Spring 1994

Flow Simulations About Steady-Complex and Unsteady Moving Configurations Using Structured-Overlapped and Unstructured Grids

James Charles Newman III

Old Dominion University

Recommended Citation

Newman, James C.. "Flow Simulations About Steady-Complex and Unsteady Moving Configurations Using Structured-Overlapped and Unstructured Grids" (1994). Master of Science (MS), Thesis, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/7tqa-be39
https://digitalcommons.odu.edu/mae_etds/205

FLOW SIMULATIONS ABOUT STEADY-COMPLEX AND UNSTEADY MOVING CONFIGURATIONS USING STRUCTURED-OVERLAPPED AND UNSTRUCTURED GRIDS

by

James Charles Newman, III
B.S., May 1993, Old Dominion University

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

MASTER OF SCIENCE
AEROSPACE ENGINEERING

Old Dominion University
May 1994

Approved by:

O. Baysal (Director)

O. A. Kandil

N. T. Frink

ABSTRACT

FLOW SIMULATIONS ABOUT STEADY-COMPLEX AND UNSTEADY MOVING CONFIGURATIONS USING STRUCTURED-OVERLAPPED AND UNSTRUCTURED GRIDS

by

James C. Newman III
Old Dominion University

Director: Dr. O. Baysal

The limiting factor in simulating flows past realistic configurations of interest has been the discretization of the physical domain on which the governing equations of fluid flow may be solved. In an attempt to circumvent this problem, many Computational Fluid Dynamic (CFD) methodologies that are based on different grid generation and domain decomposition techniques have been developed. However, due to the costs involved and expertise required, very few comparative studies between these methods have been performed. In the present work, the two CFD methodologies which show the most promise for treating complex three-dimensional configurations as well as unsteady moving boundary problems are evaluated. These are namely the structured-overlapped and the unstructured grid schemes. Both methods use a cell centered, finite volume, upwind approach. The structured-overlapped algorithm uses an approximately factored, alternating direction implicit scheme to perform the time integration, whereas, the unstructured algorithm uses an explicit Runge-Kutta method. To examine the accuracy, efficiency, and limitations of each scheme, they are applied to the same steady complex multicomponent configurations and unsteady moving boundary problems. The steady complex cases

consist of computing the subsonic flow about a two-dimensional high-lift multielement airfoil and the transonic flow about a three-dimensional wing/pylon/finned store assembly. The unsteady moving boundary problems are a forced pitching oscillation of an airfoil in a transonic freestream and a two-dimensional, subsonic airfoil/store separation sequence. Accuracy was accessed through the comparison of computed and experimentally measured pressure coefficient data on several of the wing/pylon/finned store assembly's components and at numerous angles-of-attack for the pitching airfoil. From this study, it was found that both the structured-overlapped and the unstructured grid schemes yielded flow solutions of comparable accuracy for these simulations. This study also indicated that, overall, the structured-overlapped scheme was slightly more CPU efficient than the unstructured approach.

Acknowledgments

The author wishes to acknowledge Dr. O. Baysal for providing the funding for this research. I would also like to express my sincere gratitude to my committee members Dr. O. A. Kandil and Dr. N. T. Frink for the technical advice and comments that I received during the writing of this thesis. This work was supported by NASA Langley Research Center under grants NAG-1-1150 and NAG-1-1499.

I would like to thank A. Thomas, Dr. E. Oktay, and Dr. S. Pirzadeh for generating the unstructured grids about the multielement airfoil, wing/pylon/finned store, and airfoil/store configurations, respectively. The moral and technical support of my graduate student colleagues was also appreciated, with special thanks extended to K. P. Singh. I need to also express my appreciation to, and respect for, G. W. Burgreen for serving as my *surrogate* technical advisor, friend, and support system through this ordeal. We have a kindred spirit and I will never forget you.

In life, every person needs inspiration and a role model, mine has been my father. Fathers and sons often have a hard time expressing how they feel toward one another, thus, I would like to take this opportunity to say *I love you and thank you for believing in me.* I would also like to express my love and gratitude to my mother and my parents-in-law for being supportive and nurturing.

Finally, my deepest appreciation and love are expressed to my wife Amy. Her support, love, understanding, and sacrifice has made all of this possible. I do not know where I would be today without her.

TABLE OF CONTENTS

Page

List of Figures

Nomenclature

a	speed of sound
A_{ij}	cell face area
$\mathrm{C_N}$	normal force coefficient
$\mathrm{C_p}$	coefficient of pressure
CPU	central processing unit
e	total energy
FDS	flux difference splitting
FVS	flux vector splitting
$\vec{F}, \vec{G}, \vec{H}$	flux vectors in Cartesian coordinates
$\hat{F}, \hat{G}, \hat{H}$	flux vectors in curvilinear coordinates
J	transformation Jacobian
k_{ij}	spring stiffness
$\hat{n}$	outward facing unit normal
p	pressure parameter to control spring stiffness
Q	conserved variables
R_i	residual
$R^{\pm}$	Riemann invariants
$\vec{R}$	position vector from fixed to moving frame
$\Delta\vec{r}$	radius vector
Δr	magnitude of radius vector
u, v, w	Cartesian velocities

U, V, W	contravariant velocities
$\vec{V}$	velocity vector
V_i	volume
$\vec{W}$	local cell face velocity
WPFS	wing/pylon/finned store configuration
x, y, z	coordinates of the nodes
$\Delta x, \Delta y, \Delta z$	displacement of the nodes
α, β, γ	coordinates of the fringe point
$\Delta_\pm$	forward and backward differences
$\tilde{\Delta}_\pm$	min-mod flux limiter
γ	ratio of specific heats
κ	parameter defining scheme type
$\vec{\omega}$	angular velocity vector of moving frame
ω	relaxation factor
$\vec{\rho}$	position vector from moving frame to point of concern
ρ	density
ξ, η, ζ	generalized curvilinear coordinates

Chapter 1
Introduction

1.1 Motivation

The primary justification for the implementation of Computational Fluid Dynamics (CFD) to aid in the design of a store/airframe, or even a complete aircraft configuration, is the reduced cost and time incurred to generate and test a particular model. Above is the quintessence of CFD in that it is possible to compute flows about these complex configurations for a range of flight conditions, with only the modification of a few parameters. Hence, large numerical databases of pressure, force and moment predictions for existing, as well as hypothetical, computational models may be acquired quickly and cost effectively to assist the designer. This is in accordance with the needs of industry which are seeking methods that have quick *turn around* times, methods that utilize advanced CFD technology, and methods capable of analyzing *realistic* configurations.

The first step in any flow calculation is the discretization of the physical domain. Unfortunately, this step has been a limiting factor in computing flows past complex multicomponent configurations. To circumvent the problem, many techniques for handling the computational domain have been devised and are currently being used. Some of these methods include the domain decomposition techniques (such as multiblock, zonally patched, and overlapped grids), unstructured grid approaches, and various combinations of these methods. CFD is relatively inexpensive as compared with the fabrication and the testing of experimental wind tunnel models, however, there are costs involved and expertise is needed to calculate flow fields over such complex geometries. Due to this, very few comparative studies between these methods have been undertaken. Thus, it is evident that a means of assessing the performance and limitations of each method is well overdue.

1.2 Literature Survey

With the development of CFD methods and high-speed super computers with large memory capabilities, more realistic commercial and military aircraft are being analyzed. Nearly all modern high speed and military aircraft are subject to carry some form of wing-mounted element. Typical examples of such components are nacelles, stores, or missiles. Flow simulations in the transonic and supersonic regimes, about wing/nacelle configurations, have been performed by Nishida and Bencze [1], Connell and Holmes [2], and Fouladi [3], among others. Within the realm of weapons carriage, much research has been conducted on both externally and internally carried stores. Representative work on externally captive stores may be found in Stanniland et al. [4], Arabshahi and Whitfield [5], Lijewski [6,7], Baysal et al. [7], Yen and Baysal [9], Meakin [10], Parikh et al. [11]. Newman and Baysal [12], and Noack and Bishop [13]; whereas internal store calculations have been conducted by Baysal et al. [14], Fouladi and Baysal [15], and Lohner [16]. Simulations about complete aircraft, which include the wing, fuselage, and wing mounted elements, can be found in Refs. 17-21.

In all the above cited references, some form of structured domain decomposition technique or unstructured grid methodology was used to discretize the flow field around these complex configurations. To follow is a brief summary of these techniques. It should be noted that this review is by no means comprehensive, and the interested reader may use this as a starting point.

1.2.1 Domain Decomposition Methods

The nemesis in performing a calculation over complex, or realistic, configurations has been the construction of an adequate grid on which the governing equations may be solved. For such geometries, the generation of a single structured grid is often difficult, if not impossible. Furthermore, due to the complexity of the flow usually associated with these configurations, control over grid point distribution, skewness, and clustering are imperative. Two methods which permit greater control over these aspects and also ease the overall grid generation effort are the multiblock and grid overlapping approaches.

The multiblock, or block-structured, approach [5,6,7,22] divides the computational domain into a number of blocks. Topological differences may exist between adjacent blocks, however, grid lines must be contiguous at interface boundaries; that is, grid point continuity must be maintained at block boundaries. From this condition, neighboring blocks may communicate through an extraction-injection procedure where information is extracted from one block and injected directly into another, without the need for interpolation. Hence, no modifications are needed for the evaluation of fluxes at block-block interfaces. Other features of multiblock schemes include increased control over grid point distribution, and reduced memory requirements since only one block at a time resides in computer core memory during the solution procedure. Two of the existing grid generation software packages that may be used to construct multiblocked grids are EAGLE [23] and GRIDGEN [24].

In developing a blocking strategy for the entire computational domain that obeys the point continuity restriction at boundaries, compromises must sometimes be made that deteriorate the grid quality. A block-structured system about a multicomponent configuration may be created where the grids conform to the surfaces of certain elements in the configuration, but other components must be discretized with topologically incompatible grids. For example, Lijewski [7] developed an extremely creative multiblock structure about a wing/pylon/unfinned store. This system has the desired C-O grid conforming to the pylon, unfinned body, and sting. The wing grid, however, is forced to be an H-H type, which may cause inaccuracies in the leading edge and tip regions. In general, C or O-type topologies have been found to produce more accurate solutions about wings [25].

To eliminate such deficiencies, another domain decomposition technique, known as grid overlapping, is commonly employed. Overlapping methods, for example, those derived from the Chimera scheme [8-10,12,14,18,26-27] place no restriction on interface boundaries, but does require that a sufficient region of overlapping exists between grids. The main advantage of this method is that an optimum body fitted grid may be independently generated for each component in a configuration. Such a technique immediately simplifies the grid generation required for complex multicomponent configurations. The Chimera scheme, however, requires an additional code to locate and label interpolated points as well as redundant points. The disadvantages of the Chimera scheme include the introduction of a slightly more complicated solution algorithm and, more seriously, the trilinear interpolation used in this approach is a locally non-conservative procedure. Nevertheless, this method is very attractive when dealing with moving boundary problems since after each time step or position, new grids do not have to be regenerated, rather only the new lines of communication between outer and hole boundaries need to be established.

1.2.2 Unstructured Grid Methods

Unstructured grids discretize the physical domain of interest by contiguous triangles and tetrahedra in two- and three-dimensions, respectively. These are the simplest geometrical shapes having area and volume and thus, have the capability to discretize irregularly shaped domains easier and more efficiently than structured hexahedral cells. It is interesting to note that structured domain decomposition techniques are attempts to simplify grid generation by dividing the physical domain into subdomains in which structured hexahedral cells may be readily used. This is opposed to unstructured grids which take this decomposition to its finest level. Instead of discretizing simpler subdomains with geometrically more complex cells, it uses the simplest geometric cell to discretize the entire physical domain.

Methods currently being used for unstructured grid generation are Delaunay triangulation [28,29] and the advancing front method [30-32]. It has been reported [33], however, that Delaunay triangulation based algorithms tend to be more efficient than the advancing front method, but lacks the advancing front's self sufficiency, robustness, and grid quality. Furthermore, a comparative study of the two methods may be found in Ref. 34.

Another advantageous quality of unstructured grids is that their adaptation is relatively straightforward. The current methods of adapting unstructured grids are to adaptively redistribute the nodal points based on either the solution or to a moving body, or to add and delete nodes locally when needed. The former approach is referred to as grid adaptation [35-37] and the latter as adaptive remeshing or h-refinement [38-40]. These methods have been used with great success for both steady and unsteady flow simulations.

1.3 Objectives of the Present Work

Two different methodologies are presented in this study to examine the accuracy, efficiency, and limitations of each. They are both applied to the same steady complex multi-component configurations and unsteady moving boundary problems. The first method seeks to exploit the advantageous qualities of two domain decomposition techniques for structured grids. It consists of combinations of multiblock and structured-overlapped grids, and has been previously demonstrated in Refs. 12,41. The second method discretizes the domains by an unstructured grid approach. These unstructured grids were generated by the advancing front method which has been shown to be very effective for three-dimensional complex configurations. Thus, the objectives of the present work are to compare the performance of structured and unstructured grid techniques for both steady complex and unsteady moving boundary configurations.

Chapter 2

Governing Equations and Discretization

2.1 Equations of Fluid Motion

The equations that govern fluid flow are mathematical interpretations of the physical laws which assure the conservation of mass, momentum, and energy. This coupled set of equations are known as the time dependent Navier-Stokes equations. In the absence of viscosity, heat transfer, and body forces, the Navier-Stokes equations reduce to the extensively used and well understood Euler equations. To follow, the time dependent Euler equations for dynamic meshes will be expressed in conservative law form using the integral, and the analytically equivalent differential, formulations.

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

$$\iiint_{\Omega} \frac{\partial}{\partial t}\vec{Q}\, dV + \iint_{\partial\Omega} \overline{E} \bullet \vec{N} dS = 0 \tag{2.1}$$

$\overline{E} \bullet \vec{N}$ are the inviscid flux vectors normal to the boundary $\partial\Omega$ with

$$\overline{E} = \left\{ \vec{F}, \vec{G}, \vec{H} \right\} \tag{2.2}$$

and $\vec{N}$ representing the outward pointing normal to the boundary.

The analytically equivalent differential form of the governing equations may be obtained by applying Gauss's divergence theorem to the surface integral in Eq.(2.1). Then, assuming the continuity of the integrand, the governing equations may be written in Cartesian coordinates as

$$\frac{\partial \vec{Q}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = 0 \tag{2.3}$$

In order to simplify the numerical treatment of boundary conditions for *structured* grids, the above form of the Euler equations is transformed into boundary conforming curvilinear coordinates

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = 0 \tag{2.4}$$

where

$$\hat{Q} = \frac{\vec{Q}}{J} = \frac{1}{J}\begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix}, \quad \hat{F} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ (e+p)U - \xi_t p \end{Bmatrix} \tag{2.5a}$$

$$\hat{G} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ (e+p)V - \eta_t p \end{Bmatrix}, \quad \hat{H} = \frac{1}{J}\begin{Bmatrix} \rho W \\ \rho W u + \zeta_x p \\ \rho W v + \zeta_y p \\ \rho W w + \zeta_z p \\ (e+p)W - \zeta_t p \end{Bmatrix} \tag{2.5b}$$

with the adjusted contravariant velocities corresponding to the ξ, η, and ζ directions defined as

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t \tag{2.6a}$$

$$V = \eta_x u + \eta_y v + \eta_z w + \eta_t \tag{2.6b}$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t \tag{2.6c}$$

As can be seen, the above equations are generalized from Cartesian coordinates using the following transformations

$$\xi = \xi(x,y,z,t), \quad \eta = \eta(x,y,z,t), \quad \zeta = \zeta(x,y,z,t) \tag{2.7}$$

From this, the transformation metrics and Jacobian of transformation may be expressed as

$$
\begin{gathered}
\xi_x = J\left(y_\eta z_\zeta - z_\eta y_\zeta\right), \quad \eta_x = J\left(y_\zeta z_\xi - z_\zeta y_\xi\right) \\
\xi_y = J\left(z_\eta x_\zeta - x_\eta z_\zeta\right), \quad \eta_y = J\left(z_\zeta x_\xi - x_\zeta z_\xi\right) \\
\xi_z = J\left(x_\eta y_\zeta - y_\eta x_\zeta\right), \quad \eta_z = J\left(x_\zeta y_\xi - y_\zeta x_\xi\right) \\
\zeta_x = J\left(y_\xi z_\eta - z_\xi y_\eta\right), \quad \xi_t = -x_t \xi_x - y_t \xi_y - z_t \xi_z \\
\zeta_y = J\left(z_\xi x_\eta - x_\xi z_\eta\right), \quad \eta_t = -x_t \eta_x - y_t \eta_y - z_t \eta_z \\
\zeta_z = J\left(x_\xi y_\eta - y_\xi x_\eta\right), \quad \zeta_t = -x_t \zeta_x - y_t \zeta_y - z_t \zeta_z \\
J = 1 / \left[x_\xi\left(y_\eta z_\zeta - y_\zeta z_\eta\right) - x_\eta\left(y_\xi z_\zeta - y_\zeta z_\xi\right) - x_\zeta\left(y_\xi z_\eta - y_\eta z_\xi\right)\right]
\end{gathered}
\tag{2.8}
$$

The pressure is related to the state variables via the ideal gas law as

$$
p = (\gamma - 1)\left[e - \rho\frac{\left(u^2 + v^2 + w^2\right)}{2}\right] \tag{2.9}
$$

It should be noted that for an *unstructured* grid, no such transformation is possible nor necessary, thus, the governing equations are solved in Cartesian coordinates.

2.2 Finite Volume Discretization

The finite volume formulation is derived from application of the integral conservation law expressed in Eq.(2.1). A discretization and the subsequent solution obtained using this formulation will ultimately satisfy the integral statement of conservation. Furthermore, the finite volume formulation is more attractive than the finite difference formulation due to its ability to handle arbitrary configurations. This is because the only requirement that must be adhered to is the computational domain must be divided into a finite number of non-overlapping volumes. The shape of these volumes, however, is irrelevant which leads to no ambiguity at grid singularities.

An expression for the semi-discrete approximation to the governing equations may be expressed as

$$V_i \frac{\partial Q_i}{\partial t} = -R_i \tag{2.10}$$

where Q is the cell-averaged conserved variables and R_i is the residual vector containing the inviscid fluxes. Whether on structured or unstructured grids, Eq.(2.10) forms the bases of all finite volume schemes.

For structured grid schemes, the summation in Eq.(2.10) is carried out over the six faces of the hexahedron defining the computational cell. Since structured grids have *logical indexing* of its cells, the following semi-discrete representation may be written

$$\left(\frac{\partial \hat{Q}}{\partial t}\right)_{i,j,k} = -\frac{\hat{F}_{i+\frac{1}{2},j,k} - \hat{F}_{i-\frac{1}{2},j,k}}{\Delta\xi} - \frac{\hat{G}_{i,j+\frac{1}{2},k} - \hat{G}_{i,j-\frac{1}{2},k}}{\Delta\eta} - \frac{\hat{G}_{i,j,k+\frac{1}{2}} - \hat{G}_{i,j,k+\frac{1}{2}}}{\Delta\zeta} \tag{2.11}$$

where the transformations are chosen so that the grid spacing in the computational space is uniform and of unit length (i.e., $\Delta\xi = \Delta\eta = \Delta\zeta = 1$)

The semi-discrete representation for unstructured grid schemes employing tetrahedral cells result from the direct application of Eq.(2.10), with the summation occurring over the four faces of the tetrahedra. This may be written by replacing the residual in this equation with

$$R_i = \sum_{j=k(i)} E_{ij} A_{ij} \tag{2.12}$$

Due to the random placement of the cells in an unstructured mesh, a *generalized indexing* scheme (requiring the use of a connectivity matrix which is referenced repeatedly throughout the solution process) must be used [19,42].

2.3 Upwind Discretizations

Methods currently being used to construct the inviscid flux vectors, which appear on the right hand side of Eq.(2.10), are the central and upwind differencing schemes. Central difference schemes lack dissipation and are inherently unstable. Hence, to allow shock capturing and to suppress even-odd point decoupling, artificial dissipation must be added. The most popular form of this dissipation is due to Jameson et al. [43] and Jameson and Baker [44]. It consists of a blend of second and fourth order differences of the conserved variables. This type of dissipation, however, requires user specified second and fourth order dissipation coefficients which have been found to be case dependent.

Upwind methods overcome this deficiency by modeling the underlying physics of signal propagation as dictated by characteristic theory and, thus, are naturally dissipative. These methods can generally be classified as either Flux Difference Splitting (FDS) or Flux Vector Splitting (FVS) schemes. Currently there are many FDS and FVS schemes available in the literature. A review and comparison for a number of these schemes is given in Ref. [45] and Ref. [46]. For the computations in the present work, the FDS of Roe [47,48] and FVS of van Leer [49,50] are used to evaluate the inviscid fluxes. Both of these schemes are discussed below for calculations on dynamic meshes.

2.3.1 Flux Difference Splitting

Roe's FDS is based on the approximate solution of a locally one-dimensional Riemann problem [47]. The flux across each cell face k, for Roe's scheme, is calculated using the numerical flux formula

$$F_k = \frac{1}{2}\left[F(Q_L)+F(Q_R) - \left|\tilde{A}\right|(Q_R-Q_L)\right]_k \tag{2.13}$$

where Q_L and Q_R are the conserved variables to the left and right of the interface and $\tilde{A}$ is the Roe-averaged flux Jacobian matrix. Furthermore, the Roe-averaged matrix $\tilde{A}$ is a mean value of the true flux Jacobian matrix with the following properties: (*i*) $\tilde{A}\left(Q^L, Q^R\right)$ approches $A(Q)$ as Q^L and Q^R approach Q, (*ii*) $\tilde{A}$ has a complete set of real eigenvalues and eigenvectors, and (*iii*) $\tilde{A}\left(Q^R-Q^L\right)=F\left(Q^R\right)-F\left(Q^L\right)$. Property (*iii*) results in the approximate solution being an exact solution if the right and left states can be connected by a single discontinuity parallel to the interface [45]. This explains the sharp resolution of shocks and contact discontinuities that Roe's FDS is able to obtain. Equation (2.13) essentially represents a central difference of the inviscid fluxes plus an upwind correction [19]. This upwind correction term can be written in canonical form in terms of the right and left eigenvector matrices, and the diagonal matrix of eigenvalues. This expression subsequently reduces to three ΔF flux components which may be written as

$$\left|\tilde{A}\right|(Q_R - Q_L) = \left|\Delta F_1\right| + \left|\Delta F_2\right| + \left|\Delta F_3\right| \tag{2.14}$$

where

$$\left|\Delta F_1\right| = \left|\tilde{U}\right| \left\{ \left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2}\right) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \dfrac{\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2}{2} \end{bmatrix} + \tilde{\rho} \begin{bmatrix} 0 \\ \Delta u - \Delta U\,\xi_x \\ \Delta v - \Delta V\,\xi_x \\ \Delta w - \Delta W\,\xi_x \\ \tilde{u}\,\Delta u + \tilde{v}\,\Delta v + \tilde{w}\,\Delta w - \tilde{U}\,\Delta U \end{bmatrix} \right\} \tag{2.15}$$

and

$$\left|\Delta F_{2,3}\right| = \left|\tilde{U} \pm \tilde{a}\right| \left(\frac{\Delta p \pm \tilde{\rho}\,\tilde{a}\,\Delta U}{2\tilde{a}^2} \right) \begin{bmatrix} 1 \\ \tilde{v} \pm \tilde{a}\,\xi_x \\ \tilde{v} \pm \tilde{a}\,\xi_y \\ \tilde{v} \pm \tilde{a}\,\xi_z \\ \tilde{h}_o \pm \tilde{a}\tilde{U} \end{bmatrix} \tag{2.16}$$

with

$$\Delta U = \Delta u\,\xi_x + \Delta v\,\xi_y + \Delta w\,\xi_z \tag{2.17}$$

and the adjusted Roe-averaged contravariant velocity defined by

$$\tilde{U} = \tilde{u}\xi_x + \tilde{v}\xi_x + \tilde{w}\xi_x + \xi_t \tag{2.18}$$

It should be noted that in the above equations the tilde denotes Roe-averaged quantities which may be expressed as

$$\begin{aligned}
\tilde{\rho} &= \sqrt{\rho_L \rho_R} \\
\tilde{u} &= \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right) \\
\tilde{v} &= \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right) \\
\tilde{w} &= \left(w_L + w_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right) \\
\tilde{h}_o &= \left(h_{oL} + h_{oR}\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right) \\
\tilde{a}^2 &= (\gamma-1)\left[\tilde{h}_o - \frac{\tilde{u}^2+\tilde{v}^2+\tilde{w}^2}{2}\right]
\end{aligned} \tag{2.19}$$

For the structured grid algorithm, considerable savings in computational time are realized due to Roe's FDS scheme allowing spatial factors in each direction to be approximated with a diagonal inversion. Further details of this may be found in Ref. 51.

2.3.2 Flux Vector Splitting

For van Leer's FVS scheme [49,50], the flux vectors are given in terms of the Mach number normal to the face. This results in the possible occurrences of supersonic or subsonic flow through the face. The supersonic fluxes are evaluated as

$$\hat{F}^+ = \left(\vec{F}(Q)\bullet\vec{\xi}\right)^R, \quad \hat{F}^- = \left(\vec{F}(Q)\bullet\vec{\xi}\right)^L = 0 \qquad M_\xi \geq 1 \tag{2.20}$$

$$\hat{F}^- = \left(\vec{F}(Q)\bullet\vec{\xi}\right)^L, \quad \hat{F}^+ = \left(\vec{F}(Q)\bullet\vec{\xi}\right)^R = 0 \qquad M_\xi \leq -1 \tag{2.21}$$

As for the subsonic occurrence, the fluxes through face k are split into the following contributions

$$F_k = \hat{F}^+(Q_L) + \hat{F}^-(Q_R) \tag{2.22}$$

where

$$\hat{F}^\pm = \frac{|grad(\xi)|}{J}\left\{\begin{array}{c}
f^\pm_{mass} \\
f^\pm_{mass}\left\{u + \hat{\xi}_x\left(-\bar{U} \pm 2a\right)/\gamma\right\} \\
f^\pm_{mass}\left\{v + \hat{\xi}_y\left(-\bar{U} \pm 2a\right)/\gamma\right\} \\
f^\pm_{mass}\left\{w + \hat{\xi}_z\left(-\bar{U} \pm 2a\right)/\gamma\right\} \\
f^\pm_{energy}
\end{array}\right\} \tag{2.23}$$

with

$$f_{mass}^{\pm} = \pm\frac{\rho a}{4}\left(M_{\xi} \pm 1\right)^2 \tag{2.24}$$

$$f_{energy}^{\pm} = f_{mass}^{\pm}\left[\frac{(1-\gamma)\overline{U} \pm 2(\gamma-1)\overline{U}\,a + 2a^2}{\left(\gamma^2-1\right)} + \frac{u^2+v^2+w^2}{2} - \hat{\xi}_t\frac{\left(-\overline{U} \pm 2a\right)}{\gamma}\right] \tag{2.25}$$

and

$$\begin{aligned} \hat{\xi}_x &= \xi_x/\left|grad(\xi)\right|, & \overline{U} &= U/\left|grad(\xi)\right| \\ \hat{\xi}_y &= \xi_y/\left|grad(\xi)\right|, & M_{\xi} &= \overline{U}/a \\ \hat{\xi}_z &= \xi_z/\left|grad(\xi)\right|, & \hat{\xi}_t &= \xi_t/\left|grad(\xi)\right| \end{aligned} \tag{2.26}$$

$\overline{U}$ is the adjusted contravariant velocity, which is the scalar product of the modified velocity with the normal vector to the face. A point worth noting is that van Leer's FVS is continuously differentiable, which makes it a valuable method for evaluating the inviscid fluxes in implicit algorithms where the flux Jacobians are required. Moreover, it has been found in practice that steady shocks are resolved with at most two interior zones [45,46,49].

2.3.3 Spatial Differencing

The development of a higher order scheme ultimately depends on the accurate interpolation of the state variables to the left and right of the cell interface. The manner in which this interpolation is accomplished depends on the grid type, and this is one of the major differences between the structured and unstructured grid algorithms. This is not to say that the methods used for structured grids cannot be extended to unstructured grids [52-54], it has just been found difficult to obtain CPU efficient, accurate results. Thus, techniques which exploit the geometric properties of triangles and tetrahedra have been developed and used with success for unstructured grids [19,55-57].

