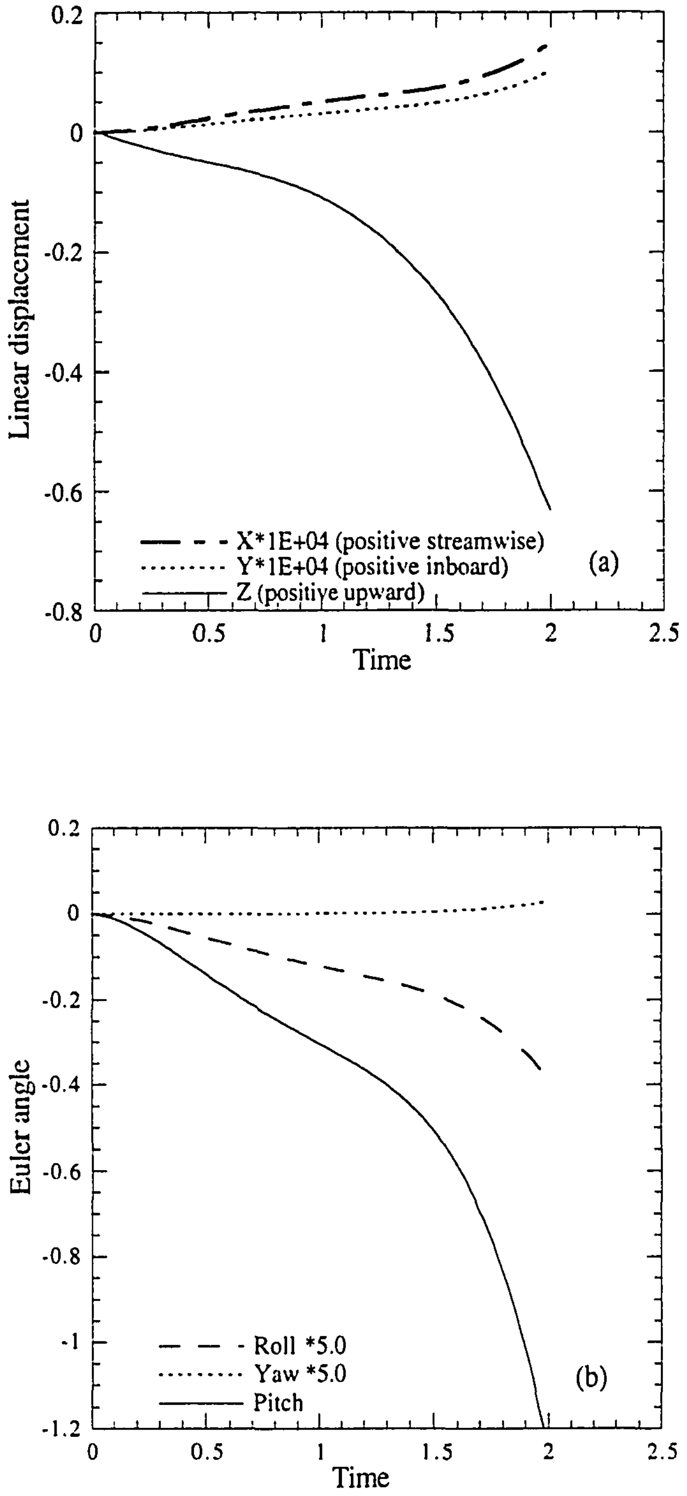


Fig. 6.12 Trajectory of store (center of gravity) separating from a delta wing (a) Displacements (b) Euler angles

