

Old Dominion University

ODU Digital Commons

Mechanical & Aerospace Engineering Theses & Dissertations

Mechanical & Aerospace Engineering

Winter 1992

Explicit Multistage Schemes for the Solution of the Three-Dimensional Compressible Euler/Navier-Stokes Equations

Alaa Ahmed Elmiligui
Old Dominion University

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



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Elmiligui, Alaa A.. "Explicit Multistage Schemes for the Solution of the Three-Dimensional Compressible Euler/Navier-Stokes Equations" (1992). Doctor of Philosophy (PhD), Dissertation, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/3gxf-m051
https://digitalcommons.odu.edu/mae_etds/235

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

**EXPLICIT MULTISTAGE SCHEMES FOR THE
SOLUTION OF THE THREE-DIMENSIONAL
COMPRESSIBLE EULER/NAVIER-STOKES EQUATIONS**

Alaa Ahmed Elmiligui

B.Sc. May 1980, Alexandria University, Egypt

M.Sc. May 1985, Alexandria University, Egypt

**A Dissertation submitted to the Faculty of Old Dominion University
in Partial Fulfillment of the Requirement for the Degree of**

**DOCTOR OF PHILOSOPHY
MECHANICAL ENGINEERING
OLD DOMINION UNIVERSITY**

December 1992

Approved by:

Robert L. Ash (Director)

Manuel D. Salas

Osama A. Kandil

Arthur C. Taylor, III

ABSTRACT

The objective of this study was to develop a high-resolution-explicit-multi-block numerical algorithm, suitable for efficient computation of the three-dimensional, time-dependent Euler and Navier-Stokes equations. The resulting algorithm has employed a finite volume approach, using MUSCL-type differencing to obtain state variables at cell interface. Variable interpolations were written in the κ -scheme formulation. Inviscid fluxes were calculated via Roe's flux-difference splitting, and van Leer's flux-vector splitting techniques, which are considered state of the art. The viscous terms were discretized using a second-order, central-difference operator.

Two classes of explicit time integration has been investigated for solving the compressible inviscid/viscous flow problems —two-stage predictor-corrector schemes, and multistage time-stepping schemes. The coefficients of the multistage time-stepping schemes have been modified successfully to achieve better performance with upwind differencing. A technique was developed to optimize the coefficients for good high-frequency damping at relatively high CFL numbers. Local time-stepping, implicit residual smoothing, and multigrid procedure were added to the explicit time stepping scheme to accelerate convergence to steady-state. The developed algorithm was implemented successfully in a multi-block code, which provides complete topological and geometric flexibility. The only requirement is C^0 continuity of the grid across the block interface.

The algorithm has been validated on a diverse set of three-dimensional test cases of increasing complexity. The cases studied were: (1) supersonic corner flow; (2) supersonic plume flow; (3) laminar and turbulent flow over a flat plate; (4) transonic flow over an

ONERA M6 wing, and (5) unsteady flow of a compressible jet impinging on a ground plane (with and without cross flow). The emphasis of the test cases was validation of code, and assessment of performance, as well as demonstration of flexibility.

DEDICATION

I dedicate this dissertation to my family whose five years of patience, endurance and sacrifice made it all possible. Their continuous love, encouragement surrounded me with a joyful and cheerful environment. It is their constant support and understanding that helped me get here. I feel fortunate to have such a wonderful family.

ACKNOWLEDGMENT

Praise and thanks be to God to whom I owe everything.

I express my sincere appreciation and deepest gratitude to my advisor Dr. Robert Ash for his technical guidance, encouragement and friendship. His knowledge, generosity and kindness will always be remembered. I also wish to thank him for his patience and tireless help in the preparation and organization of this thesis. I gratefully acknowledge Mr. Manuel Salas whose ideas, input and suggestions are reflected throughout this work. His guidance, support and understanding made this research possible. Thanks are extended to Dr. Osama Kandil for his generous input, advice, and support, and to Dr. Arthur Taylor for serving on my dissertation committee. While working on this dissertation I have received help and support from many people which I would like to take this opportunity to thank as well. Mr. Jim Keller whose constant care and assistance has been a great help to me. Mr. Duane Melson who provided countless fruitful suggestions at various stages in the project. Dr. Perry Newman, Dr. Charles Swanson, Mr. Mark Sanetrik, and Mr. Joseph Morrison for freely giving their time in discussing many technical problems. I am indebted to Dr. Ernst von Lavante and Dr. Hassan Warda for ushering me to the Computational Fluid Dynamics path.

My heartfelt thanks goes to Dr. Frank Cannizzaro for his generous input, support, and for sharing some of his excellent ideas. I have enjoyed his friendship, and the productive collaboration and technical discussions.

Financial support for this work has been provided by NASA Langley Research Center Grant No. NAG1-633.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
NOMENCLATURE	xi
Chapter 1	INTRODUCTION 1
1.1	Historical Background 1
1.2	Objective of Present Work 9
1.3	Thesis Outline 10
Chapter 2	GOVERNING EQUATIONS 12
2.1	Navier-Stokes Equations 12
2.2	Normalization of the Governing Equations. 16
2.3	Curvilinear Coordinate Transformation 18
2.4	Thin-Layer Navier-Stokes Equations 19
2.5	Reynolds-Averaged Navier-Stokes Equations 21
2.6	Baldwin-Lomax Algebraic-Turbulence Model 25
2.7	Euler Equations 26
Chapter 3	SPATIAL DISCRETIZATION 28
3.1	van Leer's Flux-Vector Splitting 33
3.2	Roe's Flux-Difference Splitting 35

Chapter 4	TIME INTEGRATION	39
4.1	Multistage Time-Stepping Scheme.	40
4.2	Predictor-Corrector Schemes.	61
Chapter 5	ACCELERATING TECHNIQUES	65
5.1	Local Time-Stepping	65
5.2	Implicit Residual Smoothing	66
5.3	Multigrid Method	70
Chapter 6	MULTI-BLOCK	76
6.1	Multi-Block Strategy	77
6.2	Multi-Block and Multigrid	79
6.3	Boundary Conditions	80
6.3.1	Solid Boundary	81
6.3.2	Symmetry Plane	82
6.3.3	Inflow/Outflow Boundary.	83
6.3.4	Interface Between Blocks	86
Chapter 7	TEST CASES	87
7.1	Background	87
7.2	Corner Flow	94
7.3	Non-Axisymmetric Jet Exhaust Plume	106
7.4	Flat Plate	121
7.5	ONERA M6 Wing	128
7.6	Jet In Ground Effect	143
7.6.1	Jet Impinging on a Ground Plane	143

7.6.2	Jet Impinging on a Ground Plane in Presence of Crossflow	153
Chapter 8	CONCLUSIONS	161
References	167
Appendix A	Full Navier-Stokes Equations in Body-Fitted Coordinates	179
Appendix B	Derivation of the Reynolds-Averaged Navier-Stokes Equations	185
Appendix C	Limiters	191
C.1	Minum-Modulus Limiter	191
C.2	Vanalbda Limiter	191
Appendix D	Multigrid Cycles	193

LIST OF TABLES

Table 1	Values of κ and the Corresponding Truncation Error. . . .	32
Table 2	Multistage Coefficients for First-Order Scheme	46
Table 3	Multistage Coefficients for Second-Order Fully Upwind Scheme	46
Table 4	Multistage Coefficients for Second-Order Fromm Scheme	46
Table 5	Multistage Coefficients for Third-Order Upwind Biased Scheme	46

LIST OF FIGURES

Figure 3.1	Schematic of Computational Cell	28
Figure 3.2	The Riemann Problem.	36
Figure 4.1	Contours of Modulus of Amplification Factor, $ G $, and Locus of Fourier Transform, P ---, for a First-Order, Four-Stage Runge-Kutta Scheme, $CFL = 2.0$	45
Figure 4.2	Contours of Modulus of Amplification Factor, $ G $, and Locus of Fourier Transform of Spatial-Operator, P ---, for Two-Stage Schemes.	50
Figure 4.3	Contours of Modulus of Amplification Factor, $ G $, and Locus of Fourier Transform of Spatial-Operator, P , ---, for Three-Stage Schemes.	52
Figure 4.4	Contours of Modulus of Amplification Factor, $ G $, and Locus of Fourier Transform of Spatial-Operator, P , ---, for Four-Stage Schemes.	54
Figure 4.5	Modulus of Amplification Factor, $ G $, as a Function of the Spatial Wave Number, β , and the CFL number, for Two-Stage Schemes.	56
Figure 4.6	Modulus of Amplification Factor, $ G $, as a Function of the Spatial Wave Number, β , and the CFL number, for Three-Stage Schemes.	58
Figure 4.7	Modulus of Amplification Factor, $ G $, as a Function of the Spatial Wave Number, β , and the CFL number, for Four-Stage Schemes.	60
Figure 4.8	Modulus of Amplification Factor, $ G $, as a Function of the Spatial Wave Number, β , and the CFL Number, for Predictor-Corrector Schemes.	64

Figure 5.1	Modulus of Amplification Factor as a Function of the Spatial Wave Number and the CFL for the Predictor-Corrector Scheme with Residual Smoother ($\epsilon = 0.5$).	69
Figure 6.1	Schematic of a Block Face with a Generic “Patch”, Accommodating Multiple Boundary Conditions	78
Figure 6.2	Schematic of Multi-Block-Multigrid Strategy	80
Figure 6.3	Schematic of a Plane in the Computational Domain	82
Figure 6.4	Inflow and Outflow Boundary Conditions	85
Figure 7.1	A Schematic Diagram of a Cross Plane for Supersonic Corner Flow.	88
Figure 7.2	Schematic of Pseudo-Two-Dimensional Jet Exhaust Plume	90
Figure 7.3	Schematic of an Isolated Jet Impinging on a Ground Plane.	92
Figure 7.4	Schematic of a Jet Impinging on a Ground Plane in Presence of Crossflow	93
Figure 7.5	Schematic of Compression Corner.	94
Figure 7.6	Mach Contours, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$	96
Figure 7.7	Pressure Contours, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$	96
Figure 7.8	Comparison of Numerical and Experimental Surface Pressure Distributions of Corner Flow.	98
Figure 7.9	Comparison of Convergence Histories of 1-2, 2-2, Four-Stage, $\kappa = 0$ Schemes Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)	100
Figure 7.10	Comparison of Convergence Histories of the Four-Stage Schemes, Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)	101

Figure 7.11	Comparison of Convergence Histories of 1-2 Scheme With and Without Multigrid Acceleration, Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)	102
Figure 7.12	Schematic of Grid for Eight Block Calculation for Flow through a 9.5° Compression Corner	103
Figure 7.13	Mach Contours for Eight Block Calculations, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$	104
Figure 7.14	Pressure Contours for Eight Block Calculations, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$	104
Figure 7.15	Comparison of Convergence Histories of Single Block and Eight Block Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)	105
Figure 7.16	Schematic of the Computed Flow Field for a Pseudo-Two-Dimensional Jet Exhaust Plume	107
Figure 7.17	Partial View of the Two-Dimensional Jet Exhaust Plume Grid.	108
Figure 7.18	Non-Axisymmetric Jet Exhaust Plume Calculations Utilizing Roe's Scheme, $M_\infty = 2.5$, $M_{jet} = 1.5$, $P_{jet} = 3.5$, and $T_{jet}/T_\infty = 3.0$	109
Figure 7.19	Non-Axisymmetric Jet Exhaust Plume Calculations Utilizing van Leer's Scheme, $M_\infty = 2.5$, $M_{jet} = 1.5$, $P_{jet}/P_\infty = 3.5$, and $T_{jet}/T_\infty = 3.0$	110
Figure 7.20	Comparison Between Roe's Flux-Difference Splitting Scheme, and van Leer's Flux-Vector Splitting Scheme and Salas' Shock-Fitting Method for a Pseudo-Two-Dimensional Exhaust Plume.	114
Figure 7.21	Comparisons Between Primitive and Conservative Extrapolations of the Roe's Scheme with the Shock-Fitting Code of Salas for a Pseudo-Two-Dimensional Exhaust Plume.	118

Figure 7.22	Comparisons Between Different Extrapolations of Roe's Scheme and a Shock-Fitting Code for a Pseudo-Two-Dimensional Exhaust Plume.	119
Figure 7.23	Comparisons of Convergence History for Pseudo-Two-Dimensional Exhaust Plume.	120
Figure 7.24	Results for the Laminar Flow Over a Flat Plate. $M_\infty =$ 0.5 , $Re_\infty = 1000$ per Unit Length.	122
Figure 7.25	Velocity Distribution Over a Flat Plate.	123
Figure 7.26	Local Skin Friction Coefficient Along a Flat Plate.	124
Figure 7.27	Convergence History For Flow Over a Flat Plate.	126
Figure 7.28	Turbulent Flow Over a Flat Plate, ($M_\infty = 0.5$, $Re_\infty = 1.0 \times 10^6$ per Unit Length.	127
Figure 7.29	Partial View of C-O Grid Topology for ONERA M6 Wing.	129
Figure 7.30	Pressure Distribution for ONERA M6 Wing $M_\infty = 0.669$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	132
Figure 7.31	Pressure Contours for ONERA M6 Wing $M_\infty = 0.84$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	134
Figure 7.32	Pressure Distribution for ONERA M6 Wing $M_\infty = 0.84$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	137
Figure 7.33	Pressure Contours for ONERA M6 Wing $M_\infty = 0.84$, and $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	138
Figure 7.34	Wall Streamlines for ONERA M6 Wing $M_\infty = 0.84$, $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	139
Figure 7.35	Pressure Distribution for ONERA M6 Wing $M_\infty = 0.84$, $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$	142
Figure 7.36	Partial View of Grid Utilized for Computing a Jet Impinging on a Ground Plane.	144

Figure 7.37	Jet Impingement on a Ground Plane. $M_{jet} = 0.5$, $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 4$, $Re = 19000$	147
Figure 7.38	Wall Jet Time-Average-V-Velocity Profiles, at Various Radial Positions, Symbol \circ Represents the Spacing of the Grid in the Normal Direction to the Ground Plane.	149
Figure 7.39	Wall Jet Time-Averaged-Static-Pressure Profiles, at Various Radial Positions, Symbol \circ Represents the Spacing of the Grid in the Normal Direction to the Ground Plane.	150
Figure 7.40	Comparison of Ground-Plane Pressure Variation for Jet Impinging on a Ground Plane	151
Figure 7.41	Comparison of Jet Centerline Velocity Decay	152
Figure 7.42	Partial View of the Grid Utilized to Compute Jet Impinging on a Ground Plane in Presence of Crossflow.	153
Figure 7.43	Velocity Vectors of the Developing Jet Flow Field in a Crossflow; $M_{jet} = 0.5$, $M_{\infty} = 0.1$, $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 3$, $Re = 1 \times 10^5$	157
Figure 7.44	Streamlines of the Developing Jet Flow Field in a Crossflow $M_{jet} = 0.5$, $M_{\infty} = 0.1$, $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 3$, $Re = 1 \times 10^5$	159
Figure 7.45	Comparison of Centerline C_p Distributions for a Jet Impinging on a Ground Plane with Crossflow; $M_{jet} = 0.5$, $M_{\infty} = 0.1$, $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 3$, $Re = 1 \times 10^5$	160
Figure D.1	V-Cycle.	193
Figure D.2	W-Cycle.	194
Figure D.3	Full Multigrid V-Cycle	195
Figure D.4	Full Multigrid W-cycle	195

NOMENCLATURE

A	flux Jacobian $\frac{\partial F}{\partial Q}$
a	local speed of sound
A^+	turbulence constant, 26.
c	chord or body length
C_{KLEB}	Klebanoff constant, 0.3
C_{cp}	turbulence constant, 1.6
C_p	Pressure coefficient
\hat{c}_p	dimensional specific heat at constant pressure
C_{wk}	turbulence constant, 1.0 {value used for transonic flow}
\hat{c}_v	dimensional specific heat at constant volume
CFL	Courant-Friedrichs-Lewy stability limit
CFL^*	standard CFL with no residual smoothing
\hat{e}_i	dimensional internal energy
E	total internal energy per unit volume
f	forcing function in finite-difference problem used in multigrid process
f, f_v	Eulerian and viscous Reynolds Favré density averaged dimensional flux vectors in the x-direction
FMG	full multigrid
F, F_v	nondimensional Eulerian and viscous flux vectors in the ξ -direction
\hat{F}, \tilde{F}	dimensional and nondimensional Eulerian flux vectors in the x-direction
\hat{F}_v, \tilde{F}_v	dimensional and nondimensional viscous flux vectors in the x-direction
FS	indicates Fuselage Station location on the afterbody
g, g_v	Eulerian and viscous Reynolds Favre density averaged dimensional flux vectors in the y-coordinate direction
G	Amplification factor
G, G_v	nondimensional Eulerian and viscous flux vectors in the η -direction
\hat{G}, \tilde{G}	dimensional and nondimensional Eulerian flux vector in the y-direction

\hat{G}_v, \tilde{G}_v	dimensional and nondimensional viscous flux vectors in the y-direction
H	total enthalpy per unit volume
h	specific enthalpy per unit volume
h, h_v	Eulerian and viscous Reynolds Favre density averaged dimensional flux vectors in the z-direction
H, H_v	nondimensional Eulerian and viscous flux vectors in the ζ -direction
\hat{H}, \tilde{H}	dimensional and nondimensional Eulerian flux vectors in the z-direction
\hat{H}_v, \tilde{H}_v	dimensional and nondimensional viscous flux vectors in the z-direction
I_h^{2h}	restriction operator from h-spacing to 2h-spacing
I_{2h}^h	prolongation operator from 2h-spacing to h-spacing
J	grid cell Jacobian of transformation
K	kinetic energy
k	coefficient of thermal conductivity
k_1	von Karman's constant 0.4
K_2	Clauser constant, 0.0168
k_T	turbulent thermal conductivity
L	finite-difference operator
M	Mach number
N	n-stage modified Runge-Kutta time integration
\hat{p}, p	dimensional and nondimensional static pressure
P_r	Prandtl number
P_{rT}	turbulent Prandtl number
PE	potential energy
q	normalized contravariant velocity
q	Reynolds Favre density averaged dimensional vector of conservative variables
Q	nondimensional vector of conservative variables in body-fitted coordinates
\hat{Q}, \tilde{Q}	dimensional and nondimensional vectors of conservative variables in Cartesian coordinates
R	Reynolds number
R^+, R^-	non-negative and non-positive Riemann invariants
S^*	isentropically derived entropy value
S_ℓ	left eigenvector matrix relative to the ℓ -direction

S_ℓ^{-1}	right eigenvector matrix relative to the ℓ -direction
t	nondimensional time
\hat{t}	dimensional time
\hat{T}, T	dimensional and nondimensional static temperature
U	solution to finite-difference problem used in multigrid process
u	approximate solution to finite-difference problem used in multigrid process
U, V, W	contravariant velocities in the ξ -, η -, and ζ -coordinate directions
\bar{u}_ℓ	local scaled contravariant velocity in the ℓ -direction
\hat{u}, u	dimensional and nondimensional Cartesian velocity in the x-direction
V	error in approximate solution to finite-difference problem used in multigrid process
\hat{v}, v	dimensional and nondimensional Cartesian velocity in the y-direction
Vol	volume of cell
\hat{w}, w	dimensional and nondimensional Cartesian velocity in the z-direction
x, y, z	nondimensional Cartesian coordinate directions
$\hat{x}, \hat{y}, \hat{z}$	dimensional Cartesian coordinate directions
y^+	length scale for law of the wall
α	angle of attack
β	Spatial wave number
β	$\frac{\hat{c}_p}{c_v} - 1 = \gamma - 1$
β_ℓ	variable coefficient for residual smoothing in the ℓ -direction
γ	ratio of specific heats $\frac{\hat{c}_p}{c_v}$
λ	indicates eigenvalue
Λ	diagonal eigenvalue matrix
μ_T	turbulent eddy viscosity
$\hat{\mu}, \mu$	dimensional and nondimensional molecular viscosity
ω	vorticity
$\hat{\rho}, \rho$	dimensional and nondimensional density
σ	$\frac{1}{\beta(\mu + \mu_T)} \left(\frac{\mu}{P_r} + \frac{\mu_T}{P_{rT}} \right)$
τ	shear stress tensor
τ^R	Reynolds stress tensor

τ^T	combined stress tensor of $\tau^R + \tau$
ε	$\frac{1}{(\gamma-1)P_\tau}$ for laminar equations or small parameter
ξ, η, ζ	nondimensional body-fitted coordinate directions
∂	partial derivative
\mathcal{R}	gas constant
∇	backward differencing or gradient
Δ	forward differencing

Subscripts

g	ghost cell
int	interior cell
ℓ	represents ξ, η, ζ
L	values obtained from the left side of a cell face
n	normal
R	values obtained from the right side of a cell face
ref	reference
x, y, z	indicates derivatives relative to these Cartesian directions
∞	free-stream value

Superscripts

h	grid spacing
n	time level
v	indicates viscous-dependent values
\wedge	dimensional Cartesian value
$\hat{}$	Favré density-averaged fluctuating value
$-$	non-positive eigenvalues
$+$	non-negative eigenvalues
\sim	nondimensional Cartesian value or Roe averaged variable
\sim	Favré density-averaged time-independent value

CHAPTER 1

INTRODUCTION

1.1 Historical Background

Recent developments in numerical methods and their applications permit the solving of complex, realistic geometries and configurations for compressible flows. Currently, computational fluid dynamics (*CFD*) is used effectively to unravel and elucidate fluid flow phenomena which are difficult to study in the laboratory. The demand to solve finely detailed models of physics has challenged many researchers to come up with new and efficient tools. This demand has resulted in revolutionary concepts in computer architecture designs and software development.

The birth of *CFD* can perhaps be linked to the early work of the English mathematician Richardson in 1917 [1]. He attempted to integrate the meteorological equations numerically. It is interesting to note that he started this process, which evolved into a new science, as an ambulance driver during World War I. He made the computations by hand, [1]. His attempts were unsuccessful due to a limited theoretical understanding of the stability of numerical methods, and to a lack of computing power. Richardson's failure outlined the areas which needed to be developed. In 1928 Courant, Friedrichs, and Lewy [2] introduced their famous stability condition, which became subsequently the *CFL* number, and represented a landmark mathematical result that has had a massive impact on computational research.

The practical birth of *CFD* came in the late 1960's when significant computing power became available. Since then, there has been considerable progress in the field of *CFD*.

The growing field of aerodynamics, and the aviation industry have been the catalyst for the revolutionary force of CFD. In this section, a brief review of previous Computational Fluid Dynamics work related to the present work is presented. For a more general review of CFD, the interested reader should review Refs. [1, 3–6].

One of the first major advances in Computational Fluid Dynamics was the work by Hess and Smith [7], who introduced panel methods to solve the linearized potential flow equation. Later the panel method was extended to lifting flows by Rubbert and Saaris [8] and supersonic flows by Woodward [9]. In 1986, Kandil and Yates [10] extended the method to solve the steady, full-potential equation. In 1987, Kandil and Hong [11], successfully formulated the vortex-panel method in a moving frame of reference.

In the early seventies, two major breakthroughs were reported which allowed the solution of non-linear mathematical models. Murman and Cole [12], devised the idea of mixed differencing (central differencing in subsonic regions, and forward or backward differencing in supersonic regions of flow). They employed a line relaxation method for the entire flow field, which was partly elliptic and partly hyperbolic. Their work, and the work of Jameson [13], was the catalyst for developing two- and three-dimensional algorithms using the Small Disturbance Equation, and the Full Potential Equation. An interesting review of the memoirs of Murman and Cole is presented in a review paper by Hall [14]. The second major breakthrough was the work by Magnus and Yoshihara [15]. They advanced the Euler Equation in time towards a steady-state, thus transforming a mixed elliptic-hyperbolic problem into a purely hyperbolic one. Another landmark in the history of CFD came in 1970, when McCormack introduced his widely used predictor-corrector explicit difference scheme [16]. Subsequently, in 1981, McCormack [17] developed an implicit analogue of his explicit finite difference method. In 1975, Warming and Beam [18] introduced a fully upwind predictor-corrector

method, which is similar to the McCormack method. Briley and McDonald [19], and Beam and Warming [20, 21] employed an Alternating Direction Implicit (*ADI*) scheme for solving the Euler and Navier-Stokes equations. The roots of *ADI* schemes trace back to Peaceman and Rachford [22], Douglas [23], and Douglas and Gunn [24]. Steger [25], adapted the Beam and Warming scheme to general curvilinear coordinates. *ADI* evolved to an effective tool and currently is employed in state-of-the-art codes designated ARC2D and ARC3D [26].

On the other hand, another important family of time integration schemes —explicit, multistage time-stepping schemes (Runge-Kutta methods)— started to evolve in the early eighties. Jameson, Schmidt, and Turkel [27], introduced explicit, multistage time-stepping schemes, to the CFD community. Explicit-multistage schemes were developed further, and have been applied successfully to compute solutions to the Euler, and Navier-Stokes equations, for two- and three-dimensional problems [28–35]. Explicit schemes combine naturally with accelerating techniques such as: local time-stepping, residual smoothing, and multigrid accelerating techniques. They are also well suited for parallel computing [36, 37].

The restriction on the time step for explicit schemes was the catalyst to develop implicit schemes. Implicit schemes require more computation per time step (iteration), but allow a larger time step to be used. The implicit time integration scheme may be stable for any step size, according to linear theory, yet it is limited in practice by the non-linearity of the governing equations. Due to simplifications made during the development of these methods (linearization) and the frequent use of explicit boundary conditions, the maximum allowable Courant-Friedrichs-Lewy number (*CFL*) is reduced. To date, the relative merits of implicit and explicit schemes are still an open debate for steady and unsteady flow calculations.

Implicit residual smoothing extends the stability limit, and improves the damping properties of the multistage time-stepping scheme. Lerat [38], introduced the idea of residual smoothing for the Lax-Wendroff scheme [39]. Jameson and Baker [29] applied the idea of implicit residual smoothing in conjunction with the modified Runge-Kutta schemes. This procedure was developed further in Refs. [28, 33, 40–42], where they employed a central-implicit-residual-smoothing operator. The use of an upwind-residual-smoothing operator was employed by von Lavante and Gronner [43], and Blazek *et al.* [44].

Multigrid acceleration techniques were developed originally by Fedorenko [45, 46] starting in 1961. Subsequently Brandt [47] applied the technique to an elliptic set of equations. The work by Brandt and many others has led to the popular use of multigrid by many in the fields of applied mathematics and computational engineering. Excellent developments of the multigrid technique can be found in Refs. [48–50]. Multigrid was used successfully for solving the potential, Euler, and Navier-Stokes equations, Refs. [51–55]. Multigrid acceleration techniques performed well when combined with central-difference methods, but the convergence rate deteriorated with upwind spatial operators because they are less dissipative. One must ensure that the basic upwind scheme exhibits good damping of high frequencies on both fine and coarse meshes. An attempt to derive a mathematical operator to eliminate the high-frequency components of the error should be pursued.

In the early 1980's, computers were powerful enough to permit the computation of solutions to the Euler equations. A new wave of inviscid upwind and central-difference schemes evolved. Upwind schemes attempted to construct the flux by modelling the underlying physics, as dictated by the sign of characteristic waves, while central-difference schemes computed the interface flux as an average of the two adjacent cells, disregarding

characteristic theory. A comparison between central-difference schemes, and upwind schemes and how they are related is given by Swanson and Turkel [56].

A prevalent way to introduce upwinding into the governing systems for hyperbolic conservation equations has been to split the flux according to the characteristic speed ($q, q \pm c$). Steger and Warming [57], were the first to devise a conservative-second-order-flux-vector splitting-upwind scheme, without the use of limiters, for the solution of the governing equations of gas dynamics. Anderson, Thomas, and van Leer [58], developed the Monotone Upstream-centered Scheme for Conservative Laws (MUSCL) approach with limiters which was incorporated in the Steger-Warming scheme. The MUSCL approach resulted in a better shock capturing capability. The main disadvantage of the Steger-Warming-flux-vector splitting scheme was that the backward and forward fluxes were not differentiable. This leads to oscillation at shocks. van Leer [59], devised an alternative splitting scheme. The advantage of van Leer's flux-vector splitting over the Steger-Warming flux-vector splitting scheme, was that the split flux-vectors were smooth and had smooth first derivatives with respect to the Mach number, so that their eigenvalues were also smooth [58].

The inviscid flux can be split in a number of ways. The Split Coefficient Matrix, (SCM) as introduced by Chakravarthy, Anderson, and Salas [60], is a natural way of splitting the flux based on the sign of the eigenvalues of the governing system of equations. A similar scheme that is based on the theory of characteristic is Morreti's λ -scheme [61]. Both the SCM- and λ -schemes have been applied to the non-conservative form of the governing equations, and require shock-fitting techniques in the presence of shocks. The conservative form of the governing equations permit shock waves to be captured as weak solutions to the governing equation [39, 62, 63], thus avoiding the difficulty of applying shock-fitting techniques.

In 1959, Godunov [64], introduced the idea of advancing in time by solving the Riemann problem at each cell. This technique has been extended to higher order schemes which are known today as Godunov-type schemes, [65–71]. A review of Godunov-type schemes is presented by Roe [5], and Yee [72]. Currently, upwind schemes are being used on a regular basis for computing solutions to the Euler and Navier-Stokes equations. They have been implemented and validated in several state-of-the-art codes, such as CFL3D [73], ISAAC [74], PAB3D [75] and FTNS3D [76].

Alternatively, Jameson, Schmidt, and Turkel [27] have introduced multistage time-stepping schemes, coupled with a central-difference operator and explicitly added dissipation terms. The explicit dissipation term was a blend of second-order-difference and fourth-order-difference terms. Second-order-difference terms suppress oscillations in the neighborhood of shock waves, while fourth-order-difference terms are crucial for the stability and convergence to steady-state. Dissipation terms have been scaled by user defined coefficients. Detailed discussion of the influence of the dissipation terms on the performance and quality of steady-state solutions can be found in Kandil and Chuang [77], Rizzi [78], Pulliam [79], and Swanson and Turkel [51].

Currently, the state-of-the-art in computational fluid dynamics replaces scalar dissipation with a matrix-valued dissipation function. Employing matrix dissipation enhances the shock capturing capabilities of the central-difference technique, and reduces the smearing of shocks and contact discontinuities which were characteristic of the original central-difference schemes [51]. Central-difference operators, coupled with a matrix-valued dissipation function, are nearly as accurate as upwind schemes, and have the merit of being computationally cheaper and easier to program [56].

The numerical dissipation terms play an important role in the success of the computations by central-difference methods. For every new configuration, the exact (optimum)

level of artificial dissipation is not known *a priori*. The level of numerical dissipation can be turned up, by a novice user, to the point of masking the physics of the problem. A certain level of expertise with central-difference schemes and with the physical problem of interest is required to select the optimum (acceptable) level of dissipation. Central-difference schemes have been applied in state-of-the-art schemes, TLNS3D [28], ARC2D and ARC3D [26] and FLOMG [32].

The application of the above numerical methods to realistic three-dimensional configurations of significant geometric complexity is virtually impossible without the use of Domain Decomposition techniques. Here, the computational domain is divided into multiple blocks (zones) and the grid for each block is then generated. A computational grid of this type adapts more easily to the shape of the body as well as to the flow features. Typically, the transfer of information between the blocks is carried out explicitly by ensuring the conservation of fluxes across the block interfaces. The consequence of this procedure, for an implicit operator, is a significant reduction in the maximum allowable CFL number.

Generating a single body fitted grid for complex, three-dimensional realistic geometries is a difficult task to perform; for some configurations it is almost impossible [80–82]. Several grid methodologies such as overlaid grids [83], patched grids [84], blocked grids [85], and unstructured grids can be applied to simplify the grid generation, provide geometric flexibility, and even provide mesh refinement. Several methods have been investigated for unstructured grids, Refs. [86–89]. These methods require more memory and computational time and fall short of their structured counterpart in terms of efficiency and accuracy [87]. The theory and algorithms for unstructured grids have to evolve before they can be used for solving practical three-dimensional problems.

Explicit-upwind schemes appear to be a good compromise between explicit-central-difference schemes, and implicit-upwind schemes. Schemes constructed along these lines combine the advantages of: simplicity; prudent use of computational resources; and accuracy in resolving the flow field. Upwind schemes are more complex and are usually reported to be better suited for compressible viscous computation. Upwind schemes are very effective in converging to steady-state on single grids of modest complexity. Most of the currently used upwind schemes are implicit. Explicit schemes require less memory, and are easily implemented in a multi-block environment. They are also naturally suited for implementation on massively parallel computer architectures. The main drawback of explicit schemes is the limitation on the allowable time step.

If the explicit time-stepping scheme is augmented with suitable accelerating techniques, such as local time-stepping, residual smoothing and multigrid acceleration, the explicit method will be superior to its implicit counterpart. Variable coefficient residual smoothing will increase the stability range of the scheme, thus allowing the use of a higher CFL number (larger time step), which enhances the rate of convergence and removes the diffusion limit on the time step. Multigrid acceleration techniques will accelerate the convergence to steady-state by using large time steps on coarser grids, and help achieve convergence rates that are independent of the number of grid points [47]

Recent advances in computer architecture and algorithmic tools open the door for a new wave of opportunities for constructing explicit, upwind-higher-order schemes. Currently the existence of robust, multi-block, explicit, upwind schemes that can be applied on a routine basis are not available. Upwind-high-order schemes are essential tools, required to capture complicated physical phenomena associated with problems of current interest.

Explicit-upwind-schemes are still in their infancy and many basic issues are yet to be settled. In order to lay the foundation for future research, a joint analytical and numerical study should be conducted to validate and demonstrate their capabilities and performance.

1.2 Objective of Present Work

The goal of the present work was to develop a general state-of-the-art, multi-block algorithm, capable of solving the governing equations of fluid motion efficiently, for a wide range of configurations with both internal and external flow. The requisite algorithm should be simple, efficient, and robust. It is required to damp the high frequency component of the error (necessary for multigrid) effectively, while acquiring low levels of numerical dissipation for accurate predictions of viscous effects, and still maintaining high resolution on stretched grids. The developed algorithm will be used subsequently to simulate complex three-dimensional, steady and unsteady flow problems.

Hence, a control-volume, explicit-multistage-high-resolution upwind scheme, suitable for efficient computations using block structured grids, was desired. Upwind schemes were selected due to their high degree of reliability in viscous flow computations and their superior shock capturing capabilities [90]. The state variables at the cell interface have been determined by MUSCL interpolation using the so-called κ scheme. Two state-of-the-art, upwind schemes: Roe's flux-differencing and van Leer's flux-vector splitting schemes, were utilized to evaluate the inviscid flux at the cell interface. The viscous stress and heat flux terms in the governing equations have been centrally differenced.

In this study, the objective was to devise explicit, upwind time-stepping schemes that can be combined successfully with upwind-spatial operators. Explicit schemes have the merit of being computationally cheaper and easier to program and implement in a multi-block code. Two classes of upwind schemes: multistage time-stepping

schemes, and predictor-corrector schemes, were suggested and have been implemented in the developed algorithm. Modified Runge-Kutta methods with standard coefficients have been successful with central-difference spatial discretization. Yet, they have not performed as well with upwind differencing. The standard coefficients have to be modified to achieve better performance with upwind differencing.

The next objective was to augment the explicit time-stepping schemes with accelerating techniques, such as local time-stepping, implicit residual smoothing and the full approximation storage (FAS), to enhance the rate of convergence to steady-state.

Current aerodynamics designs are often quite complex (geometrically). Flexible computational tools are needed for the analysis of a wide range of configurations with both internal and external flows. Hence, another objective was the implementation of the developed algorithm in a multi-block code to allow for greater geometric flexibility.

The final goal was to validate the developed computer code on several test cases of interest to demonstrate and assess the predictive capability of the algorithm. The test cases considered were: corner flow, plume flow, laminar and turbulent flow over a flat plate, an ONERA M6 wing, and the unsteady three-dimensional flow of a jet impinging on a ground plane.

1.3 Thesis Outline

In chapter two, the mathematical formulation of the governing set of equations of motion (Reynolds-Averaged Navier-Stokes equation, Navier-Stokes equation, and Euler equation) are presented and discussed. Details of implementing the Baldwin-Lomax algebraic eddy viscosity turbulence model in the algorithm are presented. In chapter three, the finite volume formulation of the governing equations is presented. The MUSCL type differencing, and the type of discretization for the inviscid and viscous flux is discussed.

Two different upwind flux formulations —Roe’s flux-difference splitting, and van Leer’s flux-vector splitting— are presented, and practical issues concerning their implementation are discussed. In chapter four, the temporal discretization of the governing equations, which represents a major part of this work is presented. Two classes of explicit time integration schemes —multistage time-stepping schemes and predictor-corrector schemes— are presented and discussed. Details of optimizing the multistage explicit time integration scheme through local Fourier analysis of the scalar advection equation are presented. Accelerating techniques, including local time-stepping, residual smoothing, and multigrid acceleration techniques are presented in chapter five. In chapter six, the multi-block capability of the developed algorithm is presented. The interaction between multigrid and multi-block implementations are discussed. The boundary conditions employed, in the developed algorithm, are presented within the framework of multi-block. Several test cases of general interest to the computational fluid dynamics community were conducted to validate, demonstrate and assess the performance and predictive capability of the present algorithm. Results of these computations are reported in chapter seven. The Conclusions for the present research work, and recommendations for future research are presented in chapter eight.

CHAPTER 2

GOVERNING EQUATIONS

The governing equations were derived from the basic principles of conservation of mass, conservation of momentum, and energy. The conservation laws were then coupled with the thermodynamic properties and constitutive equations to yield the governing set of equations for fluid motion. The derivation of the governing equations can be found in [91, 92]. Three different sets of governing equations have been used pertaining to the different test cases investigated in this study. These sets of equations are the Euler equations, Navier-Stokes equations, and Reynolds-Averaged Navier-Stokes equations. Each set is represented ultimately as an algebraic set of equations. The three sets of governing equations have been implemented in the numerical algorithm. Coupled with the appropriate set of boundary conditions, the developed algorithm is capable of computing inviscid, laminar, and turbulent fluid flows numerically.

2.1 Navier-Stokes Equations

The time-dependent, compressible, three-dimensional, Navier-Stokes equations in Cartesian Coordinates, written in strong conservation form (neglecting the body forces and external heat sources) are:

$$\frac{\partial \hat{Q}}{\partial \hat{t}} + \frac{\partial \{ \hat{F} - \hat{F}_v \}}{\partial \hat{x}} + \frac{\partial \{ \hat{G} - \hat{G}_v \}}{\partial \hat{y}} + \frac{\partial \{ \hat{H} - \hat{H}_v \}}{\partial \hat{z}} = 0 \quad (2.1)$$

where \hat{Q} is the state vector of the dependent variables, given by:

$$\hat{Q} = \begin{Bmatrix} \hat{\rho} \\ \hat{\rho}\hat{u} \\ \hat{\rho}\hat{v} \\ \hat{\rho}\hat{w} \\ \hat{E} \end{Bmatrix} = \begin{Bmatrix} \text{Density} \\ x - \text{momentum} \\ y - \text{momentum} \\ z - \text{momentum} \\ \text{Total energy} \end{Bmatrix} \quad (2.2)$$

\hat{F} , \hat{F}_v , \hat{G} , \hat{G}_v , \hat{H} , and \hat{H}_v are the dimensional flux-vectors. They are function of the state variable vector, \hat{Q} , and are given by:

$$\hat{F} = \begin{Bmatrix} \hat{u} \\ \hat{\rho}\hat{u}^2 + \hat{p} \\ \hat{\rho}\hat{u}\hat{v} \\ \hat{\rho}\hat{u}\hat{w} \\ (\hat{E} + \hat{p})\hat{u} \end{Bmatrix}, \quad \hat{F}_v = \begin{Bmatrix} 0 \\ \hat{\tau}_{xx} \\ \hat{\tau}_{xy} \\ \hat{\tau}_{xz} \\ \hat{u}\hat{\tau}_{xx} + \hat{v}\hat{\tau}_{xy} + \hat{w}\hat{\tau}_{xz} - \hat{q}_x \end{Bmatrix} \quad (2.3)$$

$$\hat{G} = \begin{Bmatrix} \hat{v} \\ \hat{\rho}\hat{u}\hat{v} \\ \hat{\rho}\hat{v}^2 + \hat{p} \\ \hat{\rho}\hat{v}\hat{w} \\ (\hat{E} + \hat{p})\hat{v} \end{Bmatrix}, \quad \hat{G}_v = \begin{Bmatrix} 0 \\ \hat{\tau}_{xy} \\ \hat{\tau}_{yy} \\ \hat{\tau}_{yz} \\ \hat{u}\hat{\tau}_{xy} + \hat{v}\hat{\tau}_{yy} + \hat{w}\hat{\tau}_{yz} - \hat{q}_y \end{Bmatrix} \quad (2.4)$$

$$\hat{H} = \begin{Bmatrix} \hat{w} \\ \hat{\rho}\hat{u}\hat{w} \\ \hat{\rho}\hat{v}\hat{w} \\ \hat{\rho}\hat{w}^2 + \hat{p} \\ (\hat{E} + \hat{p})\hat{w} \end{Bmatrix}, \quad \hat{H}_v = \begin{Bmatrix} 0 \\ \hat{\tau}_{xz} \\ \hat{\tau}_{yz} \\ \hat{\tau}_{zz} \\ \hat{u}\hat{\tau}_{xz} + \hat{v}\hat{\tau}_{yz} + \hat{w}\hat{\tau}_{zz} - \hat{q}_z \end{Bmatrix} \quad (2.5)$$

The first row in the vector differential equation, eq. 2.1, is the conserved form of the continuity equation, while the fifth row is the conserved form of the energy equation. The second, third and fourth row are the conserved forms of the momentum equation in the x , y , and z directions, respectively. It should be emphasized that while the conservation of mass and energy are scalar equations, the conservation of momentum is a vector equation with three components. In the absolute sense only the x -momentum, y -momentum, and z -momentum are the Navier-Stokes equations of fluid mechanics. It is customary within the computational fluid mechanics arena to refer to eq. 2.1 as the Navier-Stokes equations. This terminology will be adopted in this study.

Fluid density is designated as ρ ; u , v , and w are the velocity components in the x , y , and z directions, respectively; E is the total energy; τ_{ij} are the components of the shear stress tensor; q_x , q_y , and q_z are the components of the heat flux-vector in the x , y , and z directions respectively; and T is the temperature. The superscript “ \wedge ” in vector eq. 2.1 indicates a dimensional quantity.

In this study we assume that the stress is linearly dependent on the rate of strain; i.e., the Newtonian fluid assumption is adopted. The components of the viscous stress tensor in Cartesian coordinates are given by

$$\begin{aligned}
 \hat{\tau}_{\hat{x}\hat{x}} &= 2\hat{\mu}\frac{\partial\hat{u}}{\partial\hat{x}} + \hat{\lambda}\left(\frac{\partial\hat{u}}{\partial\hat{x}} + \frac{\partial\hat{v}}{\partial\hat{y}} + \frac{\partial\hat{w}}{\partial\hat{z}}\right), \\
 \hat{\tau}_{\hat{y}\hat{y}} &= 2\hat{\mu}\frac{\partial\hat{v}}{\partial\hat{y}} + \hat{\lambda}\left(\frac{\partial\hat{u}}{\partial\hat{x}} + \frac{\partial\hat{v}}{\partial\hat{y}} + \frac{\partial\hat{w}}{\partial\hat{z}}\right), \\
 \hat{\tau}_{\hat{z}\hat{z}} &= 2\hat{\mu}\frac{\partial\hat{w}}{\partial\hat{z}} + \hat{\lambda}\left(\frac{\partial\hat{u}}{\partial\hat{x}} + \frac{\partial\hat{v}}{\partial\hat{y}} + \frac{\partial\hat{w}}{\partial\hat{z}}\right), \\
 \hat{\tau}_{\hat{x}\hat{y}} &= \hat{\mu}\left(\frac{\partial\hat{u}}{\partial\hat{y}} + \frac{\partial\hat{v}}{\partial\hat{x}}\right) = \hat{\tau}_{\hat{y}\hat{x}}, \\
 \hat{\tau}_{\hat{x}\hat{z}} &= \hat{\mu}\left(\frac{\partial\hat{u}}{\partial\hat{z}} + \frac{\partial\hat{w}}{\partial\hat{x}}\right) = \hat{\tau}_{\hat{z}\hat{x}}, \\
 \hat{\tau}_{\hat{y}\hat{z}} &= \hat{\mu}\left(\frac{\partial\hat{v}}{\partial\hat{z}} + \frac{\partial\hat{w}}{\partial\hat{y}}\right) = \hat{\tau}_{\hat{z}\hat{y}},
 \end{aligned} \tag{2.6}$$

μ is the first coefficient of viscosity, and λ is the second coefficient of viscosity. To date, the value for λ for air, and how to model it, especially for compressible flows, is still disputed [93]. In this study we employ Stokes hypothesis; i.e., we assume the bulk viscosity, K , is zero or negligible

$$K = \lambda + \frac{2}{3}\mu = 0 \tag{2.7}$$

Stokes hypothesis is not necessarily endorsed, but for lack of a better model, it has been employed. It is understood that this assumption is not valid in shock regions and in regions of high gradients [94]. Invoking Stokes hypothesis yields the following expression for

the shear stress terms

$$\begin{aligned}
\hat{\tau}_{\hat{x}\hat{x}} &= \frac{2}{3}\hat{\mu}\left(2\frac{\partial\hat{u}}{\partial\hat{x}} - \frac{\partial\hat{v}}{\partial\hat{y}} - \frac{\partial\hat{w}}{\partial\hat{z}}\right), \\
\hat{\tau}_{\hat{y}\hat{y}} &= \frac{2}{3}\hat{\mu}\left(2\frac{\partial\hat{v}}{\partial\hat{y}} - \frac{\partial\hat{u}}{\partial\hat{x}} - \frac{\partial\hat{w}}{\partial\hat{z}}\right), \\
\hat{\tau}_{\hat{z}\hat{z}} &= \frac{2}{3}\hat{\mu}\left(2\frac{\partial\hat{w}}{\partial\hat{z}} - \frac{\partial\hat{u}}{\partial\hat{x}} - \frac{\partial\hat{v}}{\partial\hat{y}}\right), \\
\hat{\tau}_{\hat{x}\hat{y}} &= \hat{\mu}\left(\frac{\partial\hat{u}}{\partial\hat{y}} + \frac{\partial\hat{v}}{\partial\hat{x}}\right) = \hat{\tau}_{\hat{y}\hat{x}}, \\
\hat{\tau}_{\hat{x}\hat{z}} &= \hat{\mu}\left(\frac{\partial\hat{u}}{\partial\hat{z}} + \frac{\partial\hat{w}}{\partial\hat{x}}\right) = \hat{\tau}_{\hat{z}\hat{x}}, \\
\hat{\tau}_{\hat{y}\hat{z}} &= \hat{\mu}\left(\frac{\partial\hat{v}}{\partial\hat{z}} + \frac{\partial\hat{w}}{\partial\hat{y}}\right) = \hat{\tau}_{\hat{z}\hat{y}},
\end{aligned} \tag{2.8}$$

The first coefficient of dynamic viscosity varies with temperature and can be approximated by an approximation of Sutherland's law [93, 95]

$$\frac{\hat{\mu}}{\hat{\mu}_{ref}} = \left(\frac{\hat{T}}{\hat{T}_{ref}}\right)^n \tag{2.9}$$

where $\hat{\mu}_{ref}$ and \hat{T}_{ref} are the dynamic viscosity and temperature at reference conditions. This formulation is simple and gives reasonably good results. The parameter n is taken to be 0.76.