2.3.3a For the Structured Grid Algorithm

The structured grid algorithm utilized in this study obtains the state variables values on the cell interfaces from the following upwind-biased interpolations

$$Q^L_{i+\frac{1}{2}} = Q_i + \left\{\frac{1}{4}\left[(1-\kappa)\Delta_- + (1+\kappa)\Delta_+\right]\right\}_i \tag{2.27a}$$

$$Q^R_{i+\frac{1}{2}} = Q_{i+1} - \left\{\frac{1}{4}\left[(1-\kappa)\Delta_+ + (1+\kappa)\Delta_-\right]\right\}_{i+1} \tag{2.27b}$$

where

$$\Delta_- = Q_i - Q_{i-1}\,, \quad \Delta_+ = Q_{i+1} - Q_i \tag{2.28}$$

with

$$\kappa = \begin{cases} -1 & \text{second order fully upwind} \\ 1/3 & \text{third order upwind-biased} \\ 1 & \text{central difference} \end{cases} \tag{2.29}$$

In regions with large flow gradients, such as shocks, flux-limiting is used to eliminate numerical oscillations for the upwind-biased scheme. For all cases, the minimum-modulus (min-mod) flux-limiter [58] was used, and may be expressed by replacing Δ_- and Δ_+ in Eqs.(2.27a and b) by $\tilde{\Delta}_-$ and $\tilde{\Delta}_+$

$$\tilde{\Delta}_- = max\left[0, min(\Delta_- sgn\Delta_+, \beta\Delta_+ sgn\Delta_-)\right] sgn\Delta_- \tag{2.30a}$$

$$\tilde{\Delta}_+ = max\left[0, min(\Delta_+ sgn\Delta_-, \beta\Delta_- sgn\Delta_+)\right] sgn\Delta_+ \tag{2.30b}$$

where

$$\beta = \frac{(3-\kappa)}{(1-\kappa)} \tag{2.31}$$

The parameter β has been referred to as a compression parameter [59].

Furthermore, it should be noted that other flux-limiters may be chosen which are not discussed here. Examples are the van Albada and Spekreijise flux-limiters which may be found in Refs. 60 and 61, respectively.

2.3.3b For the Unstructured Grid Algorithm

For the unstructured grid algorithm, a higher-order scheme is obtained by expanding the cell-centered solution to each cell face using a Taylor series expansion [55] which may be expressed as

$$Q_f^{L,R} = Q_c + \nabla Q_c \bullet \Delta\vec{r} + \vartheta\left(\Delta r^2\right) \tag{2.32}$$

where the solution gradient, , ∇Q_c at the center of the cell is found using the geometric invariant features of triangles and tetrahedra. The expression for the solution gradient at the cell center may be obtained from application of Greens theorem as

$$\nabla Q_c \bullet \Delta\vec{r} = \left[\frac{1/3\left(Q_{n1} + Q_{n2} + Q_{n3}\right) - Q_{n4}}{4\Delta r}\right]\Delta r \tag{2.33}$$

where Q_{n1}, Q_{n2}, Q_{n3} are the primitive variables at the three nodes that constitute the face through which the flux passes, Δr is the distance from the centroid of the tetrahedra to the center of that face, and Q_{n4} are the same variables at the fourth node of the tetrahedra. The data at the nodes are interpolated using inverse distance weighting of the surrounding cell centers. This, as mentioned in Ref. 19, is the only question of accuracy in the overall scheme. It should be noted, however, that in Ref. 62 the data at the nodes have been obtained by both the current inverse distance weighting and by a linear least squares fit, with no discernible differences between the two. An improved averaging scheme, moreover, has been recently implemented in USM3D. Details of this scheme may be found in Ref. 57.

Chapter 3

Solution Algorithms

3.1 Time Integration

Time integration may be done implicitly or explicitly. Implicit methods traditionally have high computational costs per iteration in terms of both CPU time and memory, however, they have less stringent stability bounds. Thus, the extra work required for an implicit scheme is usually offset by the advantages obtained by the increased stability limits. Explicit methods, on the other hand, are relatively inexpensive per iteration but have restrictive bounds on stability.

Other considerations, such as the architecture of the computer used and the physics of the flow problem to be simulated, must also be investigated when making a choice on the type of time integration scheme. For example, most of the supercomputers used today use high speed vector processors and, thus, the degree to which a certain algorithm can be vectorized becomes critical. It is well known that most explicit schemes are readily vectorizable. Implicit schemes, on the other hand, need substantial amounts of memory for temporary storage and data management to become fully vectorized.

The other consideration mentioned earlier, when deciding on a time integration scheme, was based on the physics of the flow problem to be simulated. For unsteady flows, it is imperative that time accurate methods be used and that the time steps be commensurate with the time scale of the unsteady phenomena. Both implicit and explicit methods are capable of computing time accurate solutions, but the time scales are usually so small that the stability of the explicit methods are not jeopardized. Hence, on a per iteration basis, an explicit method *appears* to be the most economical approach for time accurate unsteady simulations, and implicit methods for steady state calculations. This general assessment is further complicated when these algorithms incorporate domain decomposition techniques, multigrid methods and other forms of convergence acceleration.

In the sections to follow, the details of the time integration schemes utilized in this study, for both the structured grid algorithm as implemented in the computer code CFL3D [63-65] and the unstructured grid code USM3D [19,55-57], will be presented.

3.1a For the Structured Grid Algorithm

The structured grid algorithm used in this study advances the solution in time using an implicit method. This may be accomplished by first linearizing the inviscid fluxes in time as

$$\hat{F}^{n+1} = \hat{F}^{n} + \frac{\partial \hat{F}^{n}}{\partial Q} \Delta Q^{n} \tag{3.1}$$

similarly for $\hat{G}$ and $\hat{H}$. The linearized, backward-Euler time integration of the unsteady equations is written as

$$\left[\frac{I+\phi}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q} + \delta_\eta \frac{\partial \hat{G}}{\partial Q} + \delta_\zeta \frac{\partial \hat{H}}{\partial Q} \right] \Delta Q = -R\left(Q^{n}\right) \tag{3.2}$$

where the residual is collected as

$$R\left(Q^{n}\right) = \delta_\xi \hat{F} + \delta_\eta \hat{G} + \delta_\zeta \hat{H} - \frac{\phi}{J\Delta t} \Delta Q^{n-1} \tag{3.3}$$

For upwind methods, the inviscid fluxes in the residual are replaced with the appropriate terms from one of the desired flux splitting methods expressed earlier. For example, the FDS scheme of Roe is used to express the fluxes on the faces as

$$\left[\delta_\xi \hat{F}\right]_i \equiv \hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}} \tag{3.4}$$

where

$$\hat{F}_{i+\frac{1}{2}} = \frac{1}{2}\left[\hat{F}\left(Q^L_{i+\frac{1}{2}}\right) + \hat{F}\left(Q^R_{i+\frac{1}{2}}\right) - \left|\tilde{A}\right|_{i+\frac{1}{2}}\left(Q^R_{i+\frac{1}{2}} - Q^L_{i+\frac{1}{2}}\right)\right] \tag{3.5}$$

The fluxes in the other directions are found similarly. For the FVS scheme of van Leer, the fluxes are split into forward and backward contributions according to the signs of the eigenvalues of the Jacobian matrices. This splitting is given by

$$\begin{aligned}\left[\delta_\xi \hat{F}\right]_i &\equiv \left[\delta_\xi^- \hat{F}^+ + \delta_\xi^+ \hat{F}^-\right]_i \\ &= \left[\hat{F}^+\left(Q^L_{i+\frac{1}{2}}\right) + \hat{F}^-\left(Q^R_{i+\frac{1}{2}}\right)\right] - \left[\hat{F}^+\left(Q^L_{i-\frac{1}{2}}\right) + \hat{F}^-\left(Q^R_{i-\frac{1}{2}}\right)\right]\end{aligned} \tag{3.6}$$

where Q^L and Q^R are given in Eq.(2.27a and b).

For most three-dimensional problems, the direct numerical solution to Eq.(3.2) is impractical due to the large banded coefficient matrix of the system. To overcome this, there are a number of approximate factorizations which may be used to split this large coefficient matrix into a sequence of smaller banded matrices. Some examples of possible factorizations consist of a six-factor block bidiagonal, a two-factor eigenvalue split, a two-factor combination split, and a three-factor spatially split scheme. This latter factorization is the one used by the structured grid algorithm [63,64] in this study.

Applying the three-factor spatially split factorization to Eq.(3.2) yields the following series of sweeps

$$\left[\frac{I+\phi}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q}\right]\Delta Q^* = -R\left(Q^n\right) \tag{3.7a}$$

$$\left[\frac{I+\phi}{J\Delta t} + \delta_\eta \frac{\partial \hat{G}}{\partial Q}\right]\Delta Q^{**} = \left(\frac{I+\phi}{J\Delta t}\right)\Delta Q^* \tag{3.7b}$$

$$\left[\frac{I+\phi}{J\Delta t} + \delta_\zeta \frac{\partial \hat{H}}{\partial Q}\right]\Delta Q = \left(\frac{I+\phi}{J\Delta t}\right)\Delta Q^{**} \tag{3.7c}$$

For each of the above three equations, if the differencing on the left hand side of each equation is reduced to first order spatial accuracy, the solution of a 5 by 5 block tridiagonal system is required . The above scheme is first order accurate in time for $\phi=0$ and nominally second order for $\phi=1/2$.

3.1b For the Unstructured Grid Algorithm

The unstructured grid algorithm used in this study advances the solution in time using the explicit m-stage Runge-Kutta time integration scheme developed by Jameson et al [43]. For all the unstructured cases presented in this study, the governing equations are integrated in time using four stages which has second order temporal accuracy.

Applying this scheme to the semi-discrete finite volume representation given in Eq.(2.10) yields

$$\left.\begin{aligned} Q_i^{(0)} &= Q_i^n \\ Q_i^{(1)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{4}\frac{\Delta t}{V_i^{n+1}} R_i^{(0)} \\ Q_i^{(2)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{3}\frac{\Delta t}{V_i^{n+1}} R_i^{(1)} \\ Q_i^{(3)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{1}{2}\frac{\Delta t}{V_i^{n+1}} R_i^{(2)} \\ Q_i^{(4)} &= \frac{V_i^n}{V_i^{n+1}} Q_i^{(0)} - \frac{\Delta t}{V_i^{n+1}} R_i^{(3)} \\ Q_i^{n+1} &= Q_i^{(4)} \end{aligned}\right\} \tag{3.8}$$

where the residual is expressed in Eq.(2.12).

3.2 Geometric Conservation

To avoid grid-motion induced errors when dynamic meshes are involved, the geometric conservation law (GCL) must be satisfied concurrently with the conservation of mass, momentum, and energy (Refs. 35,66,67). The GCL is only needed for the unstructured grid algorithm since the mesh is deformed and locally changed. This is not the case with the structured-overlapped grids which move as rigid bodies on top of one another [68]. The integral statement of the GCL may be written as

$$\frac{d}{dt}\iiint_{\Omega} dV = \iint_{\partial\Omega} \vec{W}\bullet\hat{n}\, dS \tag{3.9}$$

where $\vec{W}$ denotes the local velocity of the cell faces. 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. (3.9) has been expressed in Ref. 35 which is consistent with the above unstructured solution algorithm and is given by

$$V_i^{n+1} = V_i^n + \sum_{j=\kappa(i)} \left[\xi_t\, \Delta A\right]_{ij}^{n+1} \tag{3.10}$$

Thus, this equation is used to compute the local cell volumes at the current time level as seen in Eq.(3.8).

3.3 Convergence Acceleration Techniques

For steady-state calculations, the governing equations are integrated from an arbitrary initial condition to a time-asymptotic state. Thus, when a steady-state solution is desired, it is typical to employ first order time accurate schemes and use non-time-like maneuvers in an attempt to accelerate the algorithm. Examples of some current convergence acceleration techniques being used are: (*i*) *local time stepping* [43] which can be viewed as a means of conditioning the coefficient matrix in an implicit schemes or interpreted as an attempt to use a more uniform Courant number throughout the flow field for explicit schemes, (*ii*) *mesh sequencing* which uses a good initial guess for a fine mesh by first iterating on a sequence of coarser meshes, (*iii*) *multigridding* [25,69] which damps the low-frequency errors by using a series of coarser grids constructed from the fine mesh, (*iv*) ***implicit residual smoothing*** [19,43] which, in an implicit-like manner, averages the residuals locally.

All the above techniques have been found to accelerate the convergence to steady-state. Local time stepping, mesh sequencing, and multigridding are available options in the structured grid code CLF3D; whereas, local time stepping and implicit residual smoothing are options in the unstructured grid code USM3D. For a more detailed discussion of these methods, the reader is referred to the cited literature.

3.4 Initial and Boundary Conditions

The solution to any partial differential equation is completely dependent on the choices of the initial and boundary conditions. As would be expected, the form of these conditions are different for steady and unsteady computations. To follow is a discussion of the initial conditions, physical boundary conditions for steady flows and the modifications needed for unsteady moving boundary calculations. It should be noted that all boundary conditions used in this study are specified explicitly.

The initial conditions for a steady state calculation may be arbitrary, however, a good initial guess at the flow field will ultimately reduce the CPU time needed to converge the solution. It is thus common practice for steady computations to choose freestream conditions as the initial condition. Unsteady moving boundary problems, in which time accurate solutions are sought, require meaningful initial conditions. Hence, for this type of simulation, fully converged steady state solutions are used.

At the farfield boundaries, locally one-dimensional characteristic boundary conditions are employed. Here the downstream- and upstream-running Riemann invariants are written as

$$R^{\pm} = U \pm \frac{2}{\gamma - 1} a \tag{3.11}$$

Based on the direction and magnitude of the local normal Mach number, the local normal velocity and speed of sound may be determined from the Riemann invariants. Other quantities such as density and pressure may be found using the entropy relation and the equation of state, respectively.

Wall or solid surfaces are considered to be impermeable and adiabatic. For inviscid flows this is accomplished by imposing the flow tangency condition expressed by

$$u_b = u - \eta_x V \tag{3.12a}$$

$$v_b = v - \eta_y V \tag{3.12b}$$

$$w_b = w - \eta_z V \tag{3.12c}$$

where V is the contravariant velocity. The pressure is determined by enforcing the normal pressure gradient to be zero, and the density is extrapolated.

For unsteady moving boundary problems, however, the above conditions must be adjusted since the boundary faces now posses a discernible velocity. The expression for the unsteady-corrected boundary velocities are given by the same expression as in Eq.(3.12), except that the contravariant velocity used is now the one used for dynamic grids in Eq.(2.6). The pressure gradient is no longer zero and must be found from enforcing the normal momentum equation [70,71] as

$$\frac{\partial p}{\partial n} = -\rho \hat{n} \bullet \vec{a} \tag{3.13}$$

where $\hat{n}$ is the direction normal to the boundary surface and

$$\vec{a} = \ddot{\vec{R}} + \dot{\vec{\omega}} \times \vec{\rho} + \vec{\omega} \times (\vec{\omega} \times \vec{\rho}) + \left(\ddot{\vec{\rho}}\right)_r + 2\vec{\omega} \times \left(\dot{\vec{\rho}}\right)_r \tag{3.14}$$

is the acceleration of the body. This acceleration is generalized for both translational and rotational motions. From the left to right, the terms on the right hand side of Eq.(3.14) represent the translational acceleration, the tangential acceleration, the centripetal acceleration, the relative acceleration between the moving reference frame and the point of concern, and the coriolis acceleration, respectively.

Chapter 4

Procedures for Moving Grids

For any CFD calculation, the first step is the discretization of the physical domain; which may be challenging for complex configurations. Grid models to handle store separation or moving boundary problems have the added requirement of discretizing the changing domain around these bodies. This domain may encompass several bodies with large relative movements. Fortunately, the two grid models that have been found to perform best for complex geometries are also the most promising for the moving boundary problems. These two are the structured-overlapped grids and the unstructured grids. In the following sections, the details and relative merits of each model will be discussed for applications involving moving grids.

4.1 Structured Domain Decomposition

Several types of structured domain decomposition methods exist and have been used with great success for cases involving steady, complex configurations [8,15,41,17,72,73]. For moving boundary problems, multiblock and overlapped grid approaches have been the primary areas of focus. Examples of unsteady calculations using dynamic-multiblock and dynamic-overlapped grids may be found in Refs. 5,74,75 and 9,10,20,76-78, respectively.

4.1.1 Composite Grid Construction

Construction of the composite grid and lines of communication between the global and minor grids are established by a code entitled MaGGiE [8,14], which was developed from the Chimera scheme as implemented in the computer code PEGSUS [26,27]. To follow is a brief discussion of this procedure. A more detailed presentation of this material may be found in Ref. 79.

The first task when constructing a composite mesh is to search out and flag all points that fall within the boundaries of a solid body. These points are termed hole points and must be removed from the computational domain. The cells that are immediate neighbors of the hole cells are called fringe points. The intergrid lines of communication, which may be considered as interior boundary conditions, are performed through these points. This line of communication is established by first finding a target cell, in the grid which contains the solid body, that has the smallest distance to the fringe cell. Next, a search is conducted to find seven cells which, with the target cell, forms a hexahedron around the fringe point. Information may now be transferred from the vertices of the hexahedron to the fringe point via a trilinear interpolation of the conserved variables. To perform this interpolation, the hexahedron must be mapped to a unit cube using isoparametric mapping. If the coordinates of the fringe point are denoted by α, β, and γ relative to its target cell, then the interpolation to the fringe point can be expressed as

$$Q = a_1 + a_2\alpha + a_3\beta + a_4\gamma + a_5\alpha\beta + a_6\alpha\gamma + a_7\beta\gamma + a_8\alpha\beta\gamma \qquad (4.1)$$

with

$$\left.\begin{aligned}
a_1 &= Q^*_{i,j,k}, \quad a_2 = -Q^*_{i,j,k} + Q^*_{i+1,j,k} \\
a_3 &= -Q^*_{i,j,k} + Q^*_{i,j+1,k}, \quad a_4 = -Q^*_{i,j,k} + Q^*_{i,j,k+1} \\
a_5 &= Q^*_{i,j,k} - Q^*_{i+1,j,k} + Q^*_{i+1,j+1,k} - Q^*_{i,j+1,k} \\
a_6 &= Q^*_{i,j,k} - Q^*_{i+1,j,k} - Q^*_{i,j,k+1} + Q^*_{i+1,j,k+1} \\
a_7 &= Q^*_{i,j,k} - Q^*_{i,j+1,k} - Q^*_{i,j,k+1} + Q^*_{i,j+1,k+1} \\
a_8 &= -Q^*_{i,j,k} + Q^*_{i+1,j,k} - Q^*_{i+1,j+1,k} + Q^*_{i,j+1,k} + \\
&\quad Q^*_{i,j,k+1} - Q^*_{i+1,j,k+1} + Q^*_{i+1,j+1,k+1} - Q^*_{i,j+1,k+1}
\end{aligned}\right\} \qquad (4.2)$$

where the coefficients a_i are functions of the conserved variable at the vertices of the hexahedron. This above procedure is repeated for the outer boundary points of the minor grids which also require interpolation data.

4.1.2 Dynamic-Overlapped Grids

Dynamic-overlapped grids is the approach used in this study. This scheme uses multiple overset structured grids to allow relative movement between bodies. The aircraft, or main body, may be modeled with a global grid about this structure and then minor grids used about the smaller structures such as stores or engine nacelles. The minor grids are overset on the global mesh and can be moved freely within the global grid.

For store separation sequences, the dynamic-overlapped grid scheme can be summarized as follows for a single iteration. First, the flow solver computes the flow solution on the global grid. Then, this solution is transferred to the outer boundaries of the minor grids via a trilinear interpolation. The solution on these minor grids are subsequently obtained. From these solutions, the aerodynamic forces and moments may be obtained and supplied to a trajectory code which determines the next position of that body. Next, these grids are moved to their appropriate positions and the lines of communication are reestablished. Finally, the solution on the minor grids are interpolated back onto the global grid's fringe points. At this point, if the stopping criteria has not been met, the solution procedure repeats.

4.2 Unstructured Grid Adaptation

4.2.1 Adaptation Method

The grid adaptation method used here has been previously reported by Batina (Ref. 35). The unstructured mesh about the body (or bodies) of interest is considered as a system of interconnected springs. This system is constructed by representing each edge of each tetrahedra by a tension spring. Various attempts at determining the optimum relationship for specifying the spring stiffness have been made by Chakravarthy, et al. (Ref. 80). In the present study, however, the spring stiffness is assumed inversely proportional to the length of its edge and may be written as

$$k_{ji} = 1.0 / [(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{p/2} \tag{4.3}$$

where p is a parameter used to control the stiffness of the spring. Then, for each mesh point, the external forces due to the connecting springs are summed and resolved into Cartesian components. The resulting set of linear systems are solved for the displacements of each node using several Jacobi iterations:

$$\Delta x_j^{n+1} = \frac{\sum k_{ji}\, \Delta x_i^n}{\sum k_{ji}} \tag{4.4a}$$

$$\Delta y_j^{n+1} = \frac{\sum k_{ji}\, \Delta y_i^n}{\sum k_{ji}} \tag{4.4b}$$

$$\Delta z_j^{n+1} = \frac{\sum k_{ji}\, \Delta z_i^n}{\sum k_{ji}} \tag{4.4c}$$

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 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.

Over a number of iterations, poor grid distribution and grid skewness may result from extremely large translations and rotations of the bodies. When a predefined skewness criterion (based on the aspect ratio of the cells) is violated, the unstructured grid must be remeshed, regenerated, or smoothed to alleviate this possible source of error. The present study uses a Laplacian-type smoothing [81] of the grid expressed as

$$x_j^{n+1} = x_j^n + \frac{\omega}{n}\sum_{i=1}^{n}\left(x_i - x_j\right) \tag{4.5a}$$

$$y_j^{n+1} = y_j^n + \frac{\omega}{n}\sum_{i=1}^{n}\left(y_i - y_j\right) \tag{4.5b}$$

$$z_j^{n+1} = z_j^n + \frac{\omega}{n}\sum_{i=1}^{n}\left(z_i - z_j\right) \tag{4.5c}$$

where ω is the relaxation factor and i is summed over all edges connected to node j. The number of smoothing sweeps is user specified and is chosen to be 125 in this study.

4.2.2 Adaptive Window Procedure

Having adopted the above method for adapting the unstructured mesh, computational efficiency can be improved by limiting the size of the adaptation region. Limiting the size of this region is advantageous 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. The nodal points inside this window are considered as the spring network and, thus, allowed to adapt to the body movement. By adopting this procedure, significant savings in both CPU time and memory are realized.

Creating the window may be carried out by either specifying a normal distance from the body of interest or choosing a basis shape around the body (sphere, ellipsoid, etc.). The entire domain is searched to locate the points which fall within the window, and those which do, are flagged as "window" points. 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 reconstructed on several occasions during the body's trajectory. Thus, window construction must be a quick, reliable, and automated process. In the present study a basis shape is used to specify the window, and a critical displacement is chosen to determine when a new window is needed.

Two examples illustrating the adaptive window method for moving body problems are presented. The first example is for a NACA 0012 airfoil sinusoidally oscillating about the quarter chord with an amplitude of 35 degrees. The window constructed about this airfoil is shown in Fig. 4.1. This mesh contains 1577 nodes and 3042 cells, however, the adaptation window contains 569 nodes and 1180 cells. Hence, only about 30% of the original mesh is being adapted. Detailed views of the adapted mesh are given in Fig. 4.2. To ensure the integrity of the mesh around the airfoil, the stiffness of the springs in this region are increased by increasing p in Eq.(4.3) from a value of unity to 2.05.

A second example demonstrates the applicability of the adaptive window procedure to multiple-body problems. This example illustrates how the adaptive window procedure may 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 leading edge slat. Multiple windows about the leading edge slat and vane are shown in Fig. 4.3. 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. Hence, each element could have been given different prescribed motions, and the window regions locally adapted as separate entities.

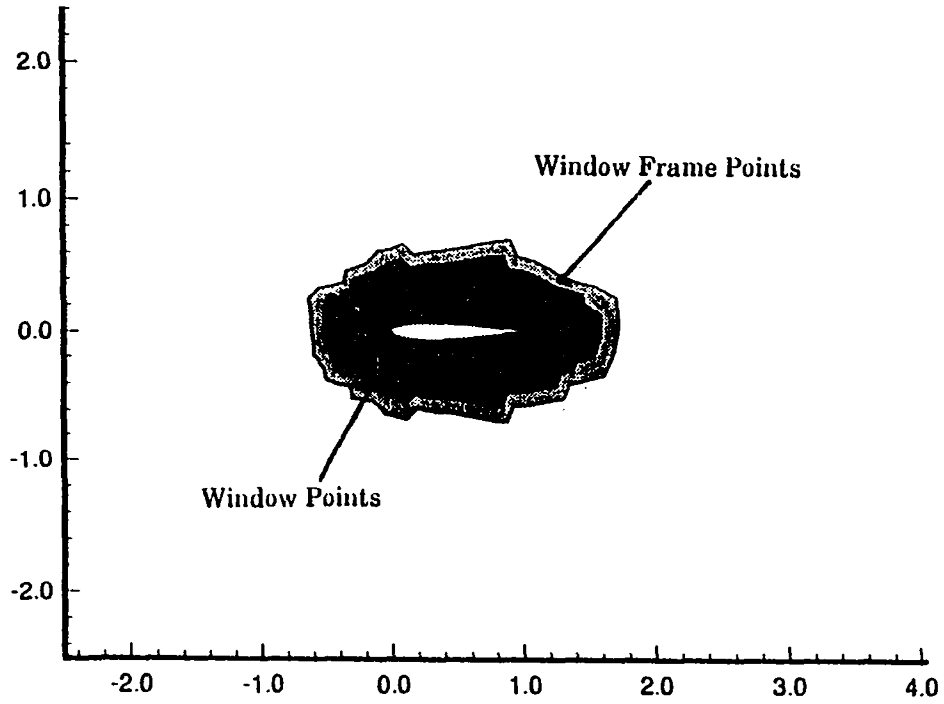
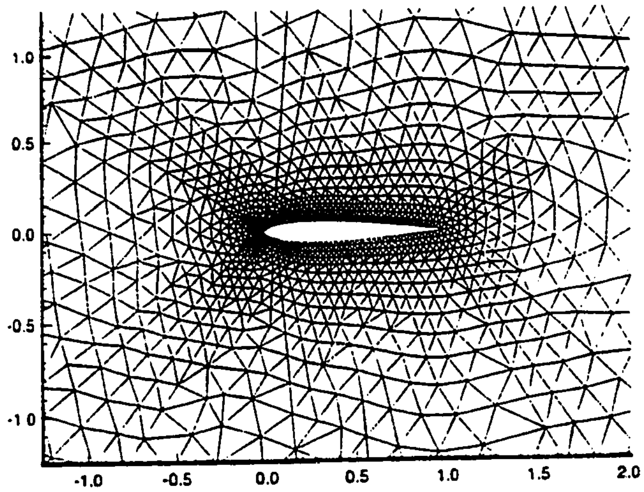
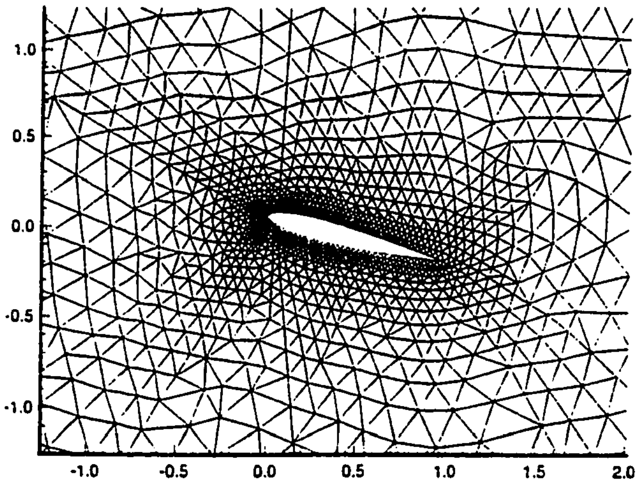


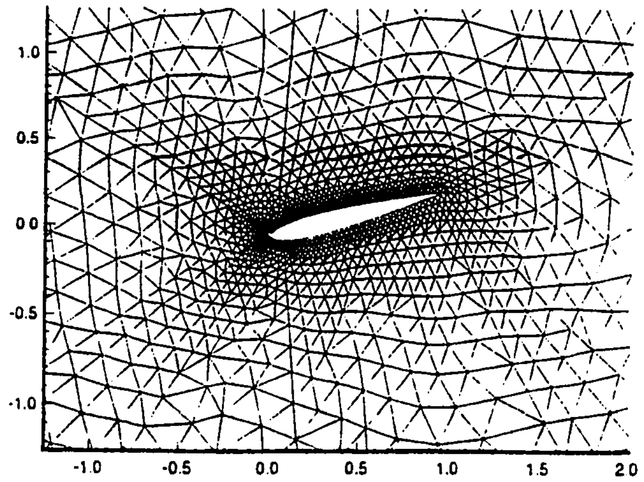
Fig. 4.1 The initial adaptation window for the NACA 0012 airfoil.