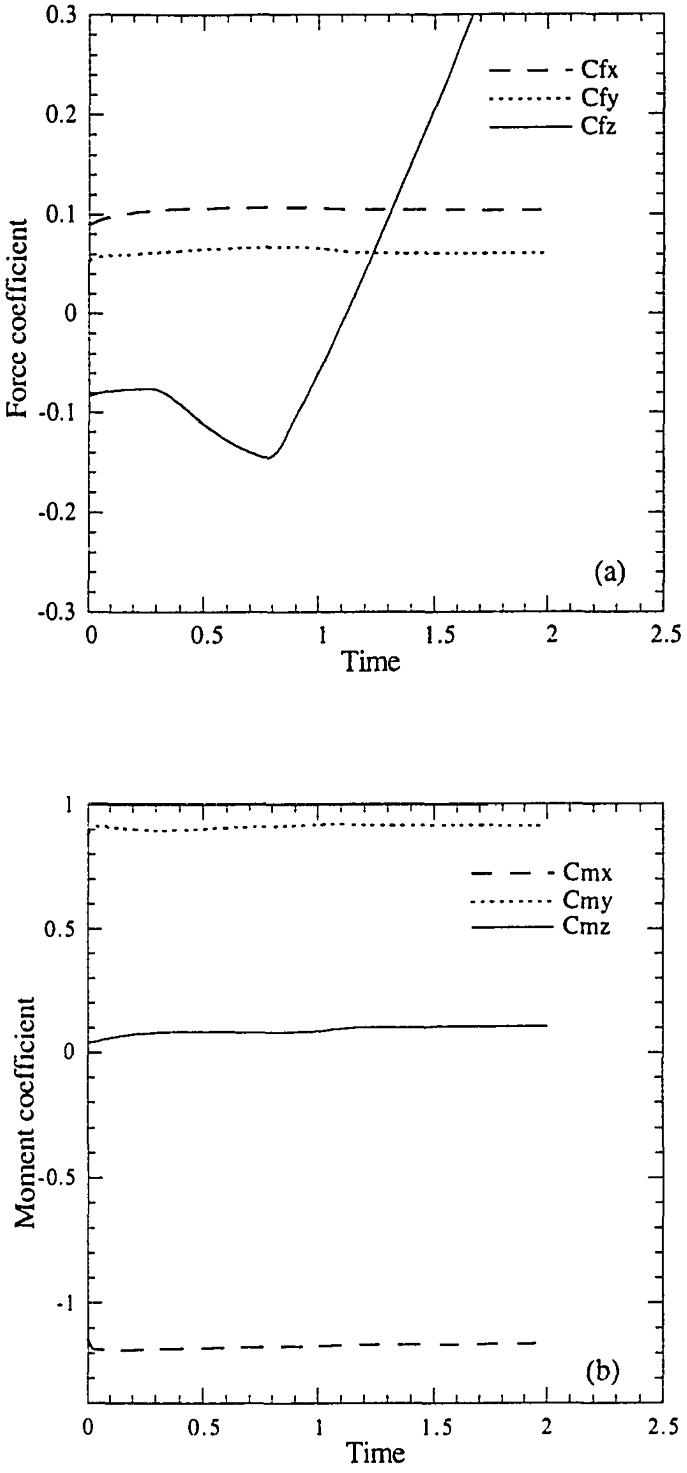


Fig. 6.13 Unsteady loads on store separating from a delta wing: a) Force coefficients at, and b) moment coefficients about store center of gravity.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 On Adaptation and Trajectory Validation

An efficient unstructured grid adaptation procedure has been developed to allow the grid to move with a moving boundary and to limit the computational costs. The grid adaptation was performed within windows put around the moving boundaries. This approach is more efficient than, for example, an adaptation applied to the entire domain or a remeshing method. For instance, without the window, the adaptation procedure took about 1.5 hours on a Cray 2 for one complete cycle of motion for the sinusoidally oscillating airfoil, whereas it took 0.6 hrs on a Cray 2 while restricting the adaptation to points inside the window. This was because only 30 percent of the total nodes was being adapted inside the window as opposed to performing the adaptation for the whole domain. For problems in which a body has small or no translational motion, creation of the window takes place only once. Otherwise, the window would be constructed at several instants during the body's motion. However, for very large amplitude motions, the window adaptation would require highly dense meshes for the major portion of the domain, otherwise, a partial remeshing procedure would be necessary.

The grid adaptation as reported earlier, was based on the spring analogy where each edge of the cell was represented by a tension spring. For the present study, spring stiffness was assumed inversely proportional to the length of the edge. Also, spring stiffness inversely proportional to normalized length was investigated and applied to the three-dimensional store separation problem.

The three-dimensional adaptation method was first tested on a coarse mesh around an ONERA M6 wing by prescribing a large sinusoidal motion to the wing and subsequently inspecting the grid quality.