The heat flux is modeled by Fourier's law of heat conduction, where

$$\hat{q}_{\hat{x}} = -\hat{k}\frac{\partial\hat{T}}{\partial\hat{x}}, \quad \hat{q}_{\hat{y}} = -\hat{k}\frac{\partial\hat{T}}{\partial\hat{y}}, \quad \text{and} \quad \hat{q}_{\hat{z}} = -\hat{k}\frac{\partial\hat{T}}{\partial\hat{z}} \tag{2.10}$$

where \hat{k} is the thermal conductivity which will be represented by

$$\hat{k} = \frac{\hat{c}_p\hat{\mu}}{Pr} \tag{2.11}$$

Here, Pr is the Prandtl number, and c_p is the specific heat at constant pressure. The Prandtl number is nearly constant for most gases ($Pr = 0.72$ for air).

The equation of state for a perfect gas relates the pressure to density and temperature and is given by $\hat{p} = \hat{\rho}\hat{R}\hat{T}$, where \hat{R} is the gas constant, which relates the specific heats for an ideal gas by:

$$\hat{R} = \hat{c}_p - \hat{c}_v \tag{2.12}$$

with \hat{c}_v representing the specific heat at constant volume. If we assume further a calorically perfect gas

$$\hat{e} = \hat{c}_v \hat{T} \quad (2.13)$$

where \hat{e} is the internal energy. Now, neglecting the potential energy, the total energy, \hat{E} , can be defined as the sum of the internal energy and kinetic energy;

$$\hat{E} = \hat{\rho} \left[\hat{e} + \frac{1}{2} (\hat{u}^2 + \hat{v}^2 + \hat{w}^2) \right] \quad (2.14)$$

If we define $\gamma = \frac{\hat{c}_p}{\hat{c}_v}$ as the ratio of specific heats, the above equations can be combined to yield the relation between pressure and total energy

$$\begin{aligned} \hat{p} &= (\gamma - 1) \hat{\rho} \hat{e} = (\gamma - 1) \left[\hat{E} - \frac{1}{2} \hat{\rho} (\hat{u}^2 + \hat{v}^2 + \hat{w}^2) \right] \\ \hat{E} &= \frac{\hat{p}}{\gamma - 1} + \frac{1}{2} \hat{\rho} (\hat{u}^2 + \hat{v}^2 + \hat{w}^2) \end{aligned} \quad (2.15)$$

2.2 Normalization of the Governing Equations.

Computing in an appropriate non-dimensional or normalized domain has the advantage that all variables are of the same order of magnitude, which enhances the performance and accuracy of numerical algorithms. Normalization eliminates the physical dimensions from the equations. Thus allowing general characteristic parameters such as Reynolds number, Mach number and Prandtl number to be changed independently. Hence parametric studies on any of these characteristic parameters can be performed easily.

Different variables or combinations of variables can be used in the normalization procedure. In this study we define the non-dimensional flow variables to be

$$\begin{aligned}
 x &= \frac{\hat{x}}{\hat{L}}, & y &= \frac{\hat{y}}{\hat{L}}, & z &= \frac{\hat{z}}{\hat{L}}, & t &= \frac{\hat{t} \hat{a}_{ref}}{\hat{L}}, \\
 u &= \frac{\hat{u}}{\hat{a}_{ref}}, & v &= \frac{\hat{v}}{\hat{a}_{ref}}, & w &= \frac{\hat{w}}{\hat{a}_{ref}}, & \mu &= \frac{\hat{\mu}}{\hat{\mu}_{ref}}, \\
 \rho &= \frac{\hat{\rho}}{\hat{\rho}_{ref}}, & p &= \frac{\hat{p}}{\hat{\rho}_{ref} \hat{a}_{ref}^2}, & T &= \frac{\hat{T}}{\hat{T}_{ref}}, \\
 E &= \frac{\hat{E}}{\hat{a}_{ref}^2}, & H &= \frac{\hat{H}}{\hat{a}_{ref}^2}
 \end{aligned} \tag{2.16}$$

Here \hat{a}_{ref} is the reference speed of sound; H is the total enthalpy, and L is a reference length. The subscript *ref* indicates the reference condition. By substituting the non-dimensional flow variables into the Navier-Stokes equations, eq. 2.1 we get;

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \{ \tilde{F} - \tilde{F}_v \}}{\partial x} + \frac{\partial \{ \tilde{G} - \tilde{G}_v \}}{\partial y} + \frac{\partial \{ \tilde{H} - \tilde{H}_v \}}{\partial z} = 0 \tag{2.17}$$

where \tilde{Q} , \tilde{F} , and \tilde{F}_v are given by

$$\tilde{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix}, \quad \tilde{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{Bmatrix}, \quad \tilde{F}_v = \frac{M_{ref}}{R_{ref}} \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \frac{\mu}{\beta P_r} \frac{\partial T}{\partial x} \end{Bmatrix} \tag{2.18}$$

here

$$\begin{aligned}
 M_{ref} &= \frac{\hat{q}_{ref}}{\hat{a}_{ref}} \\
 R_{ref} &= \frac{\hat{\rho}_{ref} \hat{q}_{ref} L}{\hat{\mu}_{ref}} \\
 \beta &= \gamma - 1
 \end{aligned} \tag{2.19}$$

where M_{ref} , and R_{ref} are the reference Mach number and Reynolds number respectively, while \hat{q}_{ref} is the velocity magnitude at reference conditions. Similar expressions can be developed for \tilde{G} , \tilde{G}_v , \tilde{H} , and \tilde{H}_v .

2.3 Curvilinear Coordinate Transformation

Most practical fluid flow problems of interest, are solved in domains with irregular shapes and boundaries. This causes difficulty in implementing the boundary conditions. In regions of high gradients, (shock waves, vortex regions, and shear layers), one needs to pack grid points in order to capture details of the flow field, and render accurate results. The uneven packing of grid points complicates the differencing operator. To avoid these difficulties, the governing equations can be transformed into a body fitted coordinate system, thus simplifying the numerical differencing and the implementation of boundary conditions.

The curvilinear coordinate system is assumed related to the Cartesian coordinates by

$$\begin{aligned}\xi &= \xi\{x, y, z\}, & \eta &= \eta\{x, y, z\}, & \zeta &= \zeta\{x, y, z\} \\ \frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta}\end{aligned}\tag{2.20}$$

The transformation matrices are given by [4];

$$\begin{aligned}\xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta), & \xi_y &= -J(x_\eta z_\zeta - x_\zeta z_\eta), & \xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta) \\ \eta_x &= -J(y_\xi z_\zeta - y_\zeta z_\xi), & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi), & \eta_z &= -J(x_\xi y_\zeta - x_\zeta y_\xi) \\ \zeta_x &= J(y_\xi z_\eta - y_\eta z_\xi), & \zeta_y &= -J(x_\xi z_\eta - x_\eta z_\xi), & \zeta_z &= J(x_\xi y_\eta - x_\eta y_\xi)\end{aligned}\tag{2.21}$$

where J is the Jacobian of the transformation;

$$J = \frac{\partial\{\xi, \eta, \zeta\}}{\partial\{x, y, z\}} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix}\tag{2.22}$$

These formulations allow the governing equations to be written as;

$$\frac{\partial Q}{\partial t} + \frac{\partial\{F - F_v\}}{\partial \xi} + \frac{\partial\{G - G_v\}}{\partial \eta} + \frac{\partial\{H - H_v\}}{\partial \zeta} = 0\tag{2.23}$$

where

$$\begin{aligned} Q &= \frac{\tilde{Q}}{J}, \quad F = \frac{1}{J}(\xi_x \tilde{F} + \xi_y \tilde{G} + \xi_z \tilde{H}), \quad F_v = \frac{1}{J}(\xi_x \tilde{F}_v + \xi_y \tilde{G}_v + \xi_z \tilde{H}_v) \\ G &= \frac{1}{J}(\eta_x \tilde{F} + \eta_y \tilde{G} + \eta_z \tilde{H}), \quad G_v = \frac{1}{J}(\eta_x \tilde{F}_v + \eta_y \tilde{G}_v + \eta_z \tilde{H}_v) \\ H &= \frac{1}{J}(\zeta_x \tilde{F} + \zeta_y \tilde{G} + \zeta_z \tilde{H}), \quad H_v = \frac{1}{J}(\zeta_x \tilde{F}_v + \zeta_y \tilde{G}_v + \zeta_z \tilde{H}_v) \end{aligned} \quad (2.24)$$

with

$$F = \begin{Bmatrix} \rho U \\ \rho u U + p \xi_x \\ \rho v U + p \xi_y \\ \rho w U + p \xi_z \\ (E + p)U \end{Bmatrix}, \quad G = \begin{Bmatrix} \rho V \\ \rho u V + p \eta_x \\ \rho v V + p \eta_y \\ \rho w V + p \eta_z \\ (E + p)V \end{Bmatrix}, \quad H = \begin{Bmatrix} \rho W \\ \rho u W + p \zeta_x \\ \rho v W + p \zeta_y \\ \rho w W + p \zeta_z \\ (E + p)W \end{Bmatrix} \quad (2.25)$$

where,

$$\begin{aligned} U &= u \xi_x + v \xi_y + w \xi_z \\ V &= u \eta_x + v \eta_y + w \eta_z \\ W &= u \zeta_x + v \zeta_y + w \zeta_z \end{aligned} \quad (2.26)$$

F_v , G_v , and H_v , are presented in Appendix A.

2.4 Thin-Layer Navier-Stokes Equations

The Navier-Stokes equations govern the motion of unsteady compressible fluid flow. The solution of the Navier-Stokes equations require a fine grid to capture the diffusive effects. Performing the computations on a fine grid requires extensive amounts of computer time and memory. At high Reynolds numbers, the effect of viscosity is confined to a thin region near solid walls where a boundary layer exists. The dominant viscous effects in the boundary layer arise from viscous diffusion normal to the body surface. A desirable approximation is to neglect the viscous terms containing derivatives in directions which are tangent to the body surface [26]. This assumption is often justified since the viscous terms containing derivatives in directions parallel to a solid boundary are usually substantially smaller than the terms with derivatives normal to the boundary. It would

also be impractical to think of a fine grid in all three directions. Viscous grids are usually dense along only the solid walls. Thus it makes sense to neglect the terms that are not properly resolved, especially if they are an order of magnitude smaller than other viscous terms.

The thin-layer Navier-Stokes equations are derived from the Navier-Stokes equations by neglecting all cross derivatives in the viscous fluxes F_v , G_v , and H_v . For example all derivatives with respect to η and ξ in the F_v viscous flux are neglected. Similarly for G_v , and H_v .

During development of the numerical algorithm, it was desired to maintain generality. Since it is not known *a priori* which direction will coincide with the solid boundary, or whether there will be more than one boundary surface, the thin shear layer approximation was applied in all three directions. The thin shear layer equations used in the developed algorithm are given by

$$\frac{\partial Q}{\partial t} + \frac{\partial \{F - F_v\}}{\partial \xi} + \frac{\partial \{G - G_v\}}{\partial \eta} + \frac{\partial \{H - H_v\}}{\partial \zeta} = 0 \quad (2.27)$$

where Q is the state vector of dependent variables. F , G , and H are the inviscid fluxes, described in eq. 2.25, and

$$F_v = \frac{M_{ref}\mu}{R_{ref}} \left\{ \begin{array}{c} 0. \\ u_\xi(\phi^2) + \xi_x \Phi \\ v_\xi(\phi^2) + \xi_y \Phi \\ w_\xi(\phi^2) + \xi_z \Phi \\ \phi^2(uu_\xi + vv_\xi + ww_\xi) + \Phi U + \frac{1}{\beta P_r} T_\xi \phi^2 \end{array} \right\} \quad (2.28)$$

$$\text{where, } \phi^2 = \xi_x^2 + \xi_y^2 + \xi_z^2, \quad \text{and, } \Phi = \frac{1}{3}(u_\xi \xi_x + v_\xi \xi_y + w_\xi \xi_z)$$

$$G_v = \frac{M_{ref}\mu}{R_{ref}} \left\{ \begin{array}{c} 0 \\ u_\eta(\theta^2) + \eta_x \Theta \\ v_\eta(\theta^2) + \eta_y \Theta \\ w_\eta(\theta^2) + \eta_z \Theta \\ \theta^2(uu_\eta + vv_\eta + ww_\eta) + \Theta V + \frac{1}{\beta Pr} T_\eta \theta^2 \end{array} \right\} \quad (2.29)$$

$$\text{where, } \theta^2 = \eta_x^2 + \eta_y^2 + \eta_z^2, \quad \text{and, } \Theta = \frac{1}{3}(u_\eta \eta_x + v_\eta \eta_y + w_\eta \eta_z)$$

$$\text{where, } \theta^2 = \eta_x^2 + \eta_y^2 + \eta_z^2, \quad \text{and, } \Theta = \frac{1}{3}(u_\eta \eta_x + v_\eta \eta_y + w_\eta \eta_z)$$

Finally,

$$H_v = \frac{M_{ref}\mu}{R_{ref}} \left\{ \begin{array}{c} 0 \\ u_\zeta(\varphi^2) + \zeta_x \psi \\ v_\zeta(\varphi^2) + \zeta_y \psi \\ w_\zeta(\varphi^2) + \zeta_z \psi \\ \varphi^2(uu_\zeta + vv_\zeta + ww_\zeta) + \psi W + \frac{1}{\beta Pr} T_\zeta \varphi^2 \end{array} \right\} \quad (2.30)$$

$$\text{where, } \varphi^2 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2, \quad \text{and, } \psi = \frac{1}{3}(u_\zeta \zeta_x + v_\zeta \zeta_y + w_\zeta \zeta_z)$$

Based on the type of problem computed one or more viscous fluxes can be neglected [76, 96].

2.5 Reynolds-Averaged Navier-Stokes Equations

Almost all flows encountered in fluid mechanics are either fully turbulent or partially turbulent. The nature of the flow and the purpose of the numerical study dictate the accuracy levels for modeling turbulent effects or the justification for neglecting turbulent effects completely (and simply assuming laminar flow). Turbulence enhances the rate of heat transfer, and alters the skin friction. Turbulence also affects the location of flow separation, the mechanism for separation, and the size of the separation bubble. Surface pressure forces, lift and drag, are also affected by the level of turbulence in the flow.

Turbulent flows are in principle still governed by the Navier-Stokes equations, eq. 2.23; however extremely fine grids and higher order schemes are required to resolve all time and length scales that accompany realistic turbulent flows. This type of computation

is known as Direct Numerical Simulation; it represents a challenge to today's computers and numerical algorithms. Direct Numerical Simulation (DNS) has been restricted to low Reynolds number flows since the number of grid points required is proportional to the $9/4$ power of Reynolds number, [97]. The significant cost of DNS calculations, even for simple flows, makes them impractical for current engineering applications. Perhaps with the development of new computer architectures, and with more parallel machines, DNS will become a practical approach. From that point of view, if the grid is coarse and must still resolve the mean details of turbulent motion, then we must resort to modeling the turbulent effects by superimposing them on the mean flow. At present, turbulence modeling forms the basis of most of the computational work in turbulent flows.

Hinze [98] best described turbulent flow as *"Turbulent fluid motion is an irregular condition of flow in which the various quantities show random variation with time and space coordinates so that statistically distinct average values can be discerned"*. Following the footsteps of Reynolds, we decompose the randomly changing flow properties into mean and fluctuating components

$$q = \bar{q} + \dot{q} \quad (2.31)$$

where q is the property being decomposed and \dot{q} is the fluctuating component; \bar{q} is the mean property defined by

$$\bar{q} = \frac{1}{\Delta t} \int_{t_0}^{t_0 + \Delta t} q \, dt \quad (2.32)$$

Δt is a time interval which is long compared to the period of any significant turbulent fluctuations, but Δt is assumed to be short compared to the time scales associated with the mean flow. If we apply the decomposition procedure to all state variables in the Navier-Stokes equations, we get the Reynolds-Averaged Navier-Stokes equations which

work well for simple incompressible flows. For compressible flows, triple correlations involving density appear in the equations. Favré [98] suggested a mass weighted decomposition for compressible flows to avoid the triple correlation involving density. The following formulations were used to decompose the flow variables in the Navier-Stokes equations 2.1.

$$\begin{aligned}\tilde{q} &= \frac{\overline{\rho q}}{\bar{\rho}}, \quad \tilde{u} = \frac{\overline{\rho u}}{\bar{\rho}}, \quad \tilde{T} = \frac{\overline{\rho T}}{\bar{\rho}}, \quad \tilde{H} = \frac{\overline{\rho H}}{\bar{\rho}} \\ \text{where } u &= \tilde{u} + \hat{u}, \quad T = \tilde{T} + \hat{T}, \quad H = \tilde{H} + \hat{H} \\ \text{note } p &= \bar{p} + \acute{p}, \quad \rho = \bar{\rho} + \acute{\rho} \\ \text{while } (\bar{\rho} + \acute{\rho})\hat{q} &= 0, \\ \text{but } \tilde{\hat{q}} &\neq 0\end{aligned}\tag{2.33}$$

Substituting the above formulations into the Navier-Stokes equations and averaging in time, we get the mass averaged Navier-Stokes equation. The details of this procedure are presented in Appendix B. The non-dimensional mass averaged Navier-Stokes equation is given by

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial (\tilde{F} - \tilde{F}_v)}{\partial x} + \frac{\partial (\tilde{G} - \tilde{G}_v)}{\partial y} + \frac{\partial (\tilde{H} - \tilde{H}_v)}{\partial z} = 0\tag{2.34}$$

where

$$\tilde{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix}\tag{2.35}$$

$$\tilde{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{Bmatrix}, \quad \tilde{F}_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_T) \begin{Bmatrix} 0.0 \\ \frac{2}{3} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ u \frac{2}{3} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) + v \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + w \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \sigma \frac{\partial T}{\partial x} \end{Bmatrix}\tag{2.36}$$

$$\tilde{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E + p)v \end{Bmatrix}, \quad \tilde{G}_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_T) \begin{Bmatrix} 0.0 \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{2}{3} \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ u \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + v \frac{2}{3} \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) + \\ w \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \sigma \frac{\partial T}{\partial y} \end{Bmatrix} \quad (2.37)$$

$$\tilde{H} = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E + p)w \end{Bmatrix}, \quad \tilde{H}_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_T) \begin{Bmatrix} 0.0 \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{2}{3} \left(2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\ u \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + v \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \\ w \frac{2}{3} \left(2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) + \sigma \frac{\partial T}{\partial z} \end{Bmatrix} \quad (2.38)$$

$$\text{where } \sigma = \frac{1}{\beta(\mu + \mu_T)} \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \quad (2.39)$$

μ is the molecular viscosity, and μ_T is the eddy viscosity. The eddy viscosity is supplied by the turbulence model. Pr is the laminar Prandtl number and Pr_T is the turbulent Prandtl number. For air, we take $Pr = 0.72$ and $Pr_T = 0.9$

Replacing μ in the Navier-Stokes equations, eq. 2.17, with $\mu + \mu_T$ and replacing $\frac{1}{\beta Pr}$ with $\frac{1}{\beta(\mu + \mu_T)} \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right)$ yields the modeled Reynolds-Averaged Navier-Stokes equations. Thus one can conclude that the mathematical formulations of the two sets of equation are similar.

2.6 Baldwin-Lomax Algebraic-Turbulence Model

In this study, the Baldwin-Lomax turbulence model was selected to model the eddy viscosity. The model is a two layer eddy viscosity model which implements a simple algebraic expression to determine the turbulent eddy viscosity [99].

The inner layer eddy viscosity model is given by ;

$$\mu_{Tinner} = \rho l^2 |\omega| \frac{R_{ref}}{M_{ref}} \quad (2.40)$$

where

$$l = k_1 y \left[1 - \exp \left\{ -\frac{y^+}{A^+} \right\} \right] \quad (2.41)$$

$$|\omega| = \sqrt{\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} \right)^2} \quad (2.42)$$

and

$$y^+ = y \sqrt{\frac{\rho_w \omega_{max}}{\mu_w}} \sqrt{\frac{R_{ref}}{M_{ref}}} \quad (2.43)$$

Where, $|\omega|$ is the magnitude of the vorticity; y is the normal distance from the nearest solid wall, k_1 is a constant equal to 0.4, A^+ is a constant equal to 26.0, ω_{max} is the maximum vorticity in a local vorticity profile along the coordinate direction normal to the wall, ρ_w is the density at the wall; and μ_w is the molecular viscosity at the wall, The original Baldwin-Lomax algebraic-turbulence model did not implement ω_{max} in the formulation but rather suggested using the shear stress at the wall, τ_w . If there is separation, implementing the shear stress at the wall yields inaccurate values for the turbulence model. If there isn't any flow separation on the wall, it can be shown that ω_{max} equals approximately τ_w . The second set of equations for the outer layer of the Baldwin-Lomax algebraic-turbulence model are given as;

$$\mu_{Touter} = K_2 C_{cp} \rho F_{wake} F_{kleb} \frac{R_{ref}}{M_{ref}} \quad (2.44)$$

where

$$F_{wake} = \min \left\{ \frac{y_{max} F_{max}}{\frac{C_{wk} y_{max} V_{dif}^2}{F_{max}}} \right\} \quad (2.45)$$

with the closure constant, $K_2 = 0.0168$, $C_{cp} = 1.6$, $C_{wk} = 1.0$ for transonic flow, and

$$V_{dif} = \left(\sqrt{u^2 + v^2 + w^2} \right)_{|_{max}} - \left(\sqrt{u^2 + v^2 + w^2} \right)_{|_{min}} \quad (2.46)$$

along the coordinate perpendicular to the surface at a particular wall location. For example, equation 2.46 would be applied along a constant x-surface, if x is the streamwise direction. The value y_{max} corresponds to the location with $F(y) = F(y)_{max}$, where

$$F(y) = y|\omega| \left[1 - \exp \left\{ -\frac{y^+}{A^+} \right\} \right] \quad (2.47)$$

In wake regions, the exponential term for $F(y)$ is set to zero. Alternatively, the Klebanoff intermittency factor is used where;

$$F_{KLEB} = \left[1 + 5.5 \left[\frac{y C_{KLEB}}{y_{max}} \right]^6 \right]^{-1} \quad (2.48)$$

and $C_{KLEB} = 0.3$.

2.7 Euler Equations

In this section the governing equations for inviscid compressible fluid flow are presented. This governing set of equations is known as the Euler Equations. The Euler equations are considered as an approximation to the Navier-Stokes Equations. They are derived by neglecting the viscous and heat flux terms from the Navier-Stokes Equations, eq. 2.23. Euler Equations are useful models in fluid flow problems where only information on the pressure distribution is required. Numerical results for the Euler equations are important in preliminary studies and design, and the Euler equations are capable of capturing shock wave interactions and contact discontinuities accurately.

The three-dimensional, time-dependent, compressible Euler equations are written in general curvilinear coordinates (ξ, η, ζ) in a non-dimensional conservative form as :

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \quad (2.49)$$

where Q is the vector of dependent variables and is given by ;

$$Q = \frac{1}{J} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} \quad (2.50)$$

F , G , and H are the inviscid flux vectors which are functions of the state vector Q and are given by

$$F = \begin{pmatrix} \rho U \\ \rho u U + p \xi_x \\ \rho v U + p \xi_y \\ \rho w U + p \xi_z \\ (E + p)U \end{pmatrix}, \quad G = \begin{pmatrix} \rho V \\ \rho u V + p \eta_x \\ \rho v V + p \eta_y \\ \rho w V + p \eta_z \\ (E + p)V \end{pmatrix}, \quad \text{and} \quad H = \begin{pmatrix} \rho W \\ \rho u W + p \zeta_x \\ \rho v W + p \zeta_y \\ \rho w W + p \zeta_z \\ (E + p)W \end{pmatrix} \quad (2.51)$$

The Euler equations are a set of first-order hyperbolic equations in time. For steady-state calculations, the mathematical nature of the equations changes from elliptic in the case of subsonic flow to hyperbolic for supersonic flow.

Another set of equations that model compressible, irrotational, inviscid fluid flow are the potential flow equations. The potential flow equations assume that the flow is irrotational ($\nabla \times q = 0$), and do not account for vorticity and entropy changes. Thus they are not capable of accurately capturing curved shocks, shock interactions or contact discontinuities [92]. Hence the Euler equations were selected as the set of equations that govern the inviscid compressible fluid flow in the numerical algorithm developed in this study.

CHAPTER 3

SPATIAL DISCRETIZATION

A control volume approach was employed in this study. The computational domain was divided into a finite number of hexahedral cells. Each cell has its own volume V and is bounded by its surface, S . Figure 3.1 shows a typical cell in the computational domain. Each cell has six quadrilateral faces; each face defined by four vertices. The

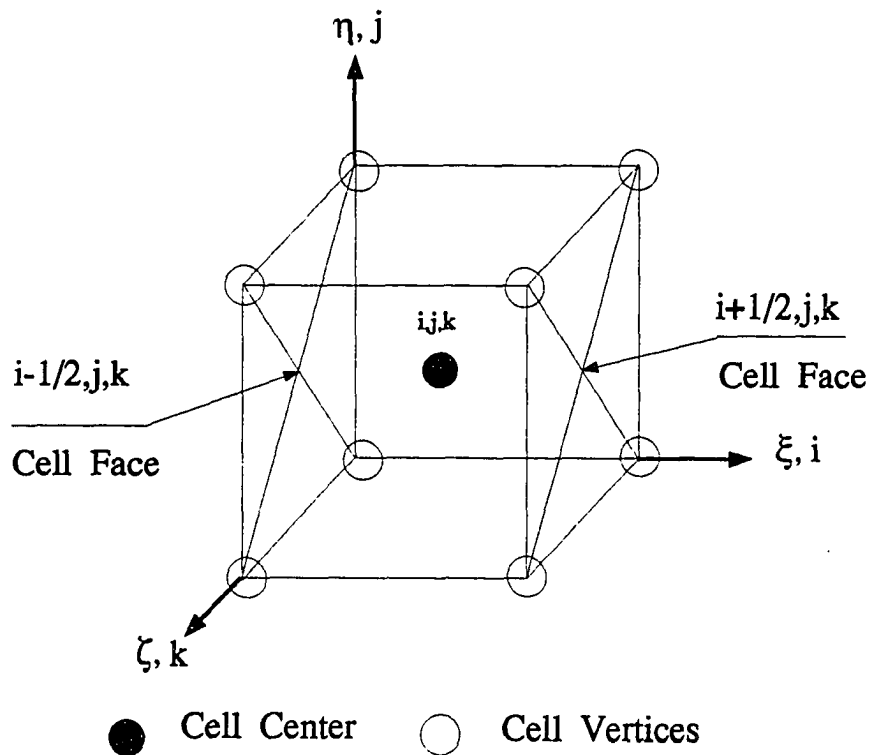


Figure 3 3.1 Schematic of Computational Cell

face of a hexahedral cell can collapse to an edge or even to a point, *i.e.*, the four vertices may lie at the same x, y, z location. By applying the basic principles of conservation of mass, momentum, and energy to a stationary cell in the computational domain we extract

the integral form of the governing equations

$$\frac{\partial}{\partial t} \int \int \int_V Q dV + \int \int_S \mathcal{F} \cdot n dS = 0 \quad (3.1)$$

where,

$$\mathcal{F} = (F - F_v)\hat{i} + (G - G_v)\hat{j} + (H - H_v)\hat{k}, \quad (3.2)$$

$$\text{and } n = n_x\hat{i} + n_y\hat{j} + n_z\hat{k}.$$

Q is the vector of conserved variables; F, F_v, G, G_v, H , and H_v are the flux-vectors which are defined in eq. 2.23; n is the outward pointing unit vector normal to the surface S , bounding the volume V . In the control volume approach, the state variables, $Q_{i,j,k}$, are stored in the center of each cell, and are considered to be a cell average rather than a pointwise value at the cell center

$$Q_{i,j,k} = \left\{ \frac{1}{V} \int \int \int_V Q dV \right\}_{i,j,k} \quad (3.3)$$

The merit of the integral form of the governing equations is that it is valid everywhere in the computational domain, even across shocks and contact surfaces, while the differential form of the equations are valid only in smooth regions. The use of the conservative law form permits shock waves to be captured as weak solutions to the governing equations where the discontinuities evolve as part of the solution, and are captured within one or more grid cells.

A semi-discrete form of the differential equation, eq. 2.23, can be written as

$$\begin{aligned} \frac{\partial Q_{i,j,k}}{\partial t} &+ (F - F_v)_{i+\frac{1}{2},j,k} - (F - F_v)_{i-\frac{1}{2},j,k} \\ &+ (G - G_v)_{i,j+\frac{1}{2},k} - (G - G_v)_{i,j-\frac{1}{2},k} \\ &+ (H - H_v)_{i,j,k+\frac{1}{2}} - (H - H_v)_{i,j,k-\frac{1}{2}} = 0 \end{aligned} \quad (3.4)$$

where $\Delta\xi, \Delta\eta$, and $\Delta\zeta$ are taken equal to unity for simplicity. If the surface integrals in eq. 3.1 are written as the sum of the contributions from the six faces of the computational

cell, then the relation between the integral form of the governing equation, eq. 3.1, and the semi-discrete form of the differential equation, eq. 3.4, becomes obvious. As a result, a geometrical interpretation of the Jacobian and metric terms of the transformation can be made. The Jacobian is calculated as the inverse of the cell volume; the vector $\frac{\nabla l}{J}$ is the directed area of the cell interface to a $l = \text{constant}$ coordinate direction, $l = \xi, \eta$, and ζ ; $|\frac{\nabla l}{J}|$ is the area of the cell interface; the unit vector $\frac{(l_x, l_y, l_z)}{|\nabla l|}$ consists of the direction cosines of the cell interface. The evaluation of the cell volume, and the directed area is given in Refs. [92, 100, 101].

The finite volume method, when coupled with an explicit scheme, has the advantage that the spatial discretization is decoupled from the temporal discretization. The temporal discretization will be discussed in detail in the next chapter, while in this chapter the construction of the interface flux, and the spatial discretization will be emphasized.

The interface flux can be constructed either by a central-difference scheme or an upwind-scheme. Upwind schemes attempt to construct the flux by modelling the underlying physics, as dictated by the sign of the characteristic waves. Central-difference schemes compute the interface flux as an average of the two adjacent cells, disregarding the characteristic theory. A comparison between central-difference schemes, and upwind schemes, as well as how they are related, is given by Swanson and Turkel in Ref. [56].

The numerical procedure employed for the evaluation of the interface flux utilized a second-order, central-difference approximation to compute the viscous flux (F_v, G_v, H_v), while a high resolution upwind shock capturing scheme was used to compute the inviscid flux (F, G, H). Roe's flux-differencing scheme and Van Leer's flux-vector splitting-scheme were applied to the conservative forms of the governing equations to evaluate the inviscid flux.

A Monotone Upstream-centered Scheme for Conservative Laws (MUSCL) was used to compute the inviscid flux at the cell interface. The state variables were extrapolated from the cell center to the cell interface and the interface flux was computed subsequently. An extrapolation of either the primitive or conservative variables can be performed. For the most part, the primitive variables were extrapolated in this study, because they render a smoother solution across shocks and slip lines and allow the use of a higher CFL number, as will be shown in the results section. The extrapolation operator of the state variables for the cell interface is based on the so-called κ -scheme formulation where the state variables to the right and left of the interface are computed as follows

$$\begin{aligned} Q_{i+\frac{1}{2}}^L &= Q_i + \frac{\phi}{4} [(1 - \kappa) \nabla_i + (1 + \kappa) \Delta_i] \\ Q_{i+\frac{1}{2}}^R &= Q_{i+1} - \frac{\phi}{4} [(1 - \kappa) \nabla_{i+1} + (1 + \kappa) \Delta_{i+1}] \end{aligned} \quad (19)$$

where, Δ_i and ∇_i are the forward and backward differences, respectively, defined as

$$\begin{aligned} \Delta_i &= Q_{i+1} - Q_i \\ \nabla_i &= Q_i - Q_{i-1} \end{aligned} \quad (20)$$

The value of the parameter κ determines the spatial accuracy of the scheme. Table 1 shows the conventional choices for κ , the corresponding accuracy and the truncation error based on one-dimensional spatial analyses [101]. The parameter, ϕ , is set equal to one for high-order extrapolations, and to zero for first-order extrapolation. Although first-order schemes possess good damping characteristics and allow a higher CFL number to be used, they are too diffusive. In the present study, $\kappa = 1/3$, $\kappa = -1$, and $\kappa = 0$ have been used.

High-order-accurate upwind schemes produce spurious oscillations and may develop instabilities near shocks and contact discontinuities. These oscillations can be reduced by the use of some kind of limiting procedures called limiters. This is achieved by imposing a constraint on the gradients of the dependent variables or on the flux functions. Several limiters are available in the literature [72, 102–104]. In the present study the min-mod

limiter and the differentiable Vanalbdba limiter have been employed. The two limiters are discussed in Appendix C. The limiters prevent numerical oscillations, but at the same time reduce the accuracy of the scheme to first order near discontinuities. The main disadvantage of limiters is that they lead to a limit cycle, with no apparent convergence. The residual will converge to a certain level and then “hang up”. To overcome this problem, the second-order strictly upwind formulation, $\kappa = -1$, was used which does not require a limiter.

In summary, the present algorithm has employed a cell-centered, finite-volume approach with the state variables at the cell interface determined by the MUSCL interpolation. The viscous and diffusion terms, at the cell interface, were discretized using a second-order, accurate, central-differencing scheme, while a higher-order-upwind-scheme (Roe’s flux-differencing scheme [66] or van Leer’s flux-vector splitting-scheme [59]) was used to construct the inviscid flux. Both Roe’s flux-differencing scheme and van Leer’s flux-vector splitting-scheme are capable of capturing relatively strong stationary —shocks within one or two interior cells— if the shock is reasonably aligned with the grid. It should be emphasized that the present algorithm employed a sequence of one-dimensional operators in all three coordinate directions ξ , η , and ζ . Thus the use of highly skewed, non-orthogonal grids should be avoided, if possible, because the one-dimensional opera-

Table 1 Values of κ and the Corresponding Truncation Error.

κ	Scheme	Second-Order Truncation Error
1/3	Third Order	0
-1	Fully Upwind	$\frac{1}{3}(\Delta\xi)^2 \frac{\partial^3 F}{\partial \xi^3}$
0	Fromm	$\frac{1}{12}(\Delta\xi)^2 \frac{\partial^3 F}{\partial \xi^3}$
1/2	Low TE Second Order	$-\frac{1}{24}(\Delta\xi)^2 \frac{\partial^3 F}{\partial \xi^3}$
0	Central	$-\frac{1}{6}(\Delta\xi)^2 \frac{\partial^3 F}{\partial \xi^3}$
-1/3	No Name	$\frac{1}{6}(\Delta\xi)^2 \frac{\partial^3 F}{\partial \xi^3}$

tors assume that the waves interact normal to cell interfaces. This method of extending one-dimensional schemes to multi-dimensions was found to be quite satisfactory for the test cases investigated in this study. A truly multi-dimensional approach is still in its infancy, and is computationally expensive [5, 105, 106].

3.1 van Leer's Flux-Vector Splitting

A prevalent way to introduce upwinding into the governing systems of hyperbolic conservation laws is to split the flux according to the characteristic speed $(q, q \pm c)$. In this study, van Leer's flux-vector splitting has been employed, because it yields sharp, crisp shock surface. A disadvantage of van Leer's scheme is that it smears slip lines because it ignores the linear waves (entropy and shear waves) [107]. A brief summary of the scheme is presented in this section. For more information about the scheme, the interested reader should review the original paper of van Leer [59].

Following Ref. [59], the flux-vectors F , G , and H can each be split into two vectors, a forward flux-vector, based on non-negative eigenvalues, and a backward flux-vector based on non-positive eigenvalues.

$$F = F^+ + F^-, \quad G = G^+ + G^-, \quad H = H^+ + H^- \quad (3.7)$$

For local supersonic Mach numbers:

$$\begin{aligned} M_l \geq 1.0, \quad F_l^+ = F_l, \quad F_l^- = 0 \\ M_l \leq -1.0, \quad F_l^+ = 0, \quad F_l^- = F_l \end{aligned} \quad (3.8)$$

where $l = \xi, \eta$, and ζ to indicate the three coordinate directions.

For subsonic local Mach numbers, $|M_l| < 1.0$ (in general notation for body fitted coordinates [53]), a local scaled contravariant velocity component \bar{u}_l is defined as

$$\bar{u}_l = \frac{l_x u + l_y v + l_z w}{\sqrt{l_x^2 + l_y^2 + l_z^2}} \quad l = \xi, \eta, \zeta \quad (3.9)$$

where the local Mach number is given as

$$M_l = \frac{\bar{u}_l}{a} \quad (3.10)$$

and a is the local speed of sound. The fluxes are:

$$F_l^\pm = \frac{1}{J} f_{mass}^\pm \left\{ \begin{array}{c} 1 \\ \hat{l}_x(-\bar{u}_l \pm 2a)/\gamma + u \\ \hat{l}_y(-\bar{u}_l \pm 2a)/\gamma + v \\ \hat{l}_z(-\bar{u}_l \pm 2a)/\gamma + w \\ f_{energy}^\pm \end{array} \right\} \quad (3.11)$$

where,

$$\hat{l}_n = \frac{l_n}{\sqrt{l_x^2 + l_y^2 + l_z^2}}, \quad n = x, y, \text{ and } z \quad (3.12)$$

$$f_{mass}^\pm = \pm \rho a \frac{1}{4} (M_l \pm 1)^2 \quad (3.13)$$

and

$$f_{energy}^\pm = \left[\frac{-\beta \bar{u}_l^2 \pm 2\beta \bar{u}_l a + 2a^2}{\gamma^2 - 1} + \frac{u^2 + v^2 + w^2}{2} \right] \quad (3.14)$$

Here;

$$F_\xi = F \quad F_\eta = G \quad F_\zeta = H \quad (3.15)$$

$$u_\xi = u \quad u_\eta = v \quad u_\zeta = w$$

and

$$\beta = \gamma - 1 \quad (3.17)$$

The “+” indicates a forward flux and the “-” indicates a backward flux.

Applying the split flux to the semi-discrete form of the governing equations, eq 2.27, gives:

$$\begin{aligned}
& \frac{\partial Q_{i,j,k}}{\partial t} + (F^+ + F^- - F_v)_{i+\frac{1}{2},j,k} - (F^+ + F^- - F_v)_{i-\frac{1}{2},j,k} \\
& + (G^+ + G^- - G_v)_{i,j+\frac{1}{2},k} - (G^+ + G^- - G_v)_{i,j-\frac{1}{2},k} \\
& + (H^+ + H^- - H_v)_{i,j,k+\frac{1}{2}} - (H^+ + H^- - H_v)_{i,j,k-\frac{1}{2}} = 0
\end{aligned} \tag{3.18}$$

The present formulation, when applied to transonic and low supersonic flows, does not require the use of flux limiters for essentially oscillation free shocks. This was pointed out by Anderson, Thomas, and van Leer [58], von Lavante and Haertl [108], Melson and von Lavante [109], and Cannizzaro, von Lavante, and Melson [110] and was explained in more detail by van Leer [59]

3.2 Roe's Flux-Difference Splitting

Roe's flux-difference splitting is an upwind scheme that approximates the Riemann problem at an interface between two cells by Roe's averaging procedure [66]. Roe's scheme provides an exact solution to an approximate Riemann problem, and is capable of handling slip lines with less smearing. The idea of advancing the solution to the next time-level by solving a Riemann problem at each cell interface was first introduced by Gudonov [64]. The Riemann problem and the different waves associated with it are illustrated in Fig. (3.2). A good review of the different Riemann solvers is given in Refs. [6, 72, 111].

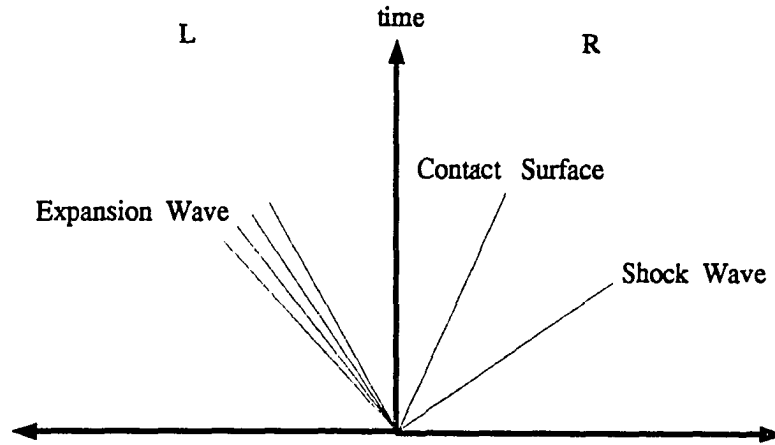


Figure 3.2 The Riemann Problem.

Roe's scheme is used widely in practical applications of computational fluid dynamics because it results in an efficient and accurate computation form. A comparison of different numerical flux formulas for the Euler and Navier-Stokes equations, Ref. [107], recommended the use of the Roe scheme. That recommendation was based on the fact that the interface flux computed by the Roe scheme includes information about all different waves —linear and non-linear— by which the neighboring cells interact. The scheme gives good results when shocks, contact discontinuities, and slip lines are aligned with the grid. The main advantage of the scheme is that it returns the exact solution whenever Q^L and Q^R lie on opposite sides of shocks and contact discontinuities. However, Roe's scheme can also represent an expansion shock due to the lack of a naturally constructed entropy condition.

The interface flux-vector F_I is evaluated as

$$\left(F_{i+\frac{1}{2}}\right)_I = \frac{1}{2}[F_R\{Q_R\} + F_L\{Q_L\} - |\tilde{A}|(Q_R - Q_L)] \quad (3.19)$$

F_L and F_R are the flux-vectors computed from the left and right states, and \tilde{A} is the Roe averaged flux Jacobian matrix

$$\tilde{A} = A[\tilde{Q}] \quad (3.20)$$

where

$$A = \frac{\partial F_l}{\partial Q} \quad (3.21)$$

and

$$|\tilde{A}| = S_\xi |\Lambda| S_\xi^{-1} \quad (3.22)$$

S_ξ and S_ξ^{-1} are the right and left eigenvectors of Roe's averaged Jacobian matrix \tilde{A} ; Λ is the diagonal matrix of the eigenvalues of \tilde{A} . The Roe averaged state is computed as

$$\begin{aligned} \tilde{\rho} &= \sqrt{\rho_R \rho_L} \\ \tilde{u} &= \frac{u_L \sqrt{\rho_L} + u_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \tilde{v} &= \frac{v_L \sqrt{\rho_L} + v_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \tilde{w} &= \frac{w_L \sqrt{\rho_L} + w_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \tilde{H} &= \frac{H_L \sqrt{\rho_L} + H_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \tilde{a} &= \sqrt{(\gamma - 1) \left[\tilde{H} - \frac{1}{2} \{ \tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2 \} \right]} \end{aligned} \quad (3.23)$$

The tildes refer to the Roe averaged quantities. The last term in eq. 3.19, i.e. $|\tilde{A}|(Q_R - Q_L)$, is a damping term due to the upwind character of the scheme and is given in detail in Ref. [55] as

$$|\tilde{A}|(Q_R - Q_L) = \begin{bmatrix} \alpha_4 \\ \tilde{u}\alpha_4 + l_x\alpha_5 + \alpha_6 \\ \tilde{v}\alpha_4 + l_y\alpha_5 + \alpha_7 \\ \tilde{w}\alpha_4 + l_z\alpha_5 + \alpha_8 \\ \tilde{a}_5 \end{bmatrix} \quad (3.24)$$

where,

$$\tilde{a}_5 = \tilde{H}\alpha_4 + \tilde{u}_1\alpha_5 + \tilde{u}\alpha_6 + \tilde{v}\alpha_7 + \tilde{w}\alpha_8 - \frac{\alpha_1 \tilde{a}^2}{\gamma - 1} \quad (3.25)$$

and

$$\begin{aligned}
\alpha_1 &= \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l| \left(\Delta p - \frac{\Delta p}{\tilde{a}^2} \right) \\
\alpha_2 &= \frac{1}{2\tilde{a}^2} \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l + \tilde{a}| (\Delta p + \rho \tilde{a} \Delta \bar{u}_l) \\
\alpha_3 &= \frac{1}{2\tilde{a}^2} \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l - \tilde{a}| (\Delta p - \rho \tilde{a} \Delta \bar{u}_l) \\
\alpha_4 &= \alpha_1 + \alpha_2 + \alpha_3 \\
\alpha_5 &= \tilde{a}(\alpha_2 - \alpha_3) \\
\alpha_6 &= \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l| \tilde{\rho} (\Delta u - l_x \Delta \bar{u}_l) \\
\alpha_7 &= \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l| \tilde{\rho} (\Delta v - l_y \Delta \bar{u}_l) \\
\alpha_8 &= \left| \frac{\sqrt{\varphi}}{J} \right| |\tilde{u}_l| \tilde{\rho} (\Delta w - l_z \Delta \bar{u}_l)
\end{aligned} \tag{3.26}$$

with $\sqrt{\varphi} = \sqrt{l_x^2 + l_y^2 + l_z^2}$, for $l = \xi, \eta$, or ζ . The Δ refers to the difference between the state variables on the left and right sides of the cell interface, such as $\Delta p = p_R - p_L$.

Here;

$$F_\xi = F \quad F_\eta = G \quad F_\zeta = H \tag{3.27}$$

and

$$u_\xi = u \quad u_\eta = v \quad u_\zeta = w \tag{3.28}$$

For more information about the Roe scheme, the interested reader should refer to the original work by Roe, Refs. [65, 66].