(a) Undeformed Mesh



(b) Adapted Mesh (+35°)



(c) Adapted Mesh (-35°)

Fig. 4.2 NACA 0012 airfoil sinusoidally oscillating about the quarter chord.

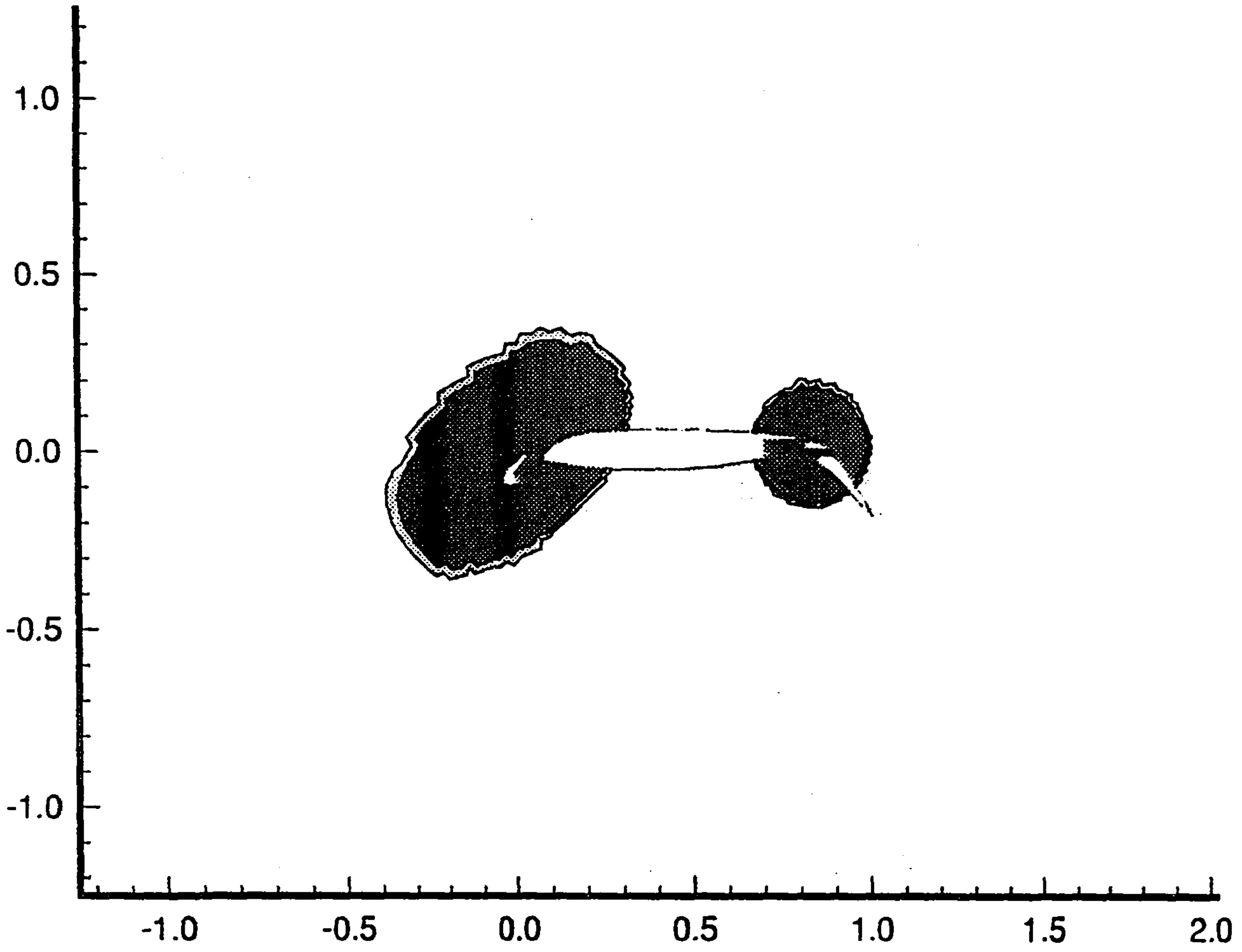


Fig. 4.3 Multiple adaptive windows for a high-lift multielement airfoil.

Chapter 5

Computational Results

5.1 Steady Computations About Complex Configurations

In the following sections, the computations of the steady flow about a two-dimensional high-lift multielement airfoil and a three-dimensional wing/pylon/finned store configuration are discussed.

5.1.1 High-Lift Multielement Airfoil

Grid Generation

The high-lift multielement airfoil used in this study consists of four components: a leading edge slat, a main airfoil, and a double-slotted flap (a vane and a main flap). Structured-overlapped grids are easily generated for two-dimensional configurations with streamlined bodies. The composite mesh consists of four grids generated about each element separately. Grids about the leading edge slat, the vane, and the main flap are of O-topology. The mesh about the main airfoil is used as the global grid and is of C-topology. This composite mesh contains 20,224 cells and is shown in Fig. 5.1a.

The unstructured mesh about this four element airfoil is shown in Fig. 5.1b. This grid is comprised of 7,614 nodes and 14,919 triangular cells. As can be seen, the unstructured mesh has a more efficient distribution of grid points. This is due to the fact that in an unstructured mesh there are no family of grid lines that must be followed. This is not the case for structured grids where the grid lines in the wake of a C-mesh continue out to the far-field.

Flow Simulation

For a freestream Mach number of 0.2 and a 16.02 degree angle of attack, the off surface Mach number contours for this configuration using structured-overlapped and unstructured grids are shown in Fig. 5.2a and b, receptively. As a result of the increased number of cells due to overlapping, better resolution is observed in the leading edge slat region of Fig. 5.2a. Surface pressure coefficients on the four elements are shown in Fig. 5.3a for the present inviscid computations, and in Fig. 5.3b for the unstructured viscous calculations of Ref. 82. From this figure it can be seen that the suction peaks are over predicted for the structured-overlapped grids and under predicted for the unstructured mesh. Since it is expected that an inviscid solution would over predict this phenomenon, it is concluded that the under prediction is due to the coarseness of the unstructured mesh (which is especially noticeable in the leading edge slat region). Furthermore, it has been asserted [82] that the inadequate resolution of these suction peaks causes the numerical generation of entropy, which is convected downstream, and may ultimately degenerate the accuracy of the solution in downstream regions. Discrepancy is also observed on the vane and flap. The most probable cause of this is due to the inviscid nature of the present computations which do not simulate the separated flow condition. Another possible source of error is the coarseness the grids in these regions. It has been shown in Ref. 83, through a grid refinement study, that the conditions aft of the flap in multielement airfoil configurations are highly sensitive to grid resolution.

The computations on the structured-overlapped grids used 2.4 Mega-words (Mw) of memory and 0.75 CPU hours to reduce the residual about 5 orders of magnitude in 2200 iterations on a Cray-2 supercomputer. A similar reduction in the residual for the unstructured grid scheme took 1700 iterations for a total of 1.65 CPU hours, and required 5.1 Mw on the same computer. It is well known that unstructured grid schemes have more intense computational needs than the structured grid schemes, however, the disparity between the two reduces as the number of subdomains in the structured domain decomposition increases.

In Ref. 84, simulations about a Modular Transonic Vortex Interaction (MTVI) model were performed using multiblocked-structured grids and an unstructured mesh. From this comparison, it was concluded that the unstructured grid scheme used significantly greater amounts of CPU time and memory than the structured grids, but quicker *turn-around* time for the generation of the mesh was observed with the unstructured grids. For a complex three-dimensional geometry, it may be difficult (if not impossible) to generate a multiblocked-structured grid about the configuration. Thus, the level/amount of structured domain decomposition must be elevated to handle the increased complexity.

5.1.2 Wing/Pylon/Finned Store Configuration

The following computations, about a generic wing/pylon/finned store (WPFS) assembly, were performed as part of a store separation analysis special session for the 1992 Atmospheric Flight Mechanics Conference. In this session, all papers [10-13,77] pertained to the analysis of the same WPFS configuration. These papers consisted of both structured and unstructured grid simulations. The Armament Directorate of the Air Force Wright Laboratory and Arnold Engineering Development Center (AEDC) conducted the CFD code validation wind tunnel tests which provided pressure, force, and moment data [85]. The WPFS configuration consists of a clipped delta wing with 45 degrees of leading edge sweep and a NACA-64A010 airfoil section. Connected to this wing is an ogive-flat plate-ogive pylon, which is located 0.07 in. above an ogive-cylinder-ogive store when in the carriage position. The store has four fins, which are NACA-0008 airfoil sections and swept at 60 degrees, located at 45, 135, 225, and 315 degrees with respect to the centerline of the pylon. Dimensions and orientation of this geometry in the captive position are depicted in Fig. 5.4.

Grid Generation

The composite mesh consists of 13 blocks and is constructed using multiblocked and overlapped grids. All component and multiblock volume grids in this mesh are generated using GRIDGEN 3D [24]. The strategy employed in the present study is to, first, require that all components have body-conforming grids (i.e., two sets of grid coordinate lines follow the natural surface contours of the body) and, secondly, generate block-structured grids about the components which do not move relative to one another. The second requirement ensures a conservative flux treatment about geometrical complexities in the WPFS configuration. Thus, the grid generation effort breaks into three major tasks: multiblock grid generation for the pylon group, multiblock grid generation for the finned store group, and single zone grid generation of the wing grid. As the final task prior to flow integration, these groups are interconnected using the Chimera scheme discussed in section 4.1.

The pylon group contains 7 blocks, all of which are of H-H topology. This group resolves the lower half of the wing, the pylon, and the region underneath the wing. The unique function of this grid is to envelop the proposed trajectory of the store and, thus, it extends 10.0 store diameters upstream, 18.0 downstream, and 15.0 below the nose of the store. The grid developed for this system contains approximately 425,000 points and is illustrated in Fig. 5.5. Notice that even though the blocks are of H-H topology, the leading and trailing ogive ends of the pylon have the same structure as would an O-grid. Hence, the strategy for requiring body-conforming grids has been met for all components of the wing/pylon group.

The other multiblock group developed is about the finned store. This 4-block system contains 248,132 points and is shown in Fig. 5.6. Each block has 89 grid points in the spanwise direction, 17 in the circumferential direction, and 41 normal to the body surface. Block boundaries correspond to the four fin locations, with the coalition of these blocks forming an O-O body-conformed grid about the store. Upstream, downstream, and radial outer boundaries are located at 4.0, 9.2, and 2.5 store diameters from the nose of the store, respectively.

Additional grids generated consist of a global grid about the wing and an intermediate grid to assist the interpolation in the region around the pylon. The wing grid (Grid 1), consisting of 264,450 points, is of C-O topology. This grid has farfield boundaries located at 45.0 store diameters upstream, 85.0 downstream, and 35.0 outboard of the store nose. An O-C grid, which conforms to the pylon, is also inserted to obtain a better resolution in the region of extreme interference that occurs between the lower wing surface, the pylon, and the upper surface of the finned store.

Once all the grids have been generated, they are interconnected to form a composite grid, and the associated interpolation data is established. The aforementioned computer code MaGGiE is used to accomplish this task. This composite grid contains nearly one million grid points. Due to extensive overlapping, care must be taken to ensure that all points falling within body boundaries (i.e., inside the "solid" body) be removed from the computational domain. To illustrate this, Fig. 5.7 shows the overlapped region between the finned store group and the global wing grid. Notice that holes must be created in the finned store group for the wing and pylon, whereas, the wing grid has points removed in the vicinity of the store. Observe how the pylon group is used to resolve the region of the lower wing surface, pylon and store. Since Grids 3, 4, 5 and 6 are coincident with the wing, the only hole created in this group is for the finned store body. The same cross-sections shown in the previous figure are also shown in Fig. 5.8, but with overlapping and hole boundaries between the wing grid and pylon group depicted. Figure 5.9 is intended to demonstrate the three-dimensional nature of the hole boundaries created in the wing, store, and pylon grids. Orientation and relative size of each grid in this structured composite mesh are shown in Fig. 5.10a for the finned store in the carriage position.

The unstructured grid used in this study was generated using VGRID3D [30]. It contains 68,580 nodes and 379,074 cells. A similar view to the one shown for the structured composite mesh is shown for the unstructured grid in Fig. 5.10b. This figure depicts the surface triangulation on both the WPFS body and the plane of symmetry. As can be seen, the advantage of unstructured grid methodologies is the relative ease with which complex configurations are discretized.

Flow Simulation

The WPFS assembly, and the same assembly without the fins on the store [5], have been the topic of many CFD code validations and comparative studies for complex configurations. In addition to the papers presented in the special session of the Atmospheric Flight Mechanics Conference, Lijewski [7] has performed calculations on this configuration using both structured multiblocked and overlapped grids. In that study, it was concluded that both schemes resulted in comparable accuracy, with excellent agreement with experimental data. However, the multiblocked grid simulation required significantly less CPU time for solving the fluid equations, but an excessive amount of *overhead* grid generation time. In the present study, the two methods that have been found to perform best for three-dimensional complex configurations, in terms of *overhead* grid generation time, are compared. These are the structured-overlapped and unstructured grid schemes.

Simulations, with the finned store in the captive position, zero degrees angle-of-attack, and a freestream Mach number of 0.95, were performed using both the structured-overlapped and unstructured grid schemes. The pressure contours on the WPFS body and plane of symmetry at an oblique angle are depicted in Fig. 5.11a and b for each method. It can be clearly seen that both schemes capture the major flow physics, however, the structured overlapped grids have crisper resolution of the shock waves. This is due to the fact that the structured grids are much finer than the unstructured mesh, and that the grid lines in this mesh are nearly aligned with the waves. The coarseness of this unstructured mesh and the resulting lack of resolution is exemplified in Figs. 5.12 and 5.13. Figure 5.12a and b illustrates the grids and Fig. 5.13a and b presents the surface pressure contours on the upper surface of the wing for the structured-overlapped and unstructured grid schemes, respectively. Shown in Figs. 5.14 and 5.15 are the same set of views for the lower surface of the wing. Once again, lack of resolution is observed due to the coarseness of the unstructured mesh. This is especially noticeable in the wave structure near the trailing edge of the pylon. It should be noted that an unstructured WPFS simulation was performed using the same flow solver [19], and presented in the special session of the Atmospheric Flight Mechanics Conference [11]. The unstructured mesh in this simulation was much finer (103,064 nodes and 567,862 cells), and a flow structure very similar to that of the structured-overlapped grids was obtained.

With the exception of the unstructured solution on the store (which is explained below), good agreement between computed and experimental data is observed on all components. Pressure coefficient comparisons at two inboard span locations on the wing are shown in Fig. 5.16. As seen in this figure, the upper surface of the wing shows little effect of aerodynamic interference, but it does demonstrate the characteristic expansion along the wing chord and the existence of a shock near the trailing edge. On the lower surface, severe interference is observed to occur between the wing and the pylon when the store is in the carriage position. This aerodynamic interference is also seen in Fig. 5.17 which depicts the pressure coefficient data at two outboard stations on the wing. It is interesting to note that since the unstructured mesh is relatively coarse, the solutions on this grid exhibit a more diffusive behavior (than would the inviscid results on a finer mesh) and, hence, resembles the viscous experimental data more closely. Thus, the present unstructured grid solutions *appear* to agree better with the experimental data in the shock regions. A more representative inviscid solution on an unstructured mesh are shown in Figs. 5.16b and 5.17b for the wing inboard and outboard stations, respectively. These computations, which are that of Ref. 11, exhibit the crisper shocks and higher pressures that are expected with an inviscid solution.

For the store, pressure data was obtained at 36 azimuthal locations, beginning at 5 and ending at 355 degrees with respect to the pylon center line. Comparisons between the computed and the experimentally measured data are presented in Figs. 5.18 and 5.19 for the upper and lower surfaces, and the inboard and outboard sides of the store, respectively. It should be noted that the surface definition of the store for the present unstructured grid is not an exact representation of that body (i.e., the surface triangulation is not a smooth ogive-cylinder-ogive store as modeled with the structured grids or used in the experiment). This is attributed to a lack of experience with the sophisticated grid generation software which was still in the developmental stages. Software [31] presently exists which would have eliminated this deficiency, and was used to correct the surface in Ref. 11. To illustrate the inviscid solution on the correct store model, the computed pressure distributions of Ref. 11 are depicted in these figures. As seen, the computed solutions of Ref. 11 are nearly indistinguishable from the present structured-overlapped solutions. Nevertheless, from these figures it is clearly evident that the highest degree of aerodynamic interference occurs at 5 degrees, which is expected, since the store and the pylon are at their closest proximity. The influence of the fins on the flow are also realized in these figures from the compressions occurring at about 60% of the store's chord.

Pressure data was measured on the inboard and outboard sides of the pylon at two vertical stations. These vertical stations correspond to constant y locations of 0.67 in. and 1.17 in. above the store. For both the inboard and outboard sides at each station, exceptional agreement with experimental data is observed with the structured-overlapped grids. The unstructured mesh also has good overall agreement, with discrepancies being attributed to the coarseness of the grid and the misrepresentation of the store. Notice that the pressure on the inboard sides is less than that on the outboard sides, which would suggest a resultant side force directed inward.

Due to the extensive degree of overlapping, with nearly 105,000 interpolated hole or outer boundary points, the structured-overlapped grids have more intense computational needs than a single or multiblock structured flow solver. To reduce the residual 5.0 orders of magnitude it took the structured-overlapped code 1700 iterations, 20.5 Cray-2 hours, and 37.5 Mw of memory. The unstructured code reduced the residual 6.2 orders of magnitude in 2250 iteration, 8.75 Cray Y-MP hours, and used 36.2 Mw of memory. It should be noted that different Cray-class computers were used for the computations, with the Y-MP being approximately 1.6 times faster that the Cray-2. Taking this into consideration, the CPU run times are roughly the same, however, the structured-overlapped grids contain over 60% more cells than the unstructured mesh.

5.2 Unsteady Moving Body Computations

In the following sections, the unsteady simulations of the flow about a two-dimensional pitching airfoil and an aerodynamically determined airfoil/store separation sequence are discussed.

5.2.1 Forced Pitching Oscillation of an Airfoil

As with the WPFS case for complex configurations, the forced pitching oscillation of a NACA 0012 airfoil has been used as the benchmark case for many code validation studies [35,37,39,48,71,76,86]. Presented is a comparative study of dynamic-overlapped grids and dynamic unstructured meshes for the unsteady pitching airfoil. To assess accuracy, the computed instantaneous pressure coefficient distributions for each method are compared with experimental data [87].

Grid Generation

The composite mesh for the structured-overlapped grids has a total of 9856 cells contained within two blocks, and is shown in Fig. 5.21a. The first block is a Cartesian mesh, and it is used as the global grid. This mesh has farfield boundaries located approximately 20 chords from the airfoil's quarter chord. The second block is of O-topology and it resolves the region directly around the airfoil. Shown in Fig. 5.21b is the unstructured mesh around this NACA 0012 airfoil. This mesh has 1577 nodes and 3042 cells. The farfield boundaries of the unstructured grid are placed approximately 15 chords from the quarter chord.

Flow Simulation

A NACA 0012 airfoil sinusoidally oscillating about its quarter chord with a mean incidence of 4.86 degrees, an amplitude of 2.44 degrees, a reduced frequency of 0.081, and a freestream Mach number of 0.6 is simulated. The initial condition for this unsteady problem was a fully converged steady-state solution. A periodic solution was obtained in three cycles of motion for both methods.

Illustrated in Figs. 5.22 through 5.29 are the instantaneous offsurface pressure contours and the unsteady pressure coefficient comparisons with experimental data for eight positions. Notice that as the airfoil oscillates, a shock is formed on the upper-surface which migrates toward the leading edge as the angle of attack is increased. As the angle of attack is decreasing, this shock migrates away from the leading edge, becoming non-existent as the angle of attack approaches mean incidence. Both methods have good overall agreement between the computed and the experimental pressure coefficient distributions at all of the eight positions. Small discrepancies over the first 5% of the airfoils upper surface are believed to be the result of neglecting viscous effects in these computations; with the maximum disparity occurring at the angles of 3.49 and 2.43 degrees. It is should be noted, however, that in Refs. 48 and 71, similar discrepancies have been observed between computed and the experimental data of Ref. 87 at selected angles of attack. In all cases, the pressure over the lower surfaces are consistently higher than that of the experiment, which would suggest that an angle of attack correction is needed. Depicted in Fig. 5.30 is the variation of the normal force coefficient with the angle of attack. As would be expected from an inviscid computation, the agreement is improved at lower angles of attack.

The CPU requirements of the structured-overlapped and the unstructured grid schemes are increased when dynamic meshes are involved. This is due to the need of reestablishing the lines of communication at hole and outer boundary points for dynamic-overlapped grids or for mesh adaptation on the unstructured mesh. For the present computations, the unstructured grid adaptation is a more CPU efficient process, however, solution adaptive remeshing (refinement) is a needed attribute which would definitely increase the computational costs.

It is not possible to give definitive CPU times for either method since it is never known a priori how many interpolated points will arise due to hole creation or how many mesh points will reside in a given window. On average, for the sinusoidally oscillating airfoil, the construction of the composite mesh required 19 μ seconds/iteration/cell and the unstructured mesh adaptation required 9.5 μ seconds/iteration/cell. The total CPU time used to complete three cycles of pitching was 4.6 Cray-2 hours for the dynamic-overlapped grids and 6.1 Cray-2 hours for the dynamic-unstructured mesh.

5.2.2 Aerodynamically Determined Airfoil/Store Separation

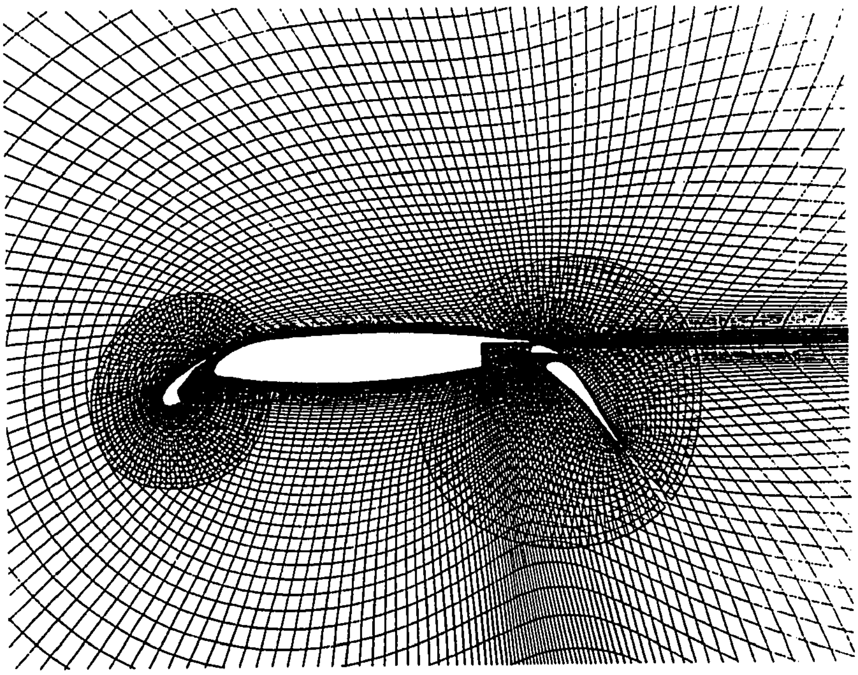
One of the primary interests for the development of dynamic mesh capabilities is the direct simulation of unsteady moving boundary problems such as store separation sequences. This case simulates the unsteady flow about an airfoil/store configuration where the store has been released and is free falling under aerodynamically determined motion. The trajectory is obtained by solving the Eulerian equations of rigid body motion for the translations and rotations of the body at each time step. The details of this trajectory code, and the unstructured simulation, have been presented by Singh et al. [37]. The following dynamic-overlapped grid computations prescribed the motion of the store to coincide with that of Ref. 37.

Grid Generation

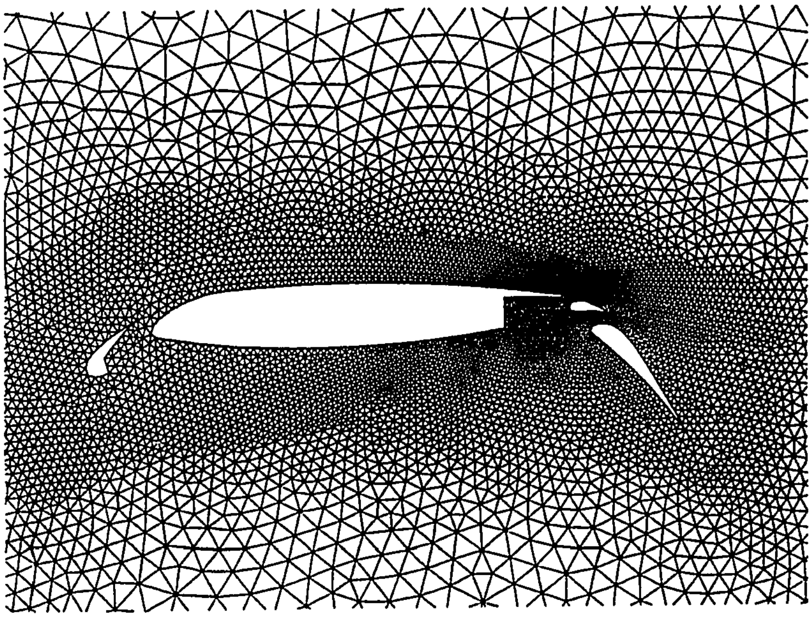
The two-dimensional airfoil/store geometry has been adapted from the three-dimensional WPFS configuration discussed earlier. The airfoil has a NACA 64A010 cross section, and the store an ogive-cylinder-ogive cross section. The composite mesh for the structured-overlapped grids is comprised of two blocks with a total of 10,368 cells, and is shown in Fig. 5.31a. The global grid of C-topology is about the airfoil, and the minor grid of O-topology is about the store. Illustrated in Fig. 5.31b are the unstructured mesh and initial adaptive window for this configuration. This mesh has 10073 nodes and 19707 cell centers. Once again, very efficient grid point distribution is obtained with the unstructured grid.

Flow Simulation

With a freestream Mach number of 0.3, the initial conditions for this unsteady store separation sequence was a converged steady-state solution. Depicted in Fig. 5.32 are the offsurface pressure contours and pressure coefficient distributions for this solution. It should be noted that the *unsightly* contours that appear in the flow field of the structured-overlapped grids are due to the plotting of the regions of overlap that end or begin at hole or outer boundaries. Figures 5.33 through 5.35 illustrate the offsurface pressure contours of three selected separated positions. Position 1 displays the beginning of a compression region below 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. Furthermore, in all figures, a similar flow structure is obtained from both methods. This includes the formation and downstream propagation of a vortex off the trailing edge of the store. By the time the store has reached its final position, the compression region between the airfoil and the store has developed into a strong normal shock, and the store-induced downward force has strengthened.