One of the major objectives of the present work was to develop a 6-DOF trajectory algorithm in order to solve for moving boundary problems where the trajectory of the moving body is governed by aerodynamic forces and moments. The experimentally determined force and moment fields were used to compute the 6-DOF trajectory of a store separating from a wing. The rotational and translational equations of motion (Eqs. (3.41) and (3.42)) along with the Euler rate equations (Eq. (2.15)) were used to compute the six-degrees-of-freedom trajectory of the store. The results compared well with the trajectory measured in the wind-tunnel tests. Some discrepancies in the results, especially in the pitch component of the rotation, can be attributed the inconsistent ejector characteristics used in the trajectory code. However, the three translational and two rotational displacements matched very well with the experimental data.

After having accomplished the trajectory validation, the trajectory code was successfully coupled with the flow solver to solve for aerodynamically determined moving boundary problems.

7.2 On Unsteady Flow Solver Validation

The two-dimensional unsteady flow solver and the dynamic mesh algorithm was validated by computing the flow past a sinusoidally oscillating airfoil for which experimental data was available. The computed unsteady pressure coefficients compared well with experimental results. Comparison of the computed results with that of the experimental results was observed to be better for lower angle-of-attack instants. The slight discrepancies in the results, mostly at higher angles-of-attack, can be attributed to the neglect of viscous effects, since at higher angles-of-attack viscous effects become important and should be accounted for.

A grid independence study was also performed by simulating the unsteady flowfield using a fine grid and a finer grid, which almost doubled the number of cells. The matching of the solutions from both of the grids confirmed the adequacy of the fine cell grid. Also, a comparative study between the present dynamic unstructured methodology and an independently developed structured methodology using dynamic overlapped grids [4, 5, 53] was performed. The results from both methodologies matched very well.

After having validated the methodology in two-dimensions, the next step was to extend this methodology to three-dimensions. The three-dimensional flow solver and adaptation scheme were validated by simulating the transonic, unsteady flow around a rigid rectangular wing undergoing a forced, periodic pitching motion for which experimental results existed. The computed real and imaginary pressure coefficients compared fairly well with the experimental data. The minor discrepancies in results were due to the neglected viscous terms and the use of a relatively coarse mesh. However, the results from the Euler computations were comparable with the Euler computations done with structured grid approach [53].

The present dynamic unstructured methodology was successfully validated for simulating three-dimensional unsteady flowfield and proves to be a viable alternative to dynamic-structured-overlapped methods [4, 5]. Among the observations made in comparison with the implicit, structured, domain-decomposition methods [9, 10, 48], were the following: 1) second-order temporal accuracy not compromised by time-linearization, or approximate factorization, or diagonalization, or interpolations, 2) less number of cells needed in the grid (about one-fifth), 3) this advantage off-set by the smaller time-step (about 20 times) requirement due to the present explicit time marching.

7.3 On Methodology Demonstrations

The overall methodology was demonstrated through a two-dimensional example: the carriage, separation, and the free-fall of a store from a wing section (airfoil). The motion and its trajectory were entirely determined by the force of gravity and the instantaneous aerodynamic coefficients provided by the unsteady flowfield computations. Although no experimental results are available for this case, the flow solution looks physically reasonable. A comparative study for this case had been previously done [59] by performing computations using the present methodology and the dynamic overlapped structured grid approach, and almost identical results had been obtained. Thus this study suggests that the present dynamic, unstructured method is a viable alternative to the dynamic, overlapped-structured-grids approaches [4, 5, 9, 15, 21].

To further demonstrate the applicability of this method, a NACA 0012 airfoil was given a three-DOF motion immersed in a transonic freestream flow. This case was chosen to test the adaptation scheme and also the flow solver for complex and large amplitude motions. This case demonstrated the robustness of the adaptive window procedure for large displacements as well as the flow solver to capture strong moving shock waves.

Among the noteworthy observations made from the present investigation were the following:

1) A dynamic unstructured method has been developed for prescribed and aerodynamically determined relative moving boundary problems. This was accomplished by developing and testing a six-degree-of-freedom trajectory algorithm, an efficient

dynamic and adaptive mesh method for small or large amplitude motions, and a time accurate, second-order method for unsteady flow equations.