CHAPTER 4

TIME INTEGRATION

In this study emphasis was placed on the simulation of complex three-dimensional steady, and unsteady fluid flow problems. The selection of a particular type of time integration technique, whether implicit or explicit, will determine the characteristics of the numerical method used to investigate the fluid problem of concern. In this study, explicit time-stepping schemes were used to construct the algorithm for solving the time-dependent, compressible, Euler, and Navier-Stokes equations.

There are a large number of explicit schemes that have either been used previously or are still in use for solving the compressible flow equations. The desired numerical method should be simple, robust, have effective damping of high frequency errors (necessary for multigrid), have low dispersion (low phase errors will reduce spurious oscillations and result in faster convergence rates), low levels of numerical dissipation for accurate predictions of viscous effects and it should maintain high resolution on stretched grids. Programing simplicity is another important issue, since the goal is to implement the time-stepping scheme in a multi-block code, and on massively parallel machines.

However, no generic time-stepping scheme was found that satisfied all of the requirements, thus it was decided to develop explicit time-stepping schemes that could be tailored to our needs. Two similar but distinct time-stepping schemes were developed for the purpose of solving the compressible, time-dependent, governing set of equations. The two schemes were the multistage time-stepping scheme, and the Predictor-Corrector Scheme. Both schemes are explicit, but they are distinct since each of the techniques

utilizes a different operator to compute the flux at cell interfaces. In order to understand, compare, and assess the two time-stepping techniques, they were applied initially to the model wave equation. There a conventional Fourier stability analysis could be carried out, yielding amplification factors, and the stability characteristics for each scheme. Discussion of the evaluation of the two schemes will be presented and discussed in this chapter.

4.1 Multistage Time-Stepping Scheme.

Modified Runge-Kutta methods, with standard coefficients, have been rather successful when used in combination with central-difference, spatial-discretization techniques. Unfortunately, they perform very poorly with upwind differencing schemes. In this section an attempt has been made to modify the standard coefficients to achieve better performance, resulting simultaneously in schemes that are, in general, of reduced accuracy in time. To explore the damping properties and extend the stability limits of the explicit multistage scheme, the scheme was first applied to the ordinary differential equation

$$\frac{dq}{dt} = -zq \quad \text{where } z \geq 0. \quad (4.1)$$

and which has the analytical solution:

$$q = q_0 e^{-z(t-t_0)} \quad (4.2)$$

Here, q_0 is the initial value of q at $t = t_0$.

A Taylor series expansion for q^{n+1} around q^n gives

$$q^{n+1} = q^n + \frac{dq^n}{dt} \Delta t + \frac{d^2 q^n}{dt^2} \frac{(\Delta t)^2}{2!} + \frac{d^3 q^n}{dt^3} \frac{(\Delta t)^3}{3!} + \dots \quad (4.3)$$

Substituting for the derivatives of q in the above equation

$$q^{n+1} = q^n \left[1 + (-z \Delta t) + \frac{(-z \Delta t)^2}{2!} + \frac{(-z \Delta t)^3}{3!} + \dots \right] \quad (4.4)$$

The above equation is the foundation of a numerical integration scheme known as the modified Runge-Kutta Scheme [27].

Consider now the model wave equation in the form

$$\frac{\partial q}{\partial t} + a \frac{\partial q}{\partial x} = 0 \quad (4.5)$$

where a is the wave speed which is assumed to be real and positive. Then

$$\frac{\partial q}{\partial t} = -a \frac{\partial q}{\partial x} = -R(q) \quad (4.6)$$

where $R(q)$ represents the right hand side of eq. 4.6. If we assume that q can be represented in an exponential form, then

$$q = q_o e^{-z(t-t_o)} \quad (4.7)$$

where q_o is the initial value of q at $t = t_o$. Therefore,

$$\frac{\partial q}{\partial t} = -q_o z e^{-z(t-t_o)} \quad (4.8)$$

$$\frac{\partial q}{\partial t} = -zq \quad (4.9)$$

The multistage explicit time-stepping can be used to advance eq. 4.9 in time from time step n to $n+1$ in the following way

$$\begin{aligned} q^0 &= q^n \\ q^1 &= q^n + \alpha_1 \Delta t R(q^0) \\ q^2 &= q^n + \alpha_2 \Delta t R(q^1) \\ &\vdots \\ q^m &= q^n + \alpha_m \Delta t R(q^{m-1}) \\ q^{n+1} &= q^m \end{aligned} \quad (4.10)$$

Comparing eq. 4.9 and eq. 4.6 yields

$$R(q) = a \frac{\partial q}{\partial x} = zq \quad (4.11)$$

If we define

$$P = \Delta t \times z, \quad (4.12)$$

combine P with eq. 4.11 and substitute into eq. 4.10, we get

$$q^{n+1} = q^n (1 + \alpha_m P + \alpha_m \alpha_{m-1} P^2 + \dots + \alpha_1 \alpha_2 \dots \alpha_m P^m) \quad (4.13)$$

By comparing the terms in eq. 4.13 and eq. 4.4, we can determine the temporal accuracy of the multistage scheme. The scheme will be first order in time if

$$\alpha_m = 1. \quad (4.14)$$

The scheme will be second-order in time if

$$\alpha_m = 1, \text{ and } \alpha_{m-1} = \frac{1}{2}. \quad (4.15)$$

The scheme will be third-order in time if

$$\alpha_m = 1, \quad \alpha_{m-1} = \frac{1}{2}, \text{ and } \alpha_{m-2} = \frac{1}{3}. \quad (4.16)$$

The scheme will be fourth-order in time if

$$\alpha_m = 1, \quad \alpha_{m-1} = \frac{1}{2}, \quad \alpha_{m-2} = \frac{1}{3} \text{ and, } \alpha_{m-3} = \frac{1}{4}, \quad (4.17)$$

and so on.

One can continue this progression and arrive at higher order schemes. It should be emphasized that the leading coefficient α_m should always be 1.0 for the scheme to be at least first order in time.

The amplification factor of the multistage explicit time-stepping scheme, G , can be derived from eq. 4.13 and will take the following form

$$G = \frac{q^{n+1}}{q^n} = 1 + \alpha_1 P + \alpha_1 \alpha_2 P^2 + \dots + \alpha_1 \dots \alpha_m P^m \quad (4.18)$$

The stability and damping properties of the multistage scheme are associated with complex polynomial G . G is a function of the coefficients α_s and of P . The complex function P is a function of the spatial operator used to interpolate q at the cell interface. Thus the stability properties of the multistage scheme are tied to the spatial operator used to compute the flux at the cell interface.

In this study a control volume approach was implemented where the spatial discretization of the wave equation takes the form:

$$R(q) = a \frac{\partial q}{\partial x} = a \frac{[q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}]}{\Delta x} \quad (4.19)$$

The extrapolation of the state variables to the cell interface is based on the so-called κ -scheme where,

$$\begin{aligned} q_{i+\frac{1}{2}}^n &= q_i^{n-1} + \frac{l}{4}[(1 - \kappa)\Delta_i + (1 + \kappa)\nabla_i] \\ q_{i-\frac{1}{2}}^n &= q_{i-1}^{n-1} + \frac{l}{4}[(1 - \kappa)\Delta_{i-1} + (1 + \kappa)\nabla_{i-1}] \end{aligned} \quad (4.20)$$

such that

$$\Delta_i = q_i - q_{i-1} \quad \text{and} \quad \nabla_i = q_{i+1} - q_i \quad (4.21)$$

As mentioned in the previous chapter κ determines the spatial accuracy of the scheme; $\kappa = -1$ is a fully upwind second-order accurate scheme; $\kappa = 0$ is an upwind biased, second-order Fromm scheme; $\kappa = 1/3$ is an upwind, biased third-order accurate scheme, and $\kappa = 1$ is a second-order accurate, central-difference scheme. The first-order scheme is obtained by setting l to zero. For simplicity, in the present stability analysis the limiter was not included in the κ -scheme.

If we now assume the data to be harmonic

$$q_j = q_{x=j\Delta x} = q_0 e^{I\beta j} \quad (4.22)$$

where β is the spatial wave number ranging from 0 to π , and I is $\sqrt{-1}$. Values of β between $\pi/2$ and π are considered to be high frequencies. Combining eq. 4.19, eq. 4.7 with eq. 4.20 yields

$$P = -CFL \left(1 - e^{-I\beta}\right) \left\{ 1 + I \left[\frac{1-\kappa}{4} \left(1 - e^{-I\beta}\right) - \frac{1-\kappa}{4} \left(1 - e^{I\beta}\right) \right] \right\} \quad (4.23)$$

Fourier transform of the spatial-operator (P) is a function of the CFL number and the wave number, $P(CFL, \beta)$. The expression for the amplification factor, given by eq. 4.18, defines the stability region of the scheme. The stability of the multistage scheme requires that the modulus of the amplification factor $|G|$ be less than unity. This expression gets complicated if we attempt to substitute the expression for the complex polynomials, P , into the expression for G and define the stability region of the scheme analytically. An alternative way to determine the stability region of the scheme is to plot the modulus of the amplification factor for the multistage explicit scheme and identify the stability limit graphically.

The stability of the multistage scheme depends on the complex polynomial P and the coefficients α_s of the multistage scheme. The locus of the Fourier transform, P , superimposed on the contours of the amplification factor can be used to optimize the coefficients of the explicit, multistage time-stepping scheme to better suit the upwind schemes, and achieve better rates of convergence to steady-state. The modulus of amplification factor $|G|$ with the locus of Fourier transform for a first-order, four-stage standard Runge-Kutta scheme are shown in Fig. 4.1. The influence of the coefficients on the contours of the amplification factor had to be fully understood to facilitate the selection of the optimum coefficient set. Optimization of the coefficients, α_s , was carried

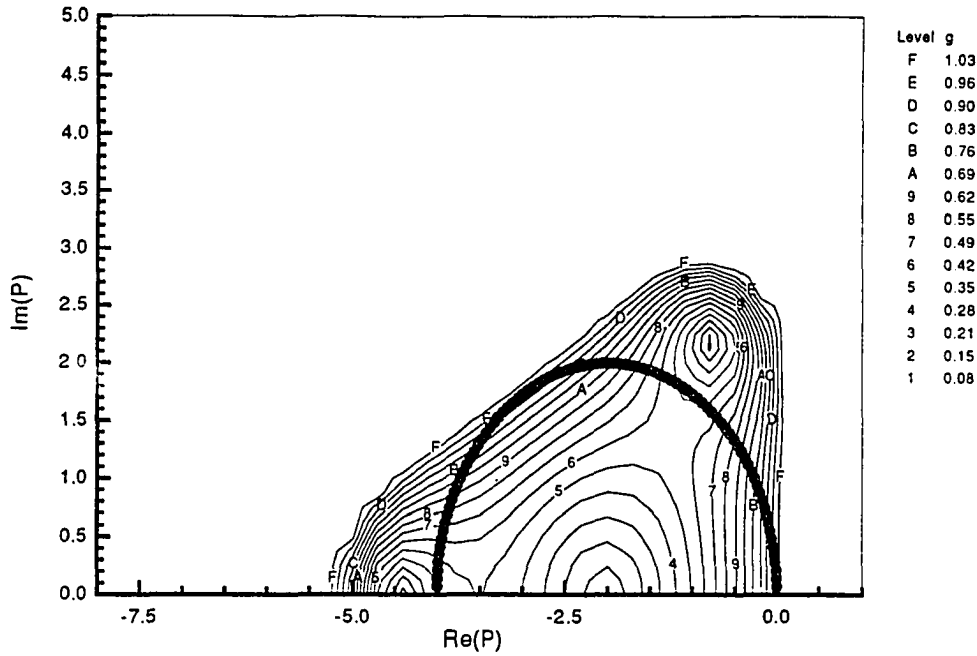


Figure 4.1 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform, P —●—●—, for a First-Order, Four-Stage Runge-Kutta Scheme, CFL = 2.0

out by displaying the stability plots on a computer terminal. The changes in the shape of the contours of $|G|$ were observed in real time as the coefficients were changed. The “islands” of low values of $|G|$ correspond to the roots of eq. 4.18. The main purpose of the optimization was to find a combination of the coefficients, α_s , such that, for the largest possible CFL values, there would be good high frequency damping (low values of $|G|$) over a large range of CFL. That is, the optimal coefficients should maximize the size of the islands, and make them as close to the real axis as possible. The optimization was performed for the two-stage, three-stage, and four-stage schemes. For each of the mentioned schemes the optimization was conducted for four different spatial operators: first-order; second-order fully upwind ($\kappa = -1$); second-order Fromm Scheme ($\kappa = 0$); and third-order upwind biased ($\kappa = 1/3$). Tables 1–4 list the optimized coefficients for the spatial operators mentioned above.

Table 2 Multistage Coefficients for First-Order Scheme

Number of Stages	Multistage Coefficients			
	α_1	α_2	α_3	α_4
Two-Stage Scheme	1.0	0.22		
Three-Stage Scheme	1.0	0.325	0.105	
Four-Stage Scheme	1.0	0.34	0.152	0.056

Table 3 Multistage Coefficients for Second-Order Fully Upwind Scheme

Number of Stages	Multistage Coefficients			
	α_1	α_2	α_3	α_4
Two-Stage Scheme	1.0	0.22		
Three-Stage Scheme	1.0	0.4	0.15	
Four-Stage Scheme	1.0	0.42	0.24	0.091

Table 4 Multistage Coefficients for Second-Order Fromm Scheme

Number of Stages	Multistage Coefficients			
	α_1	α_2	α_3	α_4
Two-Stage Scheme	1.0	0.42		
Three-Stage Scheme	1.0	0.44	0.21	
Four-Stage Scheme	1.0	0.46	0.255	0.11

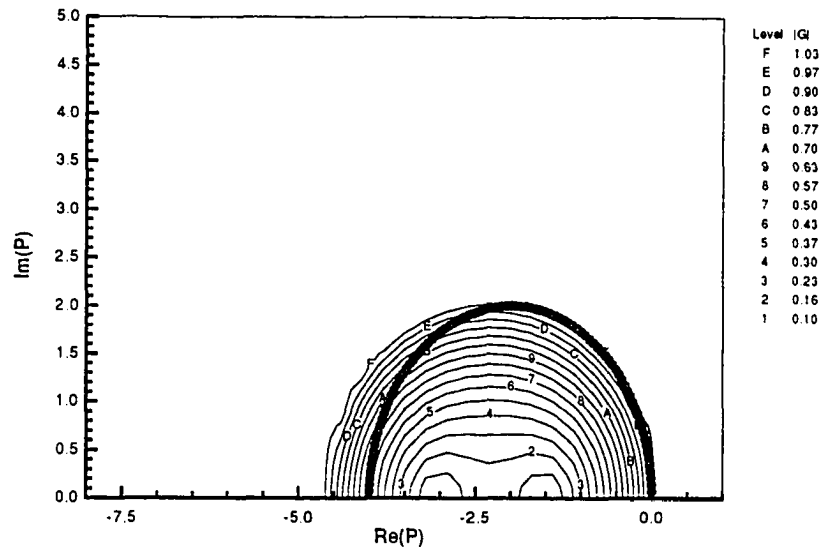
Table 5 Multistage Coefficients for Third-Order Upwind Biased Scheme

Number of Stages	Multistage Coefficients			
	α_1	α_2	α_3	α_4
Two-Stage Scheme	1.0	0.46		
Three-Stage Scheme	1.0	0.48	0.22	
Four-Stage Scheme	1.0	0.44	0.26	0.135

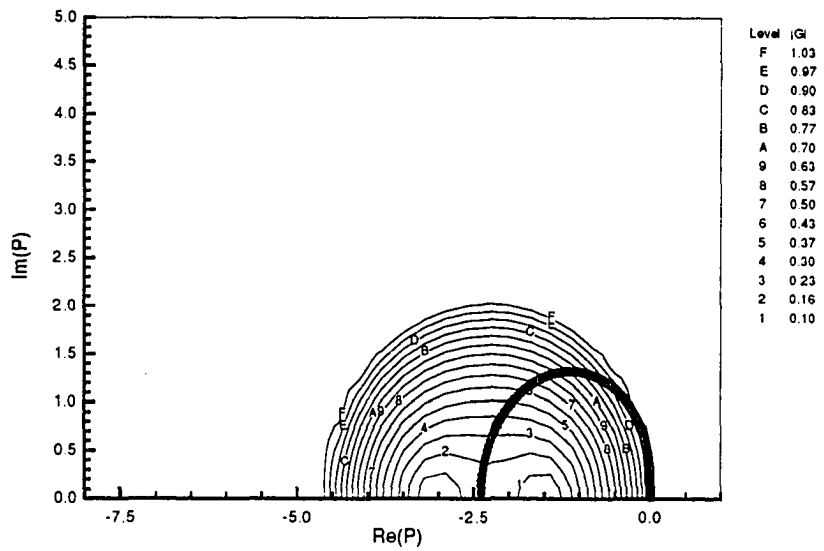
The simplest schemes to optimize were the two-stage versions, since only one coefficient can be selected freely. The first coefficient α_1 is always equal to unity to ensure that the scheme is at least first-order accurate in time. The challenge to optimize the coefficients of the explicit multistage scheme increased by increasing the number of stages, since the number of coefficients to be optimized increased. The most challenging scheme to optimize was the four-stage scheme since the optimum combination of three coefficients has to be found. The modulus of the amplification factor $|G|$ with the locus of its Fourier transform (of the spatial operator, P , corresponding to a maximum CFL) number is presented in Figs. 4.2, 4.3, and 4.4, for all the spatial operators used in this study. The resulting stability plots will be shown only in the second quadrant (upper half of the negative real part of the complex polynomial P) since they are symmetric with respect to the real axis. Figures 4.5, 4.6, and 4.7 represent the magnitude of the modulus of the amplification factor $|G|$ as a function of the spatial wave number, β , and the CFL number. By displaying the two sets of plots for a particular multistage scheme, the stability region and the damping properties of the scheme can be fully displayed. By increasing the number of stages, we are able to increase the CFL number, as shown in Figs. 4.5, 4.6, and 4.7. The time to perform a four-stage explicit scheme is twice that for a two-stage scheme. On the other hand the CFL number increased from 2 to 4, comparing with the case of a first-order scheme. This conclusion is also valid for the remaining spatial operators, as shown in the stability plots. The main advantage of going to a higher number of stages was the good damping characteristics for high wave numbers. Considering the results of the stability analysis, the most promising schemes of practical importance were the Four-Stage Fromm Scheme ($\kappa = 0$) and the third-order upwind biased Scheme ($\kappa = 1/3$). The Fromm Scheme was preferred due to its low numerical dispersion, demonstrated by results with the least oscillations around shocks.

It is important to point out that, when multigrid acceleration techniques are implemented the desire for maximum CFL number is not as important as the high frequency damping requirement. The high frequency damping (or lack of it) will affect the rate of convergence to steady-state more significantly than the CFL number. The choice of the optimum CFL number, when utilizing multigrid acceleration techniques, should be based on how well high frequencies are damped.

It should be mentioned here that, in a parallel effort, van Leer, Tai, and Powell [35], and Gaffney [112], also tried to optimize the Runge-Kutta coefficients for applications with the upwind methods. The van Leer, Tai, and Powell approach was somewhat different than the work discussed previously. Their approach assumed that a genuine and practical multi-dimensional characteristic formulation of the Euler equations could be found, and then they optimized the Runge-Kutta coefficients for only one value of the CFL number. They argued that each wave would propagate at its optimum CFL-number. Unfortunately, there is no such formulation for three dimensional cases. Generally, the maximum CFL numbers, for the van Leer, Tai, and Powell approach, were lower, and the damping was effective over a narrower range of CFL numbers.

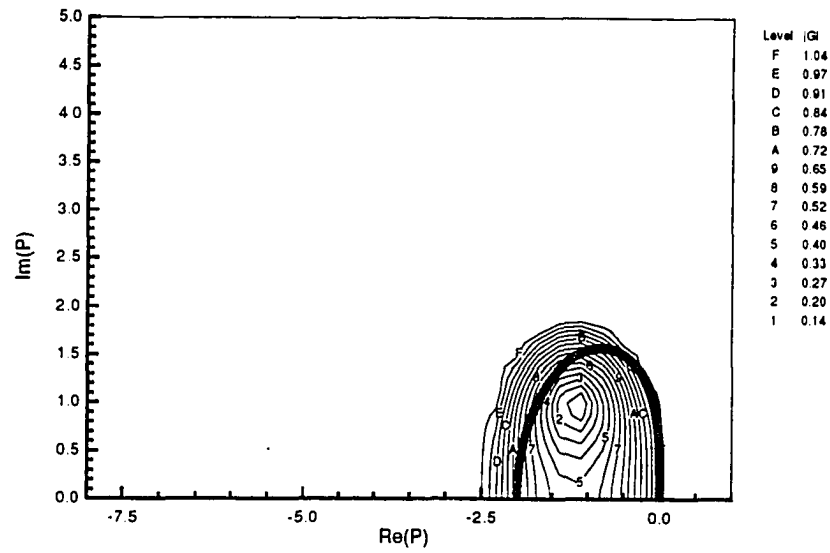


(a) First-Order, CFL = 2.0

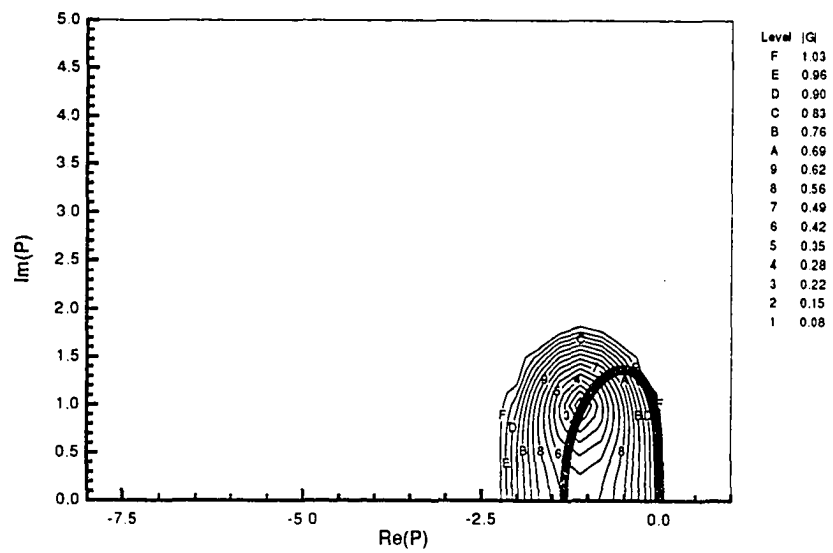


(b) Second-Order, CFL = 0.6

Figure 4.2 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P $\text{---}\bullet\text{---}\bullet\text{---}$, for Two-Stage Schemes. (Continued . . .)

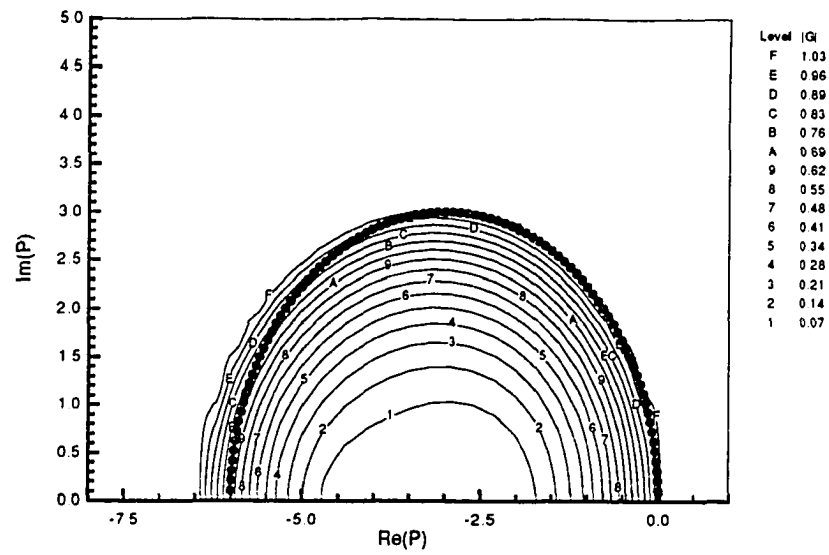


(c) Fromm, Second-Order, CFL = 1.0

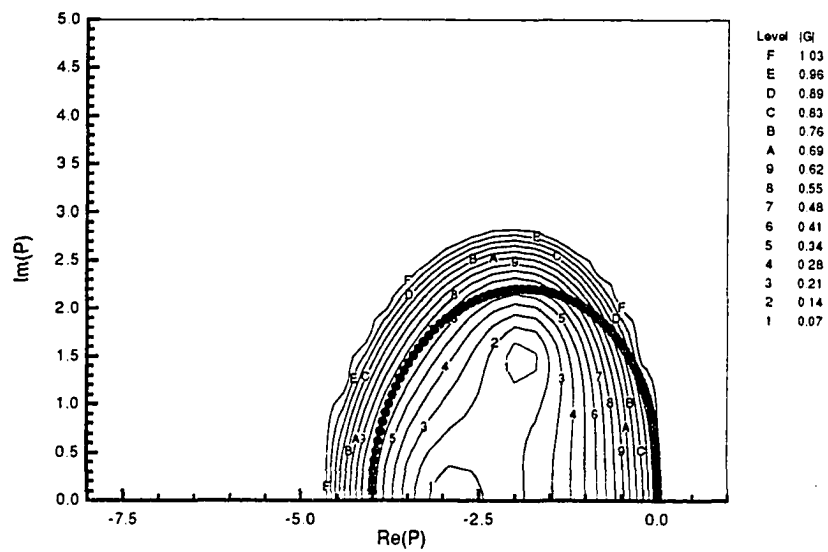


(d) Third-Order, CFL = 1.0

Figure 4.2 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P $-\bullet-\bullet-$, for Two-Stage Schemes.

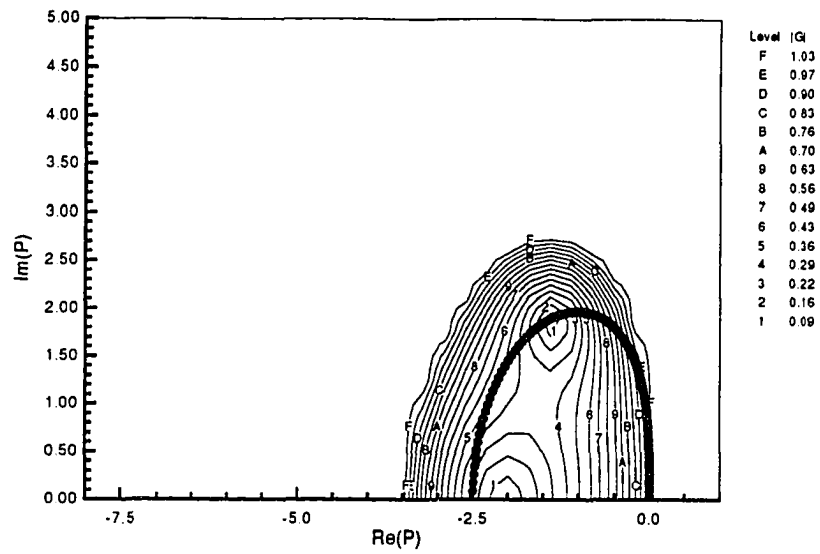


(a) First-Order, CFL = 3.0

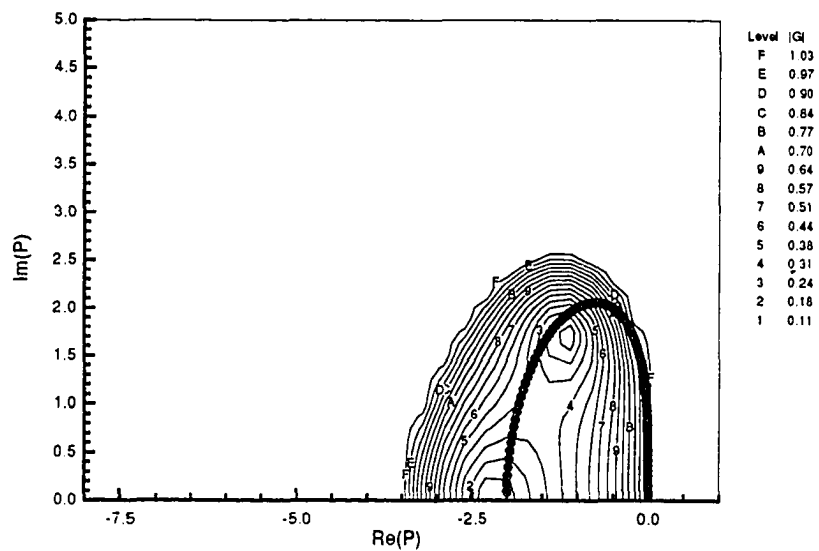


(b) Second-Order, CFL = 1.0

Figure 4.3 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P , $-\bullet-\bullet-$, for Three-Stage Schemes. (Continued . . .)

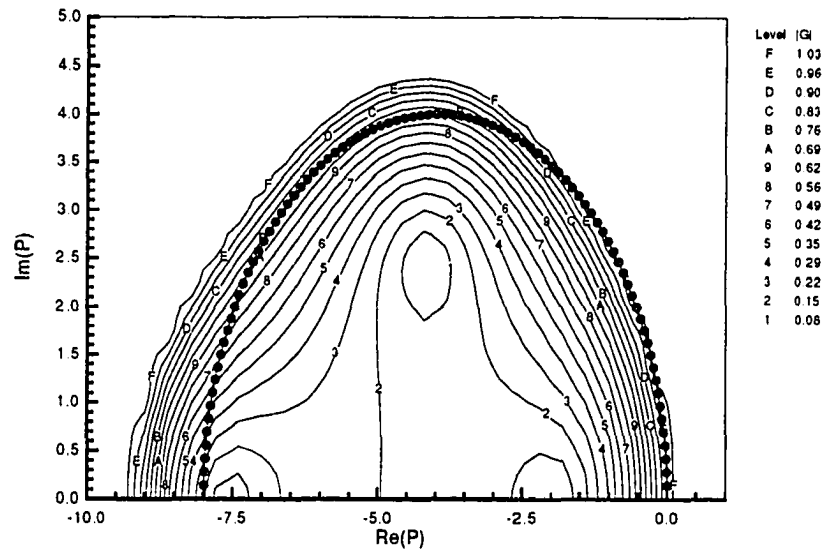


(c) Fromm, Second-Order, CFL = 1.25

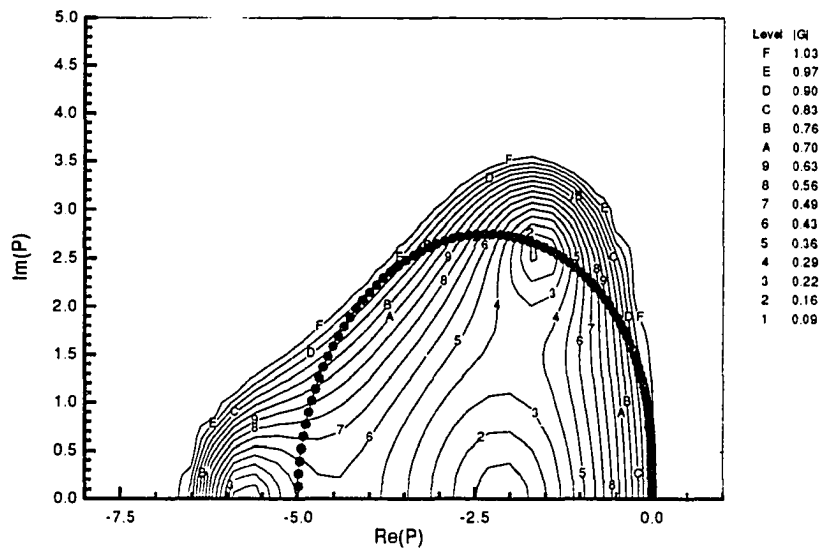


(d) Third-Order, CFL = 1.5

Figure 4.3 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P , $-\bullet-\bullet-$, for Three-Stage Schemes.

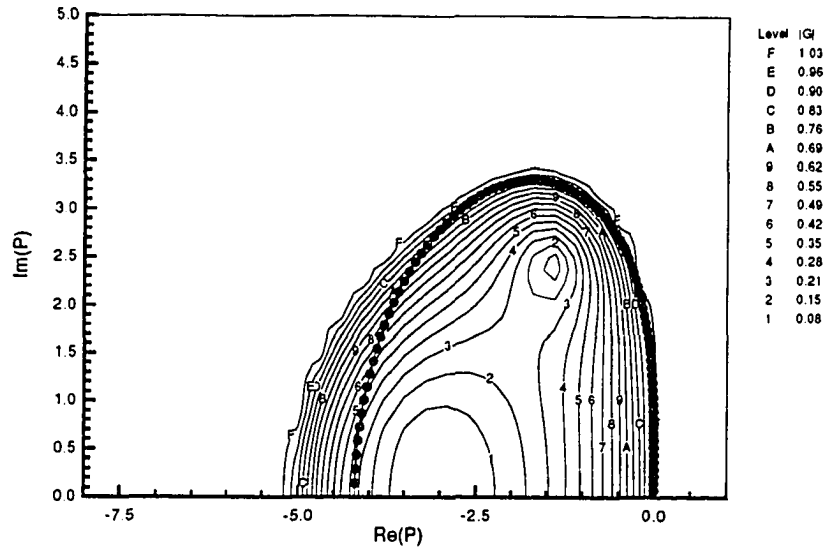


(a) First-Order, CFL = 4.0

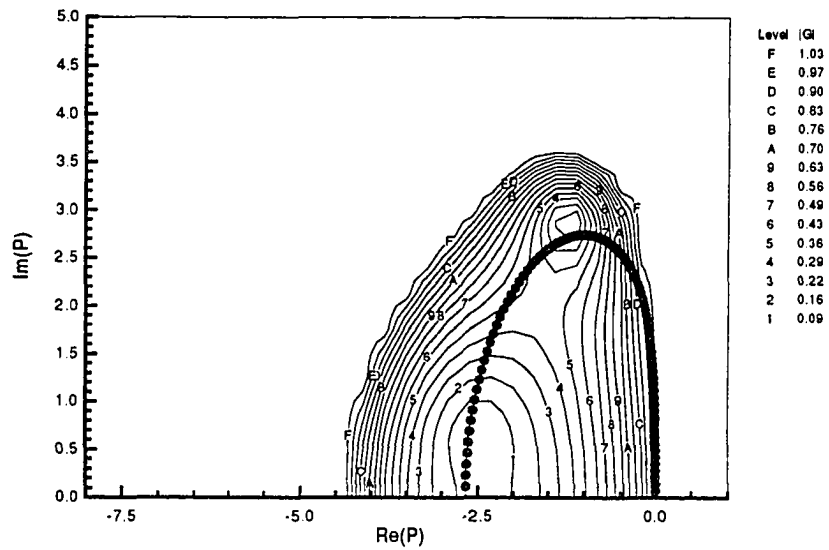


(b) Second-Order, CFL = 1.25

Figure 4.4 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P , $-\bullet-\bullet-$, for Four-Stage Schemes. (Continued . . .)

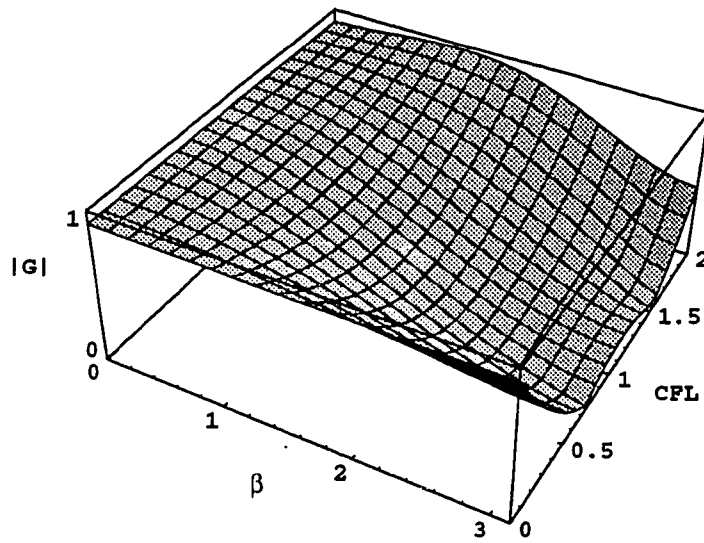


(c) Fromm, Second-Order, CFL = 2.1

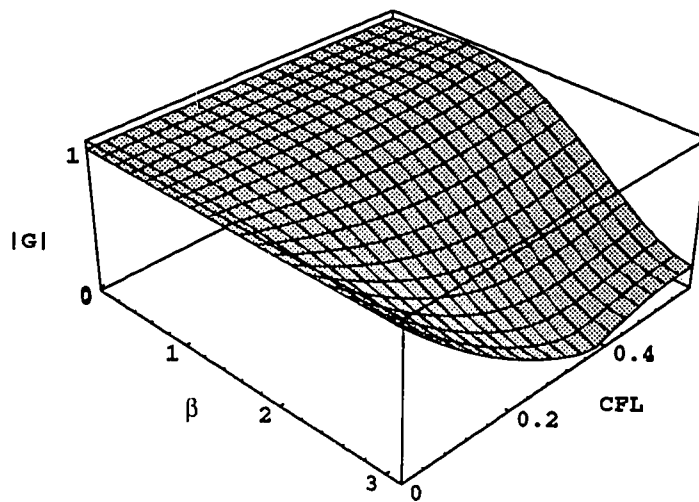


(d) Third-Order, CFL = 2.0

Figure 4.4 Contours of Modulus of Amplification Factor, $|G|$, and Locus of Fourier Transform of Spatial-Operator, P , $-\bullet-\bullet-$, for Four-Stage Schemes.

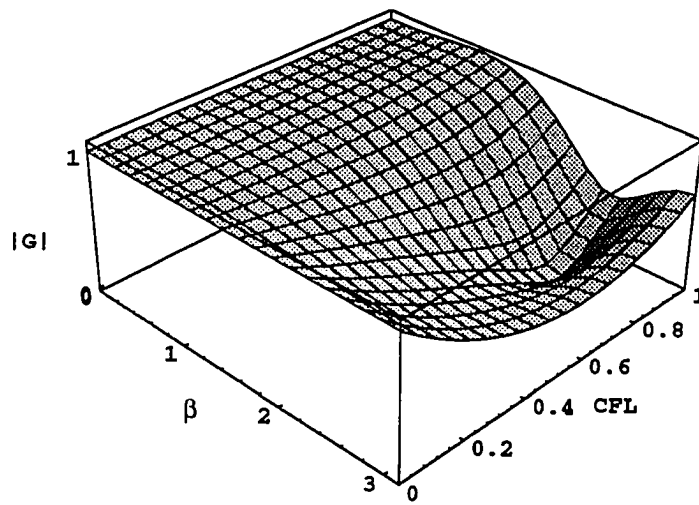


(a) First-Order

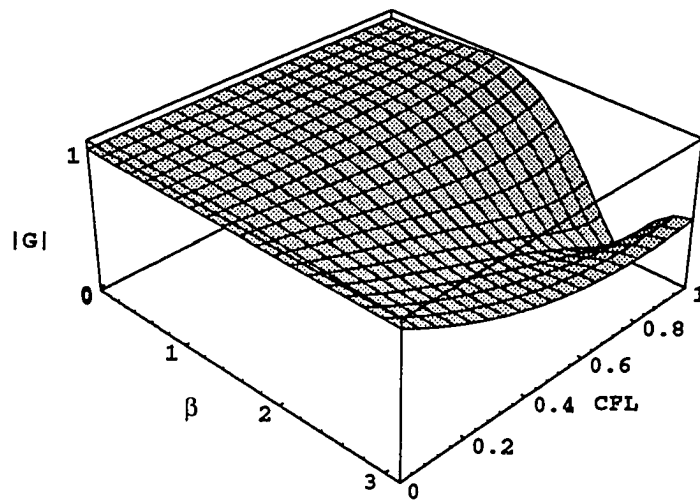


(b) Second-Order

Figure 4.5 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Two-Stage Schemes. (Continued . . .)

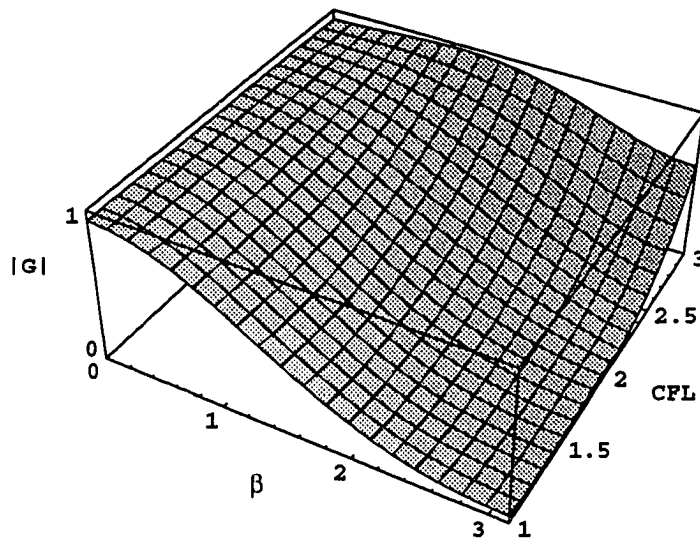


(c) Fromm, Second-Order

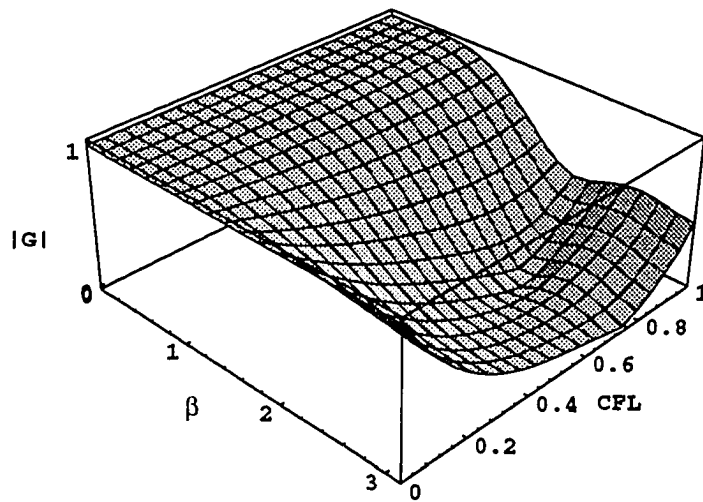


(d) Third-Order

Figure 4.5 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Two-Stage Schemes.

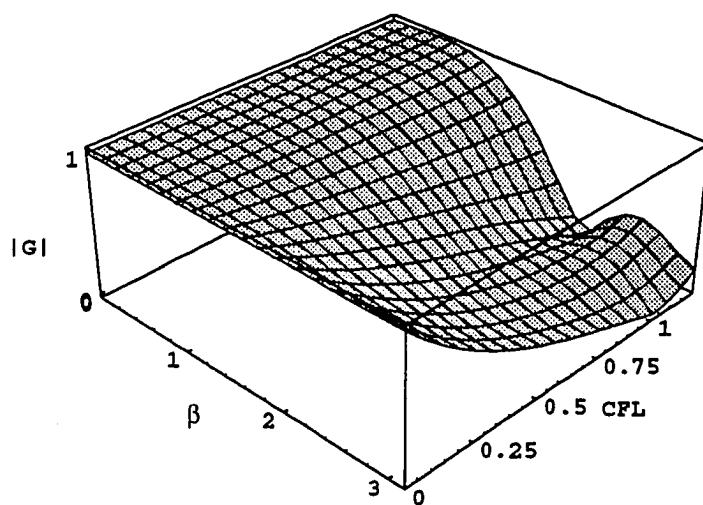


(a) First-Order

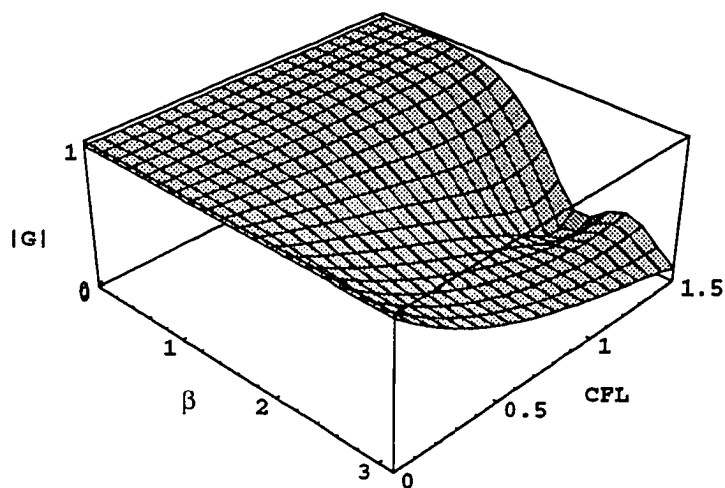


(b) Second-Order

Figure 4.6 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Three-Stage Schemes. (Continued . . .)

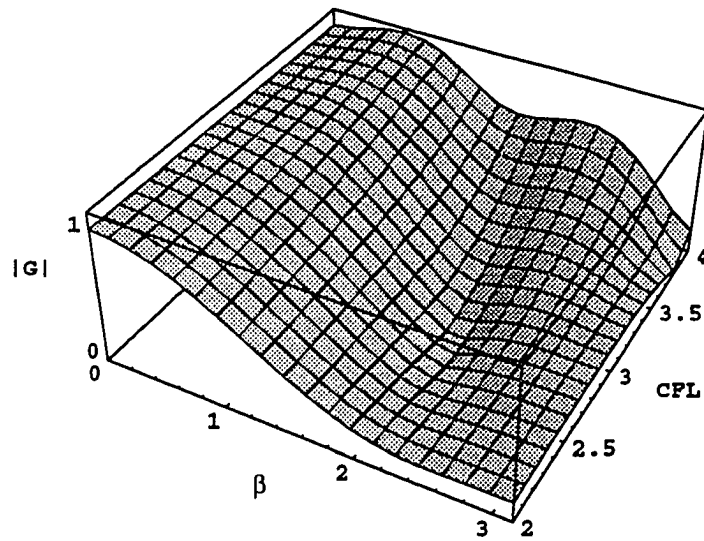


(c) Fromm, Second-Order

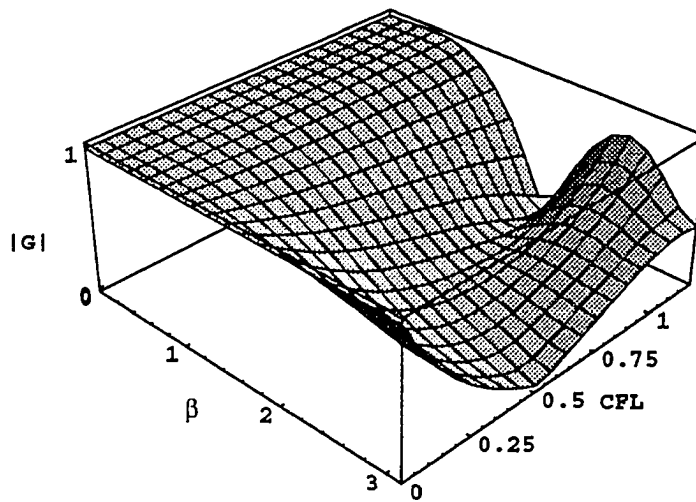


(d) Third-Order

Figure 4.6 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Three-Stage Schemes.