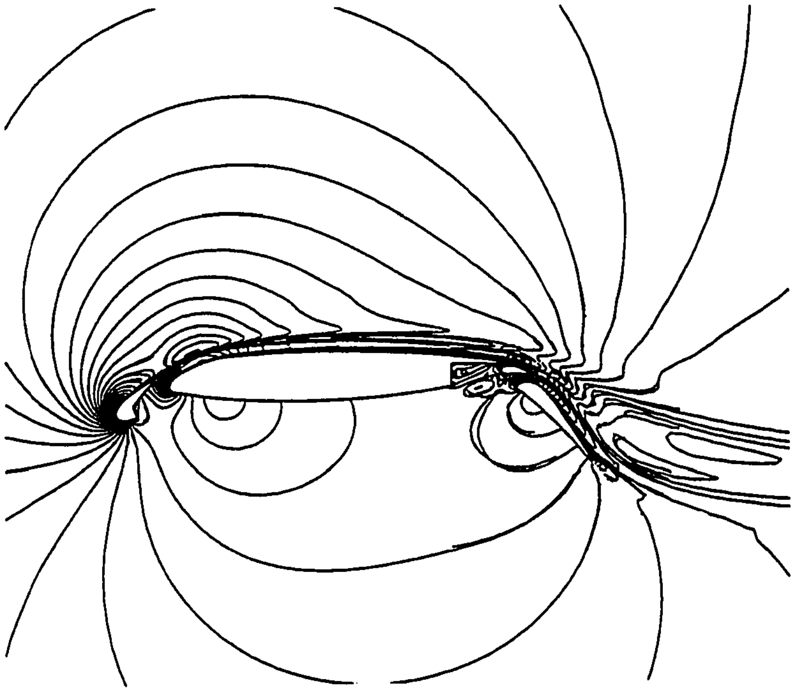


(a) Structured-overlapped grids

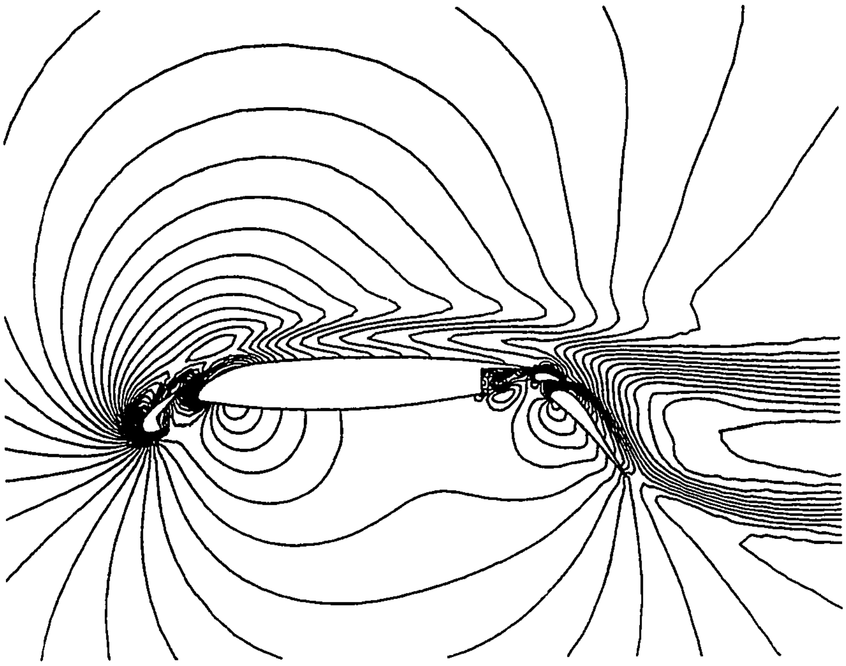


(b) Unstructured mesh

Fig. 5.1 Grids about the high-lift multielement airfoil.

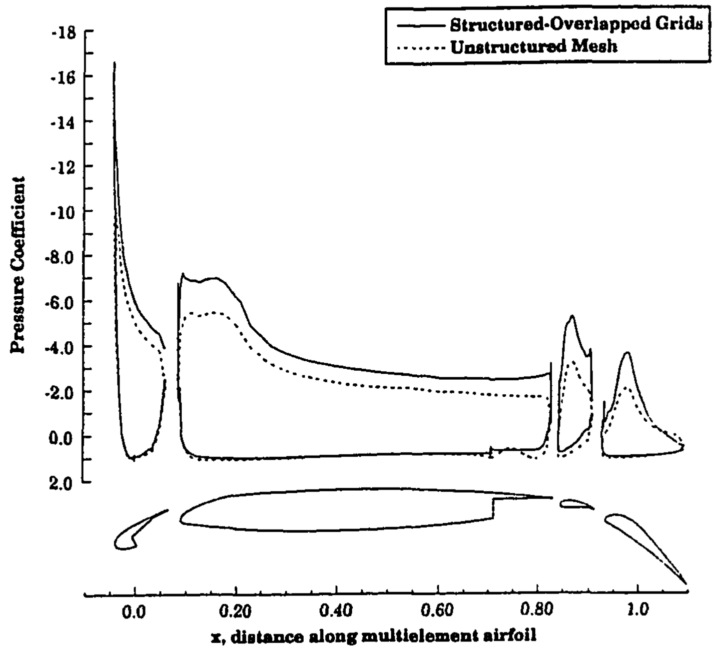


(a) Structured-overlapped grids

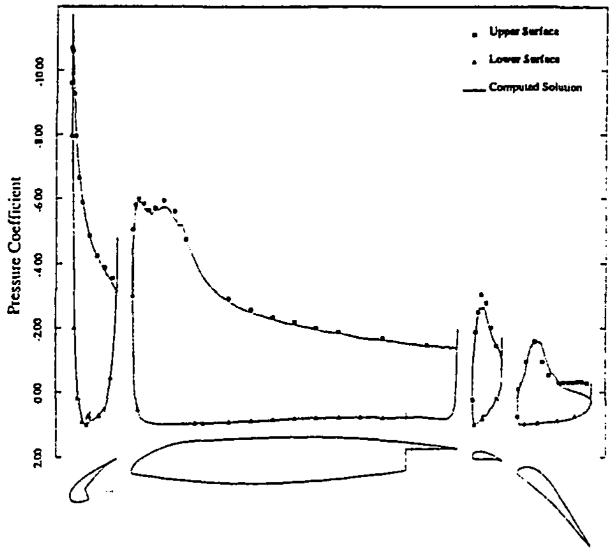


(b) Unstructured mesh

Fig. 5.2 Mach number contours for the high-lift multielement airfoil. (M_∞= 0.2, α= 16.02°)



(a) Present inviscid computations



(b) Unstructured viscous computations of Ref. 82

Fig. 5.3 Pressure coefficient distributions on the high-lift multielement airfoil.

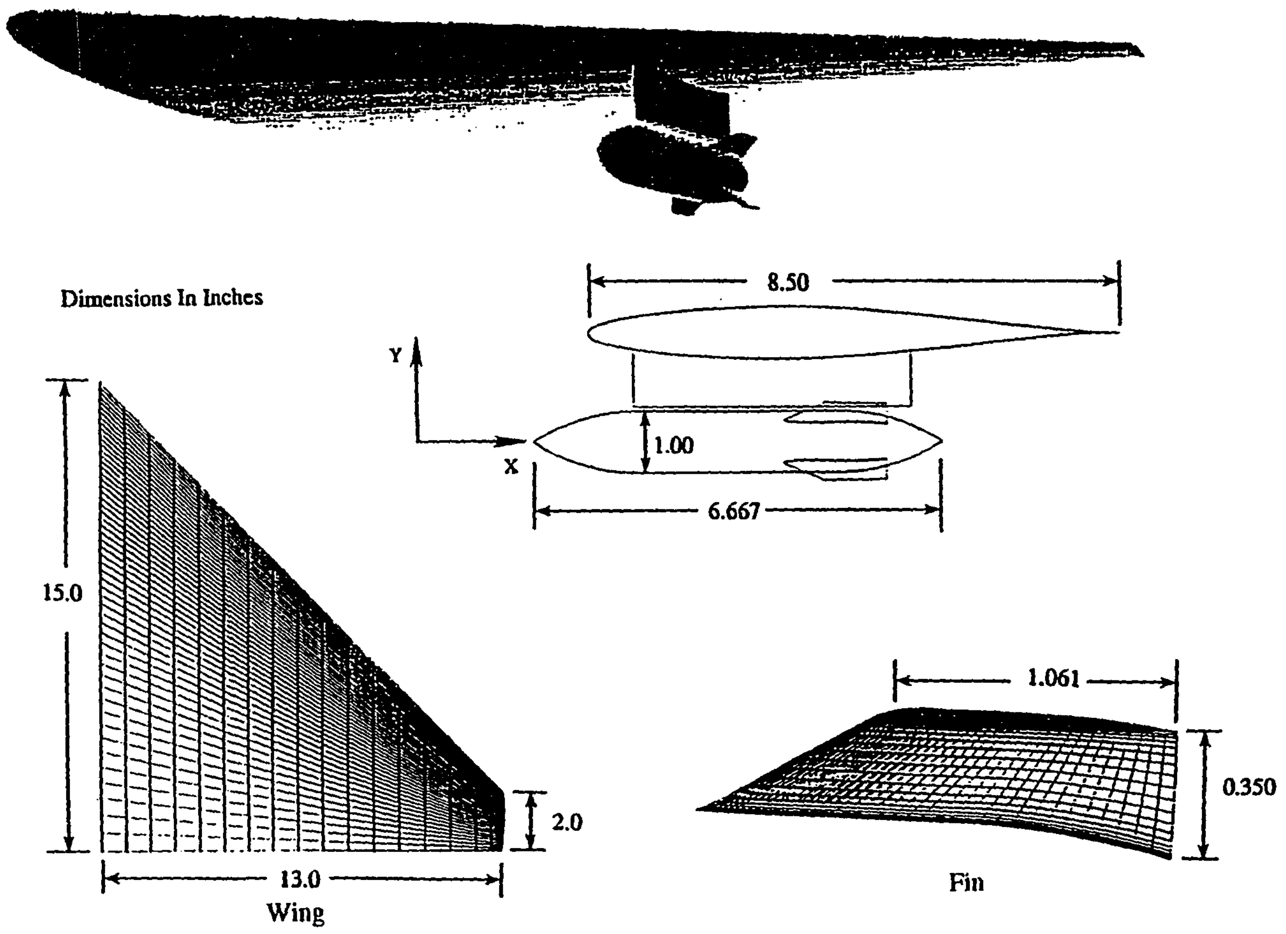


Fig. 5.4 The wing/pylon/finned store (WPFS) geometry.

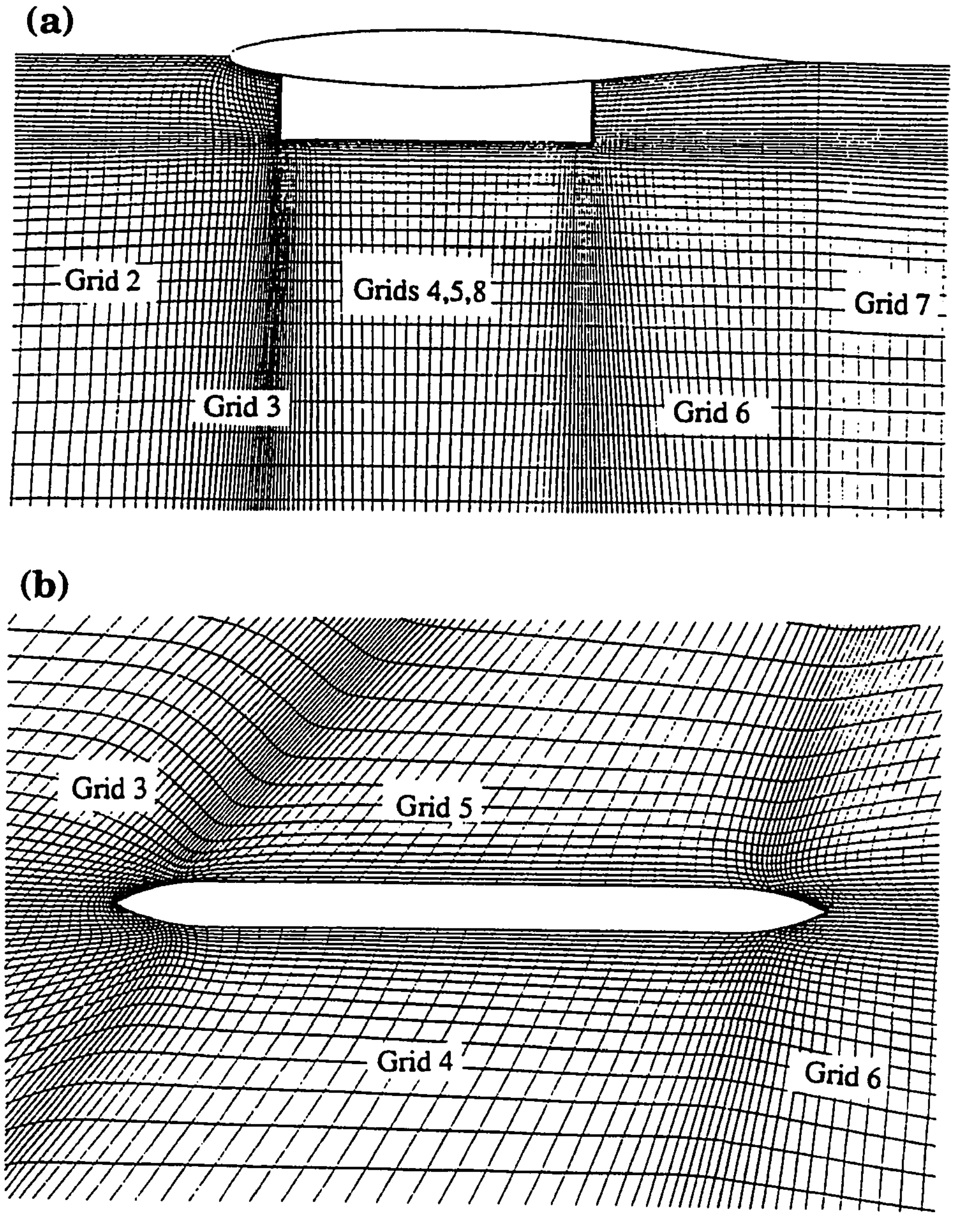


Fig. 5.5 Multiblock-structured grids in the pylon group; (a) side view, (b) enlarged bottom view.

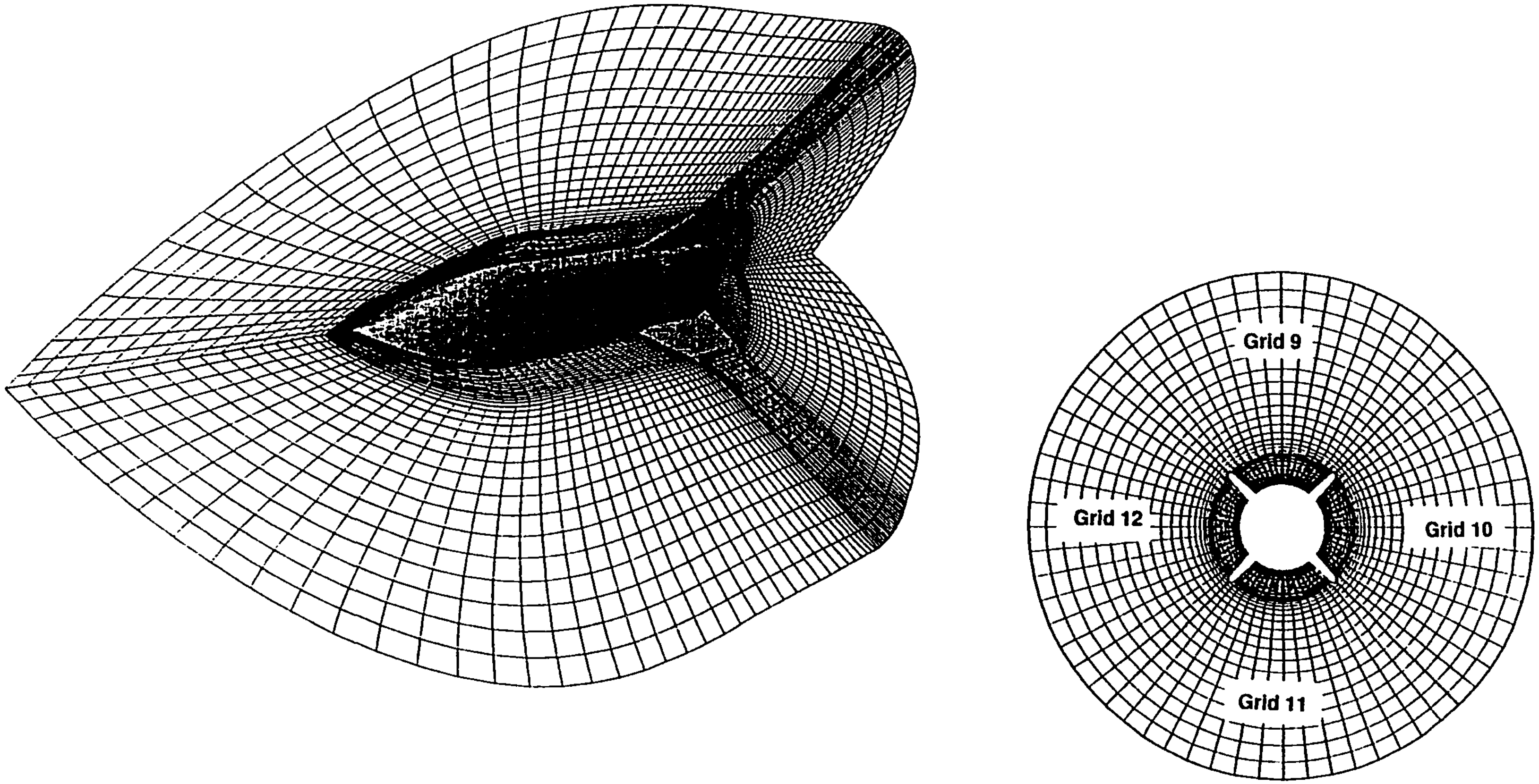


Fig. 5.6 Multiblock-structured grids in the store group.

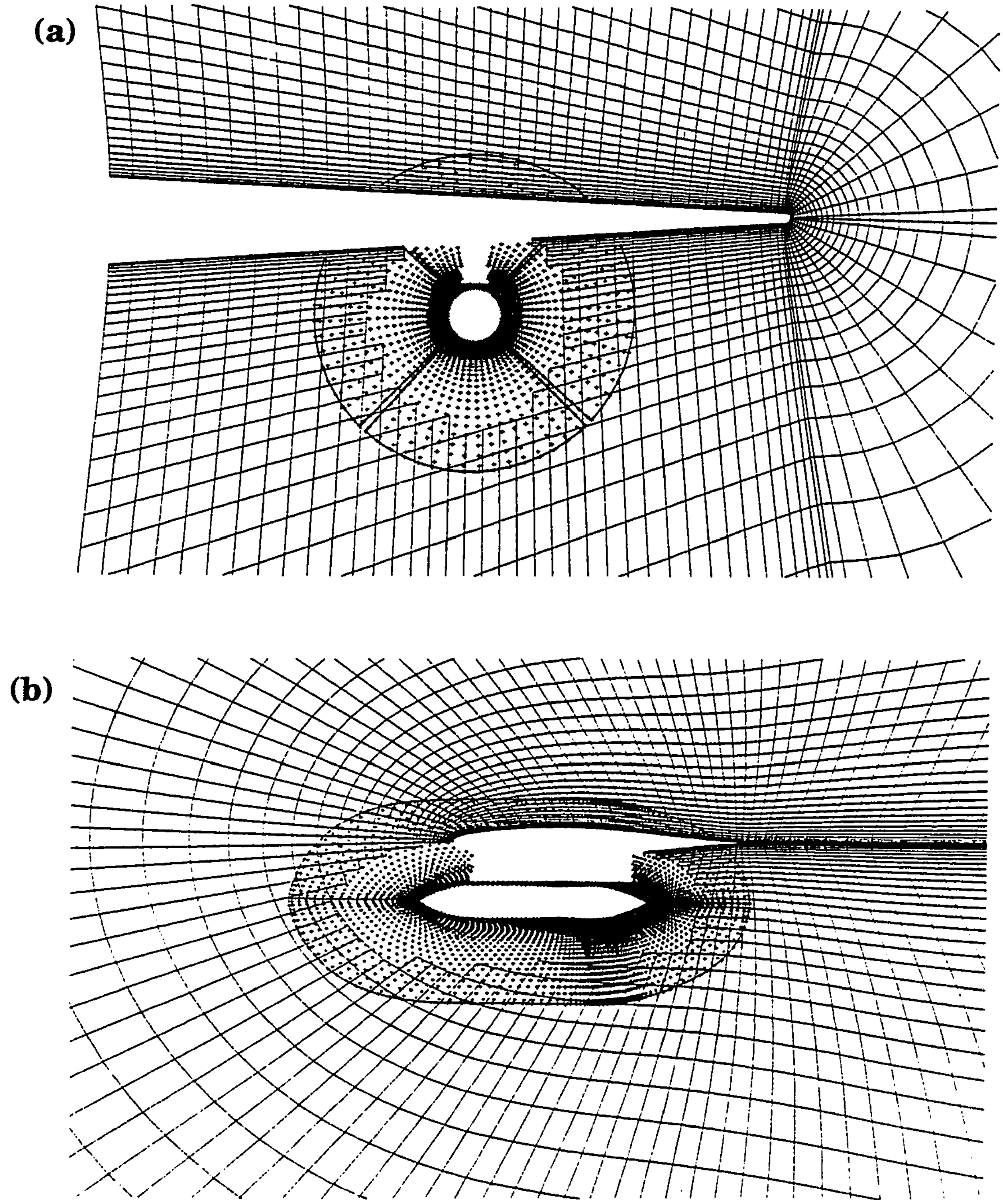


Fig. 5.7 Overlapped region between the finned store group and the global wing grid; (a) front view, (b) side view. (+) symbols indicate the overset grid.

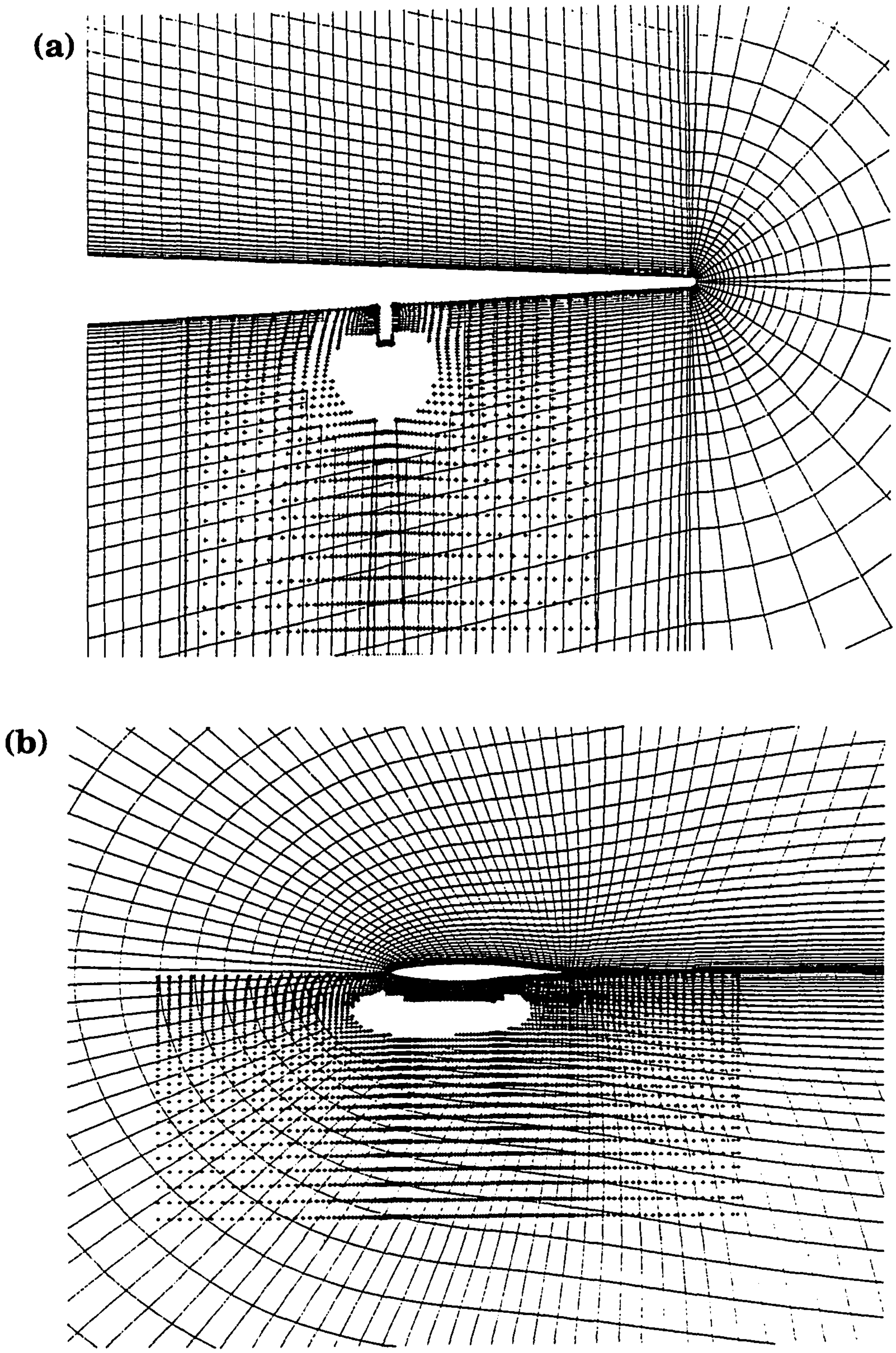


Fig. 5.8 Overlapped region between the pylon group and the global wing grid; (a) front view, (b) side view. (+) symbols indicate the overset grid.