2) The methodology has shown to be accurate, automated, easy for dynamic gridding, and relatively efficient for the required man-hours. It would take in the order of one to two weeks time to perform an unsteady flow simulation beginning with the case set-up.

3) Determining the trajectory of a free-falling object aerodynamically requires a multidisciplinary analysis, using not only adequately accurate but also computationally efficient algorithms.

4) As a cost saving measure, such simulations may be restricted to the duration of the mutual interference effects between the objects.

5) It has shown that neither the flow physics nor the unsteady aerodynamics need to be compromised since unsteady flow simulations for relative moving boundary problems (level 4) is possible with the current computational costs.

6) It has been demonstrated that the present methodology can capture the time dependent aerodynamic interference between relative moving bodies or components.

7) The computational efficiency (by further optimizing the coding and/or implicit time-marching) was deemed as the last issue to be investigated prior to proposing the method for practical and realistic applications.

7.4 Recommendations for Future Work

The dynamic unstructured methodology has proven to be a very promising approach. However, there are avenues available to extend and strengthen its capabilities. Some of the options may be listed as follows:

1. Unsteady problems require large amounts of computer time, and the use of the explicit scheme amplifies the computational time requirement. Hence, an implicit scheme or a subiterative scheme needs to be incorporated to reduce the required computational time. This should bring the times down to levels comparable with implicit dynamic, overlapped approaches [9, 10].

2. Another alternative and probably the best choice to model the unsteady moving boundary problems is the implementation of a massively parallel algorithm along with an explicit scheme. This would model the physics of the problem better owing to the small time step, thereby enhancing the temporal accuracy, and also the use of parallel algorithm which should make the computation more efficient and viable.

3. An adaptive remeshing routine should be implemented. For moving boundary problems, especially for bodies in relative motion, grid distortions are likely to occur and in order to counter that, a remeshing methodology (e.g. H-refinement) needs to be incorporated.

4. Flow adaptation, not only to the boundary motion as it is done here, but also to the solution as it evolves, should be added to capture the flow physics accurately. For instance, for moving shock problems, a flow adaptive scheme can be useful to capture the moving shock with better resolution.

5. Viscous effects and turbulence modeling may be incorporated for the type of problems where these effects are non-negligible, contingent upon vast computer resources improvements. Although some improvements are needed for superior viscous grids [60], presently, viscous grids are available and the implementation of unsteady Navier-Stokes equations, with the appropriate turbulence modeling, could extend the capability and the applicability of the present method.

6. The approach has the potential to handle flexible bodies, i.e., aereoelastic problems.

The method needs to be extended for handling complex aeroelastic problems.

REFERENCES

1. Singh, K. P., Newman, J. C., and Baysal, O., " Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," *AIAA Journal*, Vol. 33, No. 4, pp. 641-649.

2. Singh, K. P., and Baysal, O., " 3-D Unstructured Method for Flows Past Bodies in 6-DOF Relative Motion," Proceedings of Sixth International Symposium of Computational Fluid Dynamics, Sept. 1995.

3. Batina, J. T., " Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," AIAA-89-0015, Jan, 1989.

4. Baysal, O., and Yen, G. W. , " Kinematic Domain Decomposition to Simulate Flows Past Moving Bodies," AIAA-91-0725, Jan. 1991.

5. Yen, G. W., and Baysal, O., " Computing Unsteady High Speed Flows Past an Oscillating Cylinder Near a Vertical Wall," *Journal of Spacecraft and Rockets*, Vol. 31, No. 4, July/August 1994, pp. 630-635.

6. Kennon, S. R., Meyering, J. M., Berry, C. W., and Oden, J. T., " Geometry Based Delaunay Tetrahedralization and Mesh Movement Strategies for Multibody CFD," AIAA-92-4575 CP.

7. Noack, R. W., and Bishop, D. G., " A 3-D Delaunay Grid Generator and Flow Solver for Bodies in Relative Motion," AIAA-93-3349 CP.

8. Pirzadeh, S., " Recent Progress in Unstructured Grid Generation," AIAA 30th Aerospace Sciences Meeting, AIAA-92-0445, Jan. 1992.