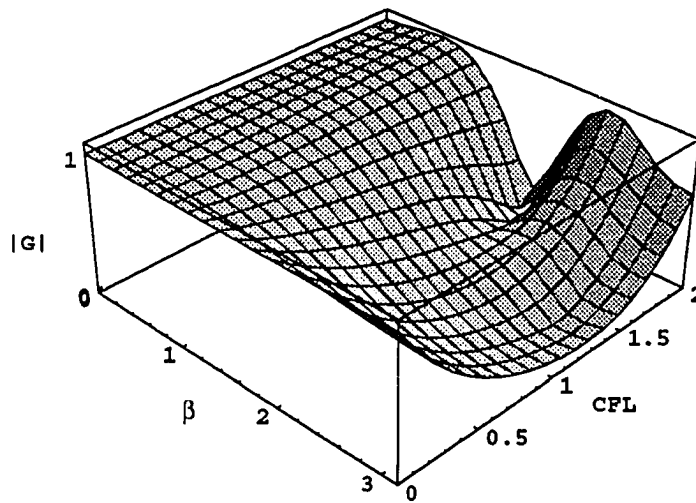


(a) First-Order

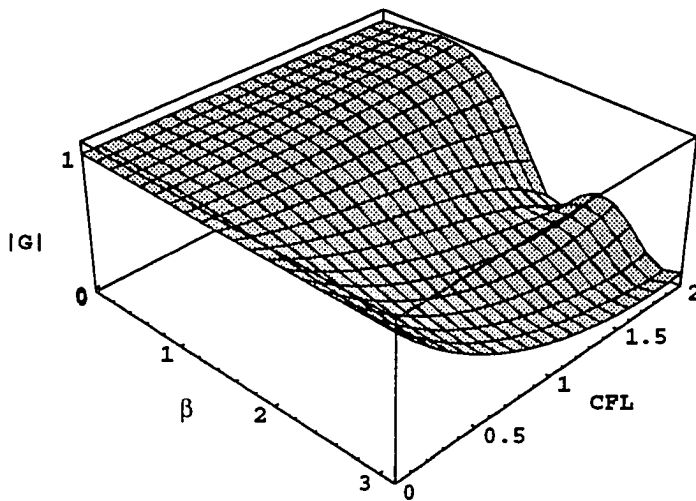


(c) Fromm, Second-Order

Figure 4.7 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Four-Stage Schemes. (Continued . . .)



(c) Fromm, Second-Order



(d) Third-Order

Figure 4.7 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL number, for Four-Stage Schemes.

4.2 Predictor-Corrector Schemes.

The main goal was to develop a two-stage explicit scheme which had good damping qualities and could allow the use of large CFL numbers. In the previous section we presented two-stage schemes for different spatial operators. These schemes offered good damping qualities but the maximum allowable CFL number was less than unity, except for the first-order spatial discretization, as shown in Figs. 4.5, 4.6, and 4.7. It was decided then to construct a two-stage scheme where the first stage (predictor step) required the use of a first-order spatial operator and the second stage (corrector step) utilized a second-order spatial operator. The scheme is a modified version of the upwind scheme of Warming and Beam [18], and is named 1–2 scheme. The two steps are given by:

Predictor Step:

$$\begin{aligned} q_{i+\frac{1}{2}}^1 &= q_i^n \\ q_{i-\frac{1}{2}}^1 &= q_{i-1}^n \\ q_i^1 &= q_i^n - \Delta t * R(q^1) \end{aligned} \tag{4.24}$$

Corrector Step:

$$\begin{aligned} q_{i+\frac{1}{2}}^2 &= q_i^n + \frac{1}{2}(q_i^1 - q_{i-1}^n) \\ q_{i-\frac{1}{2}}^2 &= q_{i-1}^n + \frac{1}{2}(q_{i-1}^1 - q_{i-2}^n) \\ q_i^{n+1} &= q_i^n - \Delta t * R(q^2) \end{aligned} \tag{4.25}$$

where, $R(q)$ is the residual and is given in eq. 4.19

A stability analysis similar to the one in the previous section, was performed. The only difference was that the type of the spatial operator used for the extrapolation to the cell interface was different. The resulting plot of the magnitude of the spatial wave number, β , between 0 and π and the CFL number between 0 and 2 is shown in Fig. 4.8. The figure shows clearly that the stability limit is higher than for the two-stage schemes

introduced in the previous section. The scheme is still second-order accurate in space and satisfies the “shift condition”. Introducing the first-order spatial operator in the predictor step allowed us to increase the CFL number to a value of 2. The scheme is a very simple and efficient method consisting of only two steps.

There were, however, some problems with the scheme. Since the two steps are different, the steady-state result, depended on the time step. This phenomena was observed in only a few cases, represented by convergence to a residual that was larger than “machine zero”. The second problem with this scheme was the limiters. None of the flux limiters tested in this scheme converged more than two orders of magnitude. The reason most probably is due to the mixing of time levels in the extrapolation of the variables to the cell interface in the corrector step. Never the less the resulting flow fields agreed well with other, fully converged numerical results.

A more serious problem is the increase of the damping factor to 1.0 at high frequencies for CFL number 1, as shown in Fig. 4.8. In the scalar case, the CFL number can be kept at its optimum value of 1.7. But in the case of the Euler or the Navier-Stokes equations there are three distinct eigenvalues in each direction. Typically, local time-stepping can be implemented, where each cell is advanced at its optimum time step. The time step is a function of the corresponding eigenvalues at that cell. One or more eigenvalues might correspond to the CFL number ranges with minimum or no damping. This was manifested by the lack of convergence of the 1-2 Scheme when utilized in a multigrid procedure.

For multigrid applications, this scheme was modified by making the predictor step second-order accurate. Thus, the predictor step equation 4.24 is replaced by the following:

$$\begin{aligned} q_{i+\frac{1}{2}}^1 &= q_i^n + \frac{1}{2}(q_i^n - q_{i-1}^n) \\ q_{i-\frac{1}{2}}^1 &= q_{i-1}^n + \frac{1}{2}(q_{i-1}^n - q_{i-2}^n) \\ q_i^1 &= q_i^n - \Delta t * R(q^1) \end{aligned} \tag{4.26}$$

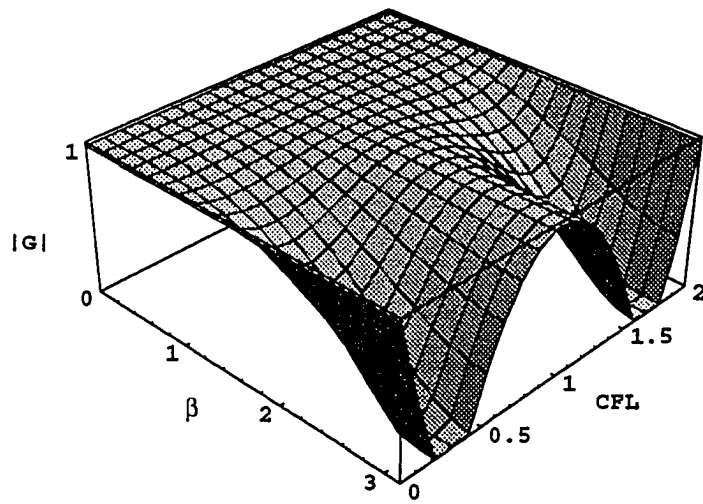
The corrector step is identical to the corrector step in the 1-2 Scheme and will only be repeated here for the sake of completeness.

Corrector Step:

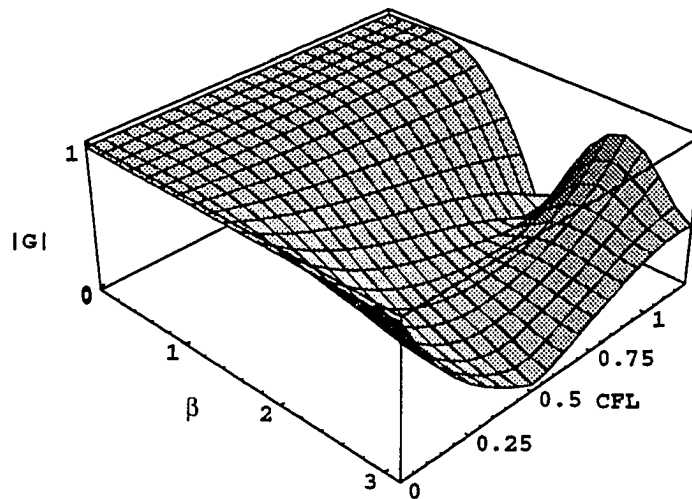
$$\begin{aligned} q_{i+\frac{1}{2}}^2 &= q_i^n + \frac{1}{2}(q_i^1 - q_{i-1}^n) \\ q_{i-\frac{1}{2}}^2 &= q_{i-1}^n + \frac{1}{2}(q_{i-1}^1 - q_{i-2}^n) \\ q_i^{n+1} &= q_i^n - \Delta t * R(q^2) \end{aligned} \tag{4.27}$$

The plot of the damping characteristics of this 2-2 Scheme is shown in Fig. 4.8. The maximum stable CFL number is now only 1.0, but its high frequency damping is improved significantly. The 2-2 Scheme performed better with multigrid.

One of the main advantages of using the explicit time-stepping schemes was their simplicity. They could be extended to higher dimensions easily. They fit into a multi-block environment naturally. They were easily implemented on massively parallel machines, such as CM-2. The main drawback was the restriction on the time step, especially on highly stretched grids (viscous grids). However, by utilizing acceleration techniques, local time-stepping, implicit residual smoothing and multigrid techniques it was possible to overcome this drawback and increase the stability region of the scheme, as will be shown in the next chapter.



(a) 1-2-Predictor-Corrector Scheme



(b) 2-2-Predictor-Corrector Scheme

Figure 4.8 Modulus of Amplification Factor, $|G|$, as a Function of the Spatial Wave Number, β , and the CFL Number, for Predictor-Corrector Schemes.

CHAPTER 5

ACCELERATING TECHNIQUES

Local time-stepping, Implicit Residual Smoothing, and Full Approximation Storage (FAS) multigrid procedure, were employed in the present algorithm to remove the stiffness from the governing set of equations and to accelerate the rate of convergence to the steady-state solution. These techniques were integrated with the explicit-multi-stage time-stepping scheme, discussed in the previous chapter, to enhance the overall computing efficiency and performance of the algorithm. For unsteady flows, the accelerating techniques employed cannot be applied and a global time step was used.

5.1 Local Time-Stepping

Local time-stepping allows each cell to advance in time by the maximum allowable local time step, as dictated by the local stability requirements. This process allows faster signal propagation through the computation domain, relaxes the stiffness of the governing equations, and hence increases the rate of convergence to steady-state. An accurate estimation of the allowable time step is of paramount importance if a robust algorithm is desired. The time step used in this study was based on both a convection and a diffusion stability limit, and is given by:

$$\frac{1}{\Delta t} \geq \frac{1}{\Delta t_\xi} + \frac{1}{\Delta t_\eta} + \frac{1}{\Delta t_\zeta} + \frac{1}{\Delta t_\xi^v} + \frac{1}{\Delta t_\eta^v} + \frac{1}{\Delta t_\zeta^v} \quad (5.1)$$

where, $\frac{CFL}{\Delta t_l} \geq \lambda_l = |U^l| + a\sqrt{\phi_l^2}$

$$\frac{CFL}{\Delta t_l^v} \geq \lambda_l^v = \frac{\mu}{\rho} \left[\phi_l \left\{ \max \left(\frac{4}{3}, \frac{\gamma}{Pr} \right) \right\} + \frac{1}{3} \{ |l_x l_y| + |l_x l_z| + |l_y l_z| \} \right]$$

$$\phi_l = l_x^2 + l_y^2 + l_z^2; \quad l = \xi, \eta, \text{ and } \zeta$$

The first three terms on the right hand side of equation 5.1 result from stability limitations on the inviscid flux, while the last three terms are due to stability limitations on the diffusion terms. The expression for the time step is based on the stability analysis study presented in [33]. The diffusion limit on the time step $\frac{1}{\Delta t_l^v}$, makes the scheme more robust on fine grids, and in boundary-layer type flows [113].

For unsteady flows a global time step was used which was required to be the smallest maximum time step calculated within the computational domain.

5.2 Implicit Residual Smoothing

Implicit residual smoothing extends the stability limit, and improves the damping properties of the multistage time-stepping scheme. Lerat [38] introduced the idea of residual smoothing for the Lax-Wendroff scheme. Jameson and Baker [29] applied the idea of an implicit residual smoothing in conjunction with modified Runge-Kutta schemes. This procedure has been developed further by [28, 33, 40–42], by employing a central-implicit-residual-smoothing operator. The use of an upwind-residual-smoothing operator was employed also [43, 44]. The smoothing operator modified the basic k-stage explicit scheme of eq. 4.10 in the following manner

$$\begin{aligned}
 \frac{\partial Q}{\partial t} &= - \left[\frac{\partial(F - F_v)}{\partial \xi} + \frac{\partial(G - G_v)}{\partial \eta} + \frac{\partial(H - H_v)}{\partial \zeta} \right] \\
 Q^0 &= Q^n \\
 Q^1 &= Q^n - \alpha_1 R^*(R(Q^0)) \\
 Q^2 &= Q^n - \alpha_2 R^*(R(Q^1)) \\
 &\vdots \\
 Q^k &= Q^n - \alpha_k R^*(R(Q^{k-1})) \\
 Q^{n+1} &= Q^k
 \end{aligned} \tag{5.2}$$

The smoothed residual R^* is a function of the unsmoothed residual R . The smoothing operator can be either an explicit or an implicit operator. In the present study an implicit operator was employed. For three-dimensional flow, the smoothing operator is given by:

$$(1 - \epsilon_\xi \partial_{\xi\xi})(1 - \epsilon_\eta \partial_{\eta\eta})(1 - \epsilon_\zeta \partial_{\zeta\zeta})R^* = R \quad (5.3)$$

where ϵ_ξ , ϵ_η , and ϵ_ζ are the smoothing coefficients for the ξ , η , and ζ coordinate directions respectively. The second-order-central difference operators are $\delta_{\xi\xi}$, $\delta_{\eta\eta}$, and $\delta_{\zeta\zeta}$; where

$$\partial_{\xi\xi}R = R_{i+1,j,k} - 2R_{i,j,k} + R_{i-1,j,k} \quad (5.4)$$

Similar expressions for $\delta_{\eta\eta}$, and $\delta_{\zeta\zeta}$ apply. An efficient tri-diagonal solver, the Thomas Algorithm, was applied sequentially in all three directions to evaluate the smoothed residual R^* such that:

$$\begin{aligned} (1 - \epsilon_\xi \delta_{\xi\xi})R^\xi &= R \\ (1 - \epsilon_\eta \delta_{\eta\eta})R^\eta &= R^\xi \\ (1 - \epsilon_\zeta \delta_{\zeta\zeta})R^* &= R^\eta \end{aligned} \quad (5.5)$$

Following the same guidelines and notation used in performing the stability analysis in chapter four, an analytical investigation of the one-dimensional wave equation was conducted to study the effect of the residual smoothing operator on the stability limit of the basic explicit time-stepping scheme. The implicit residual smoothing operator, eq 5.3, for a one-dimensional problem, in the x-direction, is given by

$$(1 - \epsilon_x \delta_{xx})R^* = R \quad (5.6)$$

which yields the following form when expanded

$$-\epsilon_x R_{i-1}^* + (1 + 2\epsilon_x) R_i^* - \epsilon_x R_{i+1}^* = R, \quad (5.7)$$

If we assume that $R^* = \hat{R}^*(t)e^{I\omega x}$ and $R = \hat{R}(t)e^{I\omega x}$ and substitute in eq. 5.7, we get

$$\hat{R}^* = \frac{\hat{R}}{1 + 2\epsilon_x \{1 - \cos(\beta)\}}; \quad \text{where } \beta = \omega x \quad (5.8)$$

The above equation shows that the smoothed residual is a function of the unsmoothed residual, the smoothing coefficient, and the spatial wave number.

The implicit residual smoothing performed well, when combined with the modified Runge-Kutta scheme, and the 2-2 scheme. The implicit residual operator damped the high frequency errors and allowed the use of a higher *CFL* number which improved the rate of convergence to steady-state. On the other hand, the overall performance was rather disappointing when combined with the 1-2 Scheme, and no gain from using the implicit operator was achieved. The plot of the damping characteristics for the 2-2 Scheme, when combined with the smoothing operator and a smoothing coefficient of 0.5 is shown in Fig. 5.1. Comparing Fig. 5.1 and Fig. 4.8 for the unsmoothed scheme clearly demonstrates that residual smoothing not only increased the stability limit of the scheme and allowed the use of a higher *CFL* number, but it also provided good high frequency damping.

In the early stages of the present work, a constant scalar residual smoothing coefficient was used. The smoothing operator was employed after every stage of the time-stepping scheme and was activated uniformly in all three directions. The value of the residual smoothing was selected to be between 0.1 and 0.5, depending on the case investigated, the computational grid, and the *CFL* number. A higher value for the smoothing coefficient was used in a coordinate direction where the grid was highly stretched. Increasing the

value of the smoothing coefficient allowed the use of a higher CFL number. Increasing the CFL number by a factor of two, usually gave the best rate of convergence. It should be emphasized that changing the value of the smoothing coefficients changed the shape of the amplification map as well as the stability range of the scheme.

In the next stage of development, an adaptive implicit residual smoothing technique was employed. This procedure was originally suggested by Martinelli [40], and developed further in by Swanson, Turkel, and White [33] for two-dimensional, central-difference schemes. It was extended subsequently to three-dimensions, and yields the following expression for the smoothing coefficients [42, 114]:

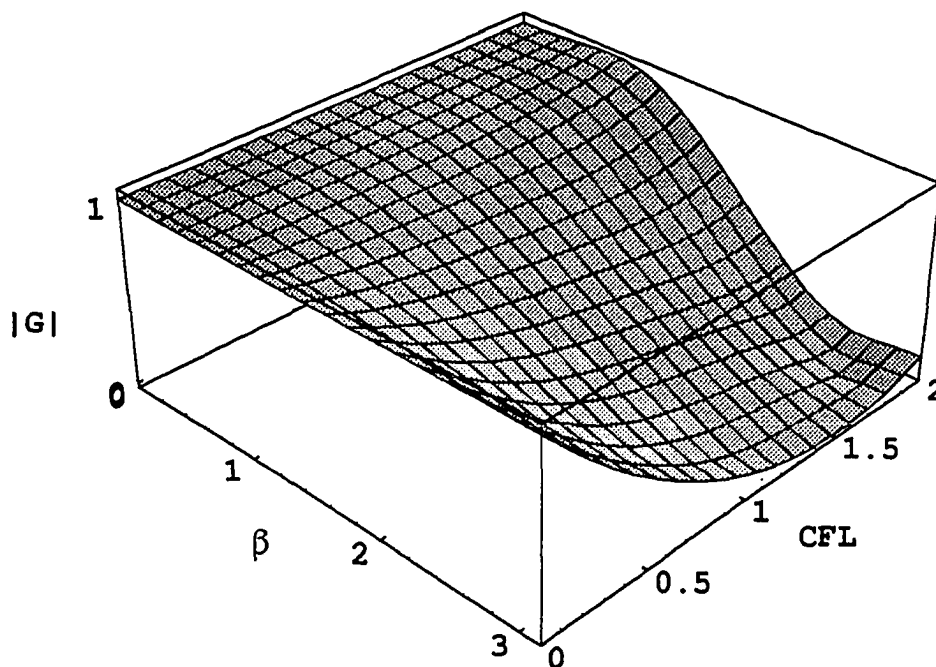


Figure 5.1 Modulus of Amplification Factor as a Function of the Spatial Wave Number and the CFL for the Predictor-Corrector Scheme with Residual Smoother ($\epsilon = 0.5$).

$$\begin{aligned}
\epsilon_\xi &= \max \left\{ \frac{1}{4} \left[\left(\frac{CFL^*}{CFL} \frac{\lambda_\xi}{(\lambda_\xi + \frac{1}{16}(\lambda_\eta + \lambda_\zeta))} \right)^2 - 1 \right], 0 \right\} \\
\epsilon_\eta &= \max \left\{ \frac{1}{4} \left[\left(\frac{CFL^*}{CFL} \frac{\lambda_\eta}{(\lambda_\eta + \frac{1}{16}(\lambda_\xi + \lambda_\zeta))} \right)^2 - 1 \right], 0 \right\} \\
\epsilon_\zeta &= \max \left\{ \frac{1}{4} \left[\left(\frac{CFL^*}{CFL} \frac{\lambda_\zeta}{(\lambda_\zeta + \frac{1}{16}(\lambda_\xi + \lambda_\eta))} \right)^2 - 1 \right], 0 \right\}
\end{aligned} \tag{5.9}$$

Here, ϵ_ξ , ϵ_η , and ϵ_ζ are the adaptive residual smoothing coefficients which are functions of the grid aspect ratio and the spectral radii λ_ξ , λ_η , and λ_ζ ; $\frac{CFL^*}{CFL}$ is the ratio of the CFL number of the smoothed scheme to that of the basic explicit scheme. Increasing the ratio of $\frac{CFL^*}{CFL} > 2.0$ caused the high frequency damping of the scheme to vanish, which was detrimental to multigrid convergence. The smoothing operator was applied after every stage of the Runge-Kutta time-stepping scheme.

5.3 Multigrid Method

The multigrid acceleration technique has been employed in the present work to augment the time-stepping schemes, discussed in chapter four, and to enhance the performance of the developed algorithm. Multigrid is still in its infancy, and a great deal remains to be learned about its performance. The multigrid acceleration technique was developed originally by Fedorenko [45, 46] in 1961. It was further developed by Brandt [47] and applied to an elliptic set of equations. The work by Brandt and many others has led to the popular use of multigrid by many in the fields of applied mathematics and computational engineering. The basis for multigrid is the use of successively coarser grids to calculate corrections to the solution of a partial differential equation (or set of partial differential equations) on a ‘fine’ mesh. These corrections reduce the low frequency components of the error in the fine-grid solution. Since the

coarse-grids contain significantly fewer points than the fine-grid, less work is required to perform a computation there than on the fine-grid. Excellent discussions concerning the development of the multigrid technique can be found elsewhere [48–50]. The multigrid has been used successfully for solving the potential, Euler, and Navier-Stokes equations [51–53, 55]

The development and implementation of multigrid for linear problems is described by Briggs [115]. Unfortunately, many problems in engineering are described by non-linear equations or sets of equations. This is particularly true for computational fluid dynamics. Because of the non-linear nature of the equations, the Full Approximation Storage (FAS) multigrid procedure has to be used [116].

Since some understanding of the theory behind multigrid is necessary in order to use it effectively, a brief development of the Full Approximation Storage (FAS) multigrid for a non-linear problem is presented. Consider the problem

$$L^h U^h = f^h, \quad (5.10)$$

where L^h is a non-linear operator on a grid, g^h , with spacing h . The forcing function, f , is known and U^h is the solution to the problem on the grid with spacing h . Taking u^h as an approximation to U^h with an error

$$V^h = U^h - u^h, \quad (5.11)$$

Equation (5.10) can be written as

$$L^h(u^h + V^h) = f^h. \quad (5.12)$$

$L^h u^h$ is subtracted from both sides of equation (5.12) to give:

$$L^h(u^h + V^h) - L^h(u^h) = f^h - L^h(u^h). \quad (5.13)$$

If the terms are smooth, they can be represented on a coarser grid, g^{2h} with spacing $2h$. The grid g^{2h} is formed by deleting every other point in g^h ; therefore, $g^{2h} \in g^h$. Points are eliminated from g^{2h} to form g^{4h} and so forth to form g^{8h} , g^{16h} , etc. Each subsequent grid is a subset of the previous grid, which places compatibility constraints on the number of grid points in each direction. Written on the coarse-grid, g^{2h} , equation (5.13) becomes

$$L^{2h} \left(I_h^{2h} u^h + V^{2h} \right) - L^{2h} \left(I_h^{2h} u^h \right) = I_h^{2h} \left(f^h - L^h u^h \right), \quad (5.14)$$

or

$$L^{2h} \left(u^{2h} \right) = f^{2h}, \quad (5.15)$$

where

$$f^{2h} = I_h^{2h} \left(f^h - L^h u^h \right) + L^{2h} \left(I_h^{2h} u^h \right), \quad (5.16)$$

and I_h^{2h} is the restriction operator.

Since equation 5.15 is on a coarser grid than equation 5.10, the numerical solution for u^{2h} is much cheaper to obtain because fewer points are involved. Note that the operator used on the coarse-grid has the same form as the fine-grid operator, the grid spacing (h and $2h$) being the only difference. Once the values of u^{2h} are obtained, the fine-grid iterative solution is updated using the following equation:

$$\left(u^h \right)_{New} = \left(u^h \right)_{Old} + I_{2h}^h \left[u^{2h} - I_h^{2h} \left(u^h \right)_{Old} \right] \quad (5.17)$$

where I_{2h}^h is the prolongation operator.

It should be emphasized that the prolonged term on the right-hand side of equation (5.14) is the correction to be applied to the fine-grid solution. Examination of this term shows that the solution on the coarse-grid is actually a solution to the originally posed problem, which allows the use of the fine-grid boundary conditions on all the coarse-grids as well. In the developed algorithm, the non-linear FAS scheme utilizes the same operator on all the grid levels. This of course simplifies the programming of the multigrid scheme.

A grid with spacing $4h$ can then be used to find corrections to the “solution” of the problem on the grid with spacing $2h$. Successively coarser grids may be used until a grid is reached which is so coarse that a direct solution may be used (or a nearly exact solution with only a small number of iteration sweeps). The correction from the coarsest grid is then used to correct the correction on the next finer grid; and this is continued through successively finer grids until the finest level is reached and the approximate solution is updated.

The usefulness of corrections obtained on a coarser grid is dependent on the smoothness of the fine-grid error passed to the coarse-grid. Hence, it is absolutely necessary that the high-frequency components of the error on the fine-grid be minimized, if not completely eliminated. It is the responsibility of the smoother (modified Runge-Kutta and Predictor-Corrector Schemes) to damp the high frequency components of the error. The removal of the low-frequency components of the error is unimportant for all but the coarsest grid since these frequencies can be resolved on the coarser grids where they become high frequencies. If the high frequencies are not damped, then the restriction operator will pass aliased information to the coarser grid and the entire multigrid scheme will cease to converge, [52]. Obviously, the choice of the smoother is critical to the proper functioning of multigrid. Thus the choice of the modified Runge-Kutta coefficients should be tailored to improve the damping properties rather than for a slight increase in the maximum CFL number, as discussed previously. Failure of the 1-2 Scheme to damp the high frequency errors, over a wide range of CFL number (Fig. 4.8), disqualified that scheme for use in multigrid applications.

The cycle of work performed starting on the finest grid, successively treating the coarser grids, and then returning to the finest grid is called one multigrid cycle. The cycles

are repeated until sufficient convergence is obtained on the finest grid. In the present study, fixed cycles known as V- and W-cycles are used, and are given in Appendix D.

The restriction operator has two forms. One form is used to restrict the dependent variables, $I_h^{2h}(u^h)$; i.e., the flow quantities ρ , ρu , ρv , ρw , and e . For these, the volume weighted average of the values of the function at mid-cells of the eight fine-grid cells, contained in a coarse-grid cell, is used to set the value on the coarse-grid and is given by [85] :

$$u^{2h} \simeq I_h^{2h}(u^h) = \sum_{i=1}^8 \frac{Vol_i^h u_i^h}{\sum_{i=1}^8 Vol_i^h} \quad (5.18)$$

The other form of the restriction operator is for the restriction of residuals, $I_h^{2h}[L^h(u^h)]$. Following Cannizzaro *et al* [85], a simple summation of the residuals over the eight fine-grid cells composing the coarse-grid cell is performed such that;

$$L^{2h}(u^{2h}) = I_h^{2h}[L^h(u^h)] = \sum_{i=1}^8 [L^h(u^h)] \quad (5.19)$$

The restriction operations are performed for all interior points. At the inflow/outflow boundaries, only the values of the functions are restricted, with no residual restriction. The residual values are frozen to the fine-grid values and are not updated on the coarse-grids. On wall surfaces, the same boundary conditions are used for all the grids.

The prolongation operation used in the current work was a tri-linear interpolation, in the computational space, of the corrections at the eight coarse-grid cells adjacent to the fine-grid-mid-cell. A practical approach to the coding of a multigrid scheme in Fortran V is presented elsewhere [54, 117].

A constant coefficient implicit corrector smoother was used to remove high frequency errors from the coarse-grid corrections before they were applied to the fine-grid. For the

test cases investigated in this study, the correction smoothing procedure did not enhance the rate of convergence, but it should pay off for high speed flows, [114]. This operator is identical to that used to smooth the residual in the previous section given by eq. 5.3.

CHAPTER 6

MULTI-BLOCK

The demand to resolve the fine details of physical flows has challenged many researchers to find new and efficient computational tools. This challenge has pushed the development of new computer architectures and numerical techniques which permit the solving of complex, realistic geometries and configurations for compressible flow problems. Many problems in engineering are solved using body fitted coordinate systems. However, many aerodynamics designs are often quite complex (geometrically), and quite often, generating a single, body fitted grid for realistic three-dimensional geometries is a difficult task to perform; for some configurations it is almost impossible.

In the present study, a multi-block strategy is employed to allow greater geometric flexibility on structured grids. The multi-block strategy (multizone) has a number of advantages. It alleviates the problem of grid generation for complex configurations [85]. Different types of governing equations can be used on different domains [75]. Multi-block can even allow the use of different numerical techniques and grid topologies on each block [54]. Multiblock also requires less memory if each zone is solved independently.

Several grid methodologies such as, overlaid grids [83], patched grid [118, 119], and blocked grid [85, 120] can be applied to simplify the grid generation, as well as provide geometric flexibility, and mesh refinement. In the present study, blocked grids have been used because the flow properties are conserved automatically across the block interface. The result section will show that this allows discontinuities, within the computational domain, to move freely across the block interfaces. The multi-block strategy along with

the interaction between multi-block, multigrid, and time integration schemes will be discussed in the this chapter. The different boundary conditions applied on the block faces will also be discussed.

6.1 Multi-Block Strategy

A multi-block strategy is used to allow greater geometric flexibility. The solution domain is divided into multiple zones (blocks) and the grid for each zone is then generated. Each block within the computational domain is treated as a three-dimensional box. Each block can have a different grid topology. Different grid topologies are often better suited for a particular flow component or configuration within the computational domain. If the blocks, and block grid topologies are chosen appropriately, the difficulty of generating a boundary fitted grid can be reduced. Also, the placement and control of wall boundary conditions are more flexible. The trade-off is the computational overhead required for communication between the multiple blocks across their respective intersections (interfaces). In reality, the lagging of communication across the interfaces can slow convergence.

Numerical treatment of grid interfaces is of paramount importance for algorithms that employ different grids within the computational domain. Interface boundary conditions, if not handled properly, can cause the numerical solution to degrade at interfaces [121]. In the present multi-block implementation, the grid in adjacent blocks, connected across an interface, is assumed to have C^0 continuity. The grid lines at the block interface are continuous but the slopes are not necessarily continuous. Having C^0 continuity greatly simplifies the handling of the boundary conditions across the interface, and avoids the necessity of spatial interpolation of the data when loading ghost cells at the block interface. It also ensures the accuracy and conservation of flow properties across each interface.

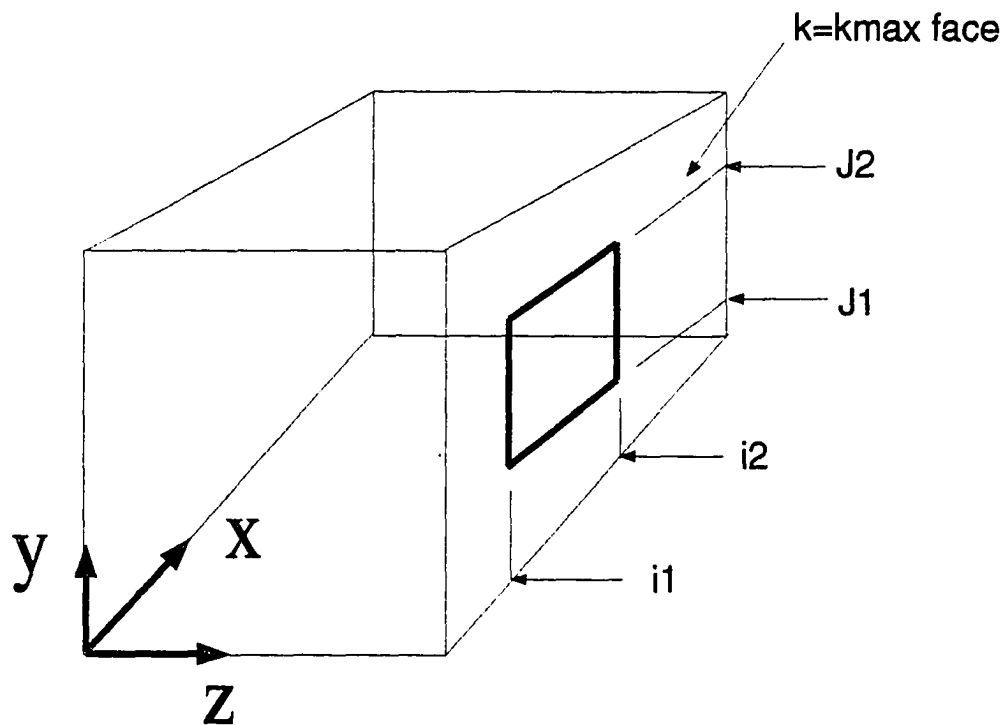


Figure 6.1 Schematic of a Block Face with a Generic
"Patch", Accommodating Multiple Boundary Conditions

This allows discontinuities within the computational domain to move freely across these interfaces as will be shown in the result section.

In the early stages of the present work, the multi-block strategy [85], used a homogeneous boundary condition for any given face of the block. (The entire face of the block had to be a wall, an inflow/outflow boundary, or an interface with another block.) That limitation has been relaxed to allow multiple boundary conditions per face [42, 117]. The face of each block can be divided into rectangular patches, where each patch can utilize a different type of boundary condition, as shown in Fig. 6.1, thus increasing the flexibility of the code for handling complex three-dimensional configurations with different boundary conditions. The boundary conditions for each patch on each block face can be specified in an input file to the algorithm, and they are not "hard-wired" in the source code.

On the block faces that have either a wall or an inflow/outflow boundary condition, standard boundary conditions are used, as will be discussed later. On faces that are interfaces, a special interface routine presets the values in two layers of ghost cells (normal to the face) equal to the latest values in the coincident interior cells in the adjacent blocks. The updates of the interface ghost cells are performed before each iteration in a given block. The iteration on each block can then proceed without the need for further information from adjacent blocks. Hence, each block is thickened by two ghost cell shells which carry the solution from adjoining blocks, as boundary conditions, into the computationally active block.

6.2 Multi-Block and Multigrid

There are two possible strategies for the implementation of multigrid with a multi-block grid structure. Either multigrid inside of multi-block, or multi-block inside of multigrid can be used. The first strategy implies that a complete multigrid cycle (or cycles) will be performed for a given block. Subsequently, computation moves to the next block and so forth until all the blocks are complete. This strategy can be advantageous since it allows the flexibility of different numbers and/or types of multigrid cycles for different blocks, and they can be adjusted to speed convergence in slowly converging blocks (assuming only steady-state results are sought). Unfortunately, communication between the blocks is reduced which slows convergence.

With the multi-block inside of multigrid strategy, the multi-block structure is just a way to update all the points on grid h in the multigrid cycle. Then a restriction is performed on all the blocks and the multigrid process is continued on each block for grid $2h$. This continues for each of the multigrid grids, and allows communication between the coarse-grids in the multigrid cycle, through updates of the interface conditions. It

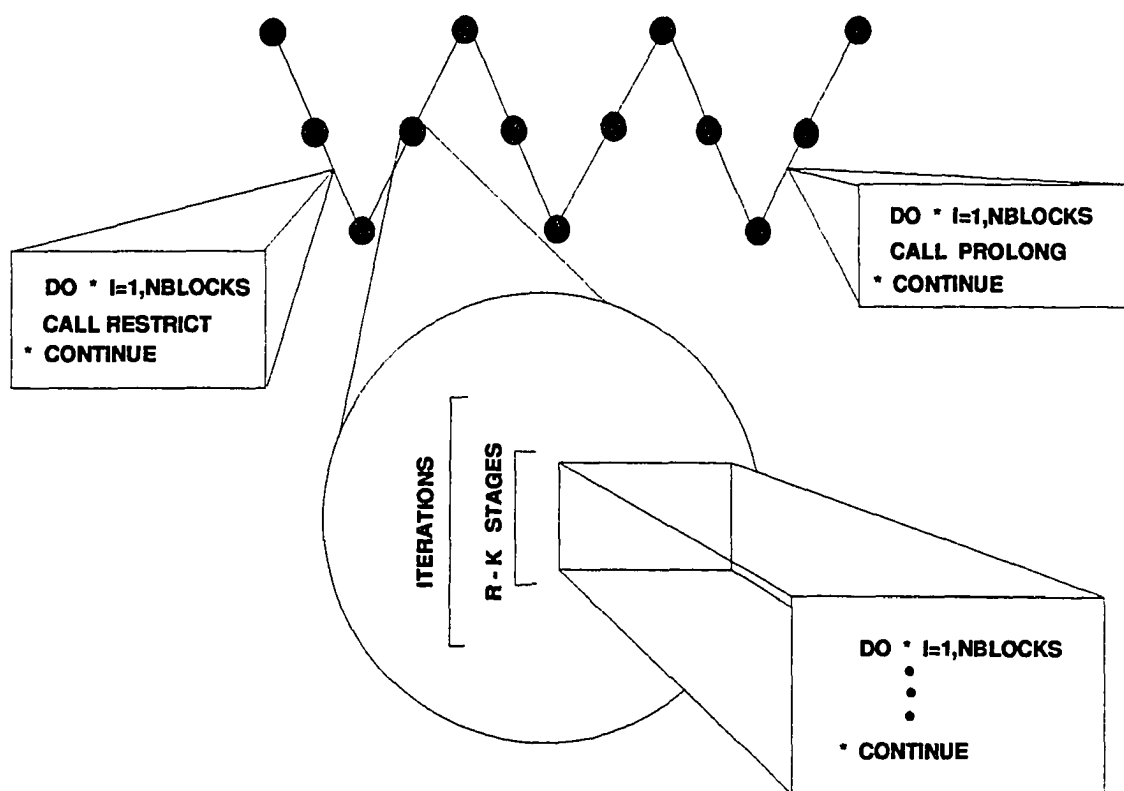


Figure 6.2 Schematic of Multi-Block-Multigrid Strategy

also eliminates the need to either freeze values for the ghost cells at their fine-grid values or invent some interface boundary condition on the coarse-grids. This method can also reproduce the convergence history of a single block solution using an explicit algorithm, and can be used to validate the multi-block logic. The multi-block inside of multigrid strategy was used in the present work. A schematic of the strategy is shown in Fig. 6.2.

6.3 Boundary Conditions

When solving computational fluid dynamic problems, several types of boundaries can be encountered. These boundaries can be real boundaries or artificial boundaries. The real boundaries can be simple solid or porous surfaces or complex wing-body junctures, while artificial boundaries can be far field boundaries or symmetry planes. These boundaries are the link by which the computational domain senses the rest of the “universe”. They drive the solution in the computational domain. An inappropriate boundary condition or

boundary procedure can have a destabilizing effect on the numerical solution. It can be the difference between fast convergence to steady-state or no convergence and instabilities. Thus it is of paramount importance, when solving fluid dynamic problems numerically, to select and implement appropriate boundary conditions and boundary procedures.

In the present study, each block within the computational domain is treated as a three-dimensional box. This box has six faces and it is on these faces that the boundary conditions have to be applied. Several types of boundary conditions have been incorporated in the developed algorithm corresponding to the different test cases investigated. These boundaries are inviscid/viscous solid walls, symmetry planes, inflow/outflow boundaries, and the interface between blocks. To facilitate the treatment of the boundary conditions and the evaluation of the fluxes at the boundaries, two layers of ghost cells (virtual/phantom cells) are used at the boundaries, as shown in Fig. 6.3. The ghost cells have their own volume and directed areas, and are similar to any other cell in the computational domain, except they are not updated during the block computations. The memory allocation for each of the blocks is increased by two planes on each of the six faces. A description of the various types of boundary conditions and their implementation in the numerical algorithm will be discussed in sections which follow.

Solid Boundary When solving the Euler Equations, the boundary conditions to be applied at solid boundaries are flow tangency conditions. The velocity component normal to the solid boundary is set equal to zero. The pressure is extrapolated linearly from inside the computational domain to the wall. One only needs the pressure at the wall to compute the inviscid flux [122]. More complex boundary conditions exist in the literature but the current approach has been robust and produced accurate results for the test cases studied.

When solving the Navier-Stokes equations, we enforce the no slip and the no injection boundary condition at the solid boundary. The pressure is extrapolated from inside the

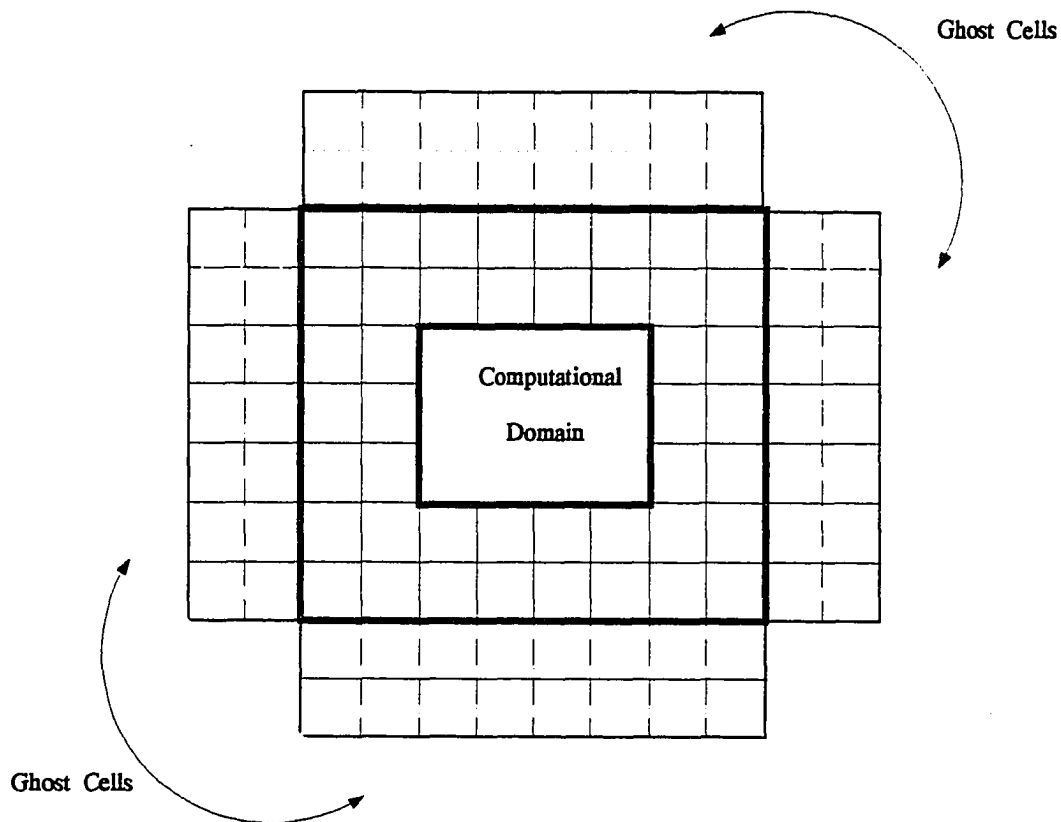


Figure 6.3 Schematic of a Plane in the Computational Domain

computational domain. The treatment of the inviscid part of the flux is similar to the treatment pursued for computing the inviscid flux in the Euler Equations. In all the viscous test cases investigated, it was assumed that heat transfer between the fluid and the solid boundary was negligible; an assumption of an adiabatic wall was made. The thermodynamic properties in the ghost cells, for physical boundaries, are set equal to the properties in the cells adjacent to the boundary. The velocity in the ghost cell is computed by requiring the average of the velocity in the ghost cell and the cell adjacent to the boundary to be equal to zero on solid boundaries.

Symmetry Plane The third type of boundary condition encountered in this study is the symmetry plane boundary condition. The values in the ghost cells are set to be the mirror image of the interior cells at the symmetry plane. The evaluation of the flux is the same

as the evaluation of flux at any point in the computation domain. The only difference is that the symmetry plane flux utilizes information stored in the ghost cells.

Inflow/Outflow Boundary. A non-reflective type of boundary condition at the far field is essential to minimize the reflection of non-physical outgoing disturbances. Thus a characteristic non-reflective type of boundary condition is used to compute the flow variables in the ghost cells at the inflow/outflow boundaries. This type of boundary condition is based on characteristic variables and the assumption that the flow is steady and locally homentropic at the boundary. The procedure for implementing the boundary condition was developed by Thomas and Salas [123] for two-dimensional flows. The derivation and application of the characteristic boundary conditions for three-dimensional flows are discussed in detail elsewhere [100].