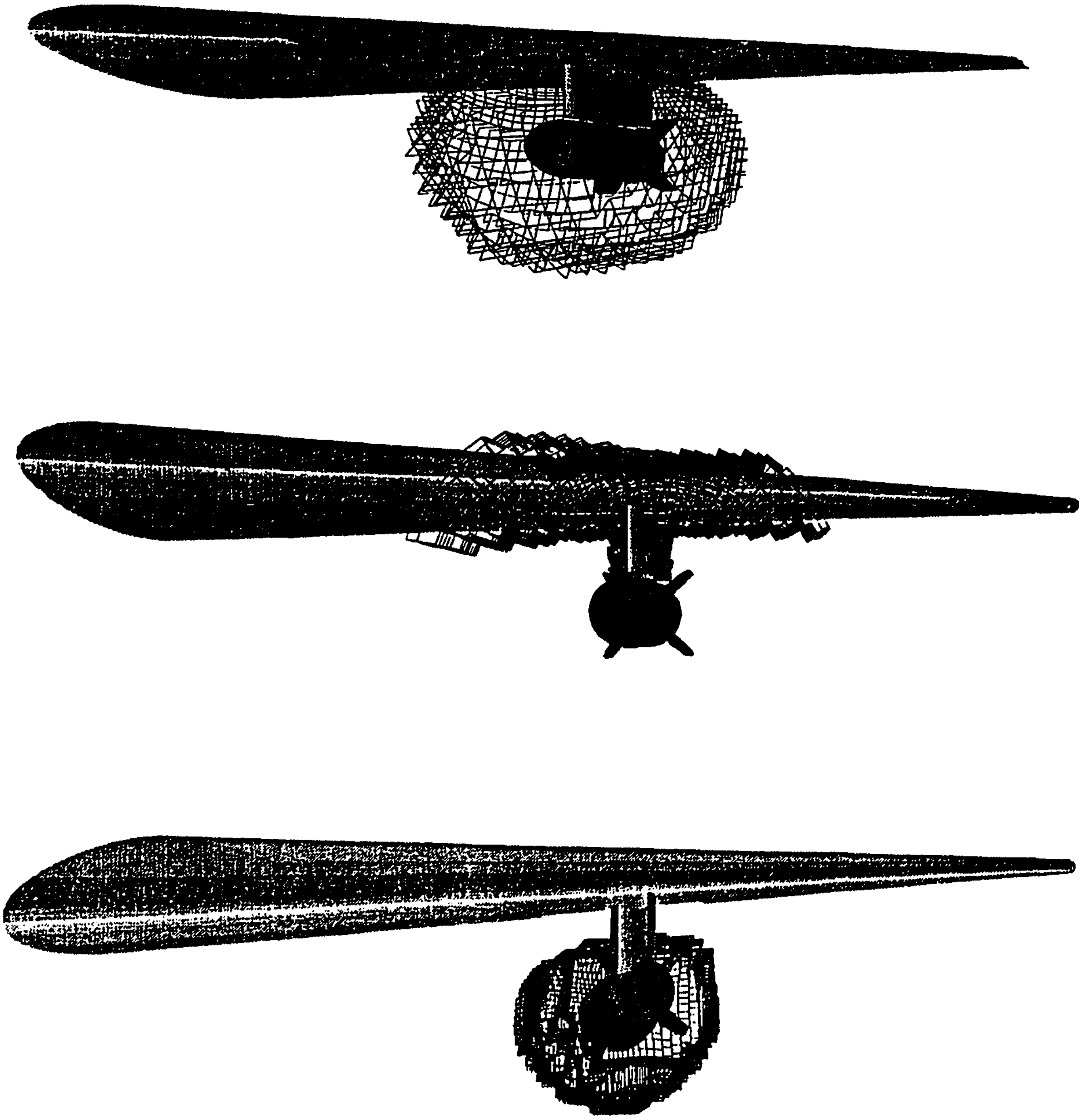
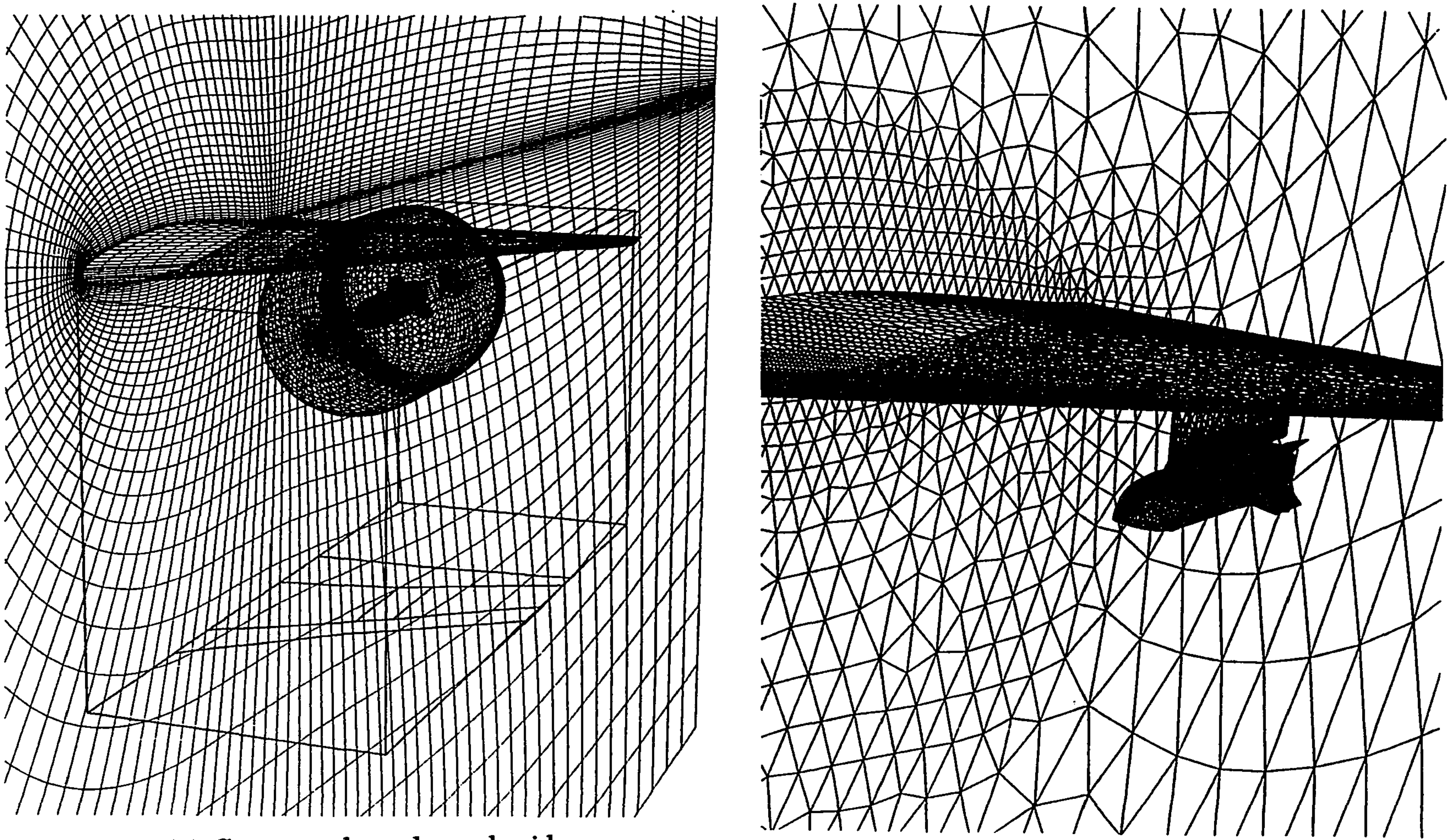


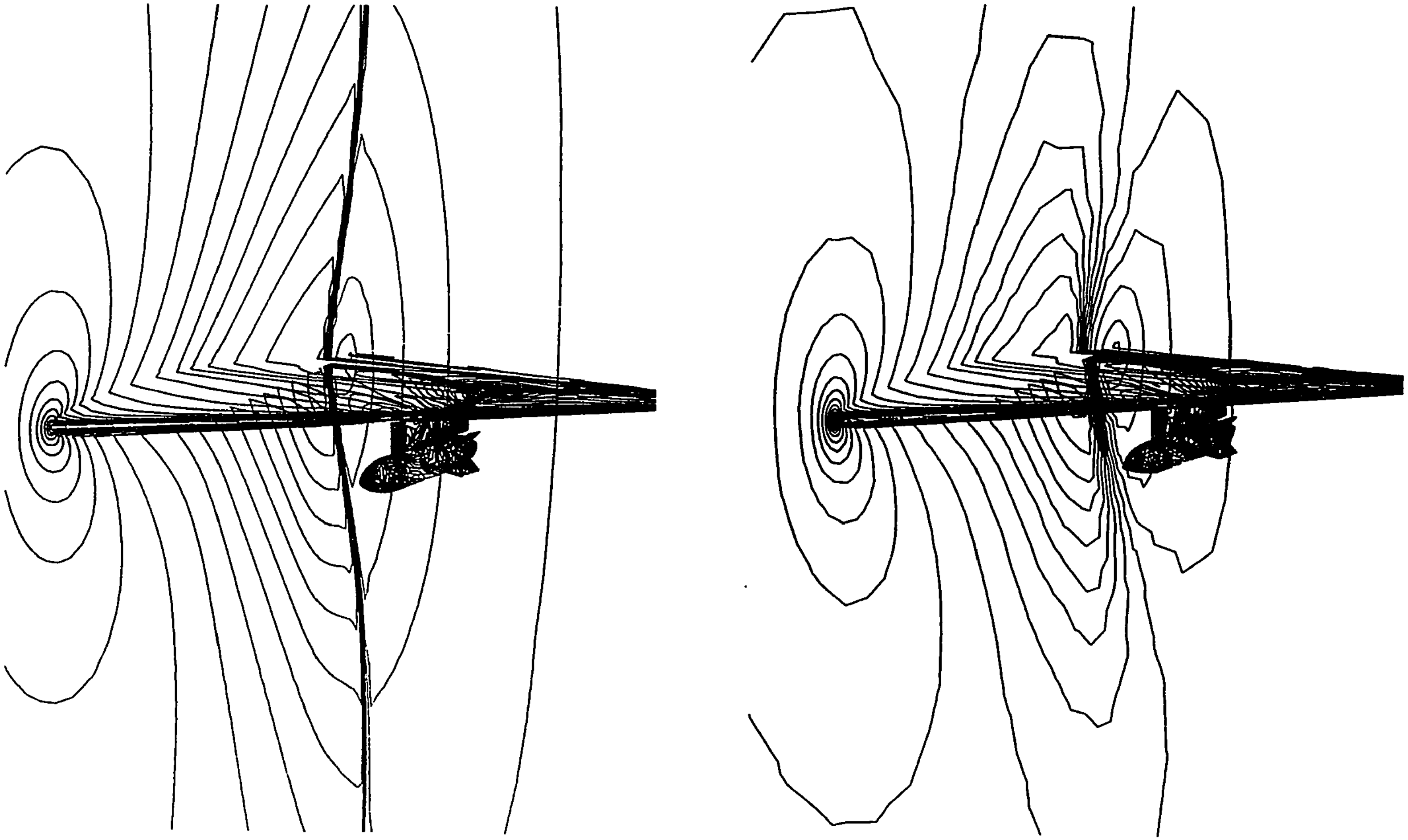
Fig. 5.9 Three-dimensional hole boundaries in the wing, store, and pylon grids.



(a) Structured-overlapped grids

(b) Unstructured mesh

Fig. 5.10 Orientation and relative size of the WPFS grids.

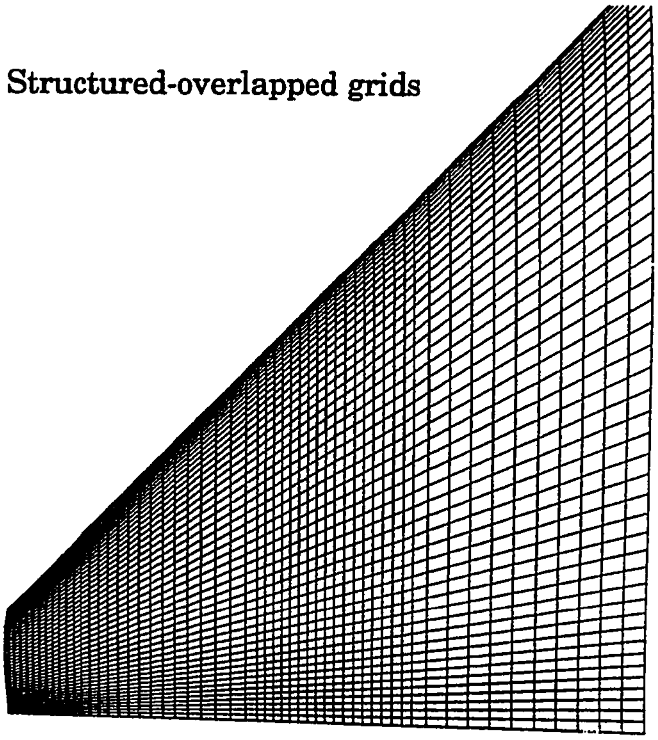


(a) Structured-overlapped grids

(b) Unstructured mesh

Fig. 5.11 Pressure contours on the WPFS surface and plane of symmetry. (M_∞= 0.95, α= 0°)

(a) Structured-overlapped grids



(b) Unstructured mesh

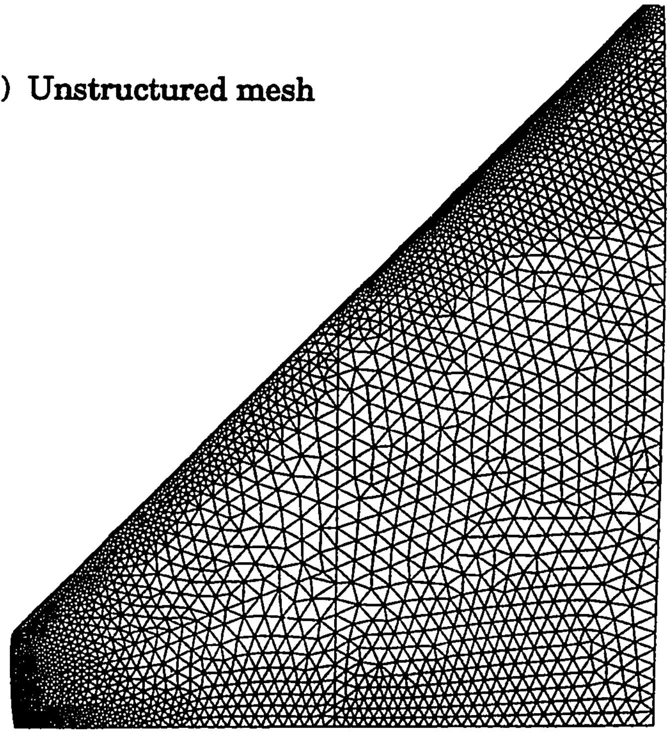
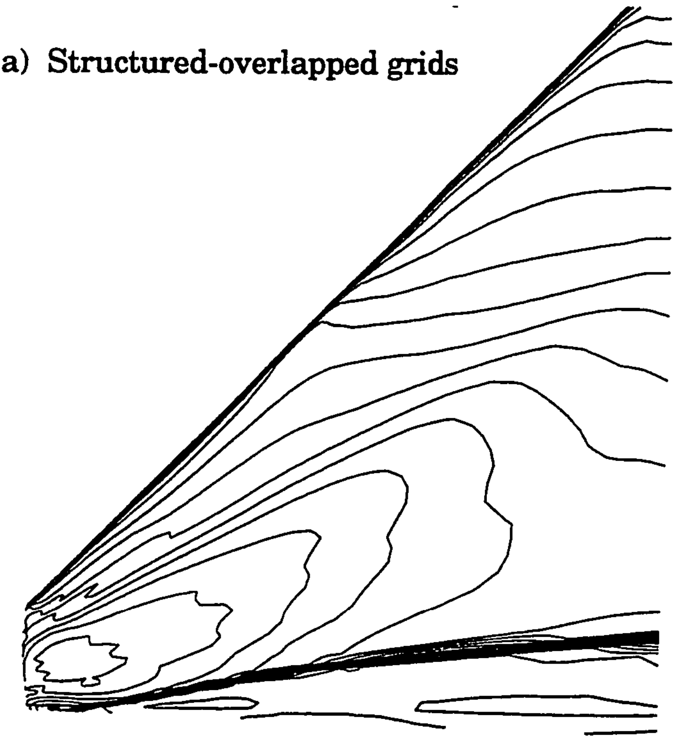


Fig. 5.12 Boundary grids on the upper surface of the wing.

(a) Structured-overlapped grids



(b) Unstructured mesh

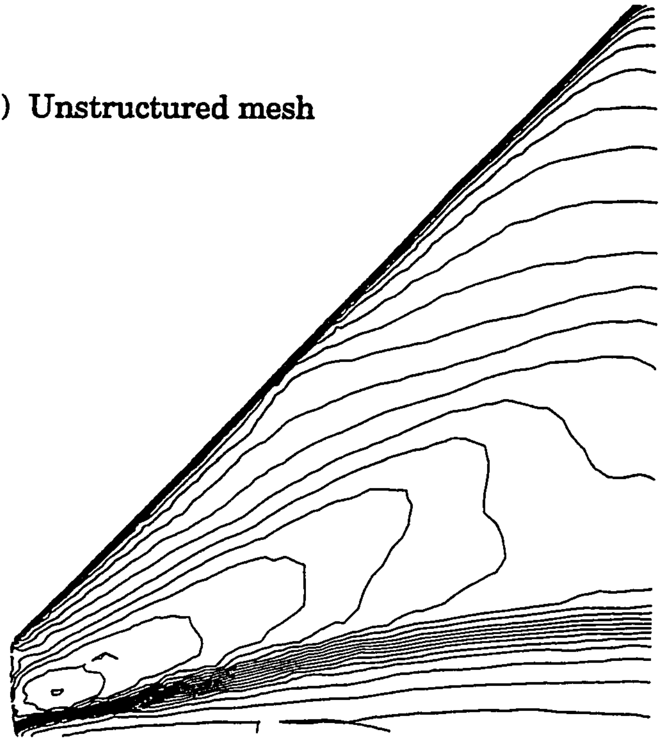
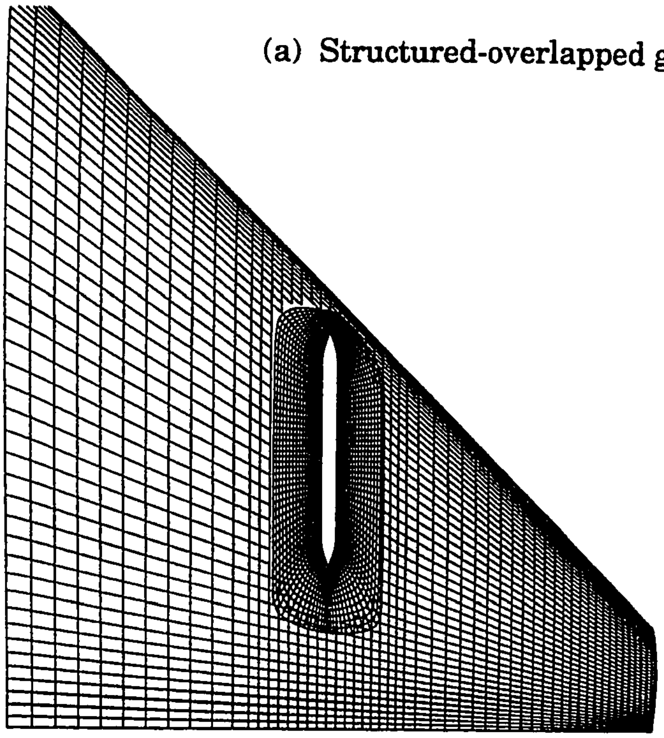
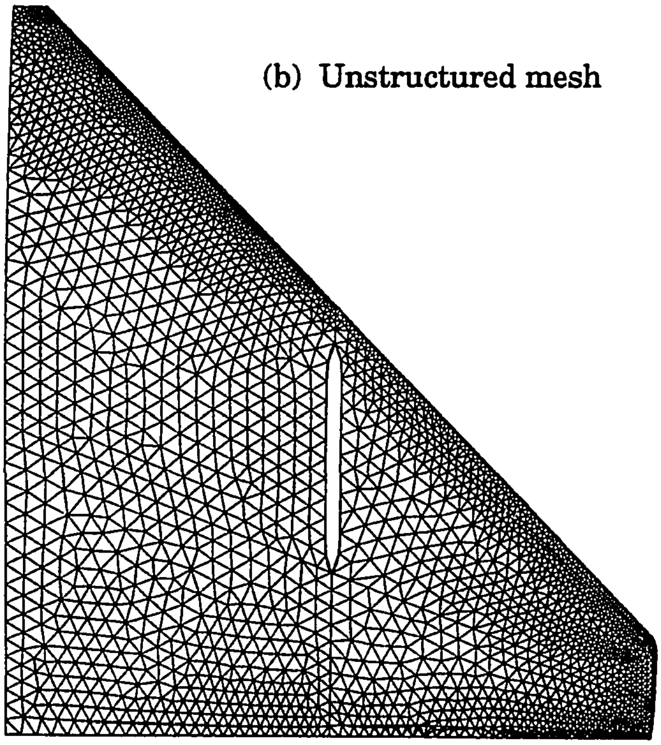


Fig. 5.13 Pressure contours on the upper surface of the wing.

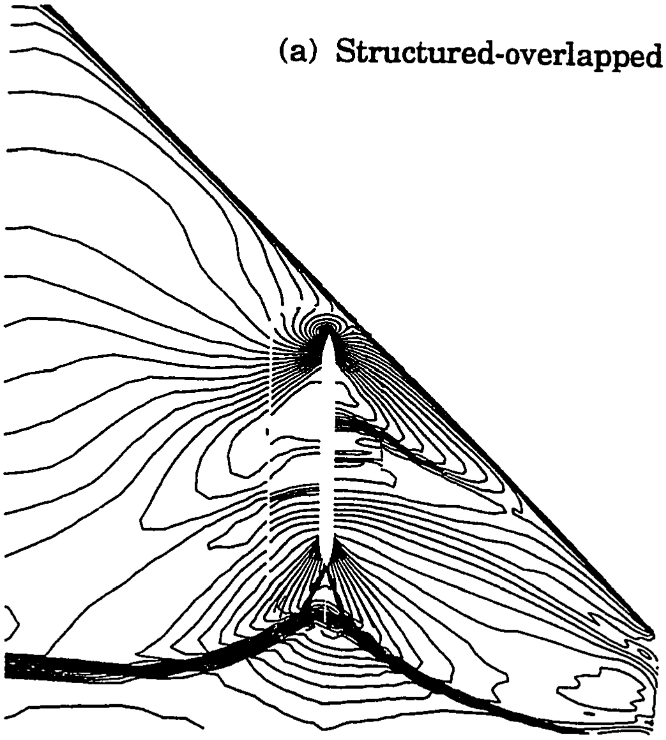


(a) Structured-overlapped grids

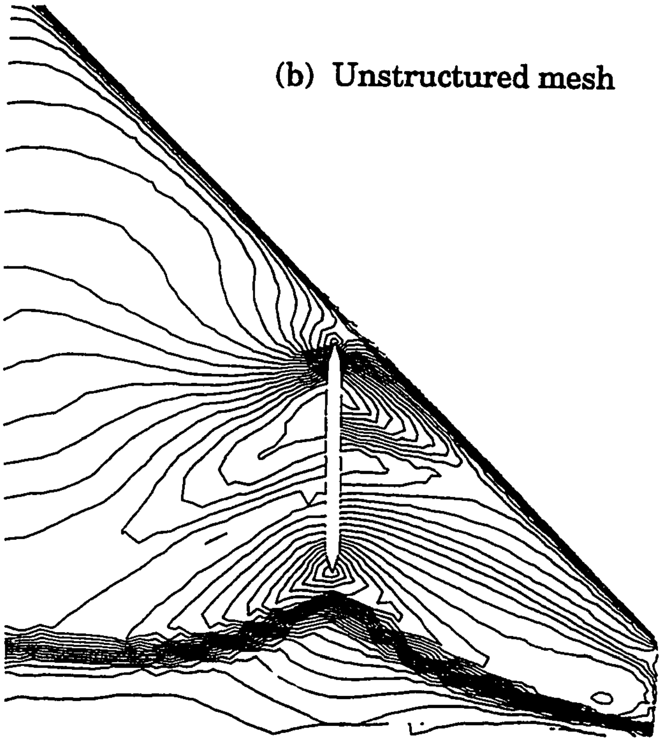


(b) Unstructured mesh

Fig. 5.14 Boundary grids on the lower surface of the wing.



(a) Structured-overlapped grids



(b) Unstructured mesh

Fig. 5.15 Pressure contours on the lower surface of the wing.

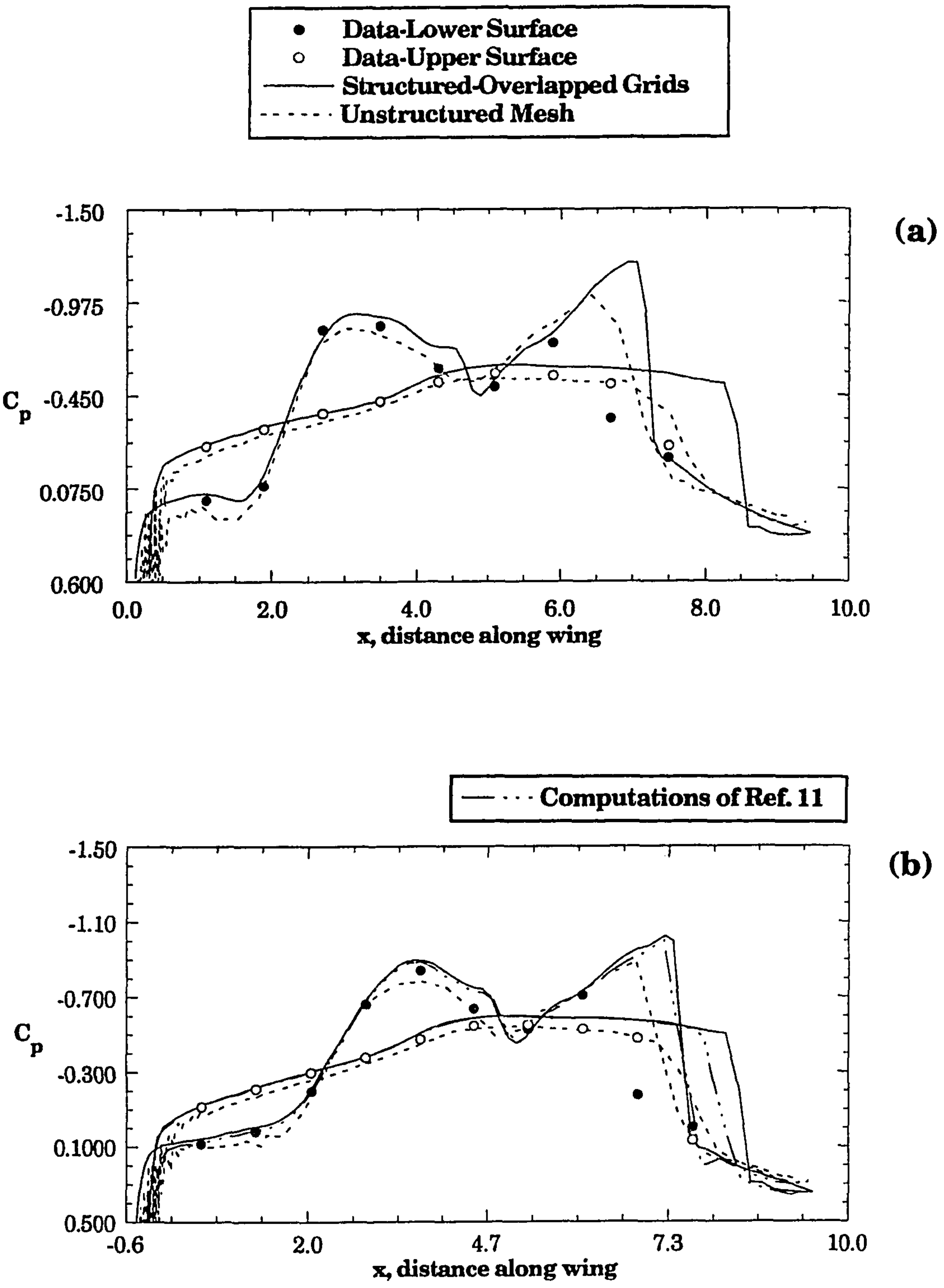


Fig. 5.16 Pressure coefficient comparison at inboard span locations on the wing; (a) z=-0.6, (b) z=-1.2.

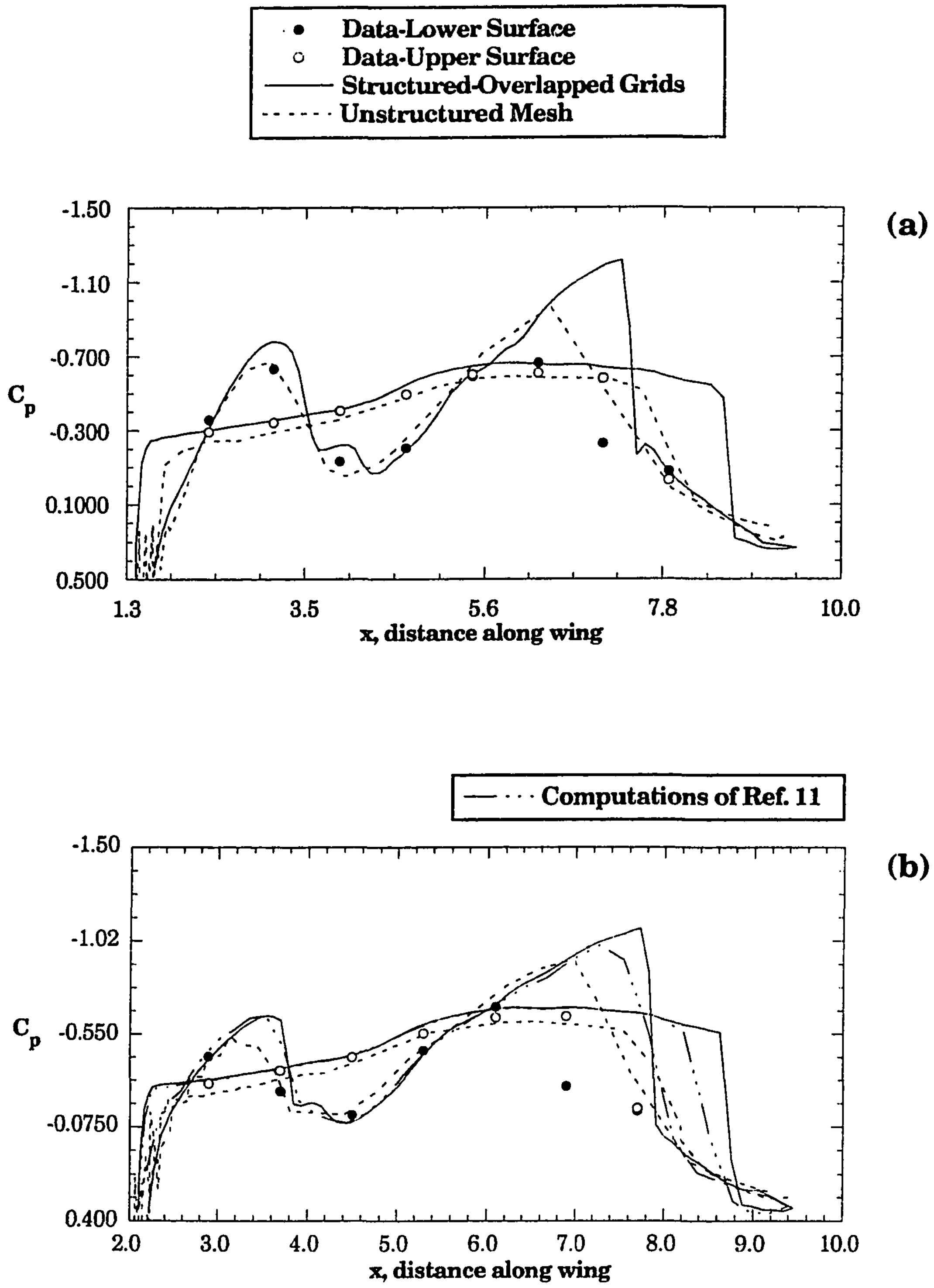


Fig. 5.17 Pressure coefficient comparison at outboard span locations on the wing; (a) z=+0.6, (b) z=+1.2.