9. Baker, T. J., " Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," AIAA-87-1124, 1987.

10. Pirzadeh, S., " Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method," AIAA-91-3233, AIAA 9th Applied Aerodynamics Conference, Baltimore, MD, Sept. 1991.

11. Parikh, P., Pirzadeh, S., and Löhner, R., " A Package for 3-D Unstructured Grid generation, Finite Element Flow Solutions, and Flow Visualization," NASA CR-182090, Sept. 1990.

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

13. Yen, G. W., and Baysal, O., " Computing Store Separation and its 6-DOF Trajectory using 3-D Dynamic Domain Decomposition Method," Unsteady Flows 1995, (Eds. Keith, Tsukumato, Baysal, Wei) FED Vol., ASME, New York, NY, Aug. 1995.

14. Yen, G. W., and Baysal, O., " Accuracy of Dynamic Overlapped Grids For Unsteady Flows," Unsteady Flows 1995, (Eds. Keith, Tsukumato, Baysal, Wei) FED Vol., ASME, New York, NY, Aug. 1995.

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

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

17. Tamura, Y. and Fujii, K., " Conservation Law for Moving and Transformed Grids," AIAA-93-3365, July 1993.

18. Donea, J., Giuliani, S., and Halleux, J. P., " An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interactions," *Computer Methods in Applied Mechanics and Engineering*, 1982, pp. 689-723.

19. Liou, Meng-Sing, " An Extended Lagrangian Method," AIAA-93-3305, 1993, pp. 92-105.

20. Löhner, R., "Adaptive Remeshing for Transient Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 75, 1989, pp. 195-214.

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

22. Yen, G. W., " Unsteady Flow Simulations about Moving Boundary Configurations Using Dynamic Domain Decomposition Techniques," Ph. D. Dissertation, Aerospace Engineering Department, Old Dominion University, May 1994.

23. Arabshahi, S., and Whitfield, D. L., " A Multiblock Approach For Solving the Three Dimensional Unsteady Euler Equations about a Wing/Pylon/Store Configuration," AIAA-89-3401, Aug. 1989.

24. Kandil, O., and Chuang, H. A., " Unsteady Navier-Stokes Computations Past Oscillating Delta Wing at High Incidence," *AIAA Journal*, Vol. 28, No. 9, Sept. 1990, pp. 1565-1572.

25. Kandil, O., and Chuang, H. A., " Computation of Vortex-Dominated Flow for a Delta Wing Undergoing Pitching Oscillation," *AIAA Journal*, Vol. 28, No. 9, Sept. 1990, pp. 1589-1595.

26. Löhner, R., Morgan, K., Peraire, J., Vahdati, M., " Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 7, 1987, pp. 1093-1109.

27. Probert, E. J., Hassan, O., Morgan, K., Peraire, J., " The Simulation of Store Separation Using Unstructured Grids and Adaptive Remeshing," Proceedings Royal

Aeronautical Society Conference on Store Carriage, Integration and Release, Bath, England, pp. 28.1-28.8, 1990.

28. Probert, E. J., Hassan, O., and Morgan, K., " An Adaptive Finite Element Method for Transient Compressible Flows with Moving Boundaries," *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 751-765.

29. Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., " Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, 1987, pp. 449-466.

30. Hassan, O., Probert, E. J., Morgan, K., Peraire, J., " Domain Decomposition Combined with Adaptive Remeshing For Problems of Transient Compressible Flow," Proc. 13th International Conference on Numerical Methods in Fluid Dynamics, Rome, 220-224, 1993.

31. Löhner, R. and Baum, J. D., " Three-Dimensional Store Separation Using a Finite Element Solver and Adaptive Remeshing," AIAA-91-0602, Jan. 1991.

32. Batina, J. T., " CFD Methods Development Considerations For Unsteady Aerodynamic Analysis," Workshop on Computational Aeroacoustics, Hampton, Va., April 1992.

33. Batina, J. T., " Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aeroelastic Analysis," AIAA/ASME/ASCE/AHS/ASC, 30th Structures, Structural Dynamics and Materials Conference, 1989.

34. Greenwood, D. T., Principles of Dynamics, Prentice-Hall Inc., Englewood Cliff, NJ, 1965, pp. 40-366.

35. D' Souza, A. F., and Garg, V. K., Advanced Dynamics-Modeling and Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1984, pp. 95-155.