In applying the characteristic boundary conditions for the developed algorithm, one can proceed by computing the Mach number normal to the boundary, at the first interior cell in the computational domain. That Mach number is used to determine the nature of the flow; subsonic or supersonic flow. If the flow is subsonic, the two Riemann invariants R^+ , and R^- , are computed as:

$$\begin{aligned} R^+ &= q_{int} + \frac{2}{\gamma - 1} a_{int}, \\ \text{and } R^- &= q_{ref} - \frac{2}{\gamma - 1} a_{ref}, \end{aligned} \quad (6.1)$$

where a and q are the speed of sound and the contravariant velocity normal to the boundary. The subscript *int* and *ref* indicate the first interior cell to the boundary and at reference conditions, respectively. Here, q_{ref} , and q_{int} are given by:

$$\begin{aligned} q_{ref} &= u_{ref} \hat{l}_x + v_{ref} \hat{l}_y + w_{ref} \hat{l}_z, \\ q_{int} &= u_{int} \hat{l}_x + v_{int} \hat{l}_y + w_{int} \hat{l}_z, \\ \hat{l}_n &= \frac{l_n}{\sqrt{l_x^2 + l_y^2 + l_z^2}}, \quad \text{where } l = \xi, \eta, \zeta, \text{ and subscript } n = x, y, z \end{aligned} \quad (6.2)$$

We can add and subtract the two Riemann invariants to compute the local velocity normal to the wall, q_g , and the speed of sound a_g such that

$$\begin{aligned} q_g &= 0.5(R^+ - R^-) , \\ a_g &= 0.25(\gamma - 1)(R^+ - R^-) \end{aligned} \tag{6.3}$$

If the computed q_g , normal velocity, is positive, the boundary is a subsonic outflow. Thus the entropy and the tangential velocity are carried outside the computational domain by the outgoing characteristic waves. The Cartesian velocity components and entropy in the ghost cells are then computed as follows

$$\begin{aligned} u_g &= u_{int} + (q_n - q_{int})\hat{l}_x, \\ v_g &= v_{int} + (q_n - q_{int})\hat{l}_y, \\ w_g &= v_{int} + (q_n - q_{int})\hat{l}_z, \\ s^* &= \rho_{int}^\gamma / p_{int} \end{aligned} \tag{6.4}$$

where s^* is an entropy related function.

If q_g , is computed to be negative, the boundary is a subsonic inflow. Thus the entropy and the tangential velocity are carried inside the computational domain by the incoming characteristic waves. The Cartesian velocity components and entropy in the ghost cells are then computed as

$$\begin{aligned} u_g &= u_{ref} + (q_n - q_{ref})\hat{l}_x, \\ v_g &= v_{ref} + (q_n - q_{ref})\hat{l}_y, \\ w_g &= w_{ref} + (q_n - q_{ref})\hat{l}_z, \\ s^* &= \rho_{ref}^\gamma / p_{ref}. \end{aligned} \tag{6.5}$$

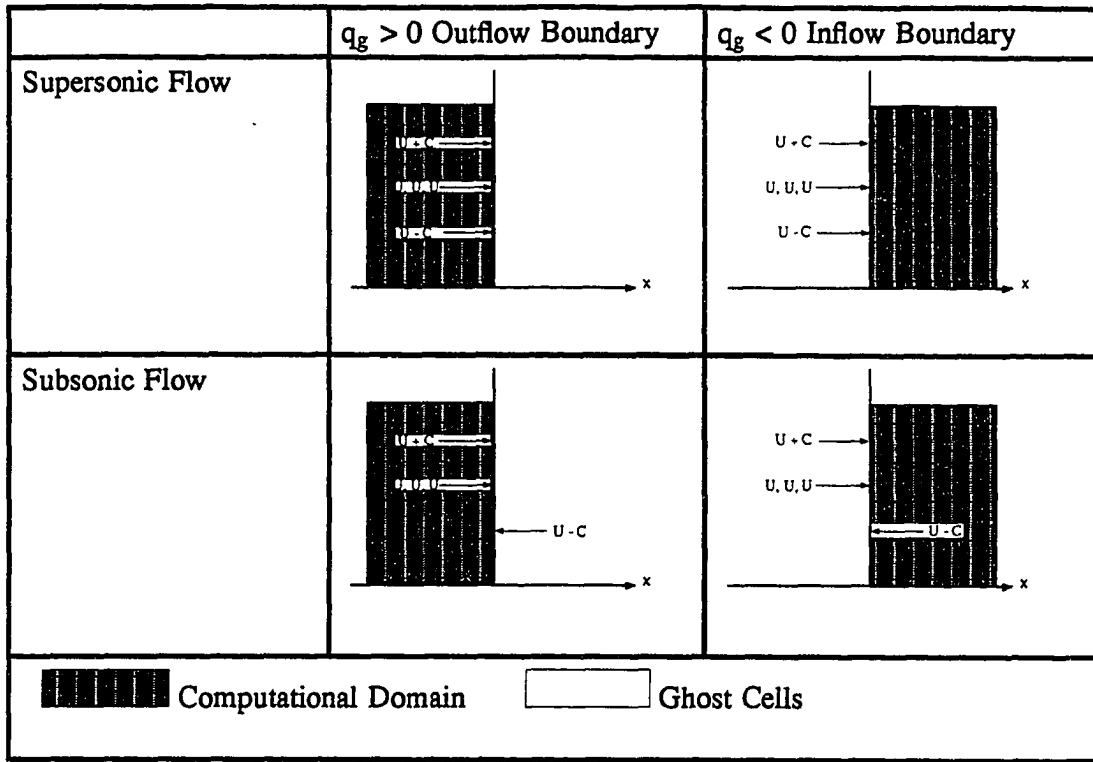


Figure 6.4 Inflow and Outflow Boundary Conditions

The density, pressure, and the total energy in the ghost cells are computed for both the subsonic inflow and outflow as:

$$\begin{aligned}
 \rho_g &= \left(\frac{a^{2s*}}{\gamma} \right)^{\frac{1}{\gamma-1}}, \\
 p_g &= \frac{a^2 \rho_g}{\gamma}, \\
 \text{and } E_g &= \frac{p_g}{\gamma-1} + \frac{1}{2} \rho_g (u_g^2 + v_g^2 + w_g^2).
 \end{aligned} \tag{6.6}$$

For supersonic inflow, all characteristic waves are entering the computational domain and the flow variables in the ghost cells are set to the freestream values. In the case of supersonic outflow, all characteristic waves are leaving the computational domain and the values in the ghost cells are extrapolated from the interior. Second-order extrapolation was always implemented. Figure 6.4 shows the four different scenarios for the inflow/outflow boundary.

It should be emphasized that the characteristic boundary conditions are valid for steady, locally homentropic flow. For viscous dominated regions, a simple, robust and reasonable procedure is to extrapolate all the variables at the downstream boundary from the interior.

Interface Between Blocks This type of boundary condition only arises in a multi-block domain at the interface between various blocks. This boundary is not a physical boundary and it is of paramount importance to treat the computational cells at the interface with the highest level of care, in order to ensure accurate transfer of data from one block to another. The interface should be transparent to the flow of information across it.

Two 'ghost' cells are used to pass all the necessary information from a neighboring block to a cell at the face of a given block without degradation of accuracy at the interface. A special interface subroutine was used to load the data from the internal cells in the neighboring blocks into the proper ghost cells. This process was performed after the other boundary conditions were enforced, and between each sub-iteration on a given multigrid level. The flux at the interface is calculated in the same way as it is calculated at any other point within the computational domain. The only difference is that the interface plane utilizes information from its neighboring block. It should be noted that the present algorithm has the capability for a block to interface with itself.

CHAPTER 7

TEST CASES

7.1 Background

The developed algorithm was utilized to compute several test cases of interest. The main objective of the computations were to validate, demonstrate, and assess the predictive capability of the algorithm. In the present chapter, results of the computation of corner flow, plume flow, laminar and turbulent flow over a flat plate, an ONERA M6 wing, and the unsteady impingement of a jet on a ground plane are reported. Each test case was computed in order to verify a certain aspect of the developed code, as will be shown in the following sections.

The first test case considered was an inviscid, three-dimensional supersonic flow through a corner of intersecting ramps. That flow has been investigated both experimentally and numerically by several researchers, [124–127]. The flow field encountered is highly complex and is dominated by complicated shock structures, shock interactions, shear layers, and shock boundary-layer interactions. A schematic of the flow structure is given elsewhere [127], and is repeated here as Fig. 7.1.

Typical examples of such flows are supersonic/hypersonic inlets and wing body junctures. In the case of supersonic and hypersonic inlets, the associated shock structure is of paramount importance, since that flow is convected through the combustion chamber and can lead to non-uniform combustion. Therefore, the development of efficient algorithms which can predict accurately the flow field, while consuming reasonable levels of computer resources is highly desirable. Clearly the present numerical scheme should

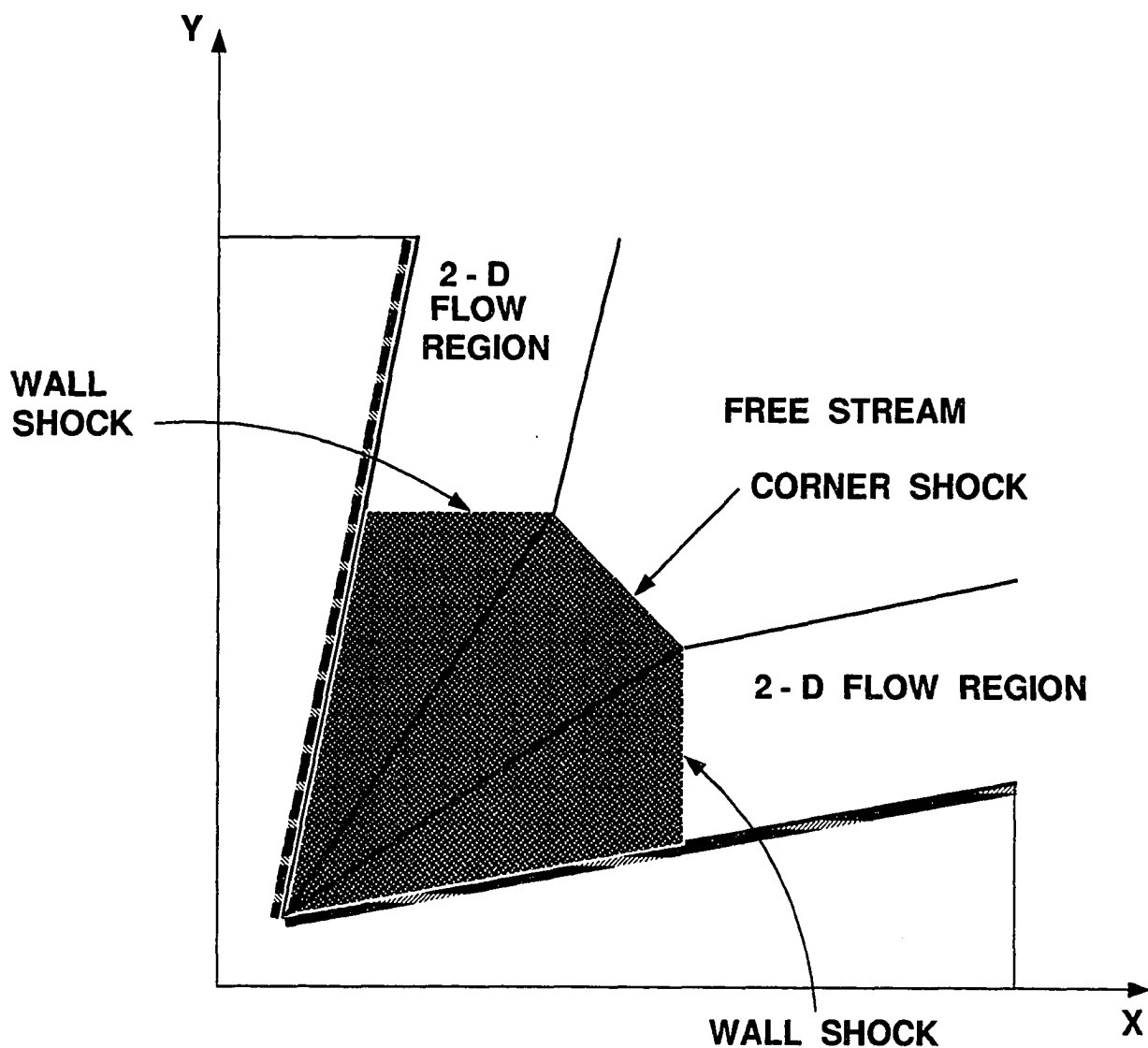


Figure 7.1 A Schematic Diagram of a Cross Plane for Supersonic Corner Flow.

be capable of accurately resolving the complicated shock structure on a reasonably fine mesh. Even in its simplest form, (inviscid formulation), the flow problem considered is of significant physical importance, and represented a challenge to the numerical algorithm.

The second test case computed was the non-axisymmetric jet exhaust plume. Several studies have been conducted, [75, 128, 129], to investigate the performance of non-axisymmetric jet exhaust plumes. The motivation for the non-axisymmetric jet flow studies has been the fact that they can provide comparable or superior level flight

performance to axisymmetric nozzles, [129]. The rectangular exit nozzle provides the jet engine with improved thrust vectoring and thrust reversing capability, which can enable a substantial reduction in landing and take off distance requirements while enhancing the maneuverability in flight. An additional advantage of the rectangular exit over the axisymmetric nozzle is that the design and structure of the rectangular nozzle is simpler and lighter for the same capabilities.

Propulsive nozzles of this type have been considered for a wide range of possible applications in airplanes such as the Advance Tactical Fighter (ATF), the short takeoff and landing (STOL) Eagle F-15 (F-15B) and the National Aerospace Plane (NASP).

A better understanding of the type of flow surrounding the nozzle and plume region is needed because of its influence on the sonic boom signature and aerodynamic performance, [128].

The non-axisymmetric jet exhaust plume flow is complex physically and represents a challenging test case for the algorithm developed in this study. Expected exhaust plume mixing (with the surrounding flow) will occur at transonic conditions, accompanied by large embedded supersonic flow regions and possibly complicated shock structures, shock interactions, slip lines and shear layers. A schematic of the flow field showing the shock wave, expansion fan, and slip line are demonstrated in Fig. 7.2.

Laminar and turbulent flows over a flat plate were computed to verify the implementation of the viscous terms and turbulence model (Baldwin-Lomax algebraic-turbulence model). The turbulent flow over an ONERA M6 wing was also computed in order to compare the performance of the developed algorithm with other three-dimensional state-of-the-art computer codes.

The ONERA M6 wing is a basic three-dimensional configuration. The flow field around the wing contains a wealth of aerodynamic features such as; shock waves,

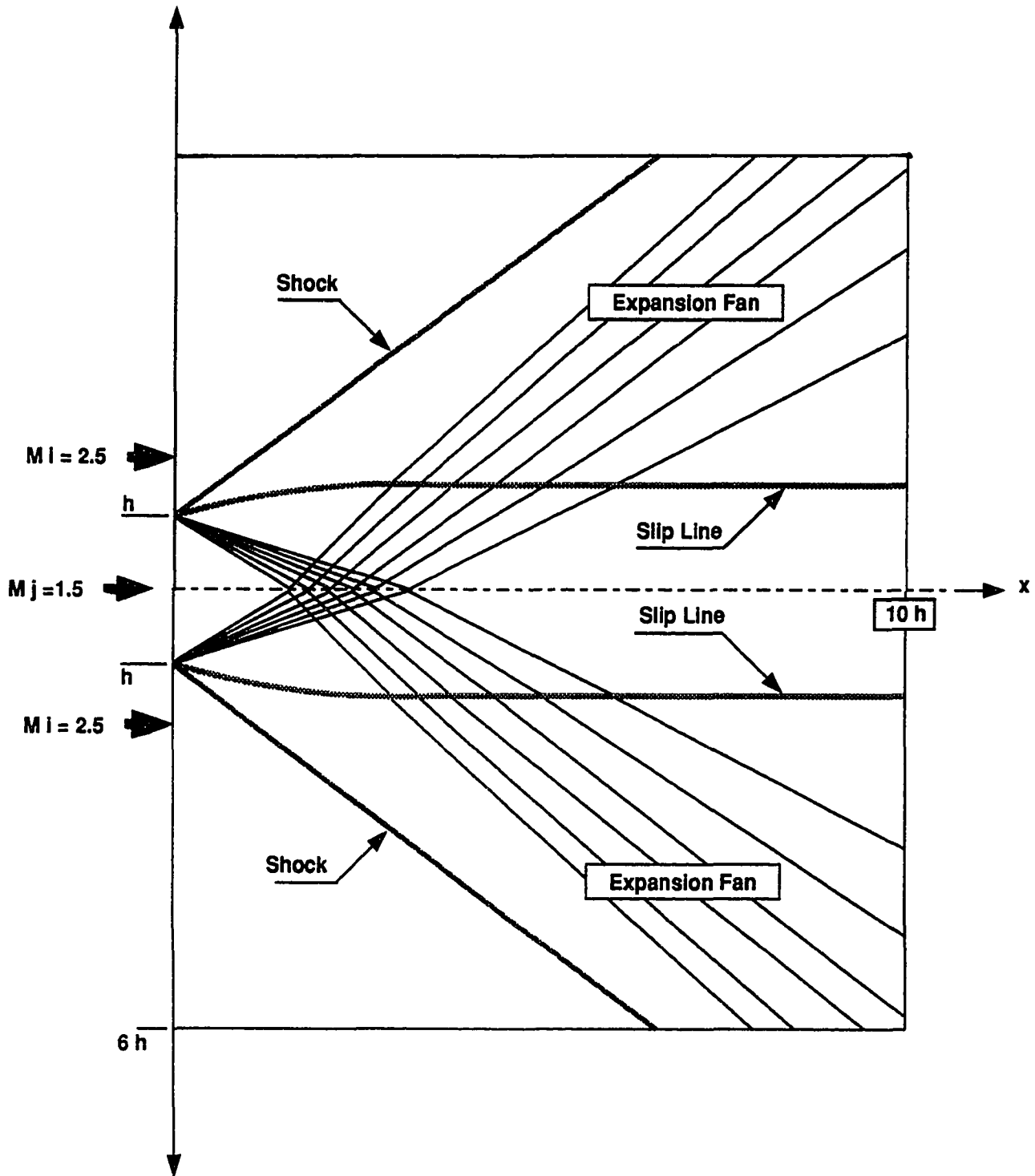


Figure 7.2 Schematic of Pseudo-Two-Dimensional Jet Exhaust Plume

spanwise variations in boundary layers, and the interaction between shock waves and the boundary layer. The wing has been developed for the experimental support of three-dimensional, transonic and subsonic flow fields and has been used extensively as a benchmark case to gauge the accuracy and performance of newly developed computational codes. Both Euler and Navier-Stokes solutions have been reported by several researchers [130–133], and there is also a large data base of experimental pressure distributions available, [134]. Hence the ONERA M6 wing has been selected to validate the capability of the developed algorithm to compute a truly three-dimensional turbulent-flow, and to assess the performance of the developed algorithm with other three-dimensional, state-of-the-art computer codes.

The flow of a jet impinging on a ground plane was the last test case conducted to validate the developed algorithm. Impinging jets, with and without cross flow, occur in a wide variety of engineering applications (cooling, heating, and drying of a variety of industrial products, tempering of glass, cooling of turbine blades, paint spraying, and welding, are just a few). Particular attention has been devoted to the problem, because of its application to V/STOL (vertical/short takeoff and landing) aircraft [135–147]. A comprehensive review, of previous experimental and numerical work, for an isolated jet impinging on the ground plane is given by Jalamani [137]. An up-to-date review of impinging jets in cross flow is given by Bray and Knowles [144]. The associated flow field of impinging jets, is highly complex, and unsteady. The flow field contains highly sheared layers, vortical regions, an impingement zone, free jet, and a wall jet. The jet impingement problem is a demanding test case. The analysis of such a complicated flow field, requires the solution of the three-dimensional, time-dependent, compressible, Navier-Stokes equations, to be capable of accurately resolving the entire flow field.

A Schematic of the flow field of an isolated jet impinging on a ground plane is shown in Fig. 7.3. The numerical simulation of such a complicated flow field will demonstrate the predictive capability of the developed algorithm, to resolve complex unsteady flows.

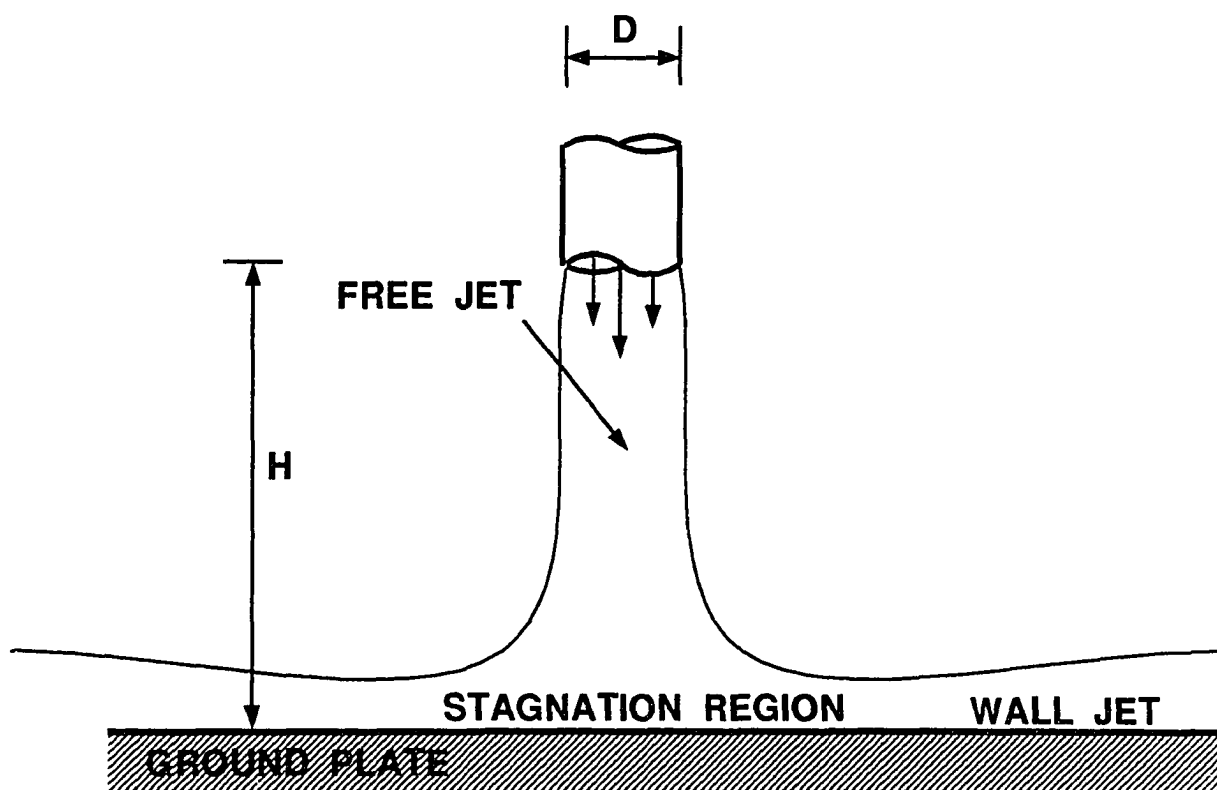


Figure 7.3 Schematic of an Isolated Jet Impinging on a Ground Plane.

When a jet impinges on a ground plane a wall jet is formed. The wall jet flows radially outwards from the zone of impingement. In the presence of a crossflow, the wall jet flowing radially outward is opposed by the crossflow (free stream) and rolls up into a horseshoe-ground vortex as shown in Fig. 7.4. The ground vortex is the primary mechanism for hot gas ingestion, creating dust clouds, and causing lift loss for the V/STOL aircraft.

There is a significant amount of scatter in the database available for jet impinging studies [144]. Extra numerical and experimental research work is highly recommended

to complete the development cycle of V/STOL aircraft. Developing a tool that can be used to simulate numerically the flow field of a jet impinging on a ground plane, with and without cross flow is highly desirable.

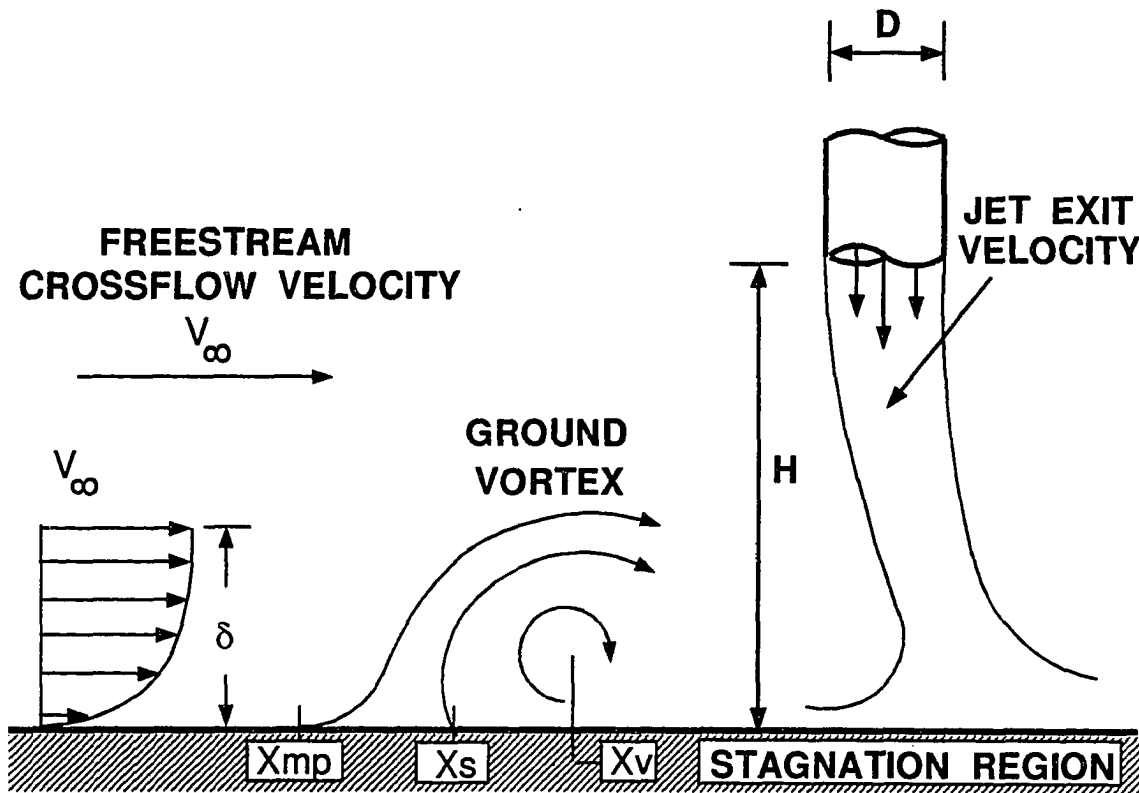


Figure 7.4 Schematic of a Jet Impinging on a Ground Plane in Presence of Crossflow

Even in its simplest form, the flow problems to be considered in the proposed work are of significant physical complexity. The computed flow fields will contain most of the rich features of fluid mechanics (shocks, rarefaction waves, shear layers, etc.). Clearly the numerical scheme should be capable of resolving all of these features accurately on a reasonably fine mesh. The developed algorithm must prove to be robust and reliable to provide the user with the necessary confidence in making design decisions based on the results obtained from this scheme.

7.2 Corner Flow

The flow through a rectangular channel configuration with two compression ramps forming a compression corner about which the channel is symmetric was the first test case computed. A schematic of the compression corner is shown in Fig. 7.5. The back and bottom walls of the channel have converging ramps, each with inclinations of 9.5° . A supersonic inlet flow Mach number of 3.0 was used for the first test case. The 2-2 Predictor-Corrector, Explicit Scheme was employed to compute the flow field. Mach contours on the two side-walls, and the exit plane are shown in Fig. 7.6, while the pressure contours are shown in Fig. 7.7. The Mach and pressure contours show clearly that three shock surfaces are generated. Two of the shock surfaces are two-dimensional

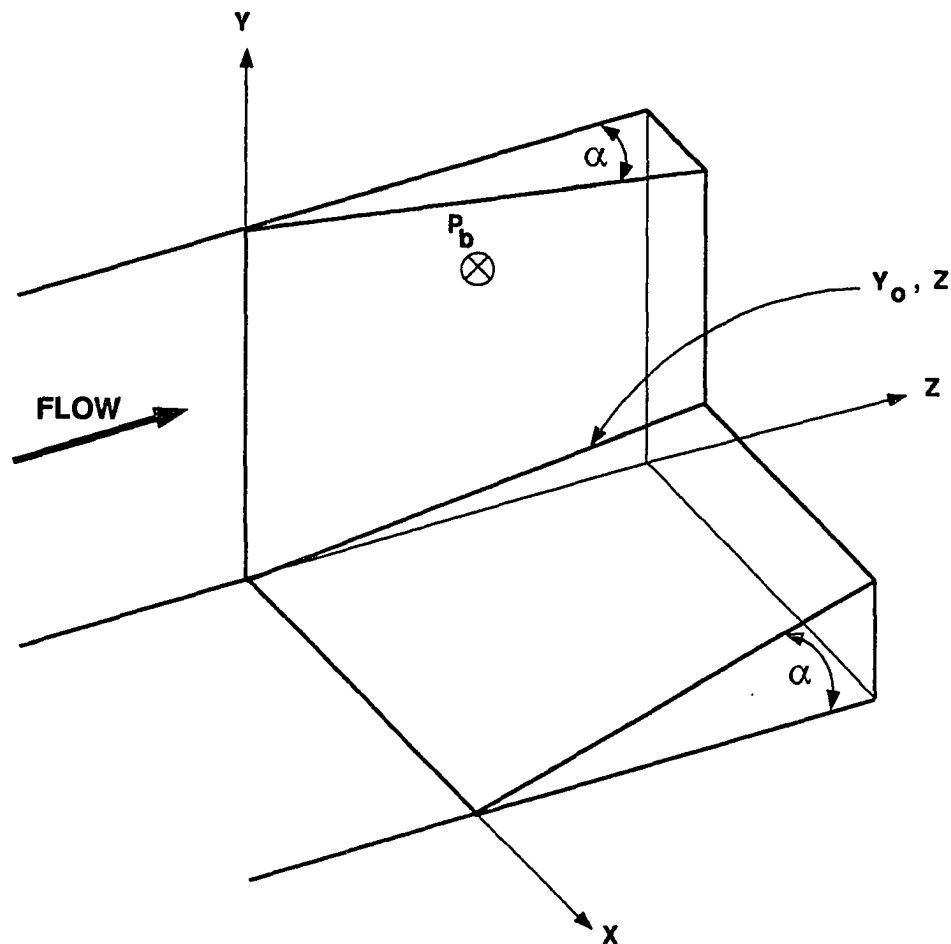


Figure 7.5 Schematic of Compression Corner.

wedge-flow shocks, which can be verified using a two-dimensional analysis based on the Mach number normal to the leading edge of the wedge. The third surface is formed where the two wedge shocks coalesce to form a three-dimensional flow region which is shaped like a cone. This can be seen in Fig. 7.6, where the Mach line contours are shown on the back and bottom walls, and on the exit plane of the channel. The positions of the wedge shocks are apparent on the back and bottom walls as regions of highly concentrated Mach lines perpendicular to the flow direction. Also, on these two walls the edges of the cone shaped surface can be seen. On the exit plane, four flow regions are present. In the upper right corner is free stream, which is one-dimensional flow. From the middle of the plane to the lower left corner (where the flow is three-dimensional), the bottom of the cone surface appears as a partial disc. The two wedge shock planes

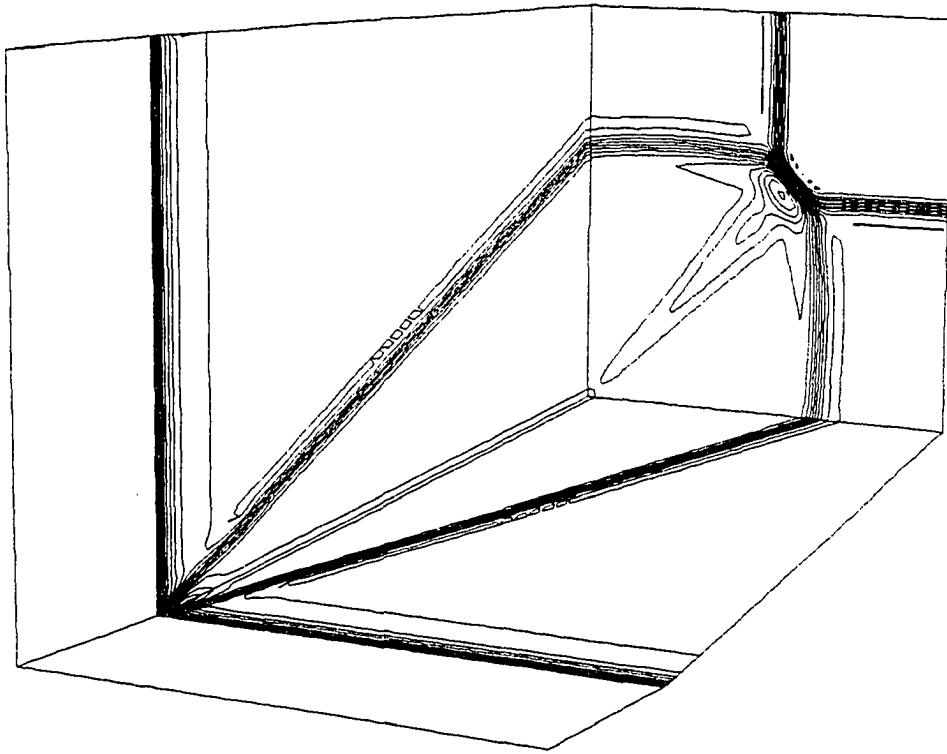


Figure 7.6 Mach Contours, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$.

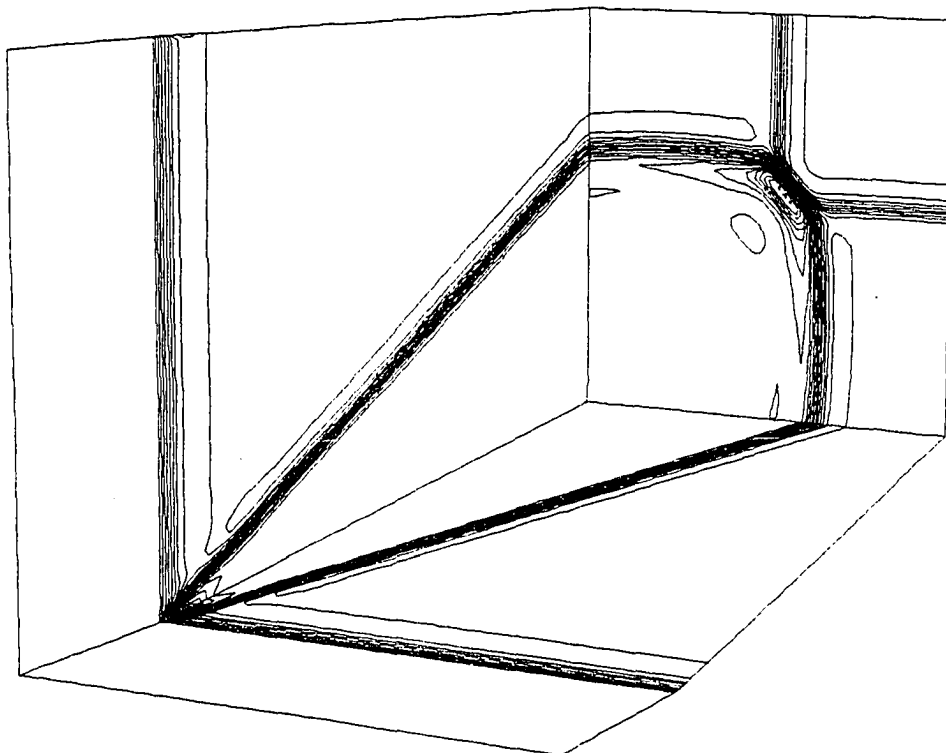


Figure 7.7 Pressure Contours, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$.

can be seen in the upper left corner and the lower right corner of the exit plane. Note that since the geometry of the channel is symmetric about the compression corner (the one joining the back and bottom walls to the exit plane of the channel), the steady-state flow field is symmetric about the corner.

A schematic of the different flow zones, at the exit plane is discussed by Marconi [127], and is shown in Fig. 7.1. The wall shock, corner shock, two-dimensional flow regions, can be identified, along with the one-dimensional free stream flow, and the three-dimensional regions in Fig. 7.1. A qualitative comparison between the computed results, Fig. 7.6, and Fig. 7.7, and the schematic diagram of the flow field [127], clearly shows that the present algorithm captured all four zones in the flow field accurately, and resolved the complicated shock structures. The triple points, where the one-dimensional free stream region meets, the two-dimensional wedge flow and the three-dimensional flow region, have been captured by the present computation, as shown in Fig. 7.6 and 7.7.

A comparison between the computed results, the two-dimensional results of Marconi [127], and Kutler [126], and the experimental data of Charwat and Redekeopp [124] is shown in Fig. 7.8. Marconi [127] used a shock-fitting code to obtain his results, while Kutler [126] used a second-order central-difference shock capturing scheme. The present results are in good agreement with the numerical and experimental results. The present numerical results, as well as Kutler's [126] results, suffer from over- and under-shoots at the shock wave, as expected. It should be emphasized that these oscillations are due to the fact that the shock wave is cutting obliquely across the grid lines. Pressure oscillations near the shock wave are evident in Kutler's results. The reference pressure, P_b , in Fig. 7.8, is the pressure value on the side wall in the two-dimensional flow region. Y_o is the point where the two compression ramps intersect, corresponding to the location of P_b , as shown in Fig. 7.5.

To compare the performance of the different time integration schemes, the inviscid corner flow was recomputed on a $33 \times 33 \times 33$ grid. Comparison between the convergence history to achieve steady-state for the explicit two-stage predictor-corrector schemes (1-2 and 2-2 Schemes) and the four-stage $\kappa = 0$ explicit scheme are shown in Fig. 7.9. It should be emphasized that the predictor-corrector schemes need only about half the CPU time per iteration, compared to the four-stage, explicit time-stepping schemes. The convergence history to steady-state for the four-stage explicit scheme is shown in Fig. 7.10. By comparing Fig. 7.9 and Fig. 7.10, it can be seen that the best performance

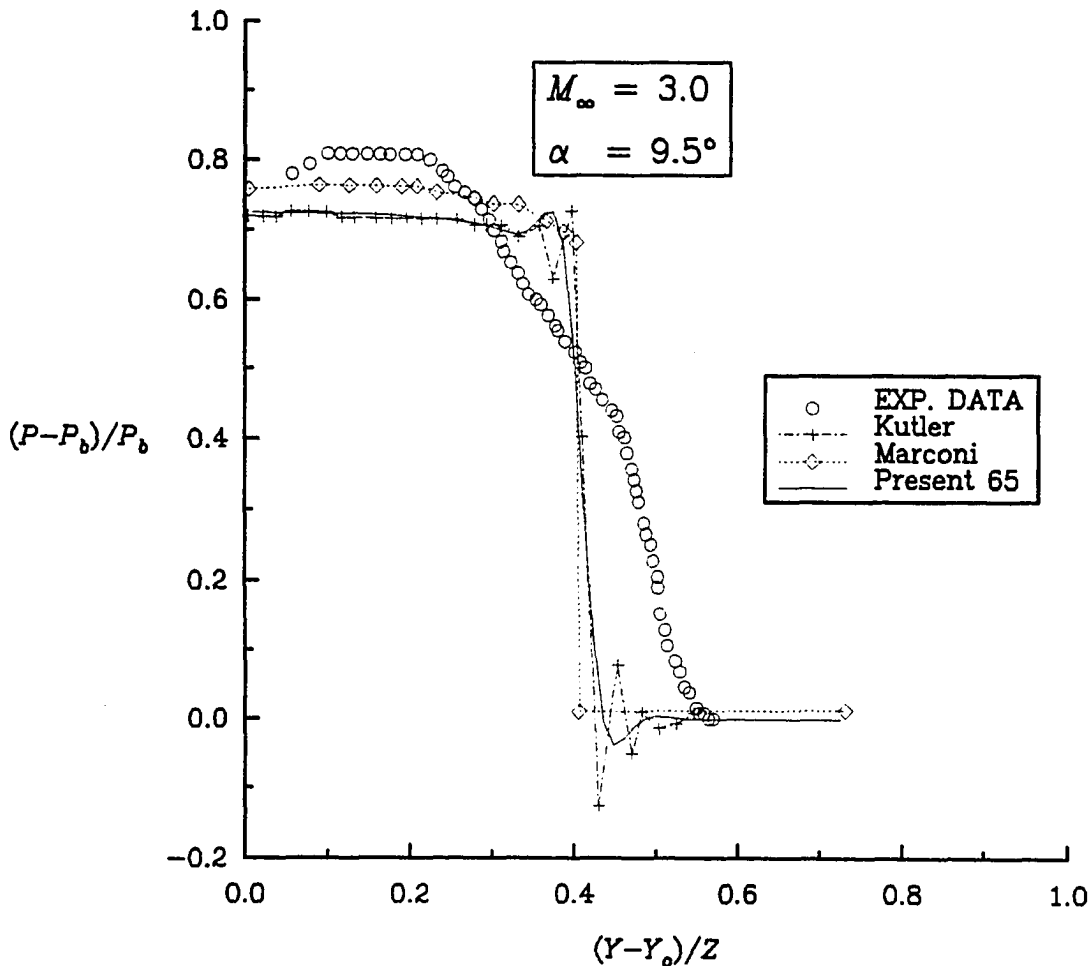


Figure 7.8 Comparison of Numerical and Experimental
Surface Pressure Distributions of Corner Flow.

was achieved with the 1-2 Scheme for this particular problem. The main drawback of the 1-2 Scheme is its incompatibility with multigrid acceleration techniques as shown in Fig. 7.11. This is in agreement with the stability analysis study reported earlier in chapter four. The second drawback was that the convergence rate of the 1-2 Scheme deteriorates on fine (viscous) grids.

To validate the multi-block capability of the algorithm, and to ensure that the block interface is transparent to the numerical scheme, the inviscid corner flow was recomputed using eight blocks. The previous single block grid was used as a starting point to generate the grids in the eight blocks by dividing the domain in half in each of the three coordinate directions (Fig. 7.12). The Mach contours on the two side walls, and the exit plane for the eight block calculation are shown in Fig. 7.13, while the pressure contours are shown in Fig. 7.14. Results of the multi-block calculation reproduce the results obtained with the single block calculation identically. As shown in Figs. 7.13 and 7.14, the block interface is transparent to the developed algorithm.

The convergence histories for the single-block and the eight-block calculations are shown in Fig. 7.15. Notice that the convergence rate for the two-step explicit scheme shows little degradation for the multiple block calculation. This is due primarily to the choice of the multi-block-inside-of-multigrid strategy which allows communication between the coarse-grids in the multigrid scheme. The small differences between the two convergence histories are due to the differences in overhead due to multi-block data transfer.