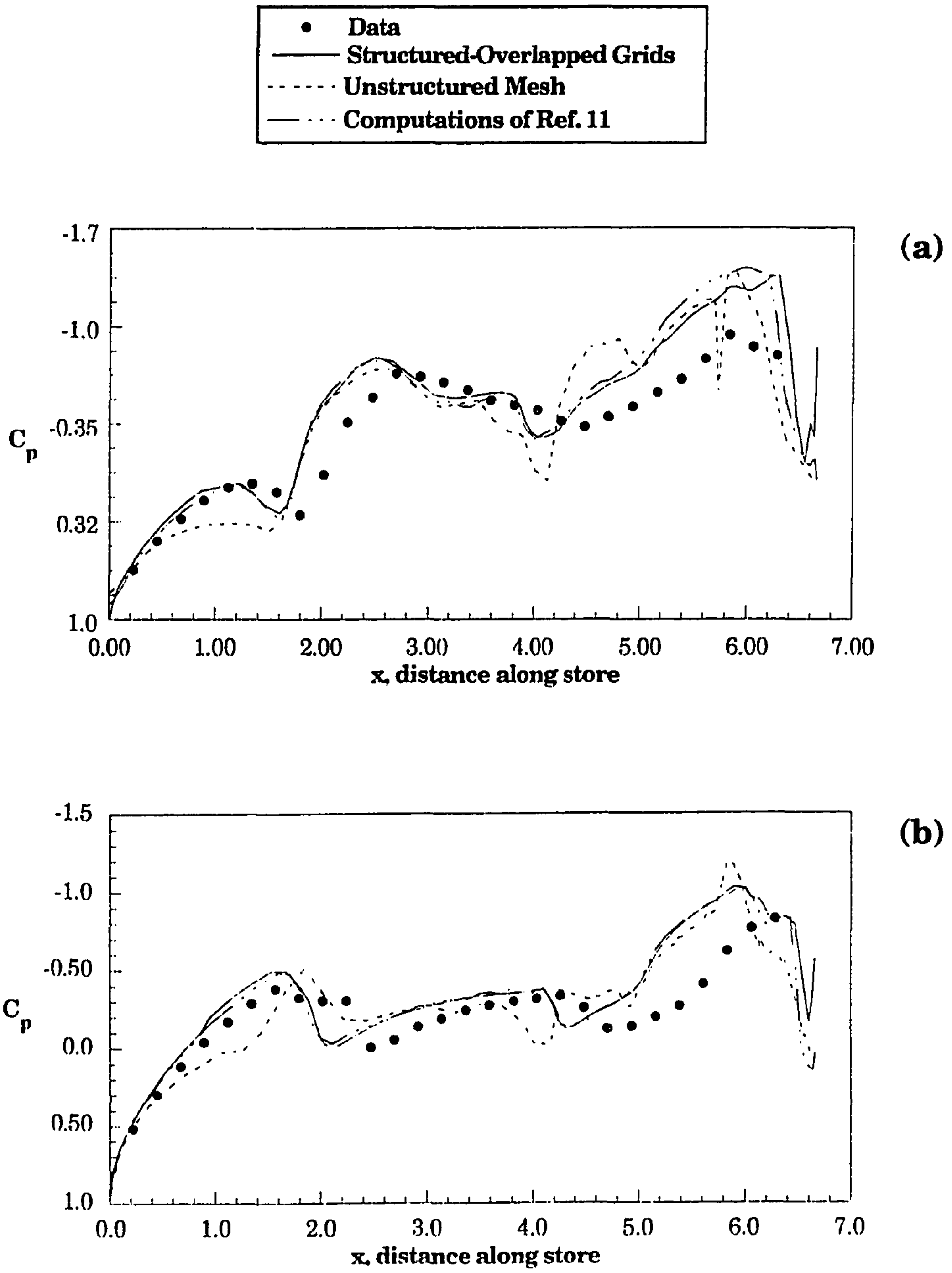


Fig. 5.18 Pressure coefficient comparison for the top/bottom of the store; (a) $\phi=5°$, (b) $\phi=185°$.

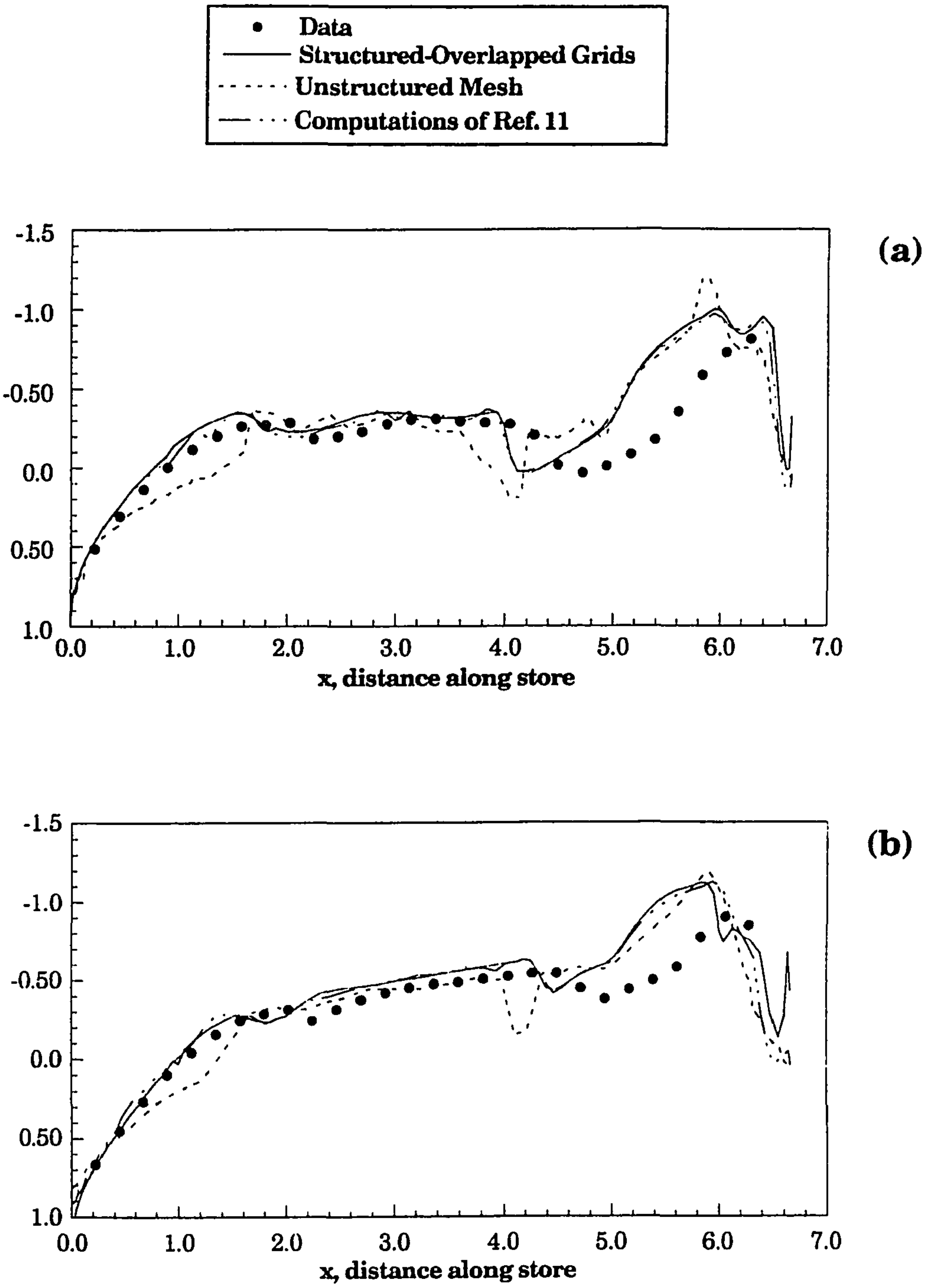


Fig. 5.19 Pressure coefficient comparison for the inboard/outboard sides of the store; (a) ϕ=95°, (b) ϕ=275°.

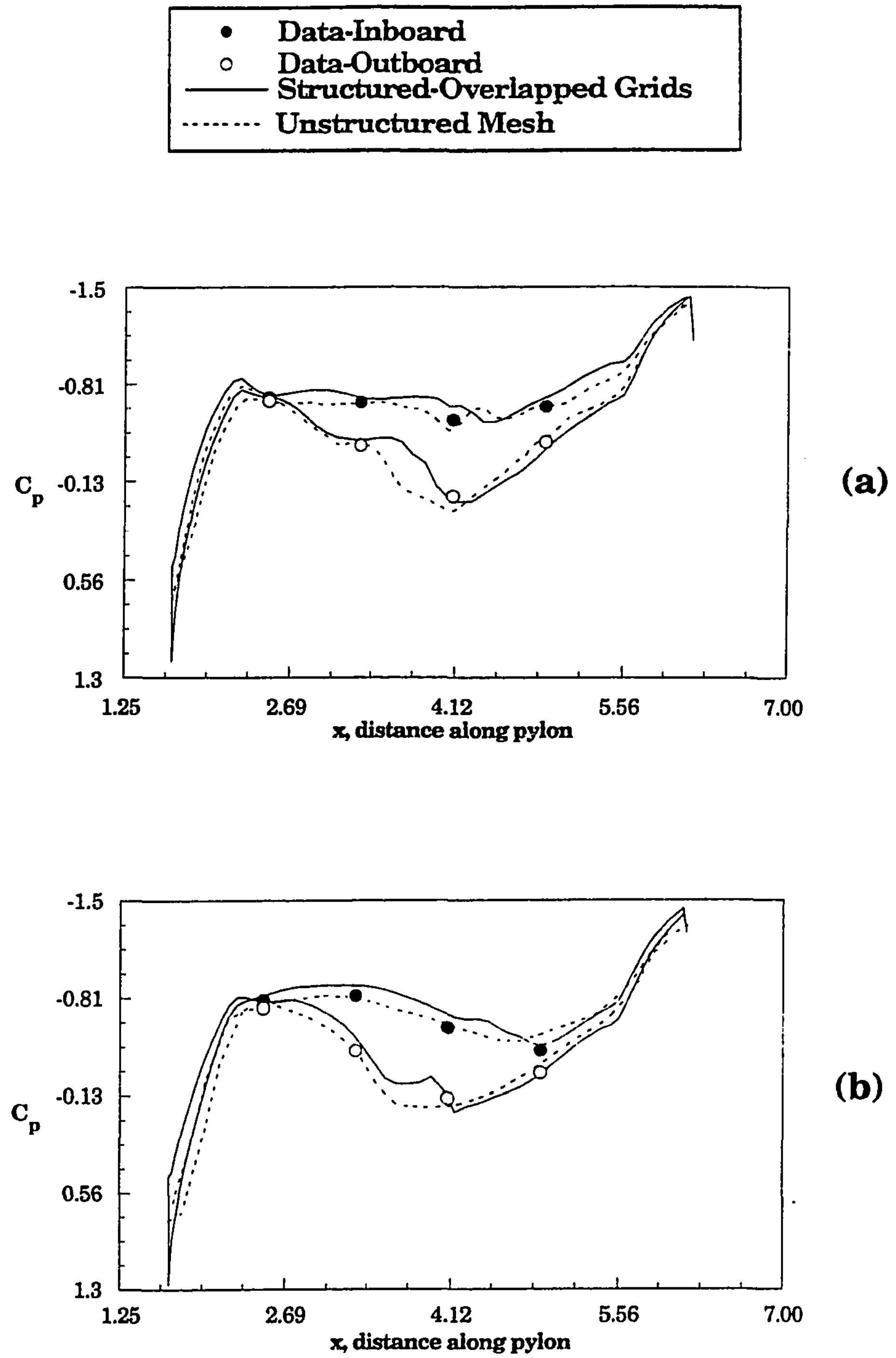
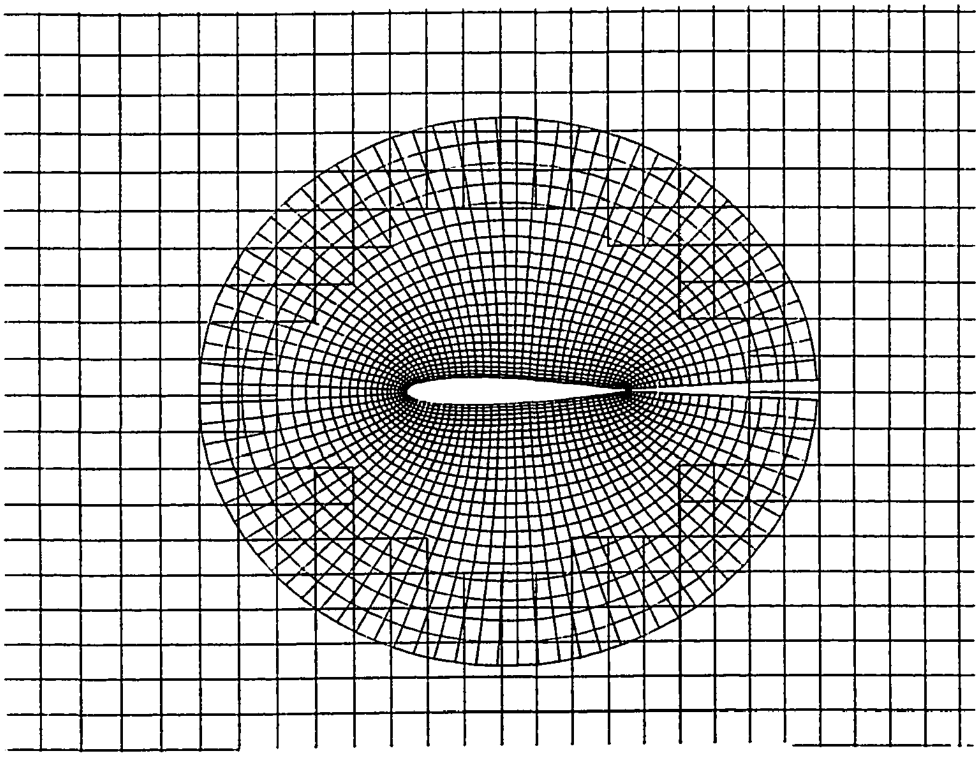
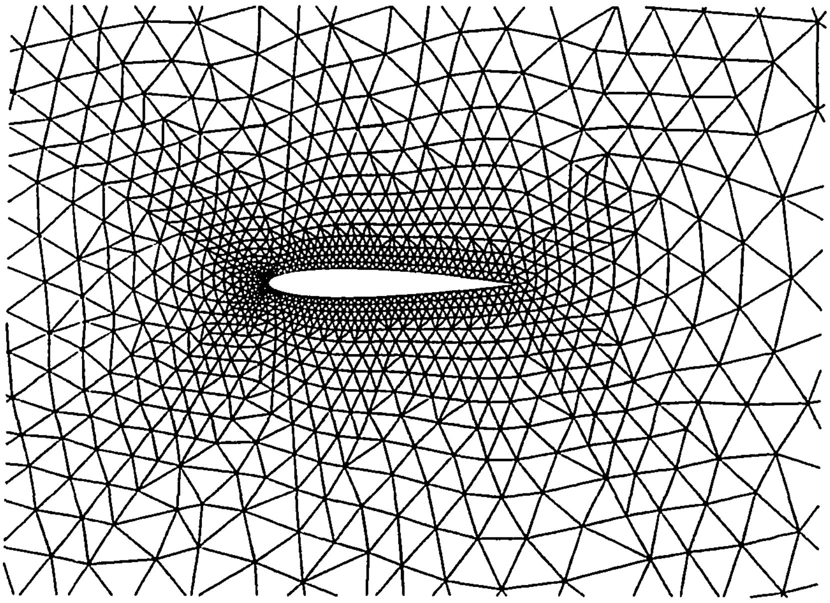


Fig. 5.20 Pressure coefficient comparison for inboard/outboard sides of the pylon; (a) y=0.67, (b) y=1.17.

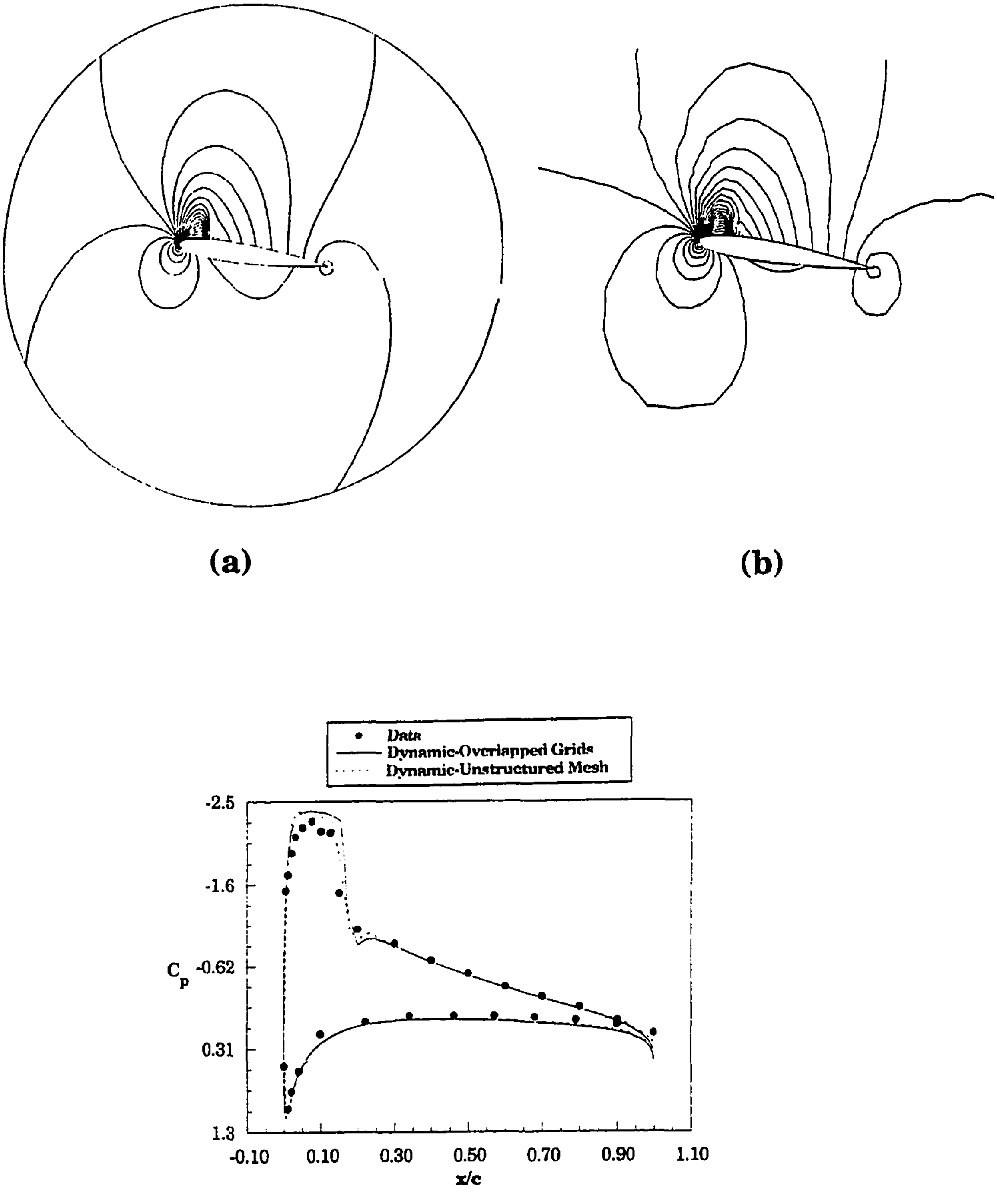


(a) Structured-overlapped grids



(b) Unstructured mesh

Fig. 5.21 Grids for the forced pitching oscillation of a NACA 0012 airfoil.

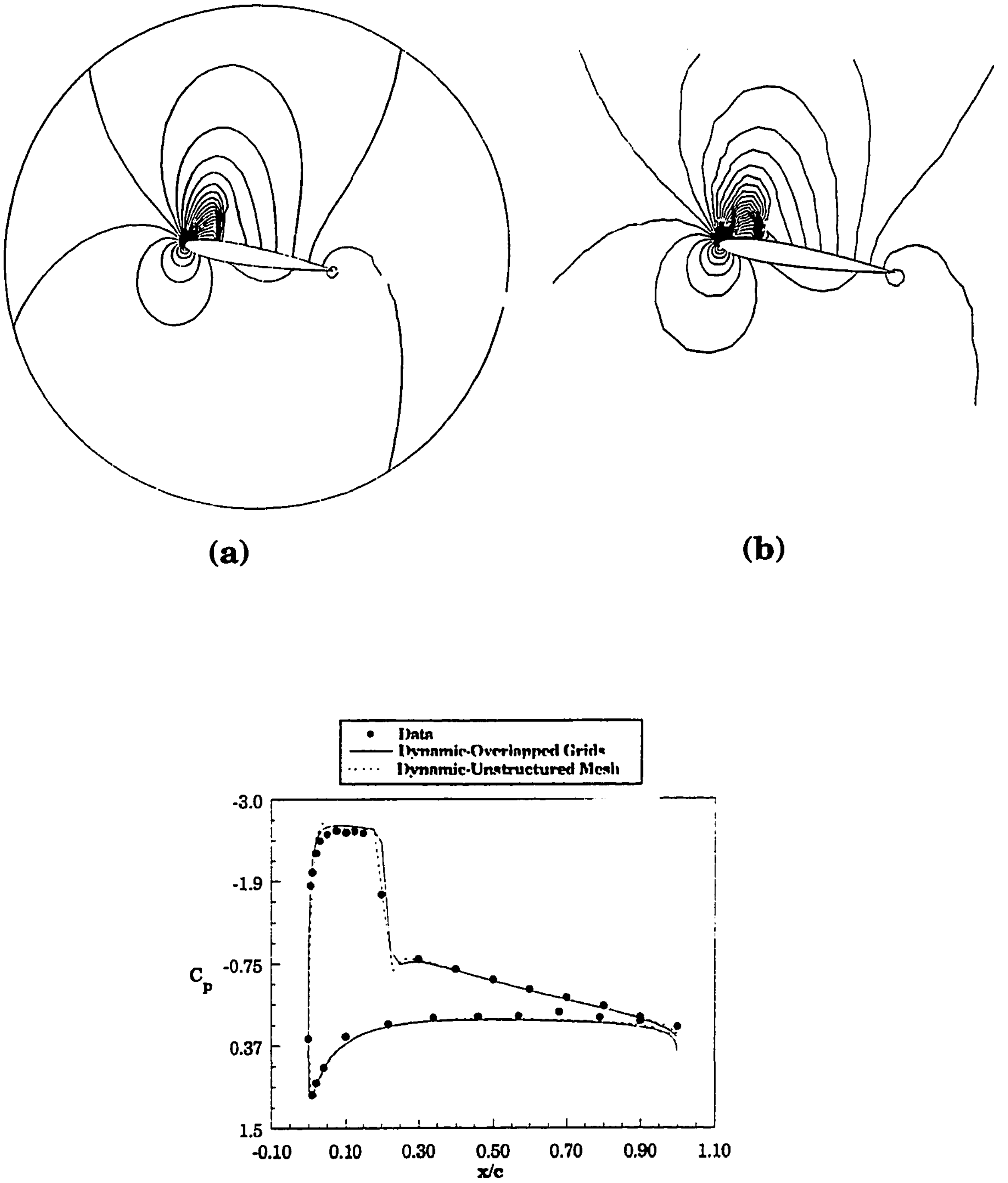


(a)

(b)

(c)

Fig. 5.22 Pressure contours and coefficient comparison for experimental position one (α=5.95°↑, M_∞= 0.6); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.



(a) (b)

(c)

Fig. 5.23 Pressure contours and coefficient comparison for experimental position two (α=6.92°↑); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.