36. Frink, N. T., " Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes," *AIAA Journal*, Vol. 30, No.1, Jan 1992, pp. 70-77.

37. Frink, N. T., " Three Dimensional Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Grids," Ph. D. Dissertation, Department of Aerospace Engineering, VPISU, Sept. 1991.

38. Jameson, A., Schmidt, W., and Turkel, E., " Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Scheme," AIAA-81-1259, June 1981.

39. Anderson, W. K., Thomas, J. L., and van Leer, B., " Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal*, Vol. 24, No. 9, Sept. 1986.

40. Roe, P.L., " Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No.2, 1981, pp. 357-372.

41. van Leer, B., " Flux Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, Springer Verlag, Berlin, 1982.

42 Godunov, S. K., " A Difference Scheme for Numerical Computation of Discontinuous Solutions of Hydrodnamic Equations," *Math Sbornik*, 47, pp. 271-306, 1959.

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

44. Rumsey, C. L., and Anderson, W. K., " Some Numerical and Physical Aspects of Unsteady Navier-Stokes Computations over Airfoils using Dynamic Meshes," AIAA-88-0329, AIAA 26th Aerospace Sciences Meeting, Jan. 1988.

45. Barth, T. J., and Jespersen, D. C., " The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA-89-0366, Jan. 1989.

46. Frink, N. T., " Recent Progress Towards a 3-D Unstructured Navier-Stokes Flow Solver, AIAA 32nd Aerospace Sciences Meeting and Exhibit, AIAA-94-0061, Jan. 1994.

47. Rausch, R. D., Batina, J. T., and Yang, T. Y., "Spatial Adaptation Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computations," NASA TM 104039, March 1991.

48. Chaderjian, N. M., and Guruswamy, G. P., " Transonic Navier-Stokes Computations For an Oscillating Wing Using Zonal Grids," *Journal of Aircraft*, Vol. 29, No.3, May-June 1992, pp. 326-335.

49. Pulliam, T. H., and Chaussee, D. S., " A Diagonal Form of an Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, No.2, 1981, pp. 347-363.

50. Kandil, O., 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.

51. Chakravarthy, S., and Szema, K. Y., " Computational Fluid Dynamics Capability For Internally Carried Store Separation," Rockwell International Science Center, Technical Report SC71039, Thousand Oaks, CA, Nov. 1991.

52. Anderson, W. K. " Grid Generation and Flow Solution Method for Euler Equations on Unstructured Grids," NASA TM-4295, Apr. 1992.

53. Newman, J. C., " Flow Simulations About Steady-Complex and Unsteady Moving Configurations Using Structured-Overlapped and Unstructured Grids," M.S. Dissertation, Aerospace Engineering Department, Old Dominion University, May 1994.

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

55. Landon, R., " NACA 0012 Oscillatory and Transient Pitching," *Compendium of Unsteady Aerodynamic Measurements*, AGARD Report 702, Aug. 1982, pp. 3.3-3.25.

56. Mabey, D. G., Welsh, B. L., and Pyne, C. R., " A Summary of Measurements of Steady and Oscillating Pressures on a Rectangular Wing," *Aeronautical Journal,* Vol. 92, No. 911, Jan. 1988, pp. 10-28.

57. Newman, J. C., III, and Baysal, O., " Transonic Solutions of a Wing/Pylon/Finned Store Using Hybrid Domain Decomposition," AIAA-92-4571, Atmospheric Flight Mechanics Conference, Aug. 1992.

58. Parikh, P., Pirzadeh, S., and Frink, N.T., " Unstructured Grid Solutions to Wing/Pylon/Store Configuration Using VGRID3D/USM3D," AIAA-92-4752, Aug. 1992.

59. Trepanier, J. Y., Reggio, M., Paraschivoiu, M., and Camarero, R., " Unsteady Euler Solutions for Arbitrarily Moving Bodies and Boundaries," *AIAA Journal,* Vol. 31, Vol. 10, 1993, pp. 1869-1876.

60. Pirzadeh, S., " Viscous Unstructured 3-D Grids by Advancing Layers Method, 32nd Aerospace Sciences Meeting and Exhibit, AIAA-94-0417, Jan. 1994.