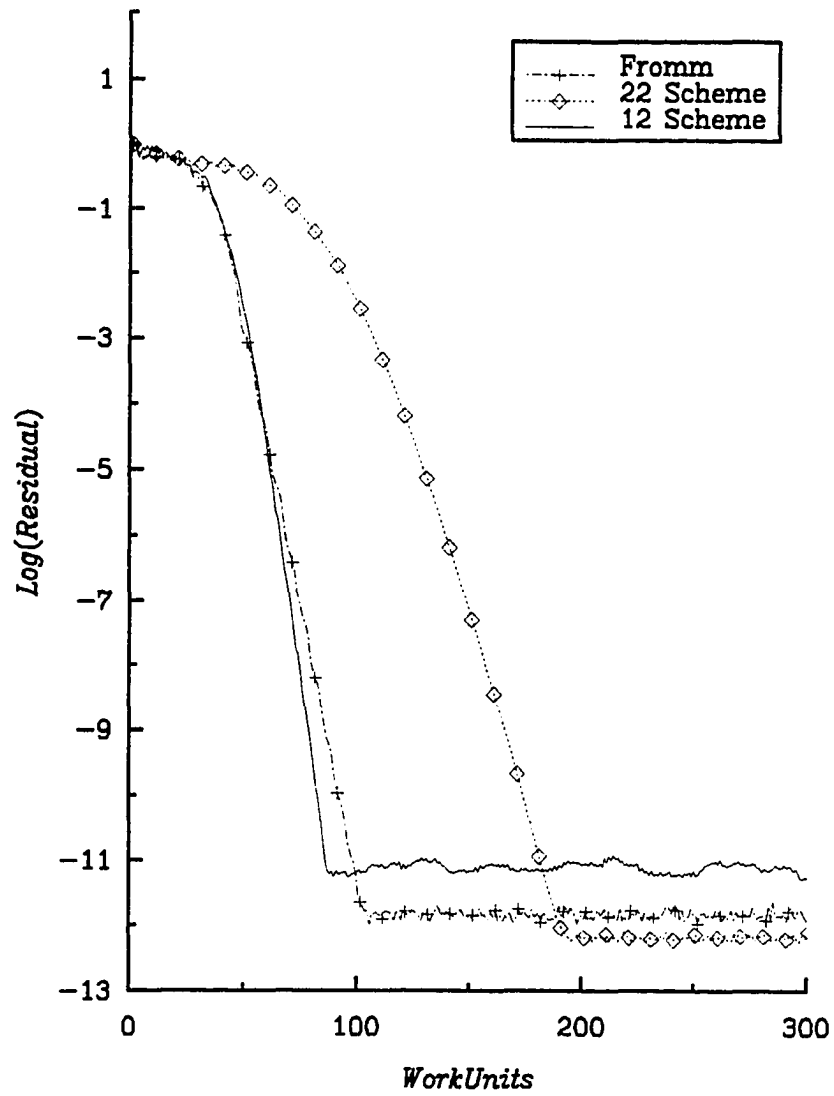


Figure 7.9 Comparison of Convergence Histories of 1-2, 2-2, Four-Stage, $\kappa = 0$ Schemes Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)

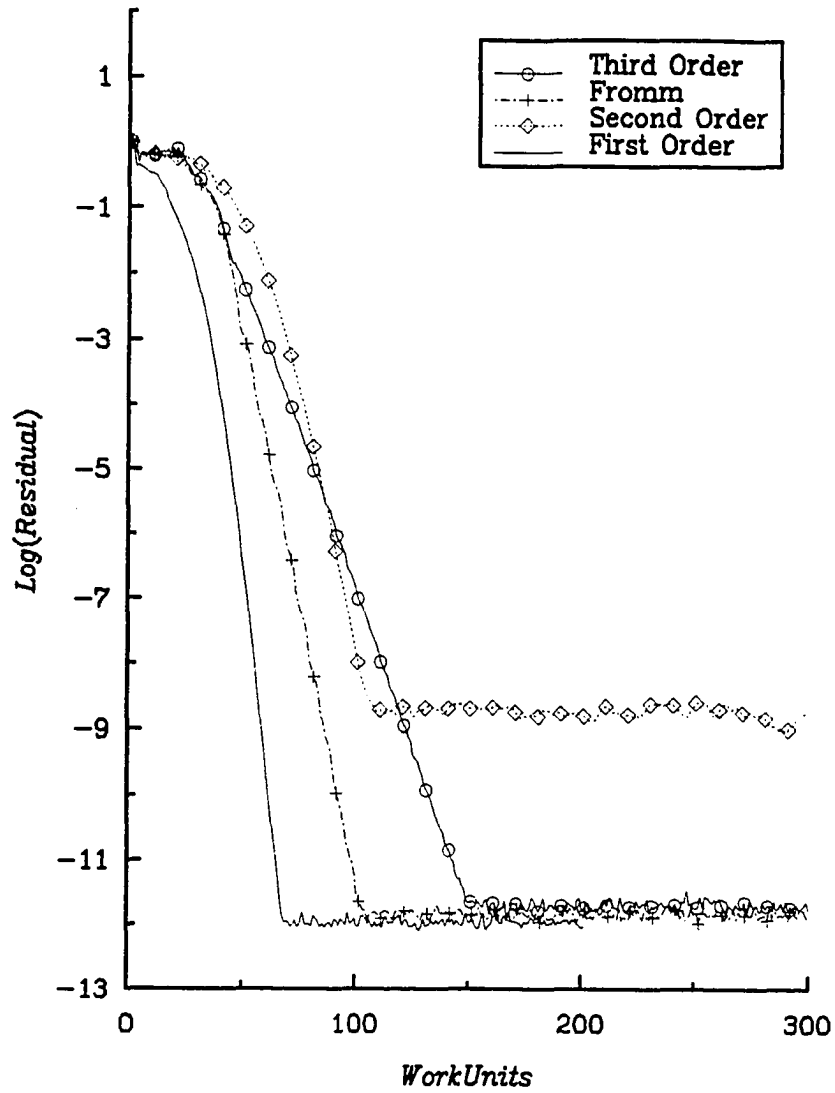


Figure 7.10 Comparison of Convergence Histories of the Four-Stage Schemes, Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)

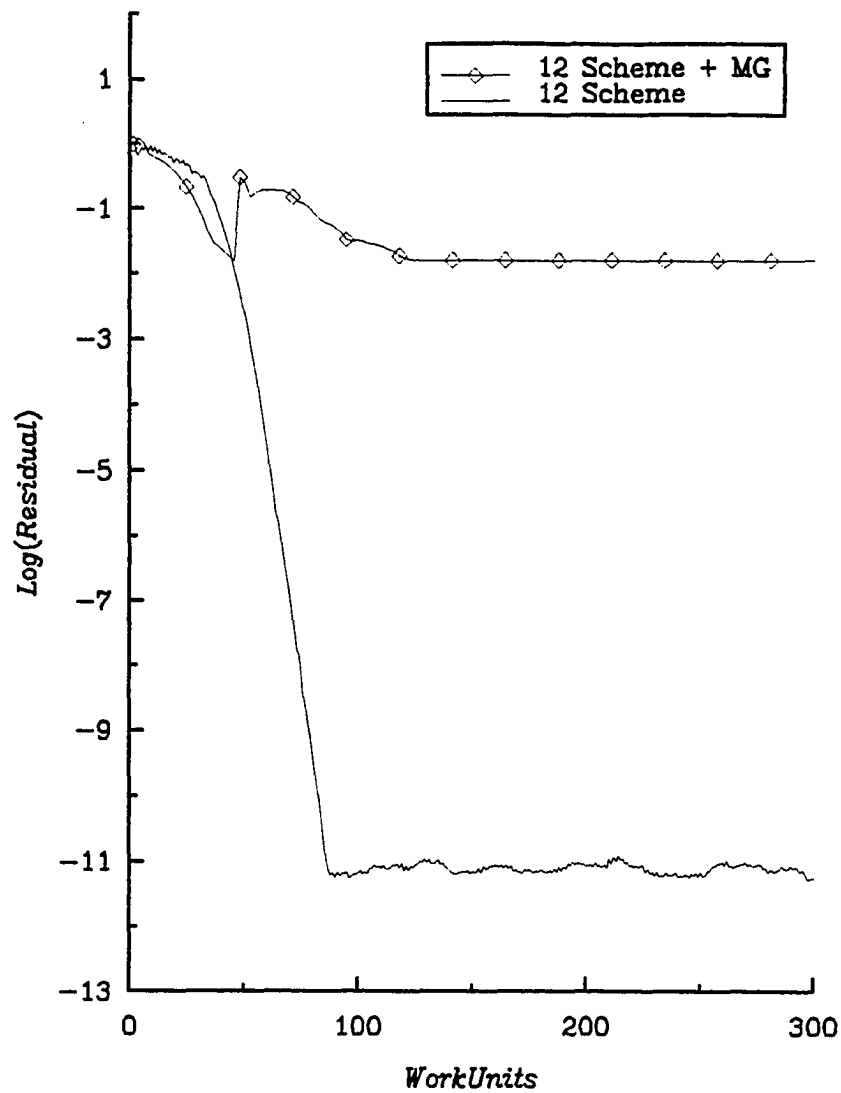


Figure 7.11 Comparison of Convergence Histories of 1-2 Scheme With and Without Multigrid Acceleration, Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)

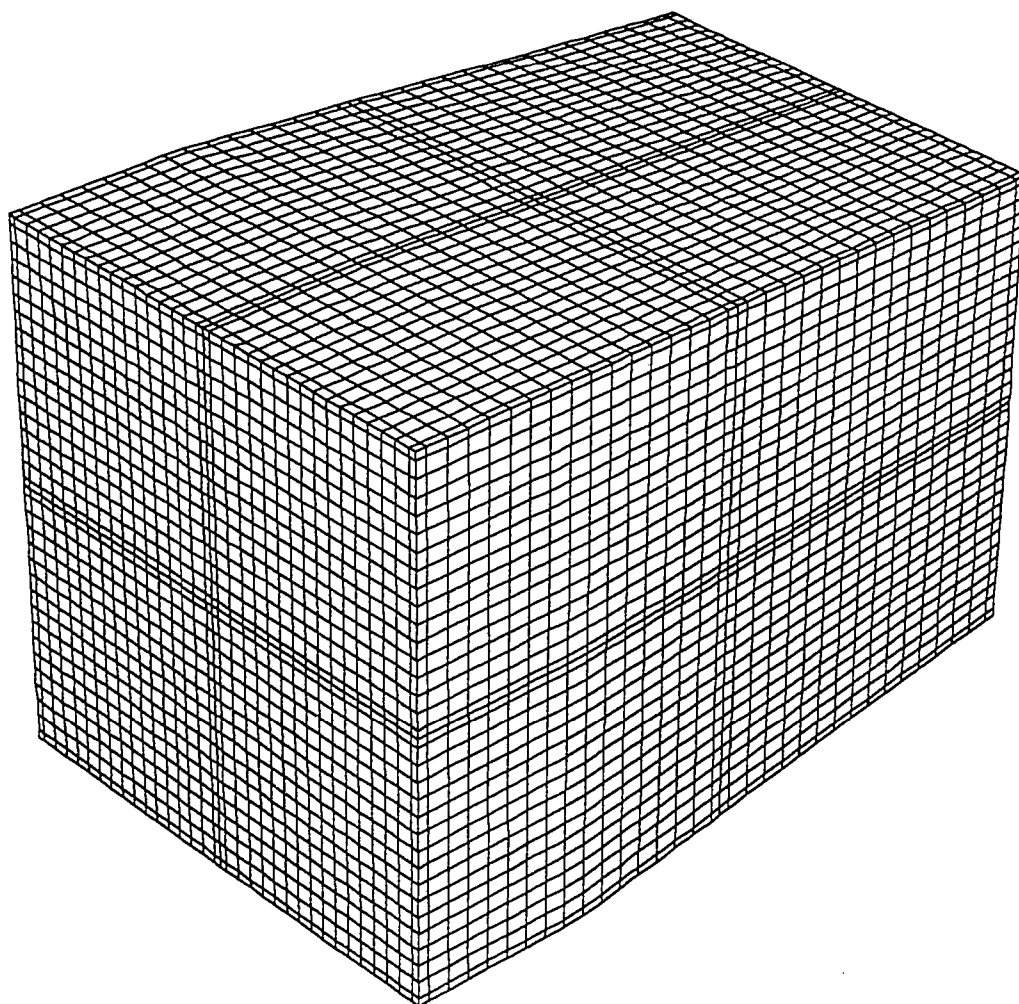


Figure 7.12 Schematic of Grid for Eight Block
Calculation for Flow through a 9.5° Compression Corner

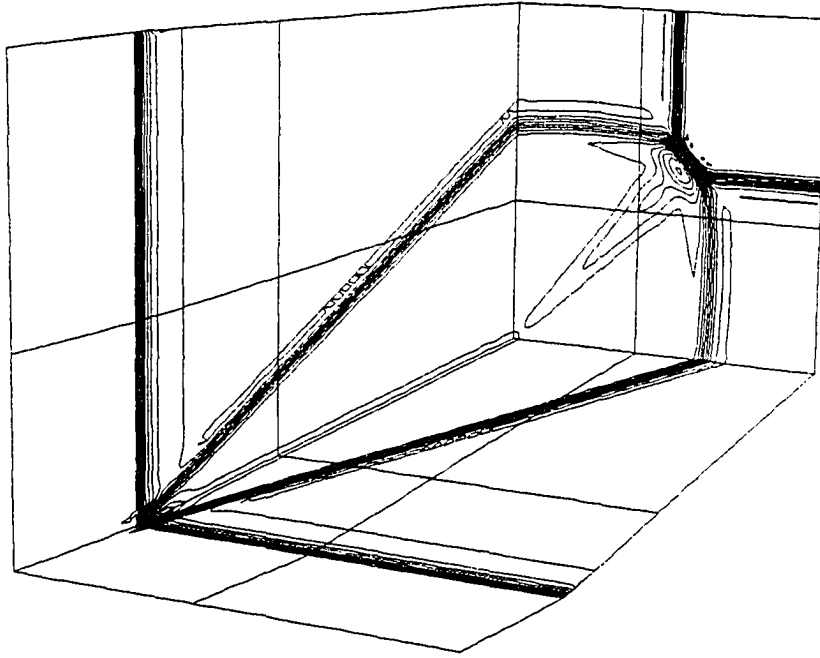


Figure 7.13 Mach Contours for Eight Block Calculations, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$.

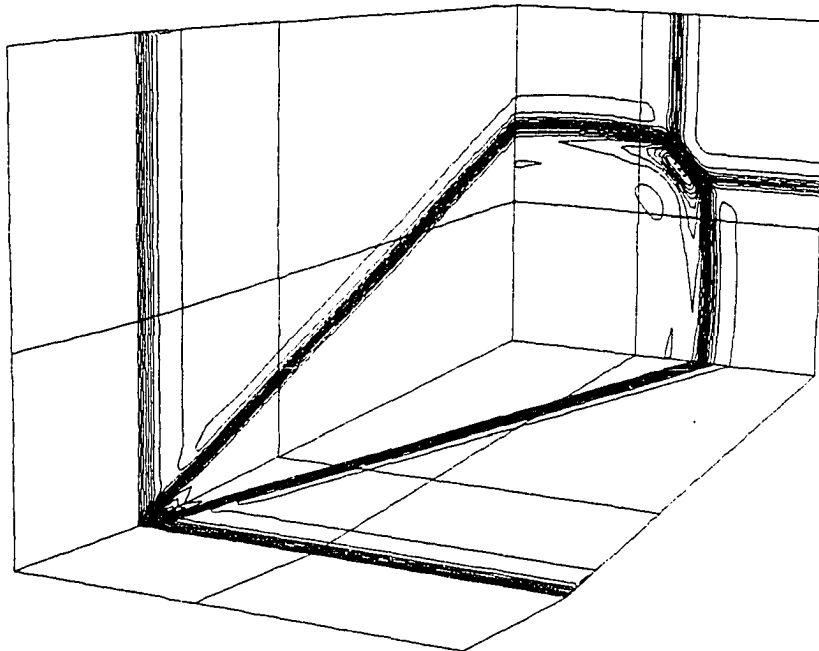


Figure 7.14 Pressure Contours for Eight Block Calculations, $M_{inlet} = 3.0$ and $\alpha = 9.5^\circ$.

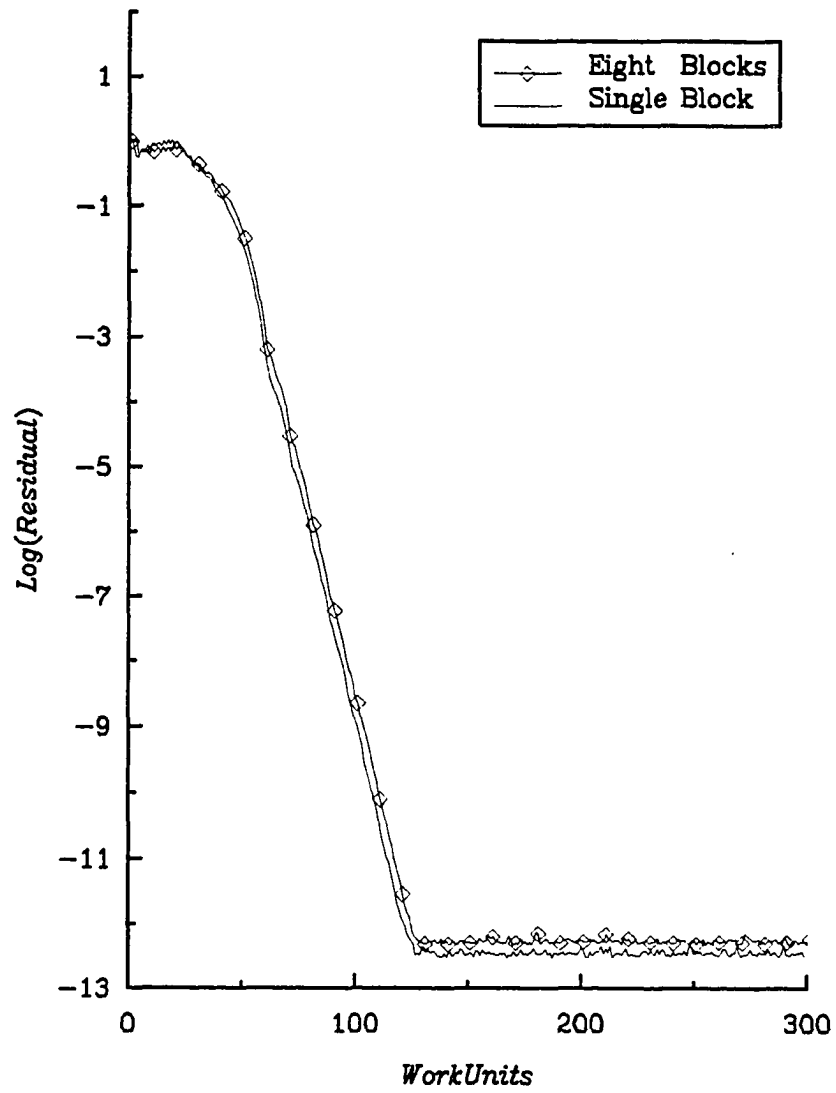


Figure 7.15 Comparison of Convergence Histories of Single Block and Eight Block Calculations of 9.5° Compression Corner Flow ($M_{inlet} = 3.0$)

7.3 Non-Axisymmetric Jet Exhaust Plume

Non-axisymmetric jet exhaust plume flows generally contain complex, multiple shocks and strong contact discontinuities or slip lines. The flow complexity is driven by both geometric complexities and Mach number, but is also influenced by static temperature, and static pressure ratios between the jet and the free-stream flow. The non-axisymmetric jet exhaust plume flow represents a challenging test case to the developed algorithm. The exhaust plume mixing with the surrounding flow will occur at transonic conditions, accompanied by large embedded supersonic flow regions and possibly complicated shock structures, shock interactions, slip lines and shear layers.

The main emphasis of this test case, besides understanding the flow physics, was to assess the performance of the Roe flux-differencing scheme and van Leer flux-vector splitting scheme, for the prediction of jet exhaust plume flows. A test case was also conducted to determine which type of extrapolation (conservative or primitive) should be utilized to evaluate the cell interface flux. This test case was used to verify the non-homogeneous boundary condition capability of the code as well.

To isolate the problems that Roe's flux-difference splitting scheme and van Leer's flux-vector splitting scheme have with shocks and contact discontinuities from problems with geometric complexities, a simple, pseudo-two-dimensional test case was considered. The test case was a flow from an infinitely wide nozzle of height 1.0. The jet Mach number was taken to be 1.5 and the free-stream Mach number was 2.5. The ratio of the jet static pressure to the free-stream static pressure (P_{jet}/P_{∞}) was 3.5 and the ratio of the jet static temperature to the free-stream static temperature (T_{jet}/T_{∞}) was 3.0.

Although three-dimensional calculations were performed using both Roe's flux-difference splitting and van Leer's flux-vector splitting schemes, they were compared with a two-dimensional, shock-fitting method [148]. The flow field is symmetric about

the x-plane and thus only half the flow field is computed numerically. Figure 7.16 is a schematic of the upper half of the flow field showing the shock wave, expansion fan, and slip line. A partial view of the grid used is shown in Fig. 7.17. When generating the grid, special attention was devoted to the alignment of the grid lines, as much as possible, with the shock wave and slip line.

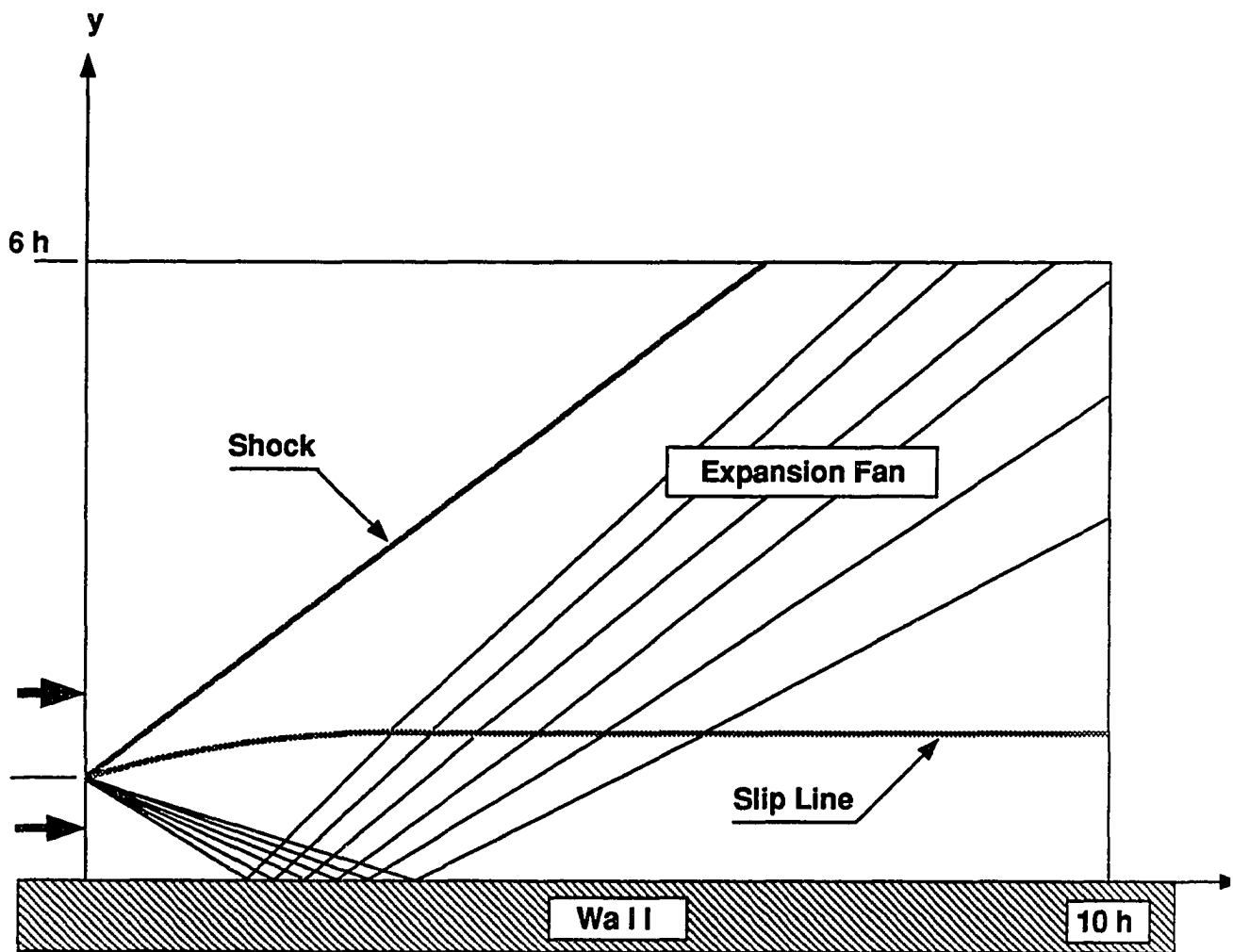


Figure 7.16 Schematic of the Computed Flow Field
for a Pseudo-Two-Dimensional Jet Exhaust Plume

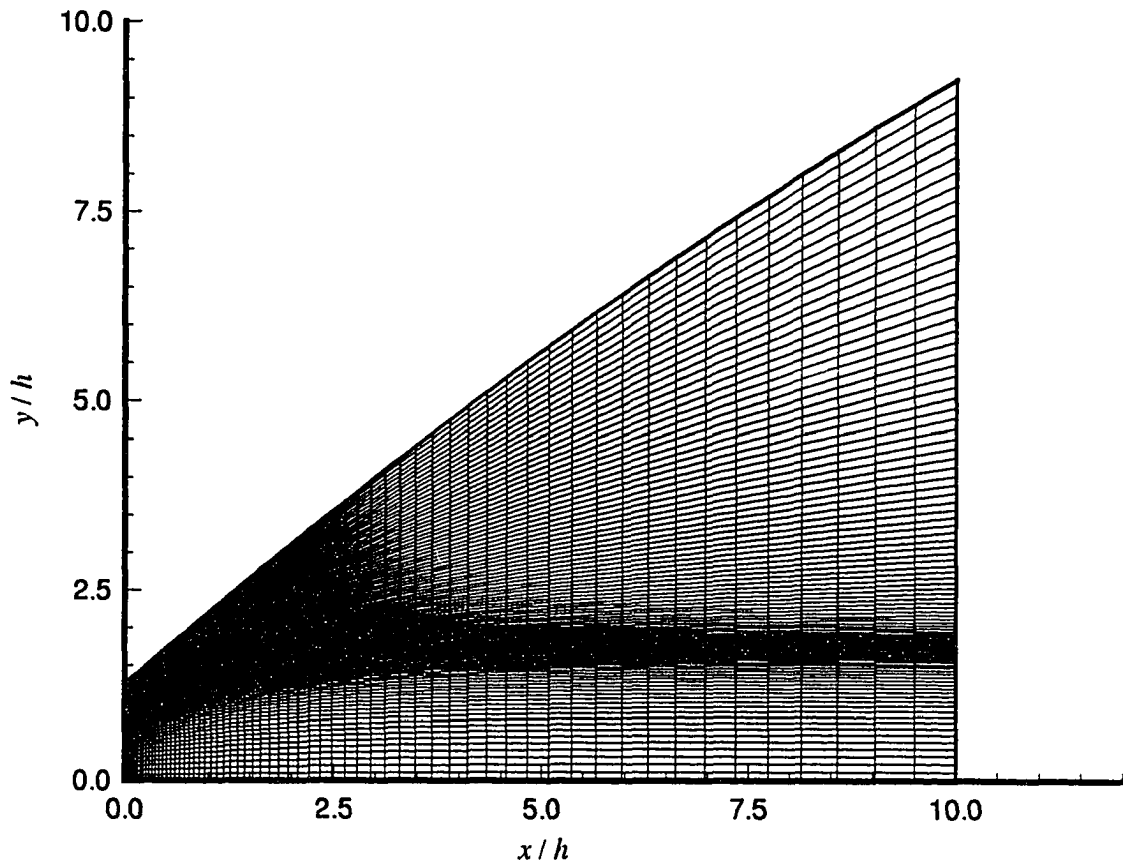
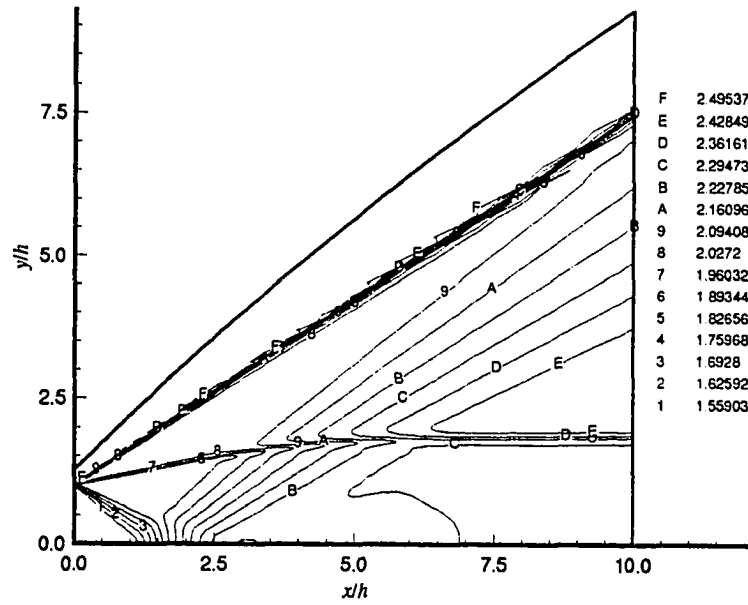


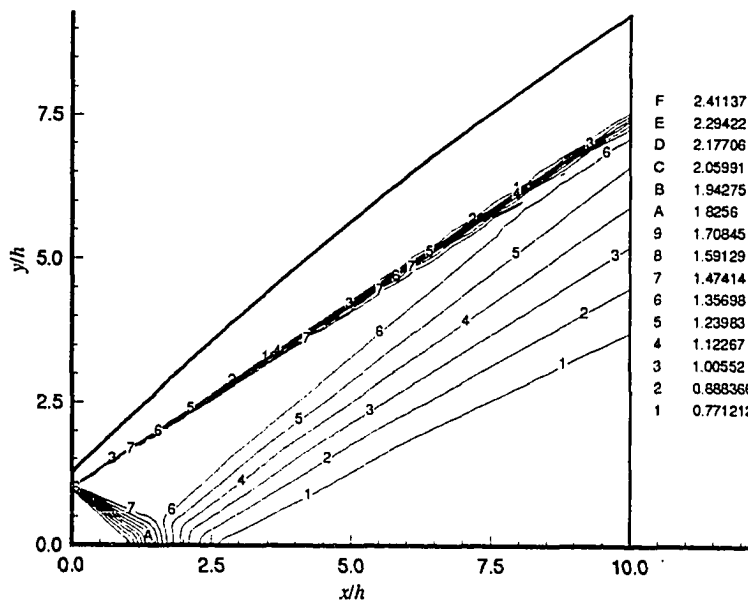
Figure 7.17 Partial View of the Two-Dimensional Jet Exhaust Plume Grid.

The Mach and pressure line contours for the Roe's flux-differencing splitting scheme are shown in Fig. 7.18, while the Mach and pressure line contours for the van Leer flux-vector splitting scheme are shown in Fig. 7.19. Both schemes employed the 1-2 explicit scheme. No flux limiters were used for the results displayed in Fig. 7.18 and Fig. 7.19.

The results obtained by the Roe flux-difference splitting and van Leer flux-vector splitting schemes are comparable. The main difference is that the van Leer flux-vector splitting scheme smears the slip line slightly. This behavior is expected since van Leers's scheme ignores the linear waves (entropy and shear layer), and does not have a mechanism to resolve the slip lines accurately [107], while Roe's flux-difference splitting is based on the idea of solving a Riemann problem at each cell interface.

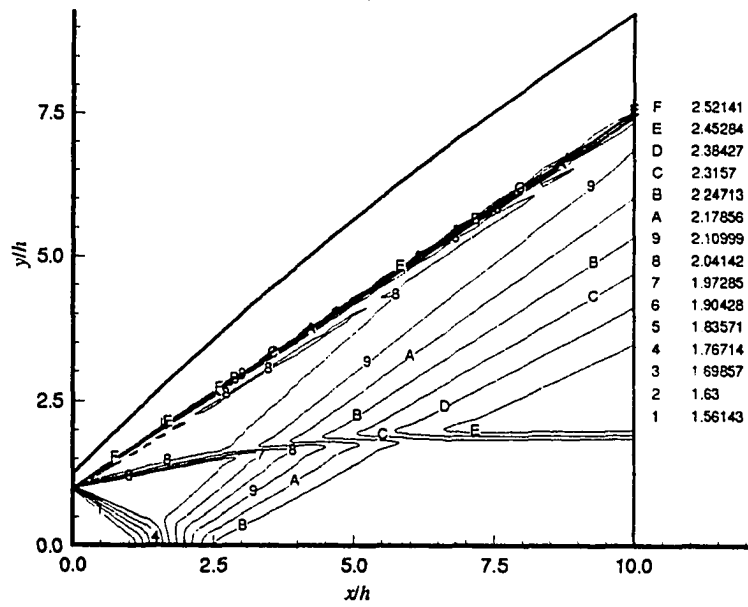


a. Mach Contours.

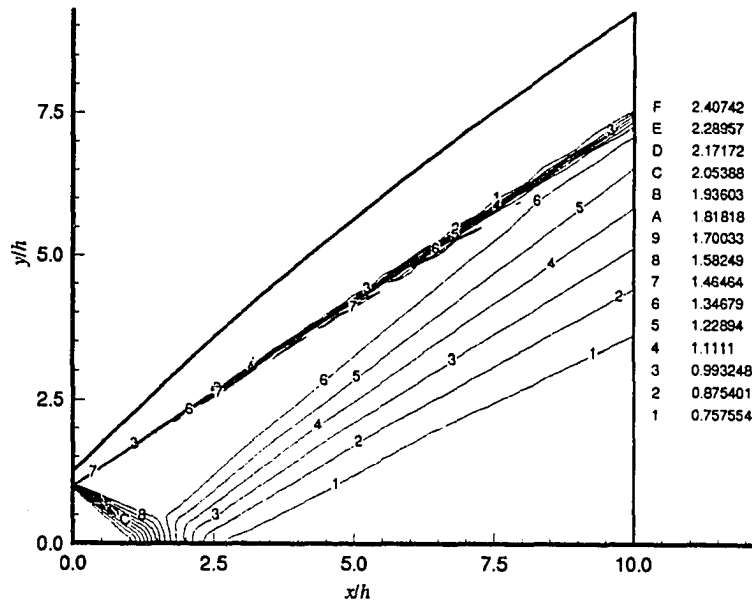


a. Pressure Contours.

Figure 7.18 Non-Axisymmetric Jet Exhaust Plume Calculations Utilizing Roe's Scheme, $M_\infty = 2.5$, $M_{jet} = 1.5$, $P_{jet} = 3.5$, and $T_{jet}/T_\infty = 3.0$.



a. Mach Contours.



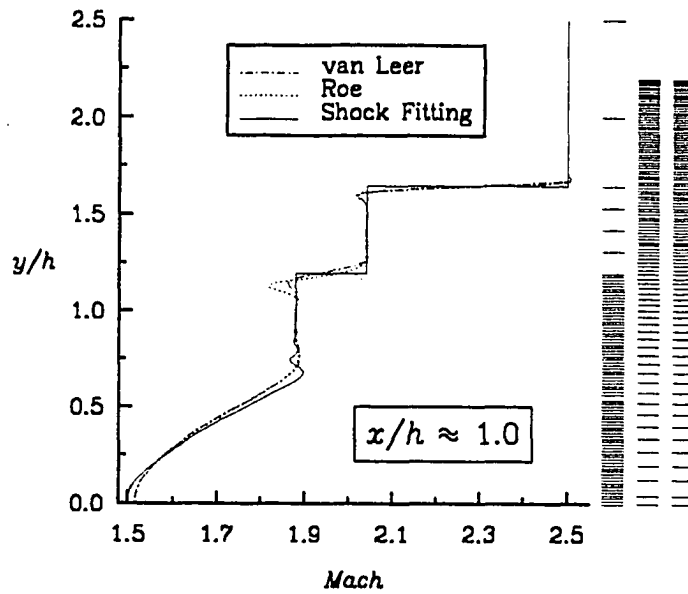
b. Pressure Contours.

Figure 7.19 Non-Axisymmetric Jet Exhaust Plume Calculations Utilizing van Leer's Scheme, $M_\infty = 2.5$, $M_{jet} = 1.5$, $P_{jet}/P_\infty = 3.5$, and $T_{jet}/T_\infty = 3.0$.

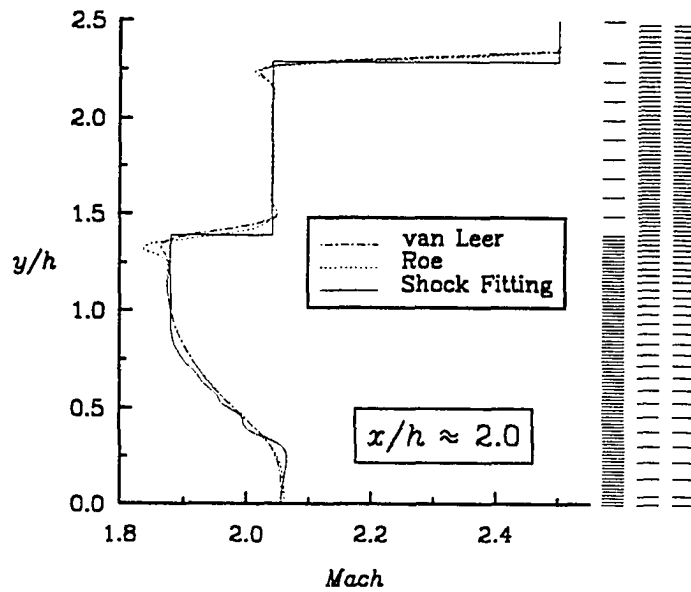
To allow detailed comparisons between the Roe, and van Leer schemes and the shock-fitting method of Salas [148], Mach number versus dimensionless height, y/h , was examined at the various downstream (x/h) locations. These plots are presented in Figs. 7.20a-f. The vertical grid spacing for each of the calculations is represented symbolically on the right of the plot. The grid spacing for the shock-fitting algorithm is shown as the left column. The shock-fitting algorithm generates its own grid, depending on the test case. The van Leer, flux-vector splitting scheme and the Roe flux-difference splitting scheme were computed using the same grid which was shown in Fig. 7.17. The spacing in the y -direction for both schemes is shown along the legend for consistency.

In Fig. 7.20a, the Mach number distribution at $x/h \approx 1.0$ is presented. At this location, the expansion fan is just reaching the lower wall. Along the expansion fan, a region of uniform flow is present, and a contact discontinuity is observed at $y/h \approx 1.2$. Notice that both the van Leer and Roe schemes smear this feature equally. This smearing is a result of the slip line cutting diagonally across the grid. After another region of uniform flow, shock occurs at $y/h \approx 1.65$. Smearing and overshoots occur with both the van Leer and Roe schemes due to the skewed grid relative to that shock wave.

At $x/h \approx 2.0$ (Fig. 7.20b), the expansion fan has reflected off the bottom wall and is moving out toward the slip line. Again, both the slip line and shock are smeared and predicted with overshoots by the two upwind schemes. At $x/h \approx 2.5$ (Fig. 7.20c), the expansion fan has moved further out toward the slip line. At $x/h \approx 3.0$ (Fig. 7.20d), the expansion fan is split by the slip line, as is apparent from the shock-fitting solution. This is not obvious in either the van Leer or Roe schemes, due to smearing and overshoots. At $x/h \approx 5.0$ (Fig. 7.20e), the slip line clearly splits the expansion fan as indicated by all three methods. Notice that the shock-fitting method predicts spurious oscillations near

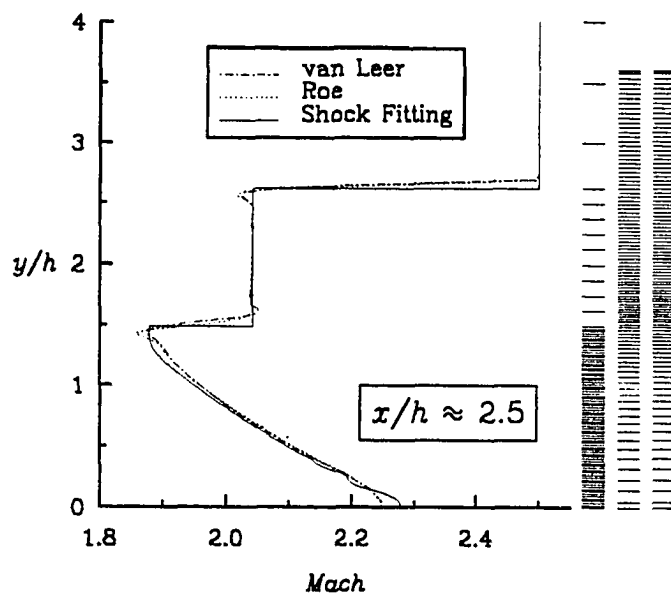


(a)

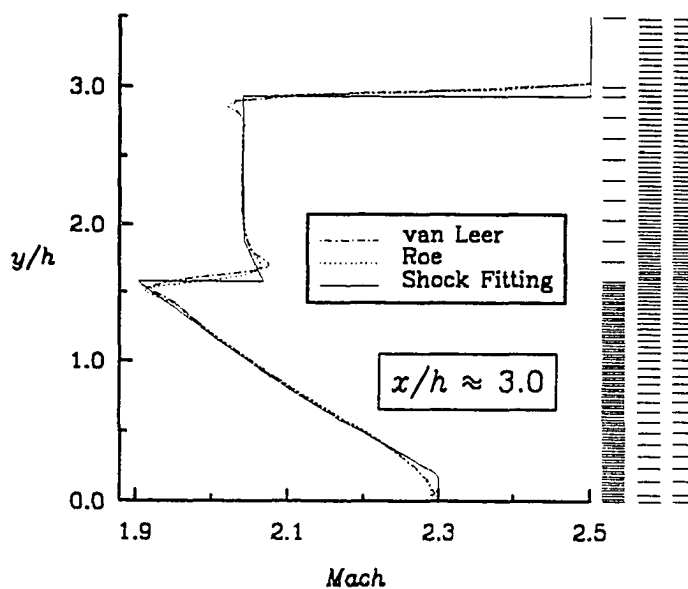


(b)

Figure 7.20 Comparison Between Roe's Flux-Difference Splitting Scheme, and van Leer's Flux-Vector Splitting Scheme and Salas' Shock-Fitting Method for a Pseudo-Two-Dimensional Exhaust Plume. (Continued . . .)



(c)



(d)

Figure 7.20 Comparison Between Roe's Flux-Difference Splitting Scheme, and van Leer's Flux-Vector Splitting Scheme and Salas' Shock-Fitting Method for a Pseudo-Two-Dimensional Exhaust Plume. (Continued . . .)

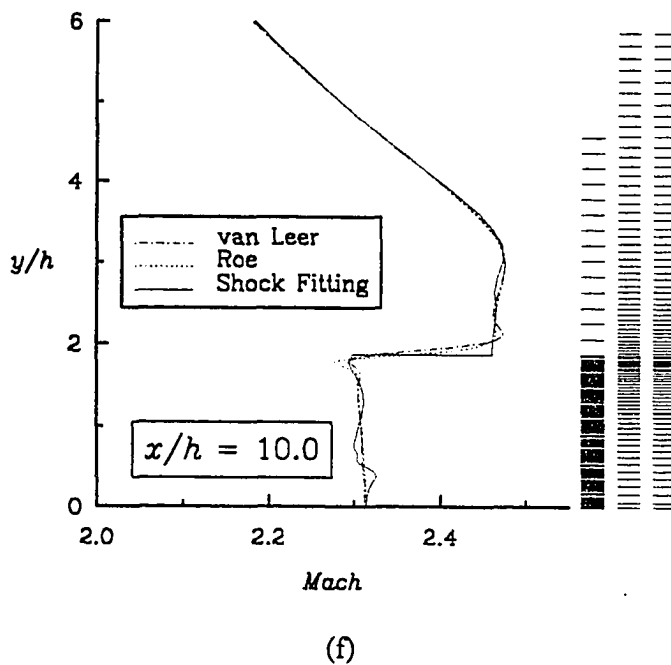
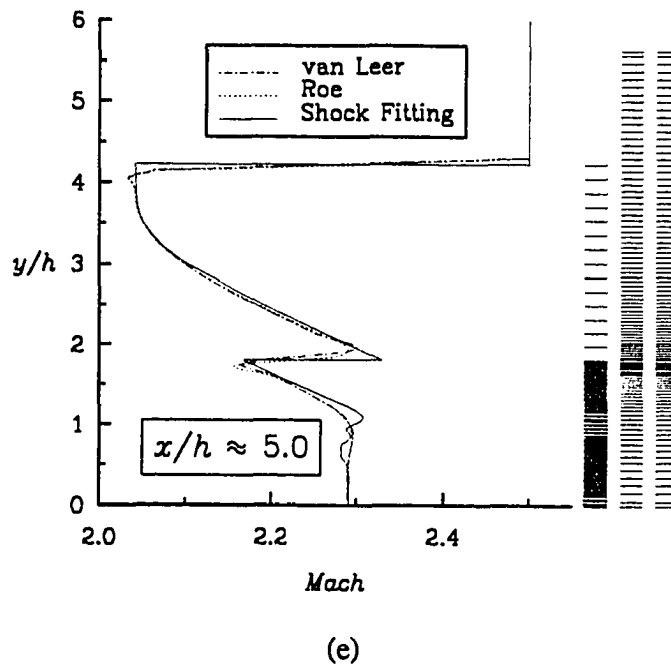
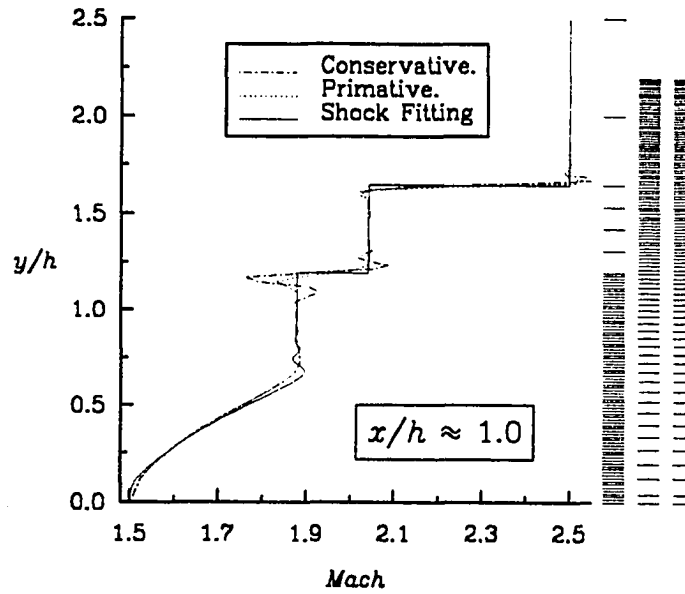


Figure 7.20 Comparison Between Roe's Flux-Difference Splitting Scheme, and van Leer's Flux-Vector Splitting Scheme and Salas' Shock-Fitting Method for a Pseudo-Two-Dimensional Exhaust Plume.

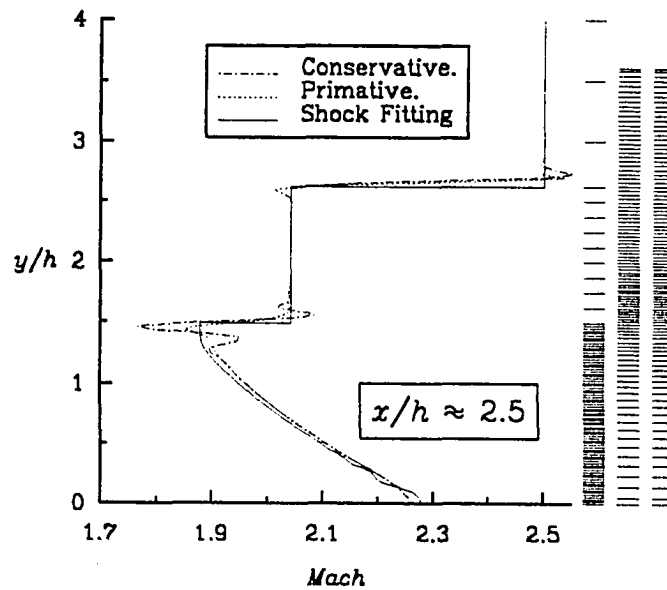
the lower wall. These wiggles are more pronounced at $x/h = 10$ (Fig. 7.20f). At this station, the shock has moved out of the computational domain.

Although overshoots and smearing are present, several important points should be noted from Fig. 7.20. Both the van Leer and Roe schemes do a reasonable job of predicting the flow and are not affected by the overshoots and smearing adversely. It was anticipated that the Roe scheme would do a better job of predicting the contact discontinuity than the van Leer scheme. This is not apparent in the present results and should be attributed to the 'poor' predictions by both methods, due to the misalignment of the grid. Obviously, a better grid can be generated but the current grid is more indicative of the type of grid that would be used for more complex geometries and flow physics. For these complex situations, proper alignment of the grid would be impossible without an adaptive grid scheme.

Evaluation of the cell interface flux requires the extrapolation of state variables to the cell interface. The developed algorithm offers the capability of extrapolating either the conservative or primitive variables. To examine the effect of utilizing either of the extrapolation methodologies, the four-stage $\kappa = -1$, Roe's flux-difference splitting scheme, with primitive and conservative extrapolation, was employed to recompute the two-dimensional plume flow. Local time-stepping, implicit residual smoothing and the multigrid acceleration technique were implemented to accelerate the rate of convergence to steady-state. Comparison between the computed Mach number distribution with the shock-fitting method of Salas [148], at various x/h -locations have been made. These plots are presented in Figs. 7.21a-f. The results show clearly that the primitive variable extrapolation renders a smoother solution across slip lines and shocks, for all x/h -locations. It should be mentioned that the results computed by the predictor-corrector explicit scheme and the four-stage $\kappa = -1$ explicit scheme were identical.