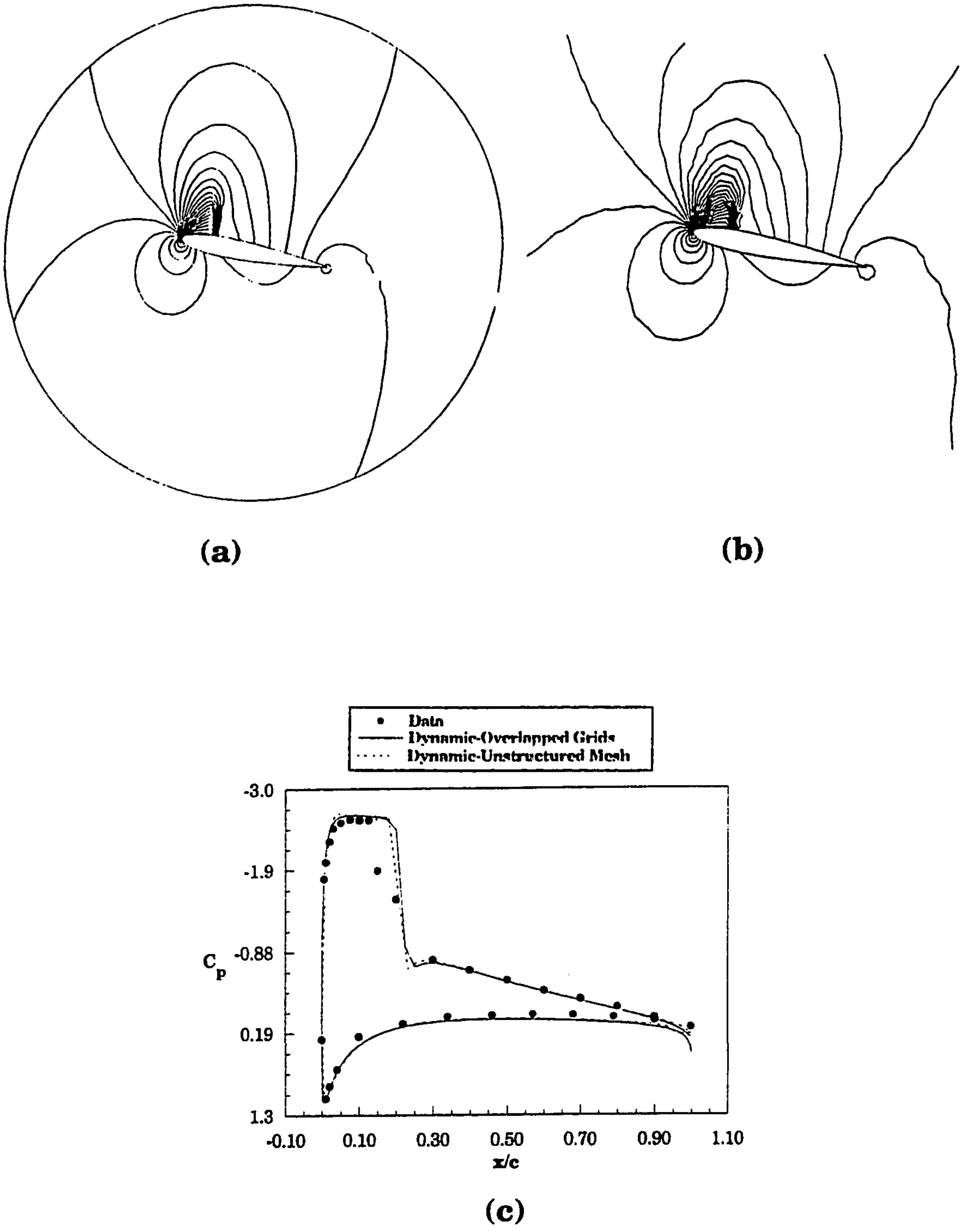
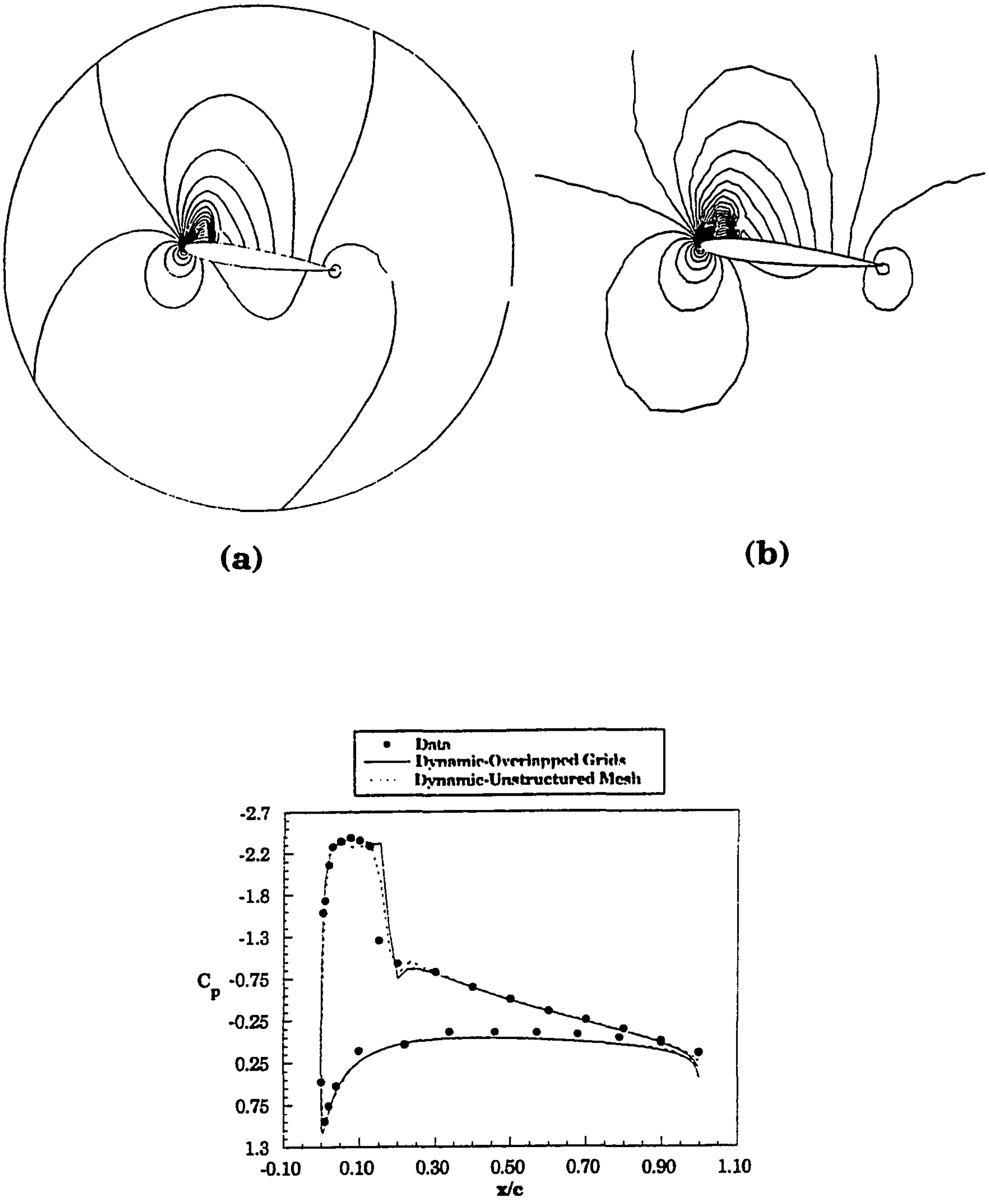


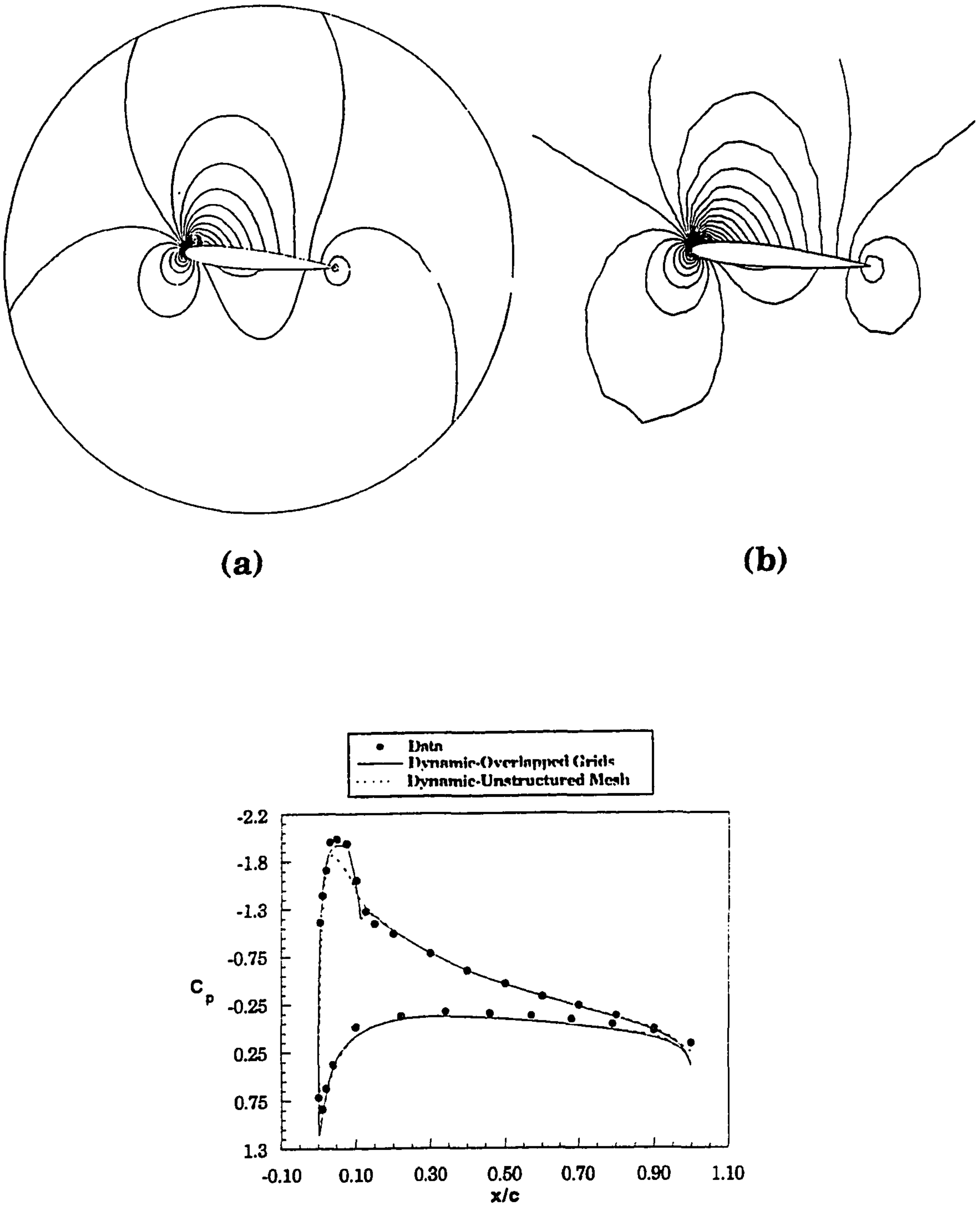
Fig. 5.24 Pressure contours and coefficient comparison for experimental position three (α=6.57°↓); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.



(a) (b)

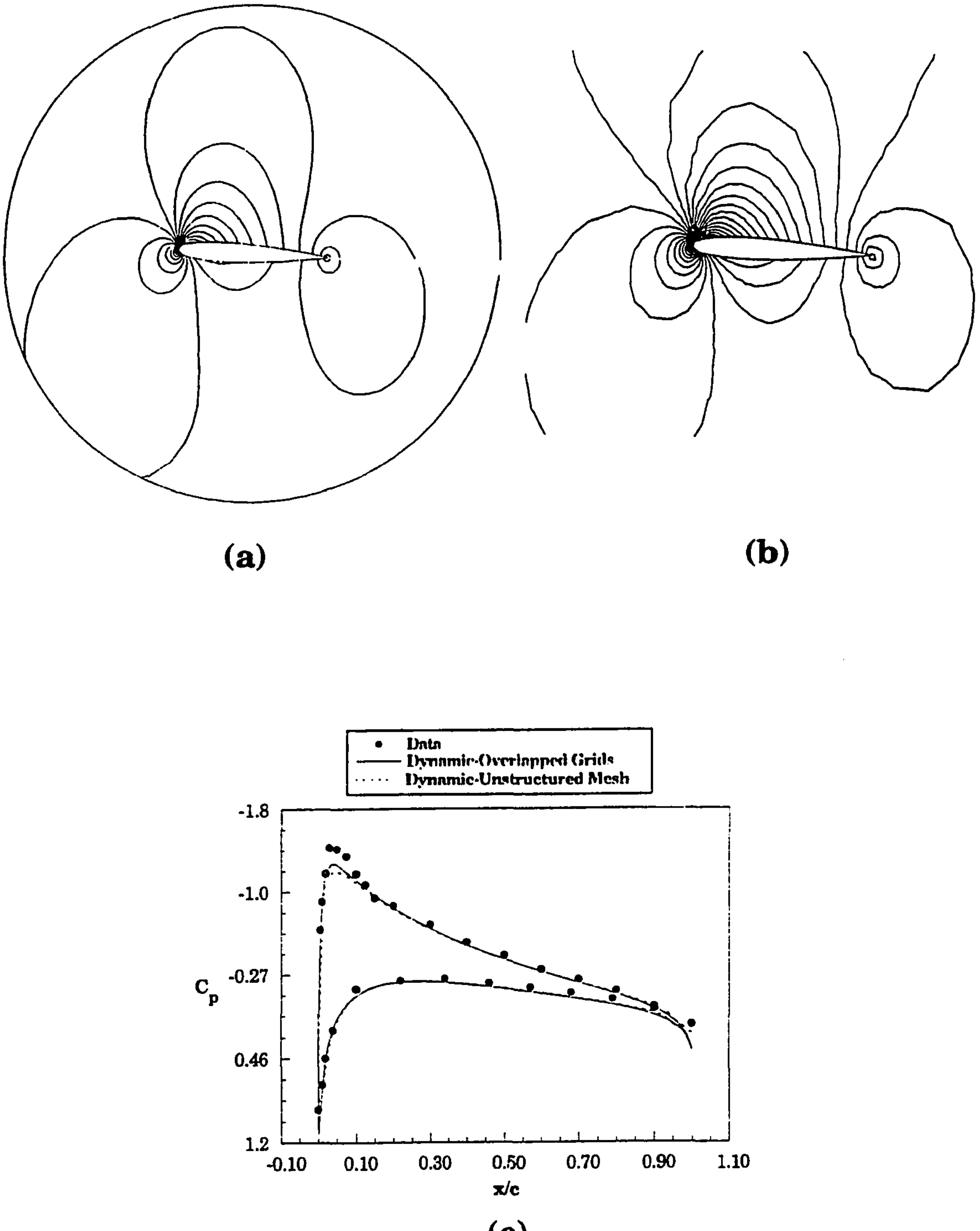
(c)

Fig. 5.25 Pressure contours and coefficient comparison for experimental position four (α=5.11°↓); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.



(c)

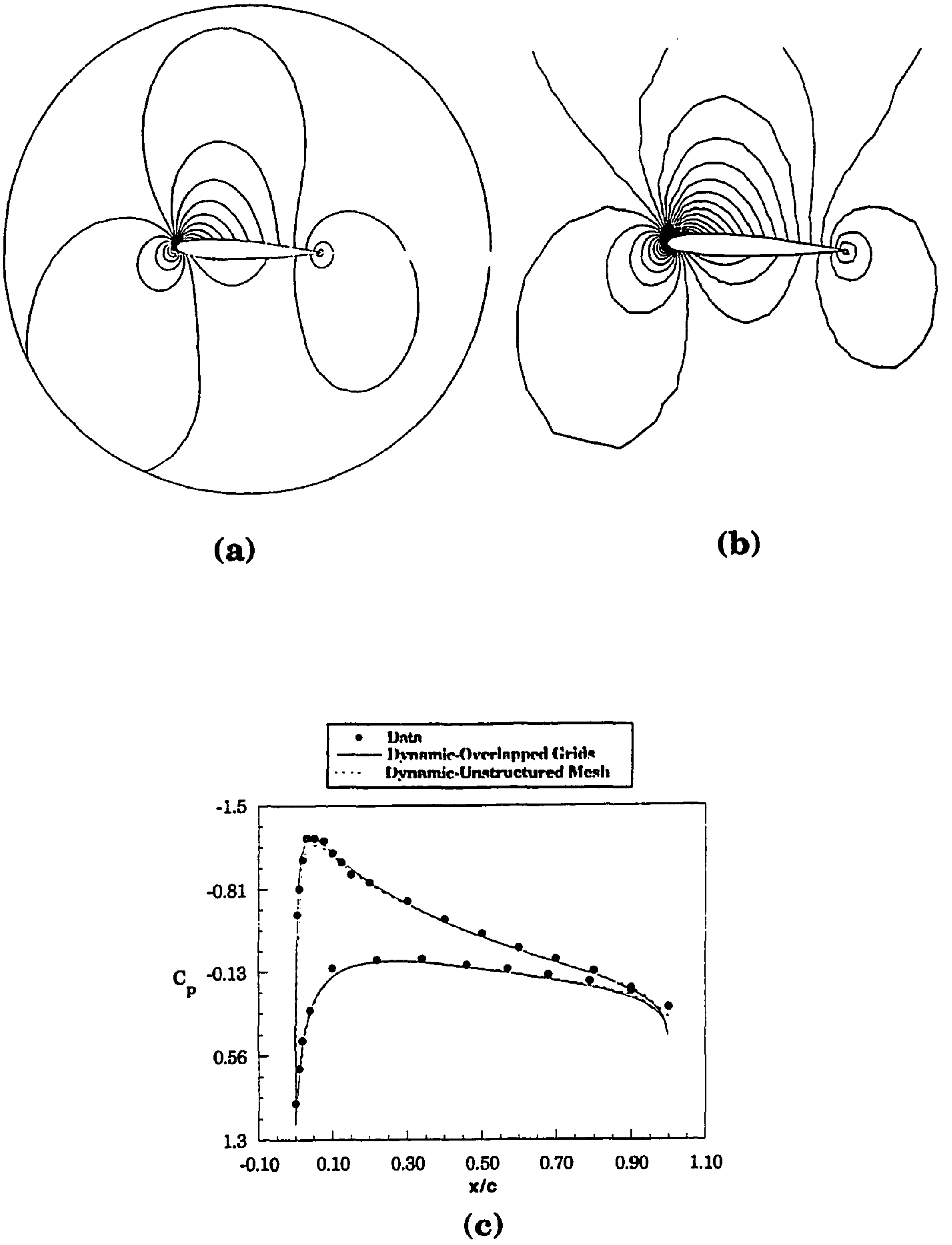
Fig. 5.26 Pressure contours and coefficient comparison for experimental position five (α=3.49°↓); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.



(a) **(b)**

(c)

Fig. 5.27 Pressure contours and coefficient comparison for experimental position six (α=2.43°↓); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.



(a) (b)

(c)

Fig. 5.28 Pressure contours and coefficient comparison for experimental position seven (α=2.67°↑); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.

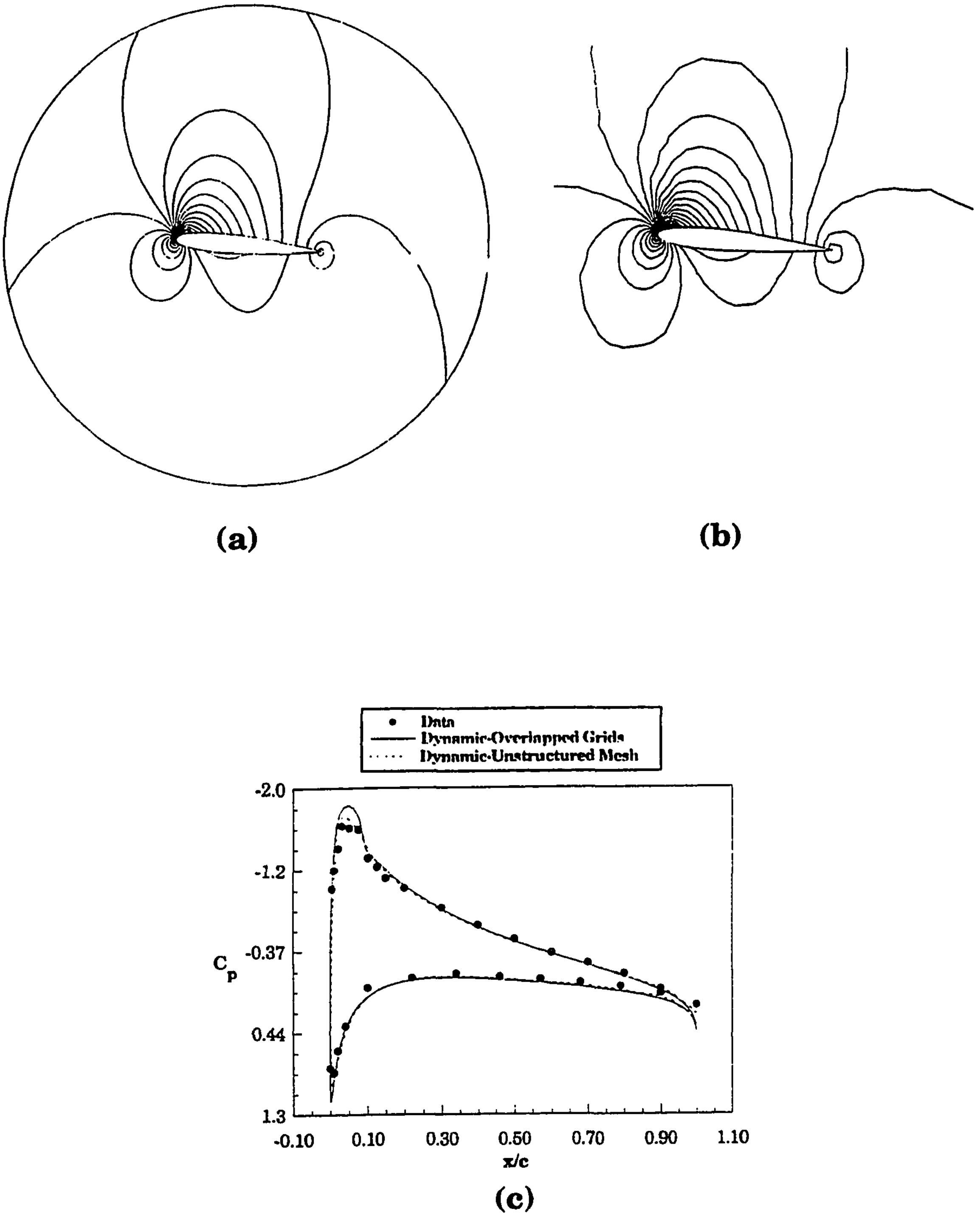


Fig. 5.29 Pressure contours and coefficient comparison for experimental position eight (α=4.28°↑); (a) structured-overlapped grids, (b) unstructured mesh, (c) pressure coefficient comparison.

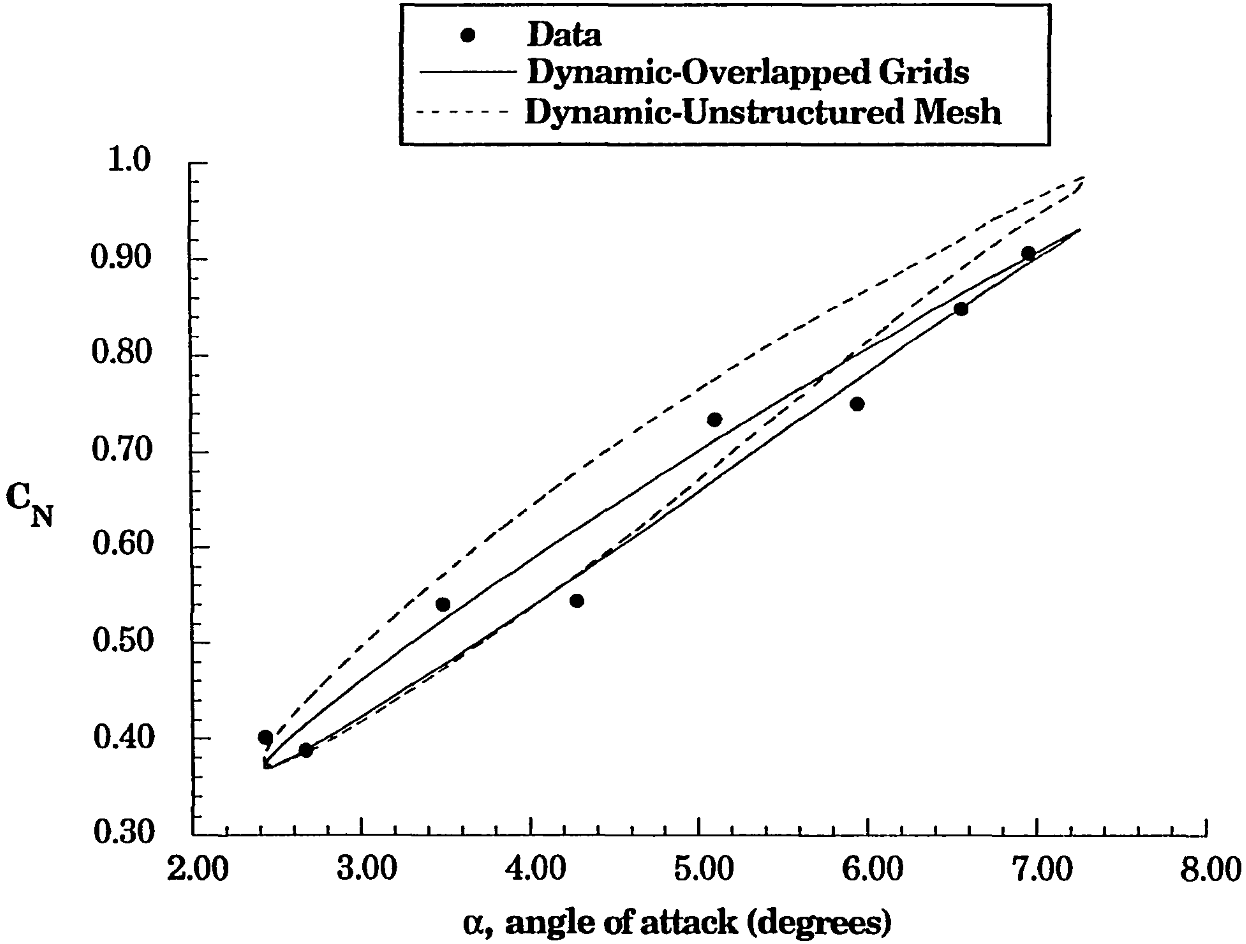
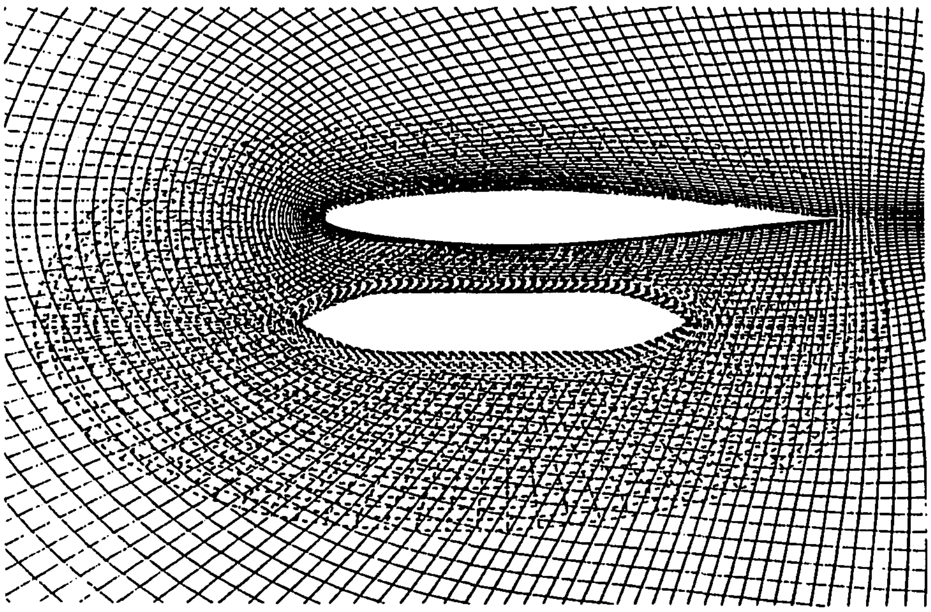
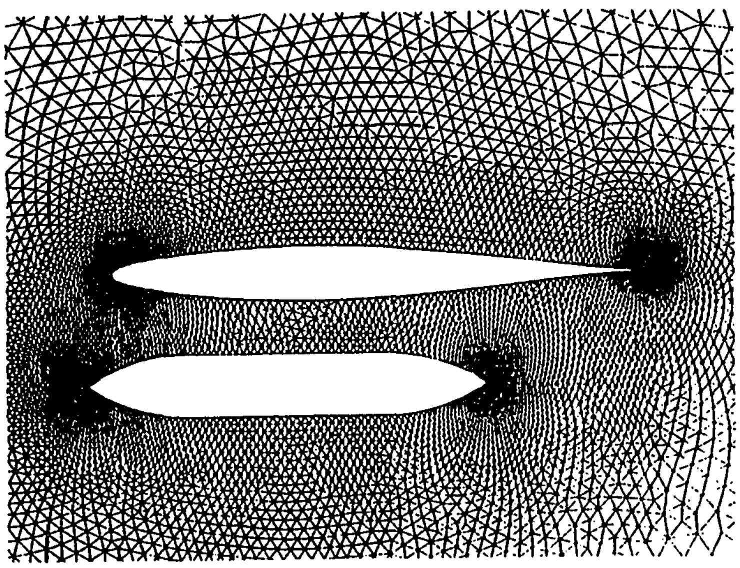


Fig. 5.30 Normal force coefficient against angle of attack.

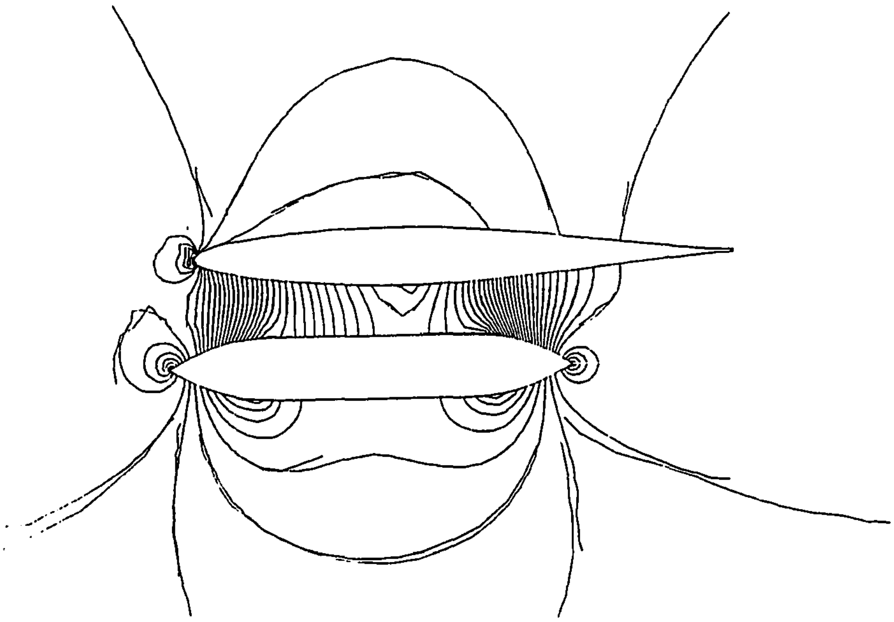


(a) Structured-overlapped grids

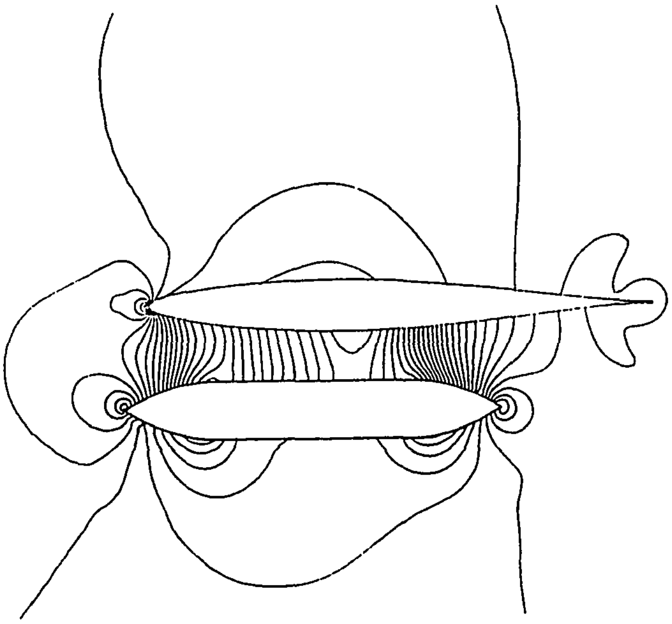


(b) Unstructured mesh

Fig. 5.31 Grids for the two-dimensional airfoil/store separation sequence.

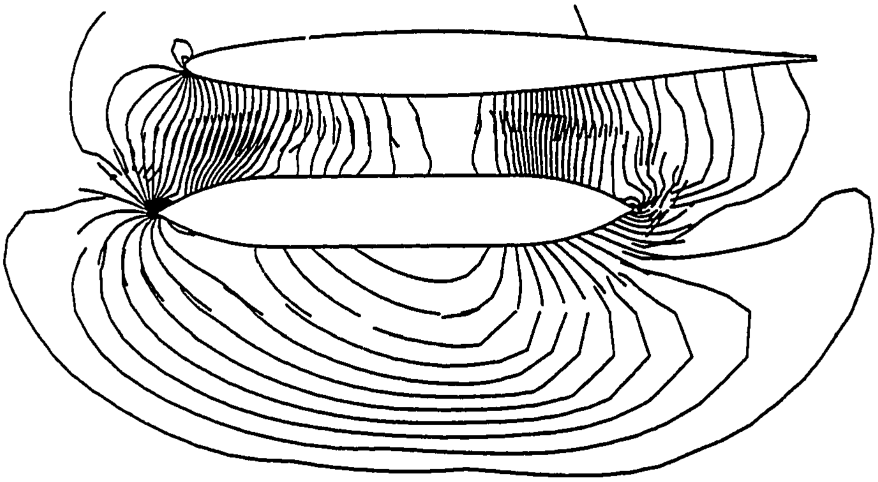


(a) Structured-overlapped grids

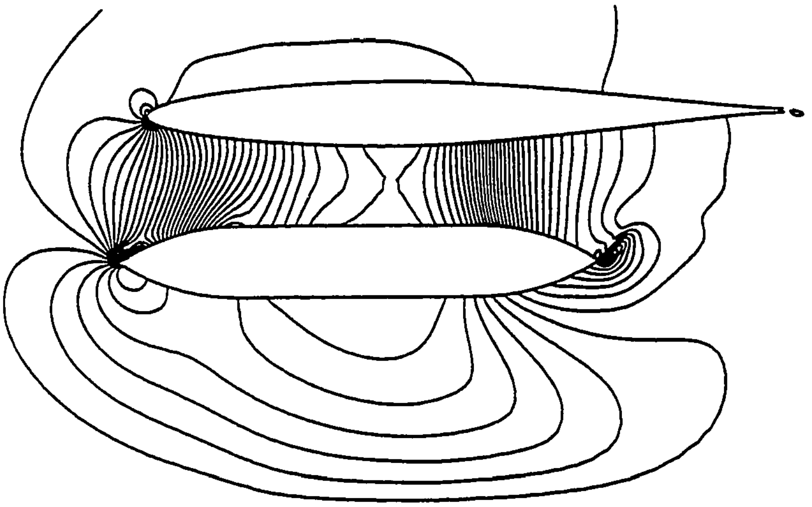


(b) Unstructured mesh

Fig. 5.32 Steady off-surface pressure contours about the airfoil/store. (M_∞= 0.3, α= 0°)

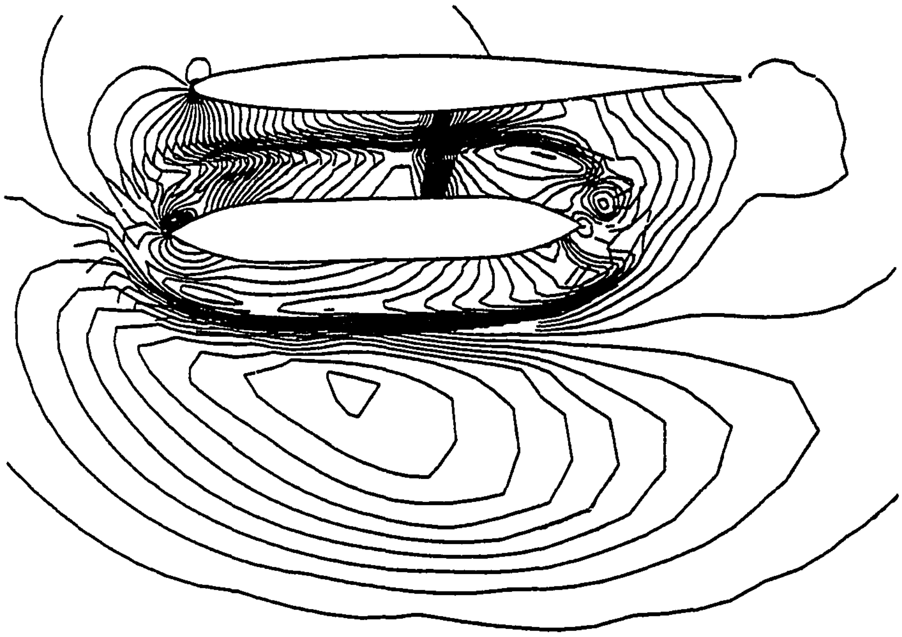


(a) Structured-overlapped grids

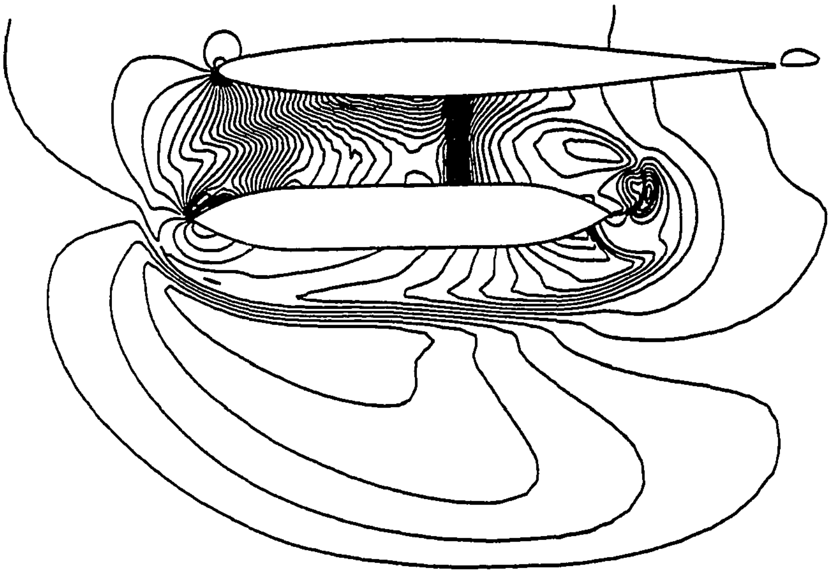


(b) Unstructured mesh

Fig. 5.33 Off-surface pressure contours about the airfoil/store for selected position one.

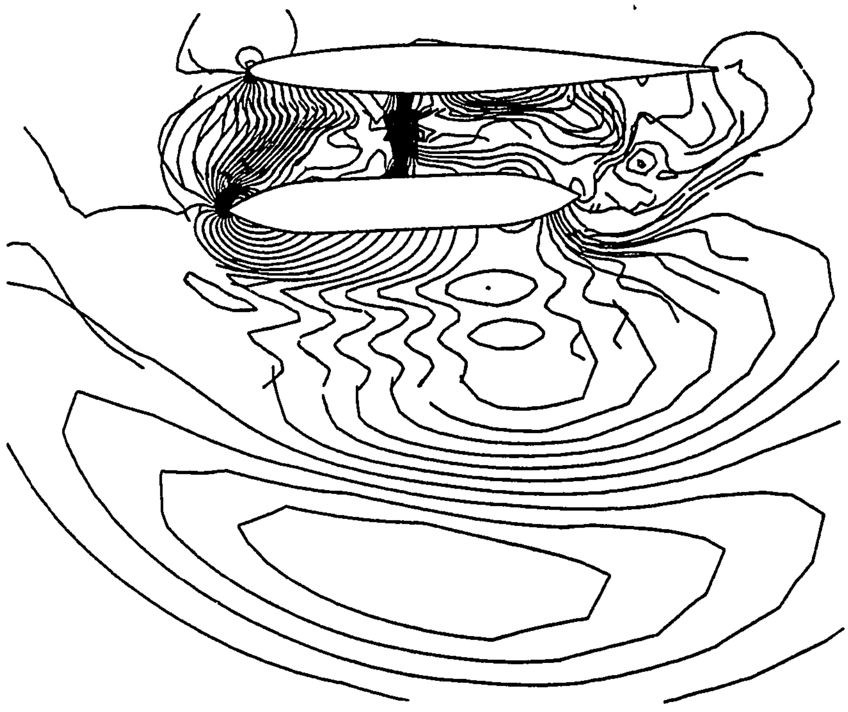


(a) Structured-overlapped grids

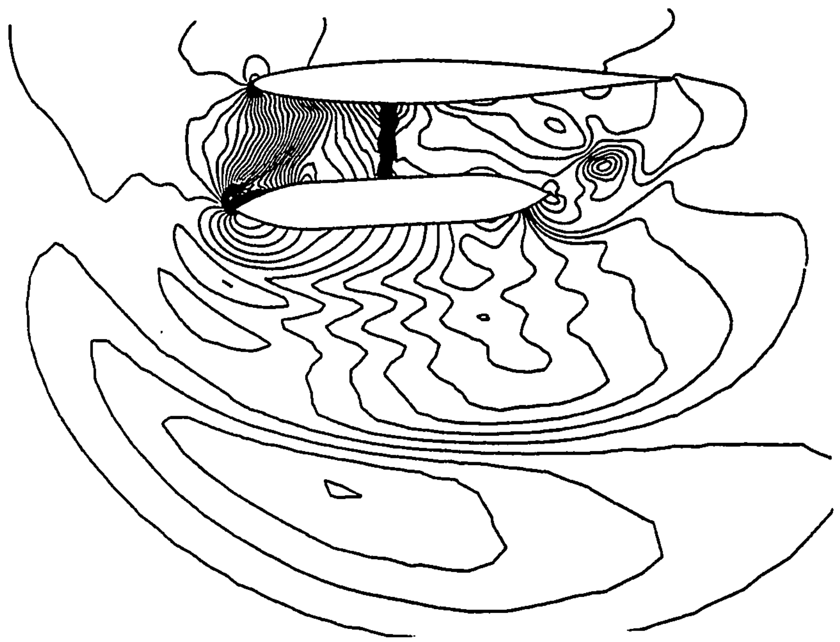


(b) Unstructured mesh

Fig. 5.34 Off-surface pressure contours about the airfoil/store for selected position two.



(a) Structured-overlapped grids



(b) Unstructured mesh

Fig. 5.35 Off-surface pressure contours about the airfoil/store for selected position three.

Chapter 6

Conclusions and Recommendations

Two CFD methodologies for treating the steady flow past complex three-dimensional configurations as well as unsteady moving boundary problems were evaluated. This comparative study was deemed timely and complementary to existing work in this research area. Other studies of this nature have been performed by Lijewski [7] for the multiblocked and the structured-overlapped grid schemes, and by Ghaffari [84] for the multiblocked and the unstructured grid schemes. It is evident from these studies that the methods which produce the quickest *turn-around* in terms of grid generation overhead time are the structured-overlapped and unstructured grids. These are, subsequently, the techniques used in this study.

It is well known that unstructured grid schemes have more computationally intense requirements in terms of CPU time and memory than do the structured grid schemes. This disparity, however, is reduced when structured-domain decomposition techniques are employed to handle complex or moving boundary configurations. This was illustrated through the simulation of the steady flow about a two-dimensional high-lift multielement airfoil and a three-dimensional wing/pylon/finned store (WPFS) assembly. The accuracy of the steady-state solution obtained from each method was assessed through the comparison of the computed and experimental pressure coefficient distributions on several of the WPFS's components. Good agreement was observed for both schemes with the exception of the present unstructured solution on the store and in shock regions. This first discrepancy was attributed to the misrepresentation of the store's geometry that occurred when the unstructured mesh was generated, and the second due to the diffusive nature of the coarse mesh used in this study.

The performance of the dynamic-overlapped and the dynamic-unstructured grids were evaluated from the unsteady simulation of an airfoil undergoing forced pitching oscillation and an aerodynamically determined airfoil/store separations sequence. The instantaneous pressure coefficient distributions about the oscillating airfoil were compared with experimentally measured data at eight positions. Once again, good agreement was observed. A qualitative comparison was shown for the airfoil/store separation. This illustrated that both methods were capable of resolving the unsteady aerodynamic interference which may occur between an aircraft and a released body. From these simulations it was seen that when the solution is carried out over many iterations, the higher cost per iteration of the unstructured flow solver is amplified. In an attempt to reduce these costs, a very efficient unstructured grid adaptation procedure was adopted.

It was demonstrated in the present study that both the structured-overlapped and the unstructured grid schemes yielded flow solutions of comparable accuracy for steady, *inviscid* CFD simulations. Moreover, favorable agreement continued to be observed between the two grid schemes when each was applied to *two-dimensional* unsteady flow problems. The present study also indicated that, overall, the structured-overlapped scheme was slightly more CPU efficient that the unstructured approach. Recommendations for future work consists of: (*i*) evaluating the performance of each method when inviscid simulations of *three-dimensional* unsteady moving boundary configurations are involved, and (*ii*) evaluate the performance of two-dimensional steady and unsteady *viscous* simulations using each method. These recommendations are currently attainable with the structured-overlapped grid schemes, but with not the unstructured grid schemes. Only recently has unstructured two-dimensional viscous grid generation become possible, with much work still needed for the reliable incorporation of turbulence models into these flow solvers.

References

1. Nishida, B.A., and Bencze, D.P, "A Multiblock/Multigrid Euler Analysis of a Propfan Transport with Wing-Mounted Nacelles Including Slipstream Effects," AIAA Paper No. 91-0706, Jan. 1991.

2. Connell, S.D., and Holmes, D.G., "A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations," AIAA Paper No. 93-3339, 1993.

3. Fouladi, K.,"Viscous Flow Past a Nacelle Isolated and in Proximity of a Flate Plate," AIAA Paper No. 92-2723, June 1992.

4. Stanniland, D.R., Baxendale, A.J., and Carr, K.G., "The Use of CFD Methods to Aid the Aerodynamic Design and Analysis of Wing/Pylon/Store Installations," Jan. 1991.

5. Arabshahi, S., and Whitfield, D.L., "A Multiblock Approach to Solving the Three-Dimensional Unsteady Euler Equations About a Wing-Pylon-Store Configuration," AIAA Paper No. 89-3401, Aug. 1989.

6. Lijewski, L.E., "Transonic Euler Solutions on Mutually Interfering Finned Bodies," AIAA Journal, Vol. 28, No. 6, June 1990, pp. 982-988.

7. Lijewski, L.E., "Transonic Euler Solutions of a Wing-Pylon-Finned Body Configuration Using Blocked and Overlapping Grid Schemes," AIAA Paper No. 91-2854, Aug. 1991.

8. Baysal, O., Fouladi, K., and Lessard, V.R., "Multigrid and Upwind Viscous Flow Solver on 3-D Overlapped and Embedded Grids," AIAA Journal, Vol. 29, No. 4, April 1991, pp. 903-910.

9. Yen, G.W., and Baysal, O., "Computing Unsteady High-Speed Flows Past an Oscillating Cylinder Near a Vertical Wall," AIAA Paper No. 92-4653, Aug. 1992.

10. Meakin, R.L., "Computations of the Unsteady Flow About a Generic Wing/Pylon/Finned-Store Configuration," AIAA Paper No. 92-4568, Aug. 1992.

11. Parikh, P., Pirzadeh, S., and Frink, N.T., "Unstructured Grid Solutions of a Wing/Pylon/Store Configuration Using VGRID3D/USM3D," AIAA Paper No. 92-4572, Aug. 1992.

12. Newman III, J.C., and Baysal, O., "Transonic Solutions of a Wing/Pylon/Finned Store Using Hybrid Domain Decomposition," AIAA Paper No. 92-4571, Aug. 1992.

13. Noack, R.W., and Bishop, D.G., "A Three Dimensional Delaunay Unstructured Grid Generator and Floe Solver for Bodies in Relative Motion," AIAA Paper No. 93-3349, July 1993.

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

15. Fouladi, K., and Baysal, O., "Viscous Simulation Method for Unsteady Flows Past Multicomponent Configuration," Journal of Fluids Engineering, Vol. 114, June 1992, pp.161-169.

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

17. Ghaffari, F., Luckering, J.M., Thomas, J.L., Bates, B.L., and Biedron, R.T., "Multiblock Navier-Stokes Solutions About the F/A-18 Wing-LEX-Fuselage Configuration," Journal of Aircraft, Vol. 30, No.3, June 1993.

18. Smith, M.H., Chawla, K., and Van Dalsem, "Numerical Simulation of a Complete STOVL Aircraft in Ground Effect," AIAA Paper No. 91-3293, Sept. 1991.

19. Frink, N.T., "Three Dimensional Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Grids," Ph.D Dissertation, Virginia Polytechnic Institute and State University, Sep. 1991.

20. Meakin, R.L., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper No. 93-3350, July 1993.

21. Jameson, A., and Baker, T.J., "Multigrid Solutions of the Euler Equations for Aircraft Configurations," AIAA Paper No. 84-0093, Jan. 1984.

22. Weatherhill, N.P., and Forsey, C.R., "Grid Generation and Flow Calculations for Complex Aircraft Geometries Using a Multiblock Scheme," AIAA Paper No. 84-1665, June 1984.

23. Thompson, J.F., "Program EAGLE- Numerical Grid Generation System User's Manual," AFATL-TR-87-15, March 1987.

24. Steinbrenner, J., Chawner, J., Fouts, C., and Nguyen, M., "The GRIDGEN 3D Multiple Block Grid Generation System," AFWAL-TR-90-0001, General Dynamics, Fort Worth Division, Jan. 1990.

25. Anderson, W.K., Thomas, J.L., and Whitfield, D.L., "Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations," NASA TM-2829, 1988.

26. Steger, J.L., Dougherty, F.C., and Benek, J.A., "A Chimera Grid Scheme," ASME Mini-Symposium on Advances in Grid Generation (Ed.:K. Ghia, U. Ghia), Houston, Texas, June 1983.

27. Benek, J.A., Buning, P.G., and Steger, J.L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper No. 85-1523, July 1985.

28. Bowyer, A., "Computing Dirichlet Tessellations," The Computer Journal, Vol. 24, No.2, 1981, pp. 162-166.

29. Baker, T.J., "Three-Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," AIAA Paper No. 87-1124, June 1987.

30. Parikh, P., Pirzadeh, S., and Lohner, R., "A Package for 3-D Unstructured Grid Generation, Finite Element Flow Solver, and Flow-Field Visualization," NASA CR-182090, 1990.

31. Pirzadeh, S., "Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method," AIAA Paper No. 91-3223, Sept. 1991.

32. Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper No. 94-0417, Jan. 1994.

33. Pirzadeh, S., "Unstructured Viscous Grid Generation by the Advancing-Front Method," NASA CR-191449, 1993.

34. Mavriplis, D.J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," NASA CR-189719, 1992.

35. Batina, J.T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," AIAA Paper No. 89-0115, Jan. 1989.

36. Trepanier, J.Y., Reggio, M., Paraschivoiu, I., and Camarero, R., "Unsteady Euler Solutions for Arbitrarily Moving Bodies and Boundaries," AIAA Journal, Vol. 31, No. 10, Oct. 1993.

37. Singh, K.P., Newman III, J.C., and Baysal, O., "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," AIAA Paper No. 94-0058, Jan. 1994.

38. Lohner, R., "An Adaptive Finite Element Scheme for Transient Problems in CFD," Computer Methods in Applied Mechanics and Engineering, Vol. 61, 1987, pp. 323-338.

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

40. Slack, D.C., Walters, R.W., and Lohner, R., "An Interactive Adaptive Remeshing Algorithm for the Two-Dimensional Euler Equations," AIAA Paper No. 90-0331, Jan. 1990.

41. Fouladi, K., Baysal, O., and Newman III, J.C., "Hybrid Domain Decomposition for Configurations with Multiple Nonsimilar Components," SIAM, Fifth Conference on Domain Decomposition Methods for Partial Differential Equations, May 1991.

42. Walters, R.W., "Compressible Flow Algorithms on Structured and Unstructured Grids," Ninth Conference on Computing Methods in Applied Sciences and Engineering, Paris, France, Jan. 29-Feb. 5, 1990.

43. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Scheme," AIAA Paper No. 81-1259, June 1981.

44. Jameson, A., and Baker, T.J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper No. 83-1929, June 1993.

45. 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.

46. van Leer, B., Thomas, J.L., Roe, P.L., and Newsome, R.W., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," AIAA Paper No. 87-1104, 1987.

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

48. Rumsey, C.L., and Anderson, W.K., "Some Numerical and Physical Aspects of Unsteady Navier-Stokes Computations Over Airfoils using Dynamic Meshes," AIAA Paper No. 88-0329, Jan. 1988.

49. van Leer, B., "Flux Vector Splitting for the Euler equations," Lecture Notes in Physics, Vol. 170, Springer-Verlag, 1982.

50. Anderson, W.K., Thomas, J.L., Rumsey, C.L., "Extension of Applications of Flux-Vector Splitting to Unsteady Calculations on Dynamic Meshes," AIAA Paper No. 87-1152, June 1987.

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

52. Whitaker, D.L., Slack, D.C., and Walters, R.W., "Solution Algorithms for the Two-Dimensional Euler Equations on Unstructured Meshes," AIAA Paper No. 90-0697, Jan. 1990.

53. Barth, T.J., and Jespersen, D.C., "The Design and Application of Upwind Schemes on Ustructured Meshes," AIAA Paper No. 89-0366, Jan. 1989.

54. Pan, D., and Cheng, J.C., "Upwind Finite Volume Navier-Stokes Computations on Unstructured Triangular Meshes," AIAA Journal, Vol. 31, No.9, Sept. 1993, pp.1618-1625.

55. 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.

56. Frink, N.T., Parikh, P., and Pirzadeh, S., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes," AIAA Paper No. 91-0102, Jan. 1991.

57. Frink, N.T., "Recent Progress Toward a Three-Dimensional Unstructured Navier-Stokes Flow Solver," AIAA Paper No. 94-0061, Jan. 1994.

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

59. Chakravarthy, S.R., and Szema, K.Y., "An Euler Solver for 3-D Supersonic Flows with Subsonic Pockets," AIAA Paper No. 85-1703, July 1985.

60. van Albada, G.D., van Leer, B., and Roberts, W.W., "A Comparitive Study of Computational Methods in Cosmic Gas Dynamics," Astronomy and Astrophysics, Vol. 108, April 1982, pp. 76-84.

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

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

63. Thomas, J.L., Taylor, S.L., and Anderson, W.K., "Navier Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings," AIAA Journal, Vol. 28, Feb. 1990, pp. 205-212.

64. Vatsa, V.N., Thomas, J.L., and Wedan, B.W., "Navier-Stokes Computations of a Prolate Spheroid at Angle of Attack," Journal of Aircraft, Vol. 26, Nov. 1989, pp. 986-993.

65. Thomas, J.L., van Leer, B., and Walters, R.W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA Journal, Vol. 28, June 1990, pp. 973-974.

66. 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, Oct. 1979, pp. 1030-1037.

67. Tamura,Y. and Fujii, K., “Conservation Law for Moving and Transformed Grids,” AIAA Paper No. 93-3365, July 1993.

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

69. Brandt, A., "Multilevel Adaptive Computations in Fluid Dynamics," AIAA Journal, Vol. 18, No. 10, Oct. 1980, pp. 1165-1172.

70. Visbal, M.R., and Shang, J.S., "Numerical Investigation of the Flow Structure around a Rapidly Pitching Airfoil," AIAA Paper No. 87-1424, June 1987.

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

72. Krist, S.L., Washburn, A.E., and Visser, K.D., "A Computational and Experimental Investigation of a Delta Wing with Vertical Tails," AIAA Paper No. 93-3009, July 1993.

73. Walters, R.W., Reu, T., McGrory, W.D., Thomas, J.L., and Richardson, P.F., "A Longitudinally-Patched Grid Approach with Applications to High Speed Flows," AIAA Paper No. 88-0715, Jan. 1988.

74. Belk, D.M., and Whitfield, D.L., "Time Accurate Euler Equations Solutions on Dynamic Blocked Grids," AIAA Paper No. 87-1127, 1987.

75. Belk, D.M., "Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids," Ph.D Dissertation, Mississippi State University, Aug. 1986.

76. Baysal, O., and Yen, G.W., "Kinematic Domain Decomposition To Simulate Flows Past Moving Bodies," AIAA Paper No. 91-0725, Jan. 1991.

77. Lijewski, L.E., and Suhs, N., "Chimera-Eagle Store Separation," AIAA Paper No. 92-4569, Aug. 1992.

78. Dougherty, F.C., and Kuan, J.H., "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme," AIAA Paper No. 89-0637, Jan. 1989.

79. Lessard, V.L., "Domain Decomposition for Multigrid, Finite Volume Flow Solvers," M.S. Thesis, Mechanical Engineering and Mechanics Dept., Old Dominion University, Dec. 1989.

80. Chakravarthy, S.R. and Szema, K-Y., "Computational Fluid Dynamics Capability For Internally Carried Store Separation," Technical Report SC71039TR, Rockwell International Science Center, Thousand Oaks, CA, Nov. 1991.

81. Mavriplis, D., and Jameson, A., "Multigrid Solutions of the Euler Equations on Unstructured and Adaptive Meshes," ICASE Report No. 87-53, 1987.

82. Mavriplis, D.J., "Unstructured Mesh Algorithms for Aeroynamic Calculations," ICASE Report No. 92-35, July 1992.

83. Anderson, W.K., and Bonhaus, D.L., "Navier-Stokes Computations and Experimental Comparisons for Multielement Airfoil Configurations," AIAA Paper No. 93-0645, Jan. 1993.

84. Ghaffari, F., "On the Vortical-Flow Prediction Capability of an Unstructured-Grid Euler Solver," AIAA Paper No. 94-0163, Jan. 1994.

85. Heim, E.R., "CFD Wing/Pylon/Finned Store Mutual Interference Wind Tunnel Experiment," AEDC-TSR-91-P4, Jan. 1991.

86. Richter, R., and Leyland, P., "Precise Pitching Airfoil Computations by Use of Dynamic Unstructured Meshes," AIAA Paper No. 93-2971, July 1993.

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