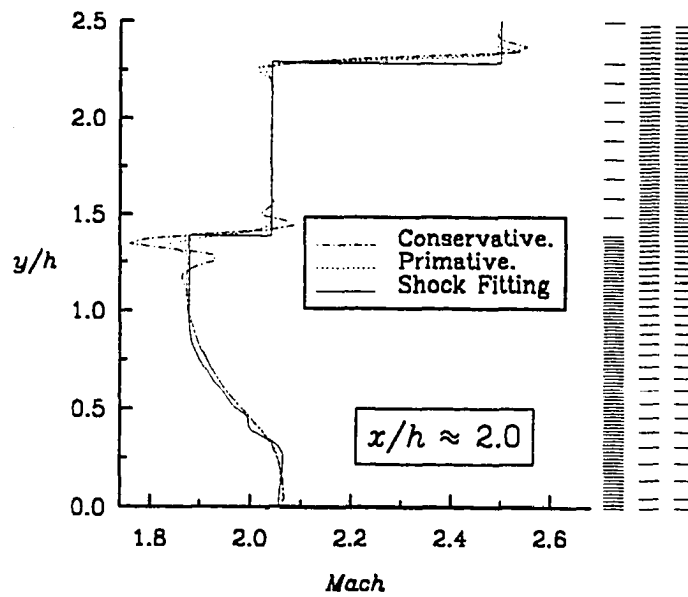


(a)

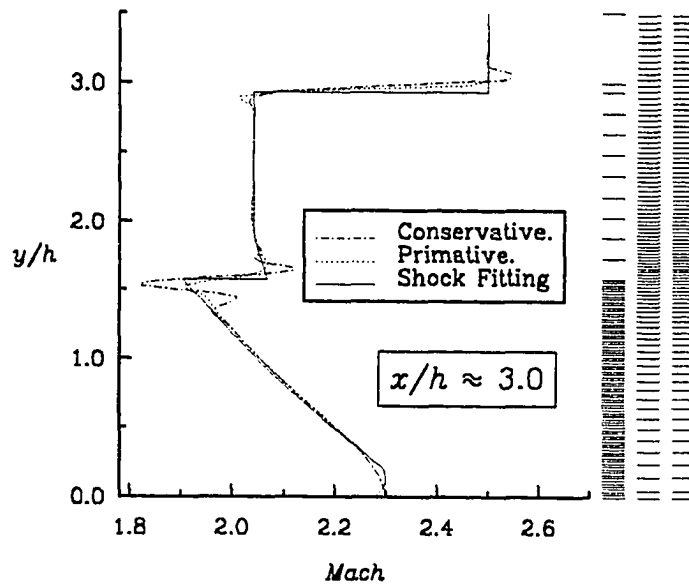


(b)

Figure 7.21 Comparisons Between Primitive and Conservative Extrapolations of the Roe's Scheme with the Shock-Fitting Code of Salas for a Pseudo-Two-Dimensional Exhaust Plume. (Continued . . .)

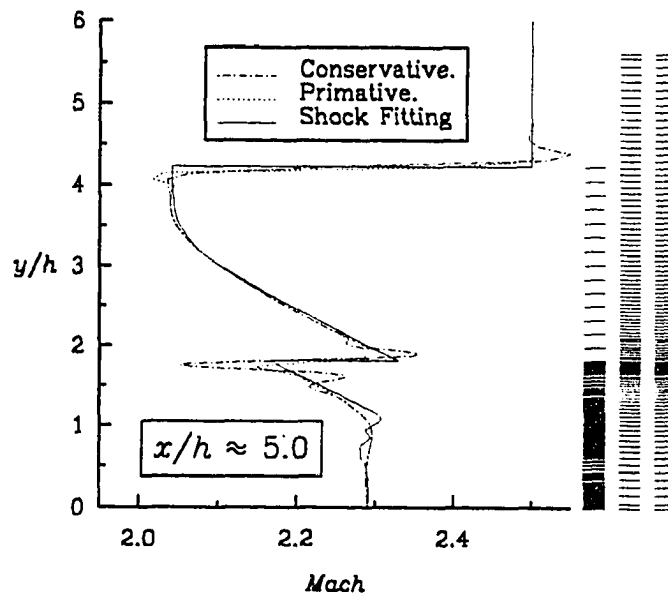


(c)

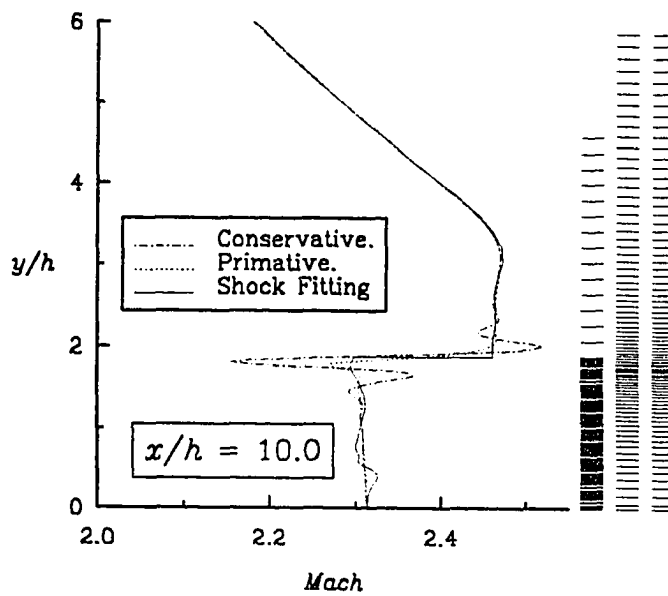


(d)

Figure 7.21 Comparisons Between Primitive and Conservative Extrapolations of the Roe's Scheme with the Shock-Fitting Code of Salas for a Pseudo-Two-Dimensional Exhaust Plume. (Continued . . .)



(e)



(f)

Figure 7.21 Comparisons Between Primitive and Conservative Extrapolations of the Roe's Scheme with the Shock-Fitting Code of Salas for a Pseudo-Two-Dimensional Exhaust Plume.

To examine the effect of using a higher order method, the Roe scheme was recomputed as first, second ($\kappa = -1$, and $\kappa = 0$), and third-order accurate ($\kappa = 1/3$). Comparison between the computed results and the shock-fitting method of Salas [18], for the Mach number versus y/h at $x/h \approx 1.0$ is presented in Fig. 7.22. The first-order Roe scheme is highly dissipative and gives essentially useless results. The second and third-order results are more accurate and are nearly identical. The second-order ($\kappa = -1$) generates a larger undershoot at the slip line, but it produces the least amount of overshoot at the shock. The third-order accurate ($\kappa = 1/3$) method generated oscillations and produced the largest overshoot at the shock. For detailed comparison of the different order of extrapolations at all other locations, the interested reader is referred to reference [54]. Comparison of the convergence history for all four types of extrapolations are shown in Fig. 7.23

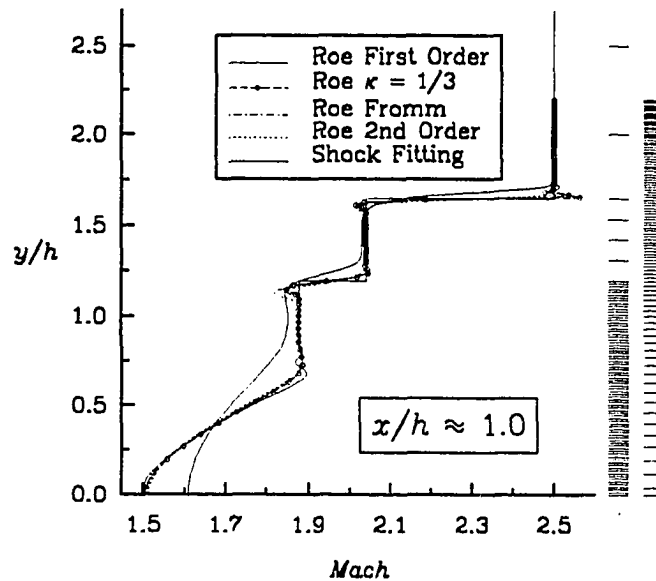


Figure 7.22 Comparisons Between Different Extrapolations of Roe's Scheme and a Shock-Fitting Code for a Pseudo-Two-Dimensional Exhaust Plume.

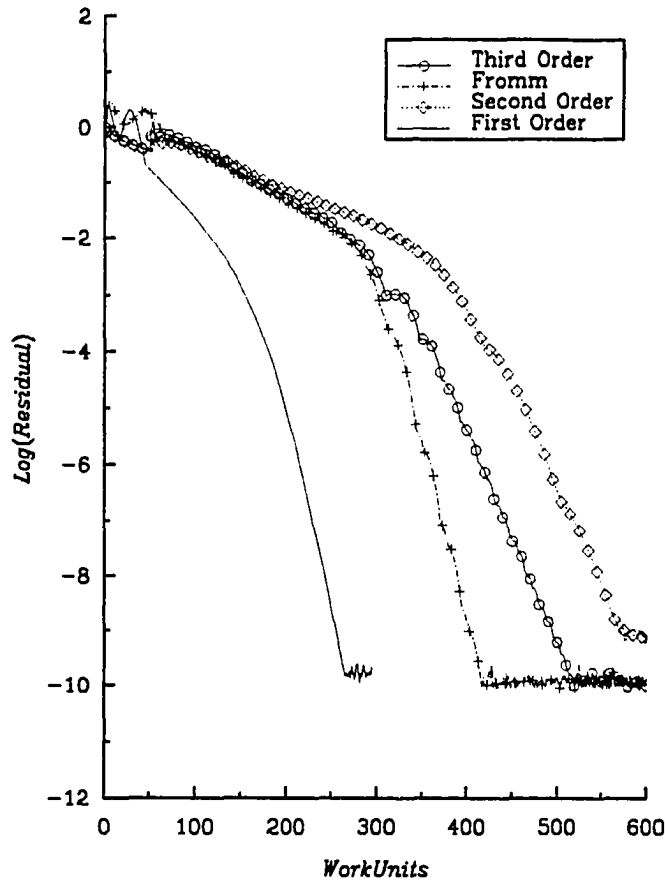


Figure 7.23 Comparisons of Convergence History
for Pseudo-Two-Dimensional Exhaust Plume.

To examine the flexibility of the multi-block structure, and the non-homogeneous boundary conditions, the flow was recomputed on a two-block structured grid. The computation domain was divided into two blocks: one block had a homogenous inflow boundary condition, which is the jet exhaust, $M_{jet} = 1.5$, and a second block which had a homogenous free stream inflow, $M_{\infty} = 2.0$, (refer to Fig. 7.16). The two block results were identical to the single block results. No modifications to the code were required to go from the single-block plume calculation, with non-homogeneous boundary conditions at the inflow, to the two-block calculation. Only the input to the program was changed to accommodate both runs.

7.4 Flat Plate

The main goal of computing the laminar, and turbulent flow over a flat plate was to verify the viscous capability of the developed algorithm, and to check the correctness of the Baldwin-Lomax algebraic-turbulence model implementation. For the laminar case the free stream Mach number (M_∞) was 0.5, and the free stream Reynolds number (Re_∞) was 1000 per unit length of the flat plate. The grid used for this test case was $65 \times 65 \times 5$ (streamwise, normal, spanwise). The normalized minimum spacing in the normal direction to the wall was, $\Delta y_1 = 1 \times 10^{-4}$. The grid was fine enough to produce at least ten vertical grid points through the boundary-layer thicknesses at all locations on the plate. Although the flat plate problem was a two-dimensional problem, five spanwise planes were employed to test the multigrid acceleration technique, and to check the viscous terms in all three directions. The computed laminar velocity profile, and variation of the local skin friction coefficient along the plate are shown in Fig. 7.24. Comparison between the computed results and the classical Blasius boundary-layer solution, [93], shows excellent agreement.

A grid refinement study was performed to determine the minimum number of points required to accurately resolve the laminar boundary layer. Three grids were employed in the grid refinement study; the first grid was a $33 \times 49 \times 5$ with a $\Delta y_1 = 0.015$, the second grid was $33 \times 65 \times 5$ with a $\Delta y_1 = 0.0075$, and the third grid was $33 \times 81 \times 5$ with a $\Delta y_1 = 0.00375$. The grid refinement was carried out by dividing the grid spacing in the normal direction only. Comparisons between the computed velocity profiles, and Blasius boundary-layer solution are shown in Fig. 7.25. The developed algorithm has resolved the laminar boundary layer successfully, even on the coarsest grid, where there were only five points in the boundary layer. Fig. 7.26 shows a comparison between the computed skin friction and Blasius boundary-layer solution, for all three grids.

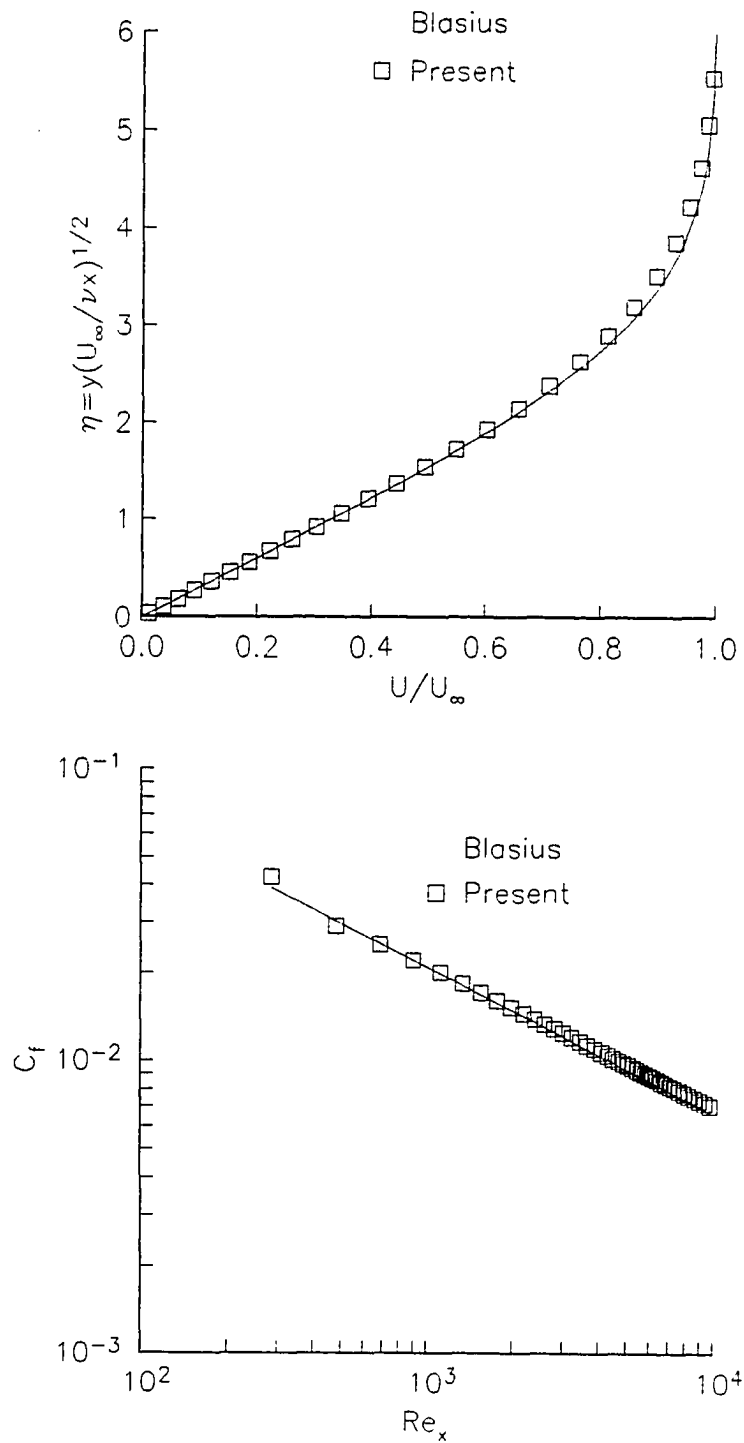


Figure 7.24 Results for the Laminar Flow Over a Flat Plate. $M_\infty = 0.5$, $Re_\infty = 1000$ per Unit Length.

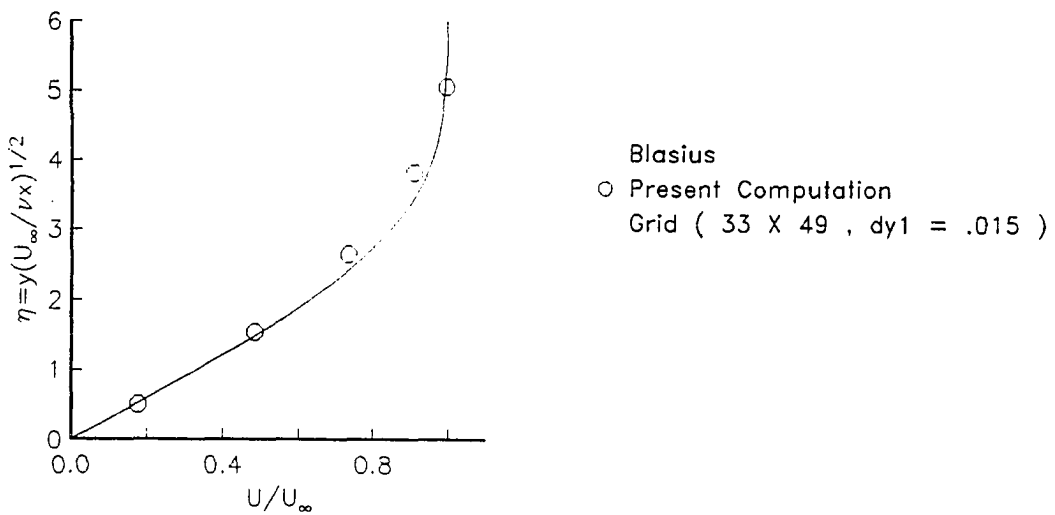
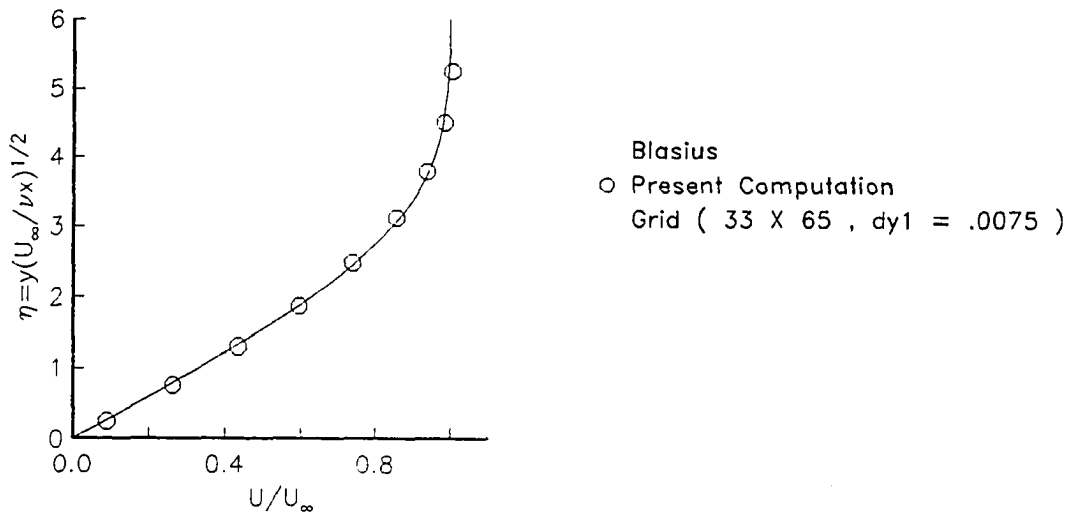
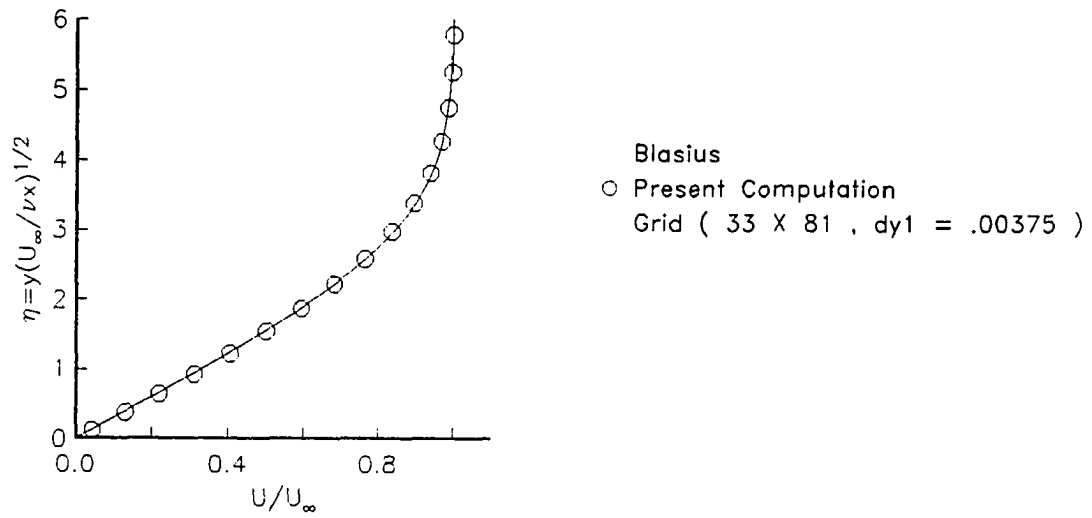


Figure 7.25 Velocity Distribution Over a Flat Plate.

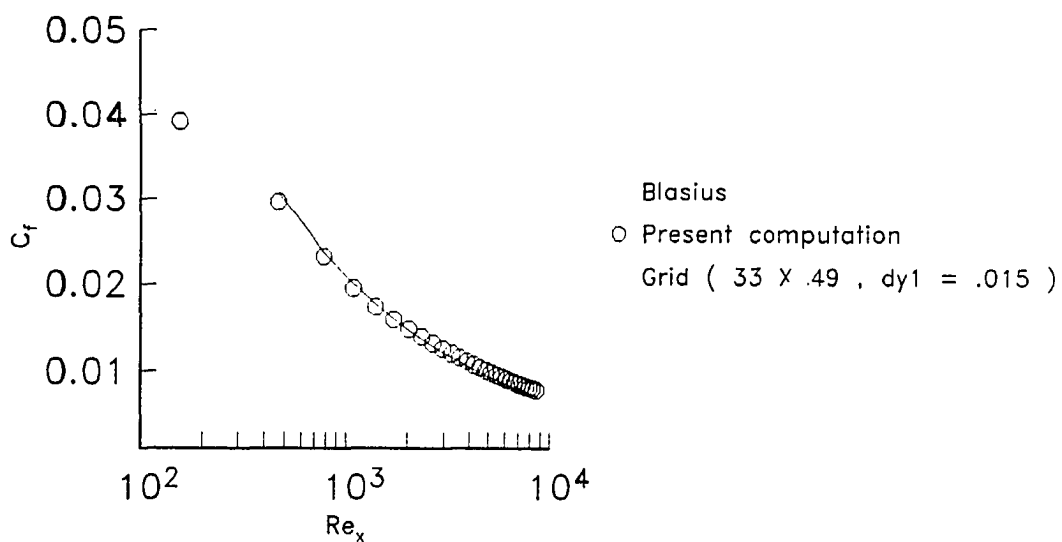
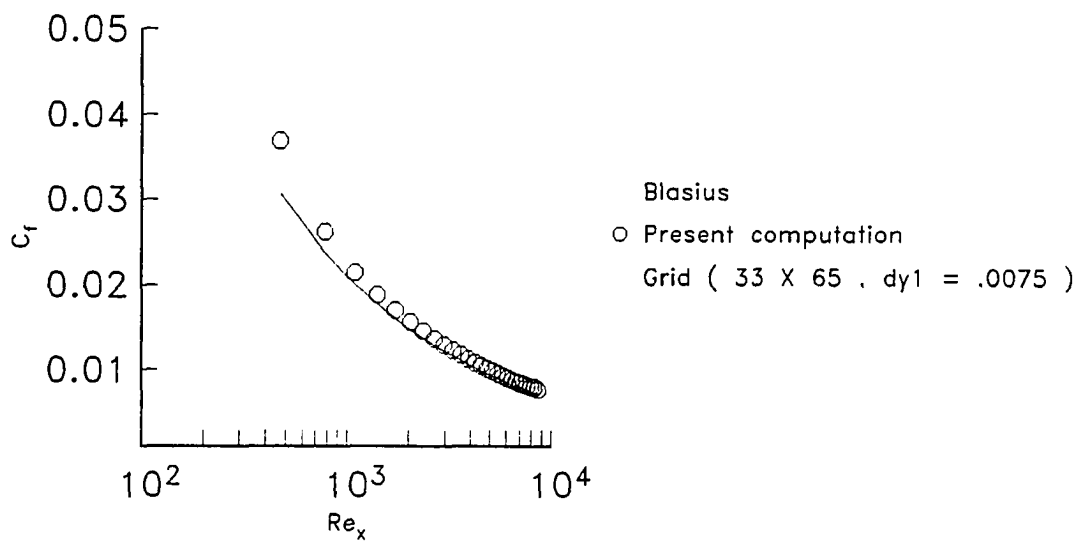
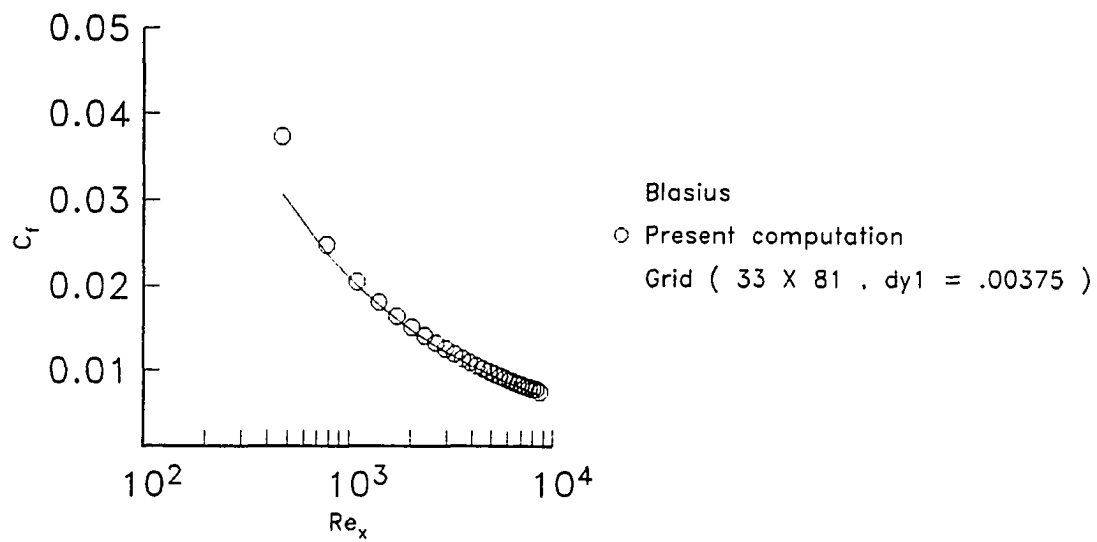


Figure 7.26 Local Skin Friction Coefficient Along a Flat Plate.

The flow over the flat plate was recomputed to evaluate the effect of the implicit residual smoothing, and the multigrid acceleration techniques on the rate of convergence to steady-state. The recomputed case employed the $33 \times 65 \times 5$ grid where, $\Delta y_1 = 0.0075$. Four sets of computations were performed. In all four sets the four-stage $\kappa = 0$ explicit time-stepping scheme was used to drive the solution to steady-state. In the first set, the basic explicit algorithm was augmented with local time-stepping. In the second set of computations, the basic explicit scheme was coupled with local time-stepping and implicit, residual-smoothing. In the third set of calculations the basic explicit scheme used local time-stepping and the full multigrid acceleration technique. In the last set of calculations, the performance of the basic explicit scheme was boosted by utilizing local time-stepping, implicit residual smoothing and the full multigrid acceleration technique. Comparison of the convergence history for all four test cases are shown in Fig. 7.27. As shown in the figure the acceleration techniques do enhance the rate of convergence to steady-state. The figure demonstrates clearly that the implicit-residual smoothing is beneficial in reducing the total number of work units required for convergence.

It should be noted, that the high frequency oscillations present on the fine mesh are slowly damped for highly skewed cells. However it was found that the convergence of the global flow field is improved significantly by using multigrid techniques, especially for complex three-dimensional grids.

For turbulent flow over a flat plate, the free stream Mach number (M_∞) was kept at 0.5, and the free stream Reynolds number (Re_∞) was set equal to 1.0×10^6 per unit length of the flat plate. The grid used in this case was $65 \times 65 \times 5$, with a $\Delta y_1 = 1.0 \times 10^{-5}$. Computations were performed using the four-stage, $\kappa = 0$, (second-order, upwind-biased) explicit, time-stepping scheme. Based on the previous experience gained when solving the laminar flat plate, local time-stepping, implicit-residual smoothing, and the full multigrid

acceleration technique were employed to enhance the rate of convergence to steady-state. Results for the turbulent flow over a flat plate are presented in Fig. 7.28. The computed results for the coefficient of friction agree well with the empirical formula $C_f = 0.0592 Re_x^{-1/2}$, [93], especially once fully developed turbulent flow is reached. Comparison between the computed velocity profile and the 1/7 power law, shows fairly good agreement. The comparison between the computed velocity profile with the law of the wall also shows good agreement.

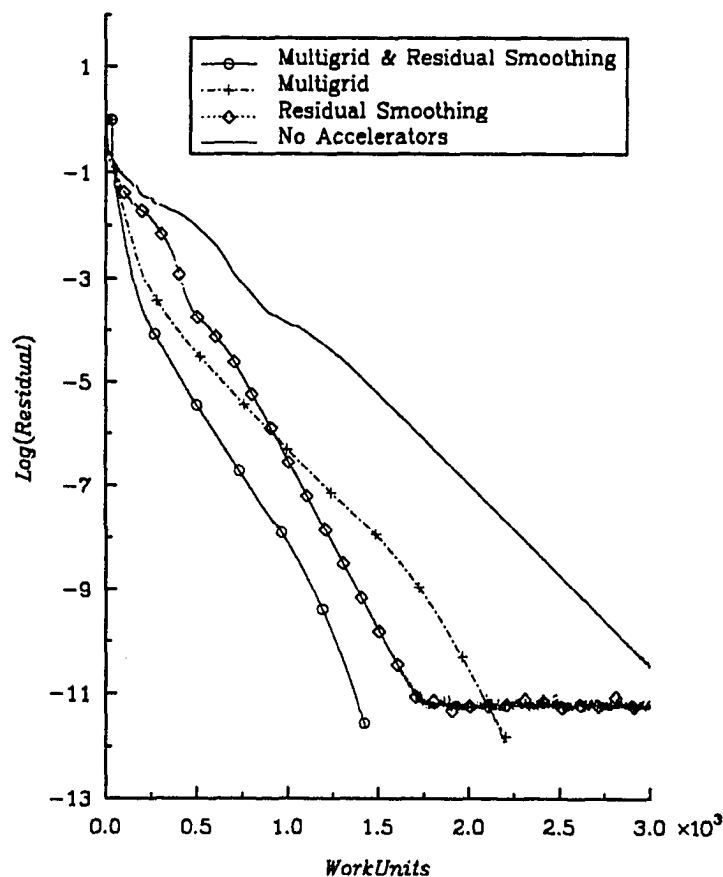


Figure 7.27 Convergence History For Flow Over a Flat Plate.

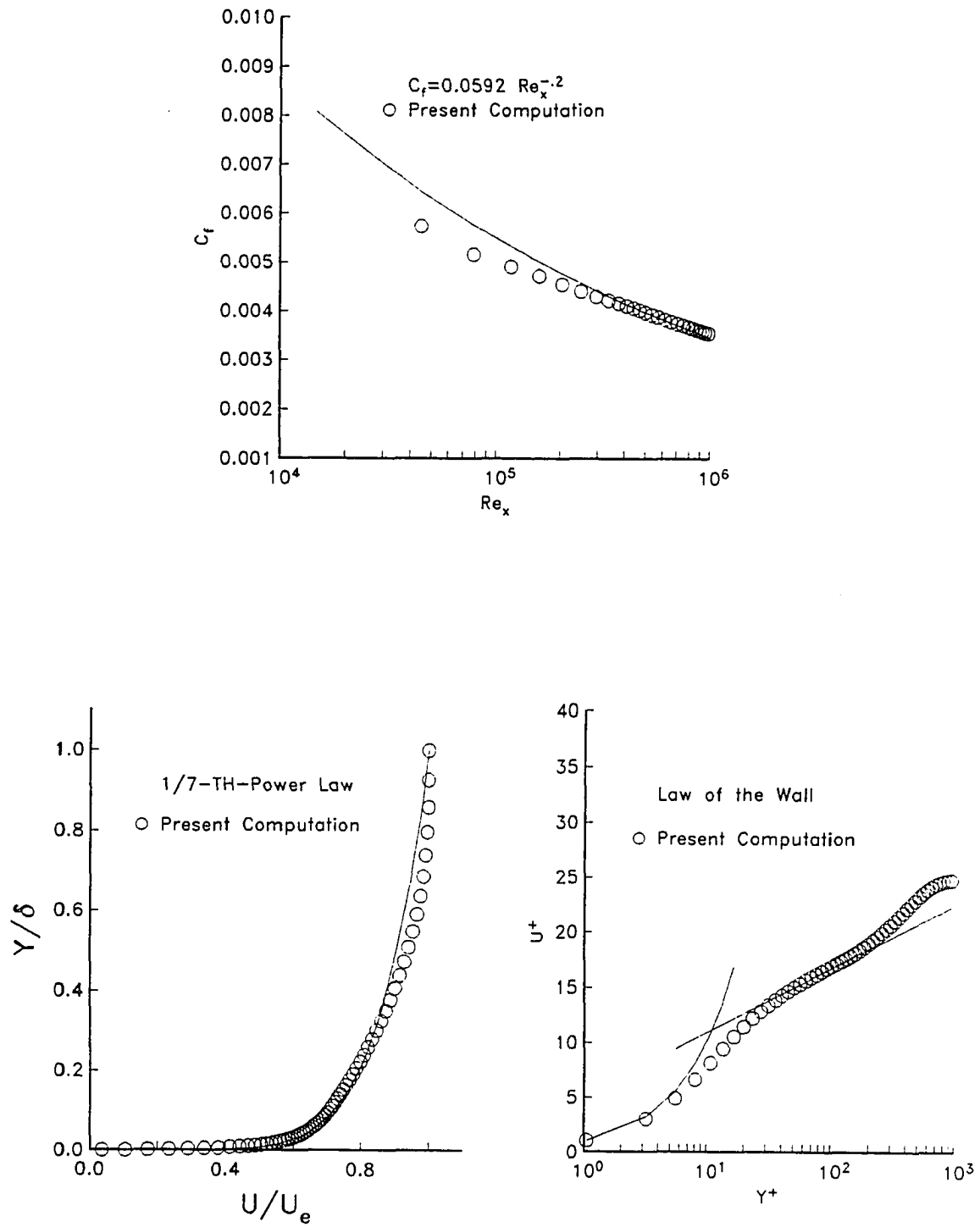


Figure 7.28 Turbulent Flow Over a Flat Plate,
 ($M_\infty = 0.5$, $Re_\infty = 1.0 \times 10^6$ per Unit Length).

7.5 ONERA M6 Wing

The ONERA M6 wing was selected to validate the capability of the developed algorithm to compute a truly three-dimensional turbulent-flow, and to assess the performance of the algorithm with other three-dimensional, state-of-the-art computer codes.

In the present study, a $193 \times 49 \times 33$ (streamwise, normal, spanwise) grid was employed. The grid employed a C-O mesh topology; C in the streamwise direction and O in the spanwise direction. The grid was generated by Bruce Wedan [149], and is shown in Fig. 7.29. Three test cases were investigated; one was a subcritical flow and two cases were supercritical flows. On all three test cases, a four-stage $\kappa = -1$ explicit time-stepping scheme is used to compute the flow field around the ONERA M6 wing. Local time-stepping, implicit residual-smoothing, and full multigrid procedures were added to the explicit time-stepping scheme to accelerate convergence to steady-state. Roe's flux-differencing scheme was employed to compute the inviscid flux, and a second-order, central-difference operator was used to evaluate the viscous and diffusion terms. The second-order upwind scheme was selected for these computations, to avoid the use of a limiter when computing the supercritical test cases. Limiters halt convergence and lead to limit cycle solution oscillation with no apparent convergence.

The first test case was a subcritical case, where the free stream Mach number (M_∞) was 0.699, the angle of attack (α) is 3.06° , and the chord-based Reynolds number (Re_∞) was 11.7×10^6 . The Reynolds number was based on free stream flow conditions and the mean aerodynamic chord. Pressure distributions along several spanwise locations of the ONERA M6 wing are shown in Figs. 7.30a-f. The computed results show good agreement with the experimental data, [134], as shown by the C_p plots in Figs. 7.30a-f. In Fig. 7.30, η represent the locations where the pressure distribution is displayed and is equal to $\frac{2Y}{B}$, where Y is the spanwise distance and is normalized with respect to the semispan $B/2$.

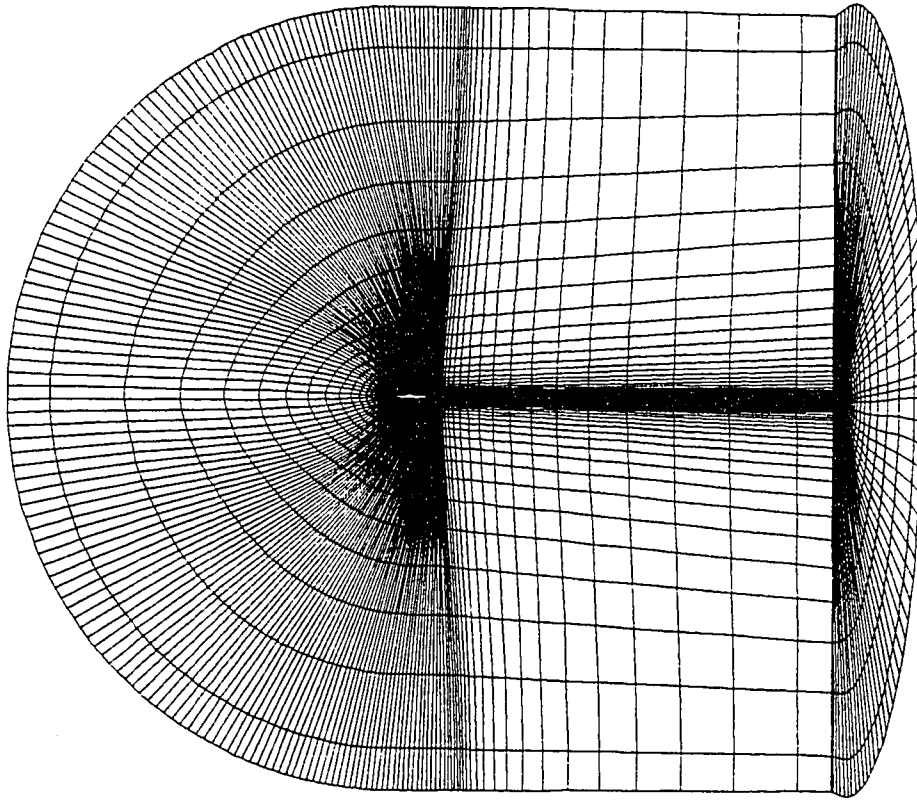


Figure 7.29 Partial View of C-O Grid Topology for ONERA M6 Wing.

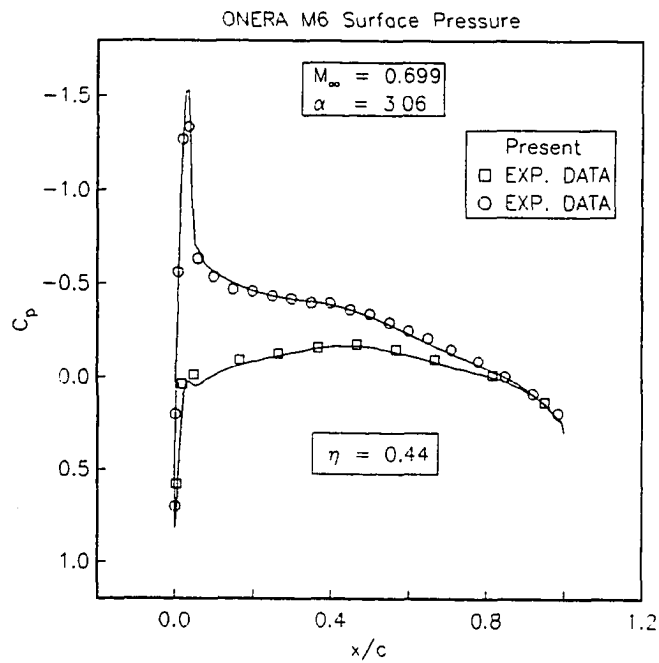
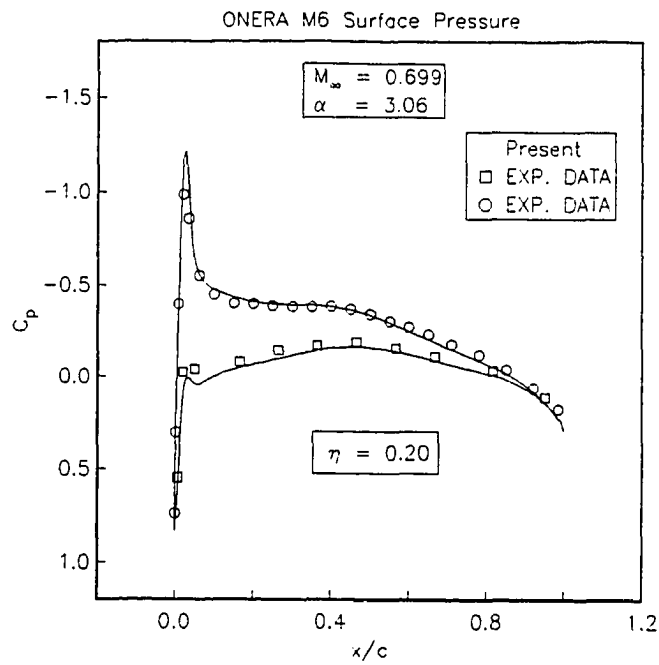
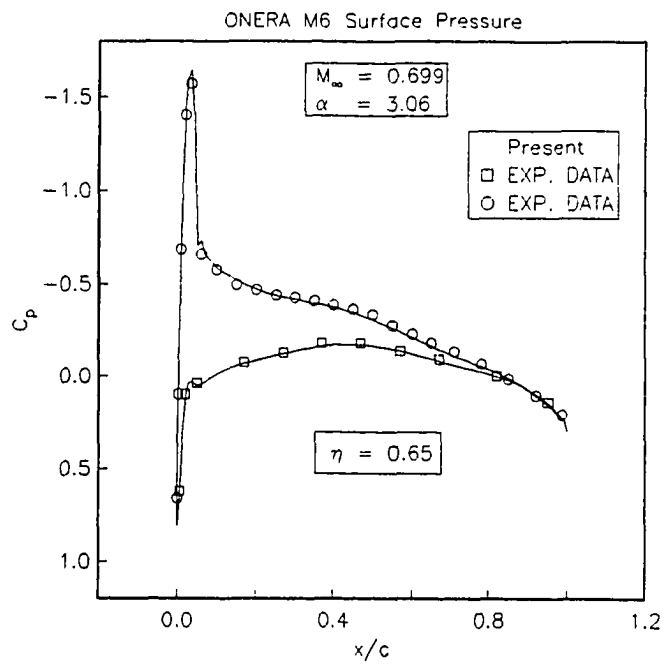
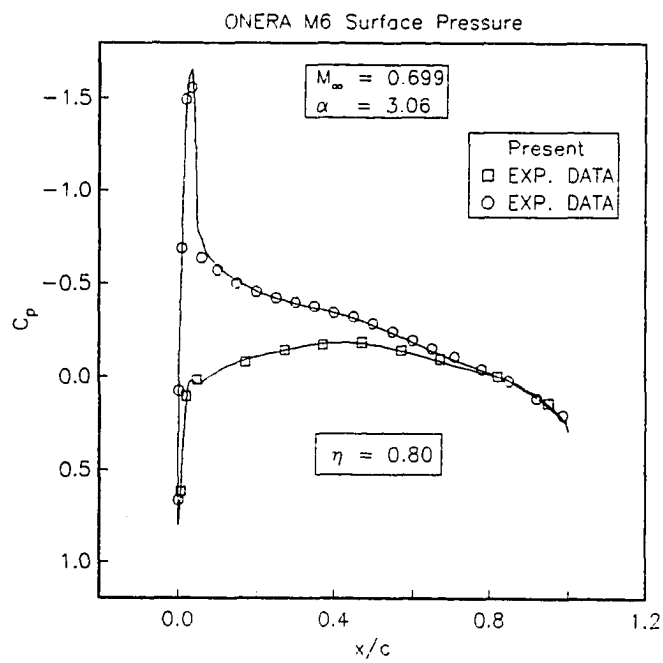


Figure 7.30 Pressure Distribution for ONERA M6 Wing

$M_{\infty} = 0.669$, $\alpha = 3.06^\circ$, and $Re_{\infty} = 11.7 \times 10^6$ (Continued . . .)



(c)



(d)

Figure 7.30 Pressure Distribution for ONERA M6 Wing

$M_\infty = 0.669$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$ (Continued . . .)

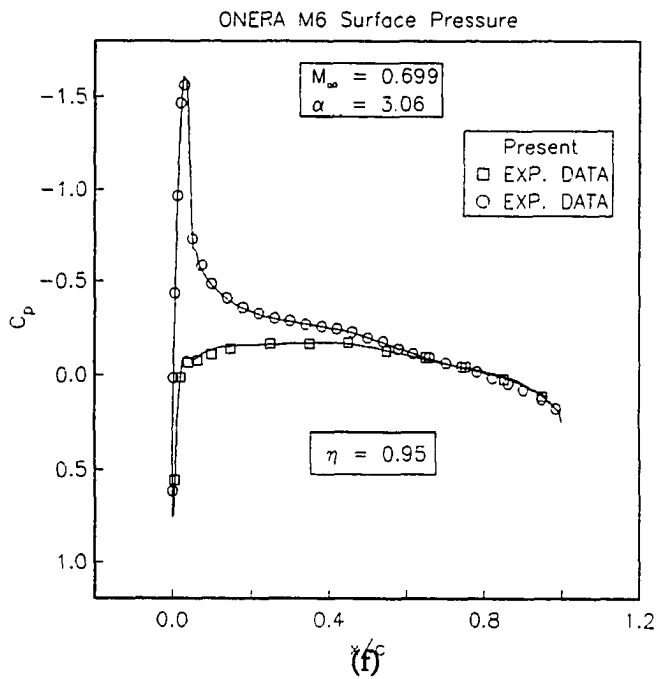
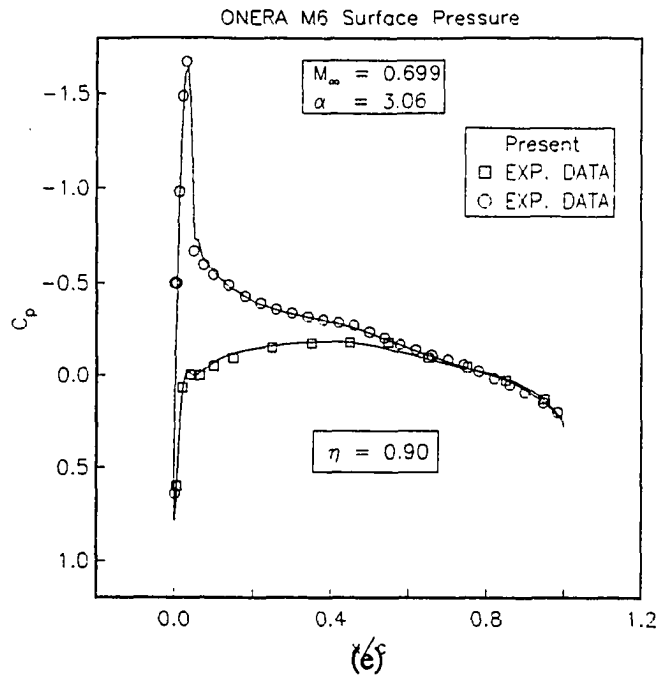


Figure 7.30 Pressure Distribution for ONERA M6 Wing

$M_\infty = 0.669$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$

The second test case was a supercritical case, with $M_\infty = 0.84$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$. This case is considered as one of the standard test cases available in the literature for validation of three-dimensional *CFD* algorithms. The same C-O grid reported earlier is used for the supercritical test cases. The flow is attached over the whole wing. Pressure contours on the upper wing surface are shown in Fig. 7.31, where a Lambda shock is clearly defined. Comparison between the computed results, experimental data [134], and the computational results from two other numerical algorithms, CFL3D and TLNS3D are displayed in Fig. 7.32. CFL3D, [73], and TLNS3D, [28], are considered to be state-of-the-art upwind and central difference schemes, respectively. The results reported here were performed using the thin-layer, Reynolds-Averaged Navier-Stokes equation and the Baldwin-Lomax, algebraic-turbulence model.

CFL3D is an upwind code which employs an implicit approximate-factorization time-marching algorithm. Roe's flux-differencing scheme is used to compute the inviscid flux with $\kappa = 1/3$. The viscous terms are evaluated through a second-order-accurate, differencing operator. TLNS3D utilizes a second-order central-difference operator for the spatial derivatives and an explicit, five-stage Runge-Kutta time marching scheme. Artificial dissipation was added to the central-difference algorithm to maintain numerical stability, and suppress oscillations in the vicinity of shock waves and stagnation points. For more detailed information about CFL3D and TLNS3D, the interested reader should consult the papers by the original developers of the two codes [73, 28]. Figure 7.32 shows good agreement between the present results and the numerical results obtained by CFL3D and TLNS3D. The computed results also agree well with the experimental data.

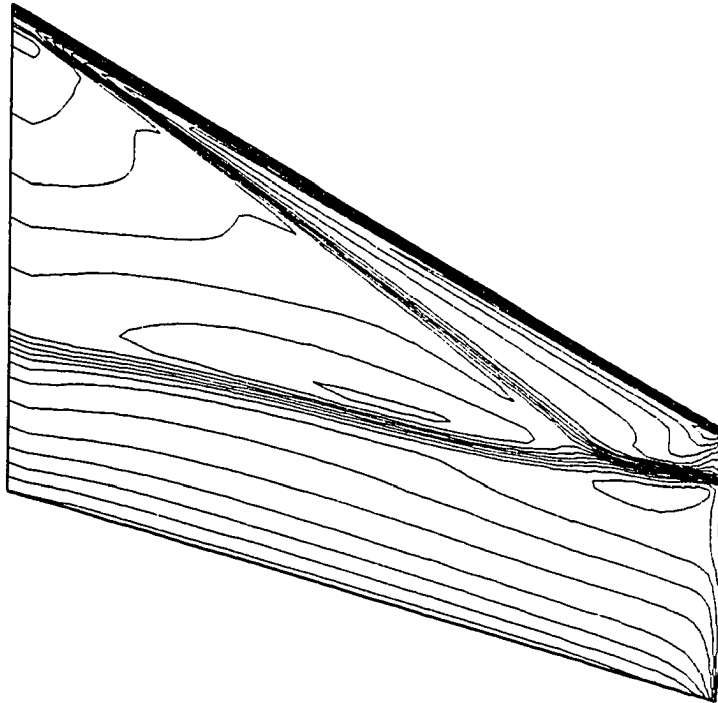
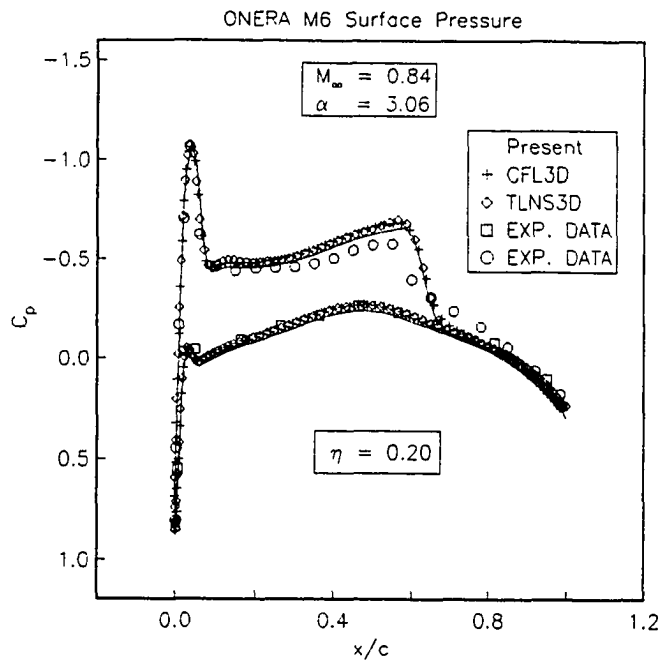
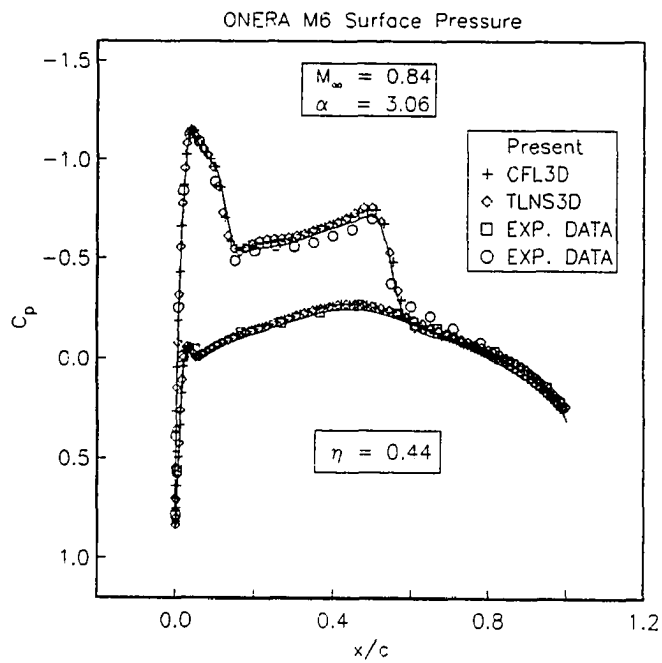


Figure 7.31 Pressure Contours for ONERA M6 Wing

$M_\infty = 0.84$, $\alpha = 3.06^\circ$, and $Re_\infty = 11.7 \times 10^6$.



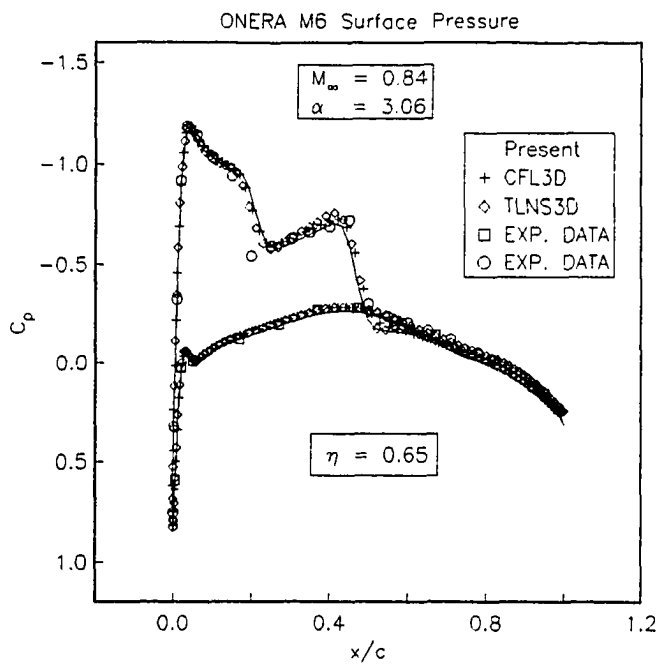
(a)



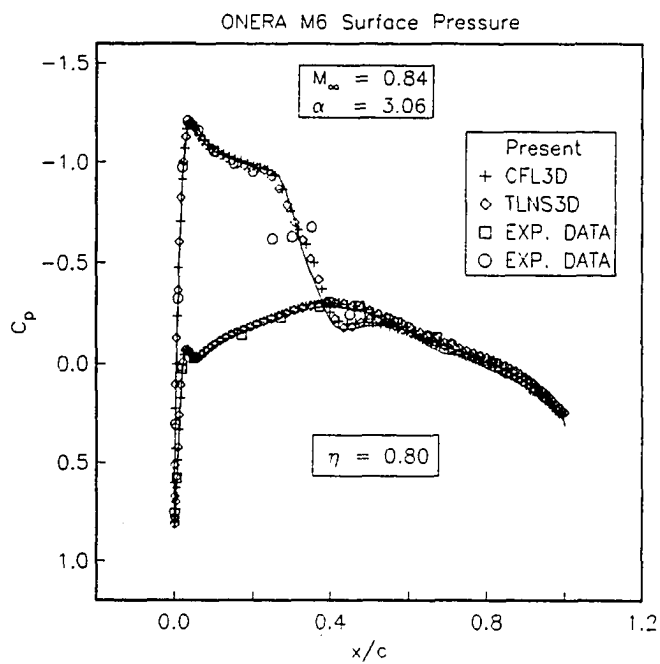
(b)

Figure 7.32 Pressure Distribution for ONERA M6 Wing

$M_{\infty} = 0.84$, $\alpha = 3.06^\circ$, and $Re_{\infty} = 11.7 \times 10^6$ (Continued . . .)



(c)



(d)

Figure 7.32 Pressure Distribution for ONERA M6 Wing

$M_{\infty} = 0.84$, $\alpha = 3.06^\circ$, and $Re_{\infty} = 11.7 \times 10^6$ (Continued . . .)

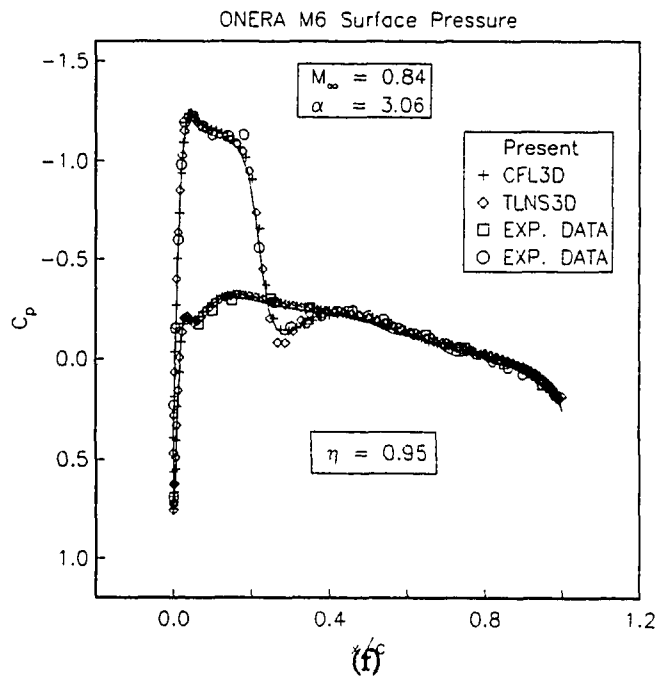
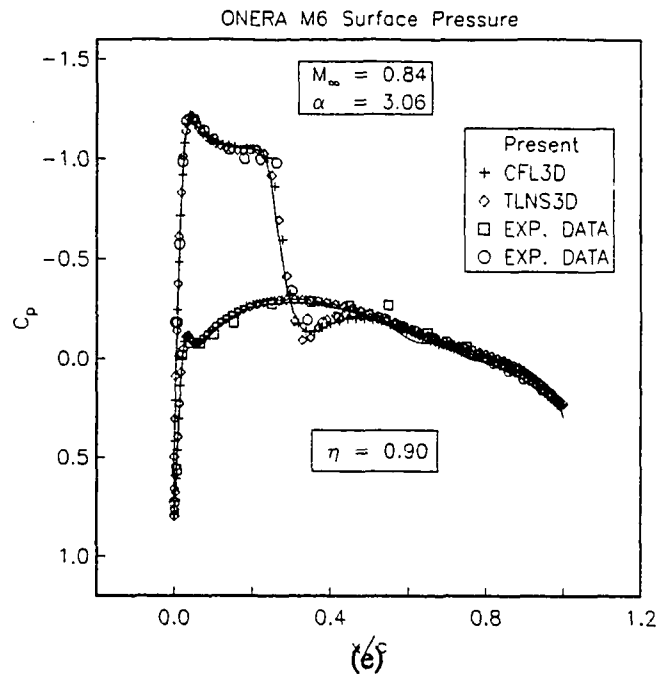


Figure 7.32 Pressure Distribution for ONERA M6 Wing

$$M_{\infty} = 0.84, \alpha = 3.06^{\circ}, \text{ and } Re_{\infty} = 11.7 \times 10^6$$

The third test case is a supercritical case, with $M_\infty = 0.84$, $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$. Again the same grid was used. This test case is more demanding than the two previous test cases, because stronger shocks develop on the upper surface of the wing, resulting in significant flow separation. The pressure contours on the upper surface are shown in Fig. 7.33. The streamlines on the upper surface of the ONERA M6 wing are shown in Fig. 7.34, where the separation region is clearly delineated. Comparison between pressure coefficients for the present computed results, previous experimental data [134], CFL3D [73], and TLNS3D [28], are displayed in Fig. 7.35. Large discrepancies between the computed results and the experiment exist, as shown in Fig. 7.35. As expected, the Baldwin-Lomax, algebraic-turbulence model cannot handle separated flow, and was incapable of capturing the global features of that flow. The size of the separated region was under-predicted, and as a result, the shock is located farther downstream.

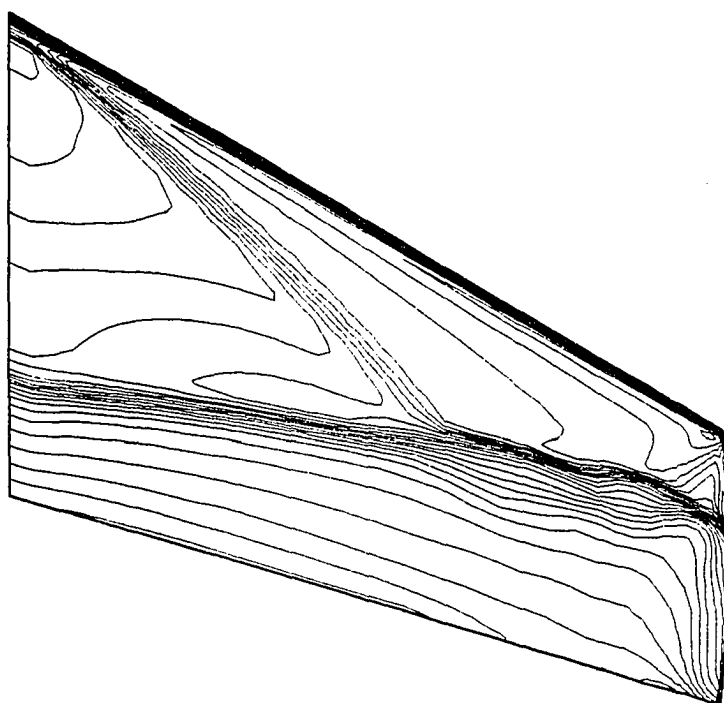


Figure 7.33 Pressure Contours for ONERA M6 Wing

$M_\infty = 0.84$, and $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$.

It should be emphasized that the performance of the present algorithm was comparable to the other two codes and the discrepancies between the computed results and experiment were not due to a lack of resolution of the grid, but due to limitations on the Baldwin-Lomax, algebraic-turbulence model. Abid, Vatsa, Johnson, and Wedan [150] reported that the agreement between TLNS3D results, obtained with the same grid, and experimental data were improved significantly when the Johnson-King turbulence model was employed.

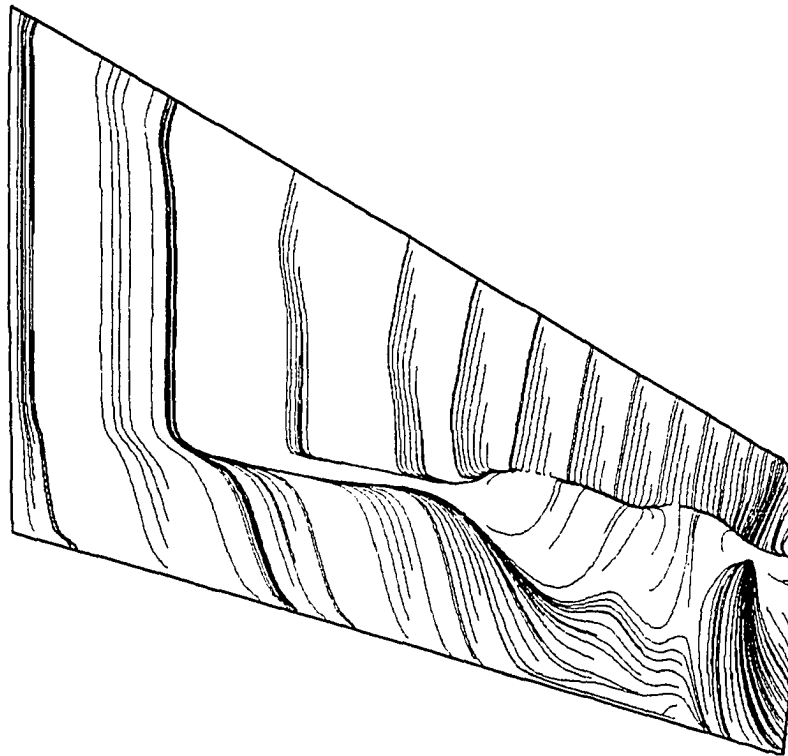
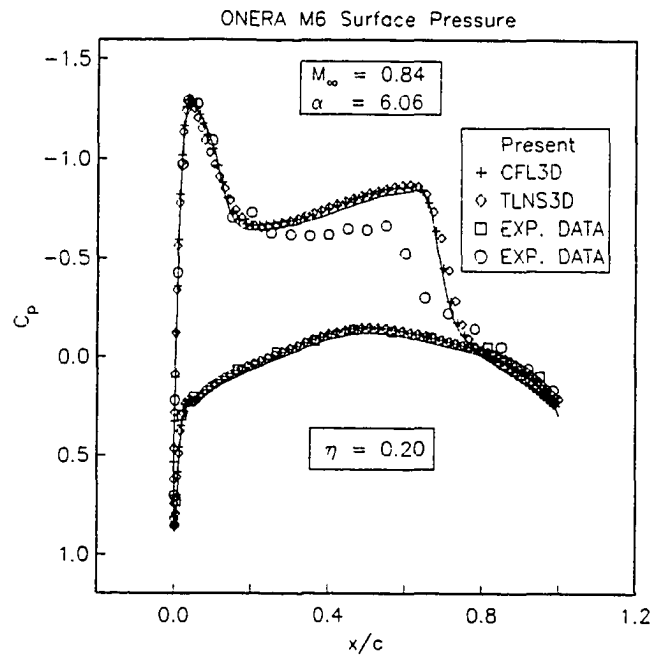
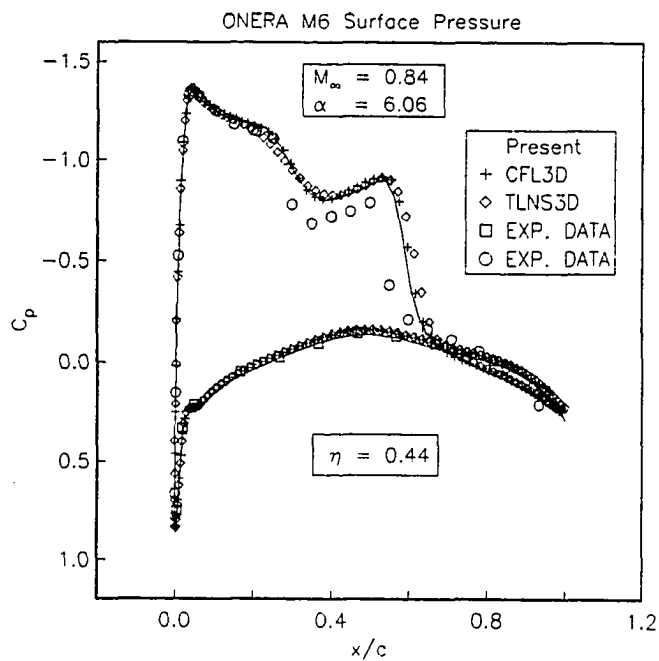


Figure 7.34 Wall Streamlines for ONERA M6 Wing

$$M_{\infty} = 0.84, \alpha = 6.06^{\circ}, \text{ and } Re_{\infty} = 11.7 \times 10^6.$$



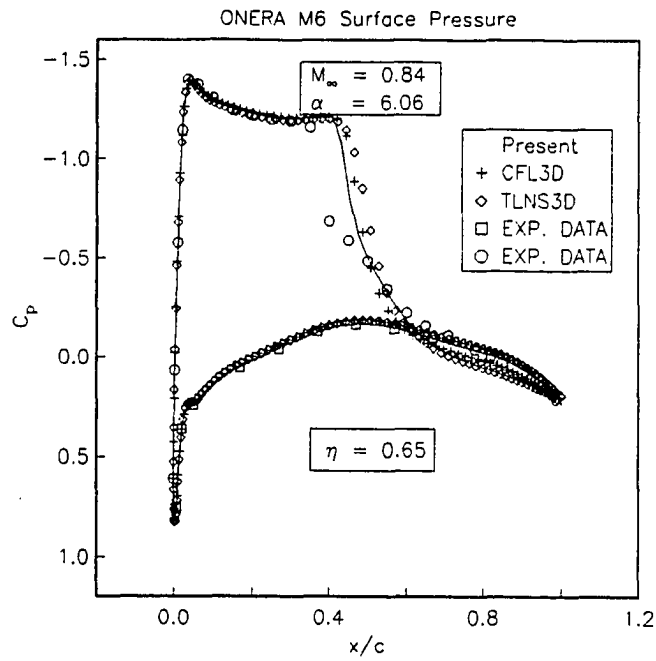
(a)



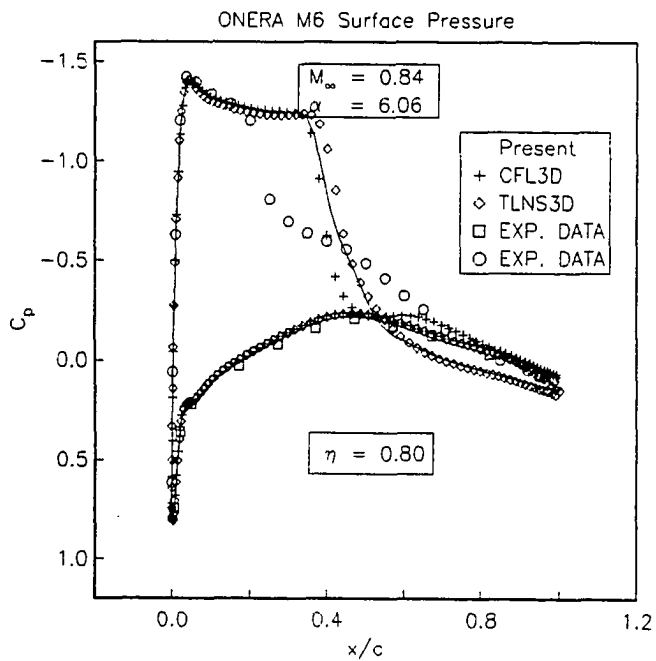
(b)

Figure 7.35 Pressure Distribution for ONERA M6 Wing

$M_{\infty} = 0.84$, $\alpha = 6.06^\circ$, and $Re_{\infty} = 11.7 \times 10^6$ (Continued . . .)



(c)



(d)

Figure 7.35 Pressure Distribution for ONERA M6 Wing

$M_\infty = 0.84$, $\alpha = 6.06^\circ$, and $Re_\infty = 11.7 \times 10^6$ (Continued . . .)

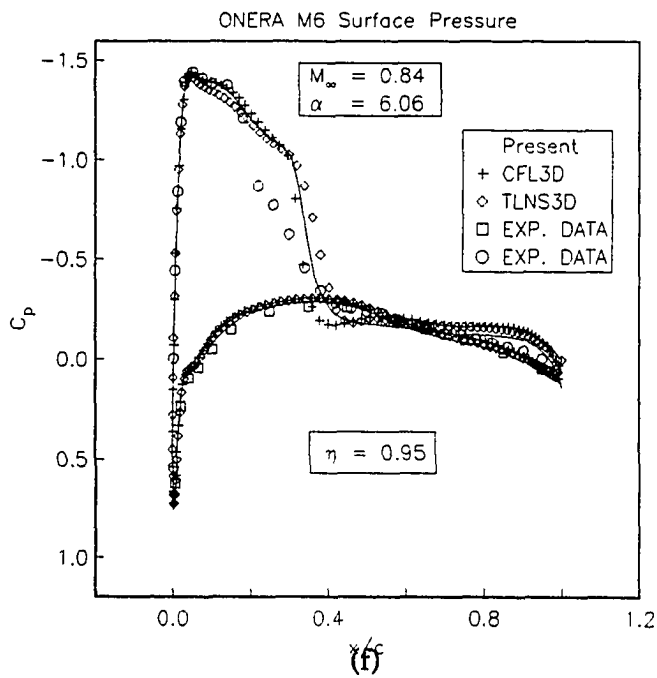
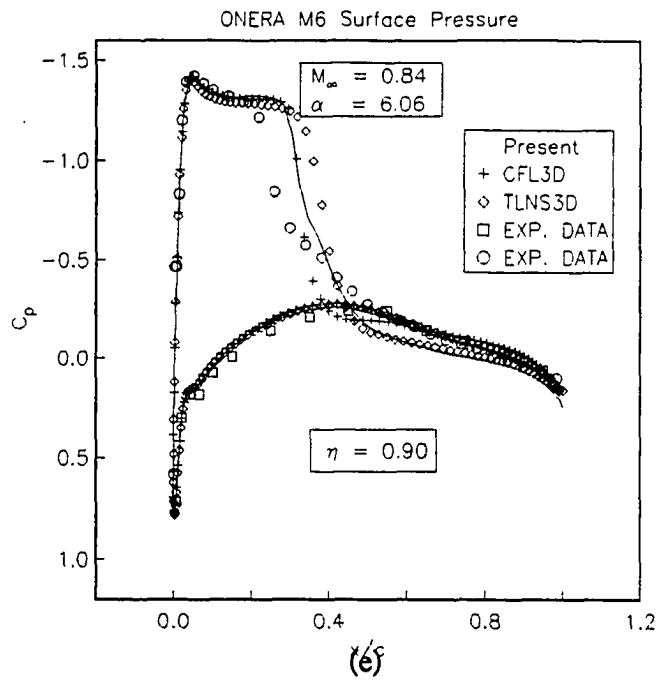


Figure 7.35 Pressure Distribution for ONERA M6 Wing

$$M_\infty = 0.84, \alpha = 6.06^\circ, \text{ and } Re_\infty = 11.7 \times 10^6$$

7.6 Jet In Ground Effect

The last test case to validate the developed algorithm was a simulation of a jet impinging on a flat plane. The analysis of such a complicated flow field, requires the solution of the three-dimensional, time-dependent, compressible, Navier-Stokes equations, in order to resolve the entire flow field accurately.

Two test cases were conducted: a single isolated-jet impinging on a ground-plane, and a jet impinging on a ground plane in the presence of an ambient crossflow. The jet impinging problem is a demanding test case. It should be emphasized that the numerical calculations which were performed here were devoted primarily to the development and verification of the algorithm for this three-dimensional, time-dependent, compressible Navier-Stokes case. Also, it was desired to demonstrate and assess the capability of the algorithm to capture the large scale phenomena associated with these unsteady flows.

The Baldwin-Lomax, algebraic-turbulence model incorporated to date in this algorithm is not capable of resolving separated flows accurately. As demonstrated in the previous test case (ONERA M6 wing at 6° angle of attack), separating and non-parallel flow features are not modeled properly. The use of classical Reynolds-averaged turbulence models to account for such a flow field is questionable and can even mask the fluid mechanics to be studied [137, 151]. Developing a universal turbulence model capable of resolving such a complicated flow field is beyond the scope of the present work, but should be pursued in the future.

Jet Impinging on a Ground Plane The numerical simulation of the flow field generated by an isolated jet impinging on a ground plane, in ambient air, was performed for a jet Mach number of 0.5 and a jet Reynolds number of 19000, based on the jet diameter. The jet exit was assumed to be 4 jet diameters away from the ground plane,

and the ground-plane was assumed to extend radially outward 15 jet diameters. The flow properties were normalized with respect to the jet inlet conditions.

A partial view of the grid used in the numerical computation is shown in Fig. 7.36. Grid lines are clustered near the solid boundary, and along the edge of the free- and wall-jet. The grid dimensions were 129 (normal) \times 97 (radial) \times 5 (circumferential) where the flow was assumed to be symmetric.

A schematic of the flowfield for the isolated jet impinging on a ground plane was shown previously in Fig. 7.3. The flow field is characterized by distinct regions, *i.e.*, free jet, shear layer, stagnation, and wall jet regions. The numerical simulation of such a complicated flow field can demonstrate the predictive capability of the present algorithm for resolving complex flows.

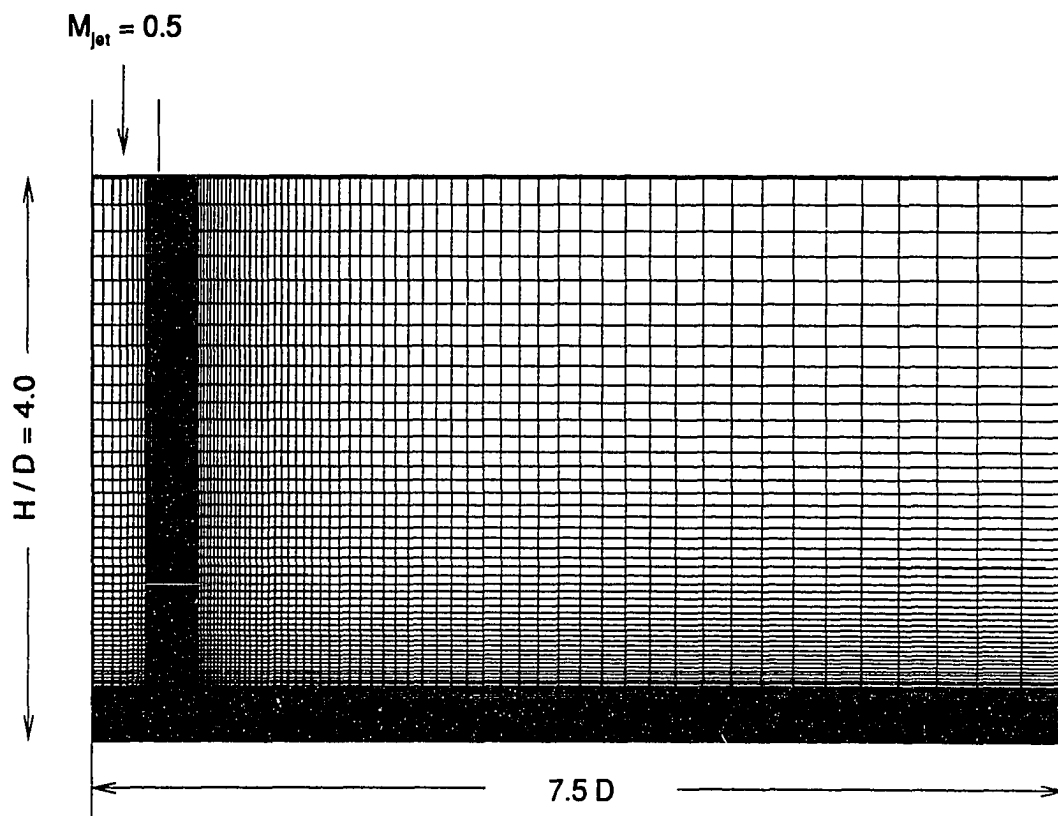
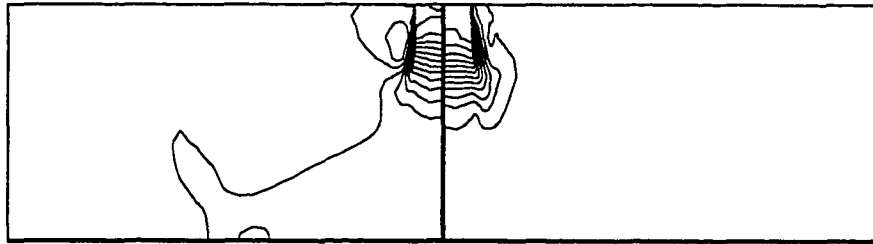


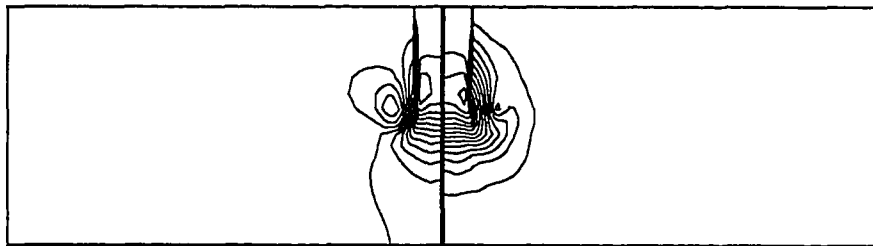
Figure 7.36 Partial View of Grid Utilized for
Computing a Jet Impinging on a Ground Plane.

The numerical simulation started with the jet entering the computational domain and interacting with quiescent air, Fig. 7.37a . The solution was advanced in time using a non-dimensional, global time step. The time step was based on the stability criterion for the scheme. Accelerating techniques, local time-stepping, residual smoothing and multigrid were not used in this test case because the flow was unsteady. Snap shots, at different time steps, of the Mach number and the stagnation pressure contours are shown in Fig. 7.37. A starting vortex developed as the free jet propagated into the computational domain, as shown in Figs. 7.37b-c. The free-jet impinged on the ground-plane, Fig. 7.37d, and is deflected to form a wall jet, as shown in Fig. 7.37e. The stagnation region creates a favorable pressure gradient which causes the wall jet to accelerate rapidly as it departs radially from the stagnation region, Figs. 7.37f-g. The primary vortex, located on top of the wall jet, produced a local unsteady adverse pressure gradient which caused the flow to separate. The wall jet, with a primary vortex located above, moves with the separation point radially outward as shown in Figs. 7.37f-h. From Fig. 7.37, the different flow fields that identify this type of flow, namely the free jet, the stagnation region, the wall jet and the moving separation, are predicted accurately and appear to be accurate depictions of the flow physics.

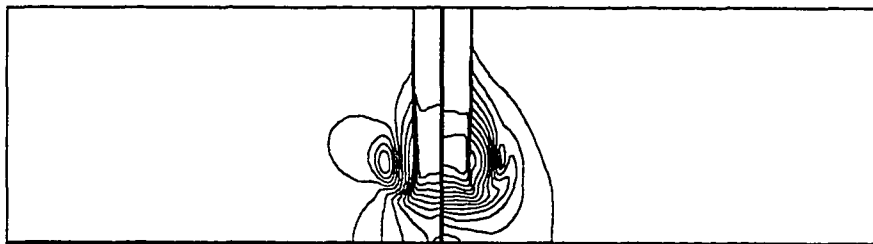
$$M_{jet} = 0.5$$



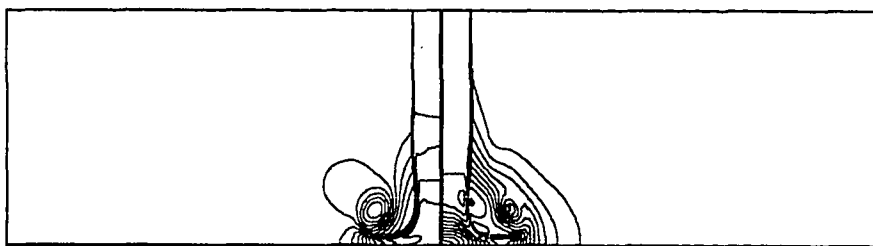
(a)



(b)



(c)



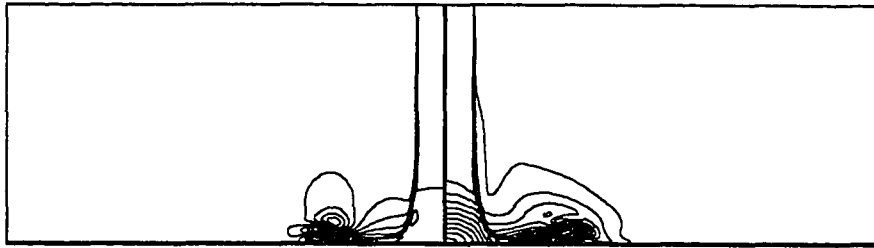
(d)

Stagnation Pressure Contours

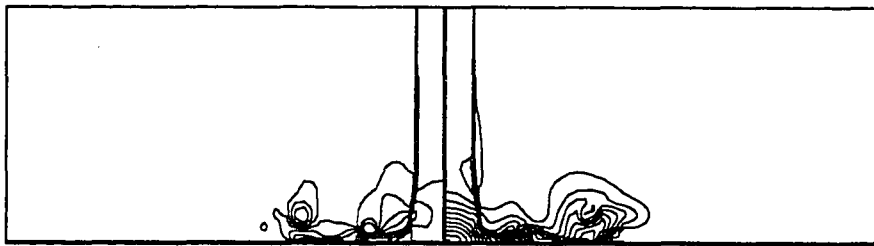
Mach Number Contours

Figure 7.37 Jet Impingement on a Ground Plane. $M_{jet} = 0.5$,
 $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 4$, $Re = 19000$. (Continued ...)

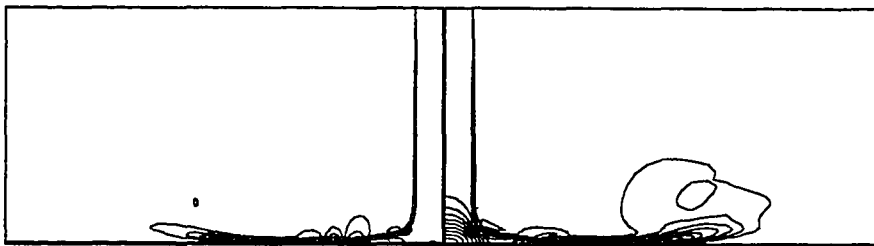
$$M_{jet} = 0.5$$



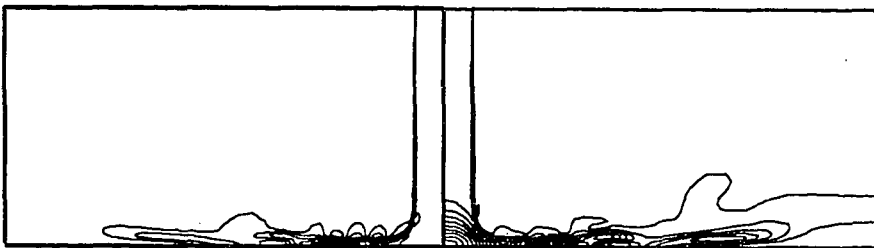
(e)



(f)



(g)



(h)

Stagnation Pressure Contours

Mach Number Contours

Figure 7.37 Jet Impingement on a Ground Plane.

$$M_{jet} = 0.5, P_{jet}/P_{\infty} = 1, \frac{T_{jet}}{T_{\infty}} = 1, \frac{H}{D} = 4, Re = 19000.$$

Profiles of the time-averaged velocity (tangential component), versus the normal wall distance off the plane are shown in Fig. 7.38. It can be observed that the favorable pressure gradient created by the jet impinging on the ground plane causes the wall jet to accelerate as it departs from the stagnation region, Fig. 7.39. As shown in Fig. 7.38, the time averaged velocity increased, from the stagnation region, to a maximum value downstream ($x/r = 2.81$). Further downstream the maximum value decreased and the boundary layer increased. The flow in the impact region (stagnation region) is dominated by the high pressure gradient, and the flow behaves in an almost inviscid manner.

Unfortunately, there are no experimental or numerical data sets available to permit comparison of the computed time-averaged profiles. However, a flow visualization study conducted by Didden and Ho [138], reported the unsteady separation in the boundary layer of an impinging jet on a flat plane. Didden and Ho, suggested that unsteady separation was induced by the primary vortex and it moved downstream in the radial direction. Jalamani [137], computed numerically the two-dimensional time-dependent, impulsively started jet issuing from a plate in proximity to the ground. His results indicated the formation of a primary vortex, and separation of the wall jet as it moved radially outward.

Comparison between the computed ground pressure distributions, and the numerical data of Bower [140], and the data of Bradbury [152] are presented in Fig. 7.40. The comparison shows good agreement between the computed results and other numerical, and experimental results. Comparison between the computed jet centerline velocity decay, and other computational results [137, 153, 154], and with experimental data [153] are presented in Fig. 7.41. The computed results compare well with experimental data. Data presented in Fig. 7.41, were obtained from Jalamani [137]. The discrepancy between numerical data are mainly because of difference in inlet jet conditions.

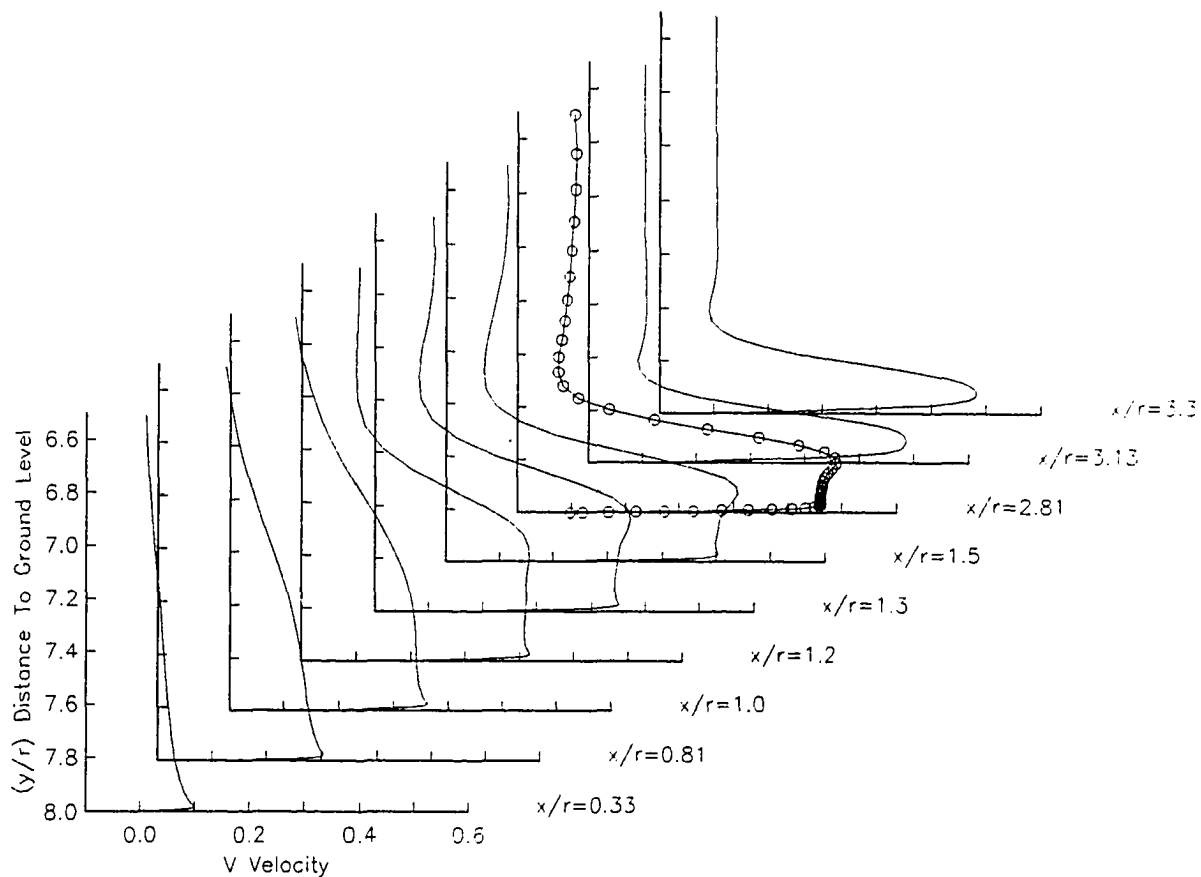


Figure 7.38 Wall Jet Time-Average-V-Velocity Profiles, at Various Radial Positions,
 Symbol ° Represents the Spacing of the Grid in the Normal Direction to the Ground Plane.

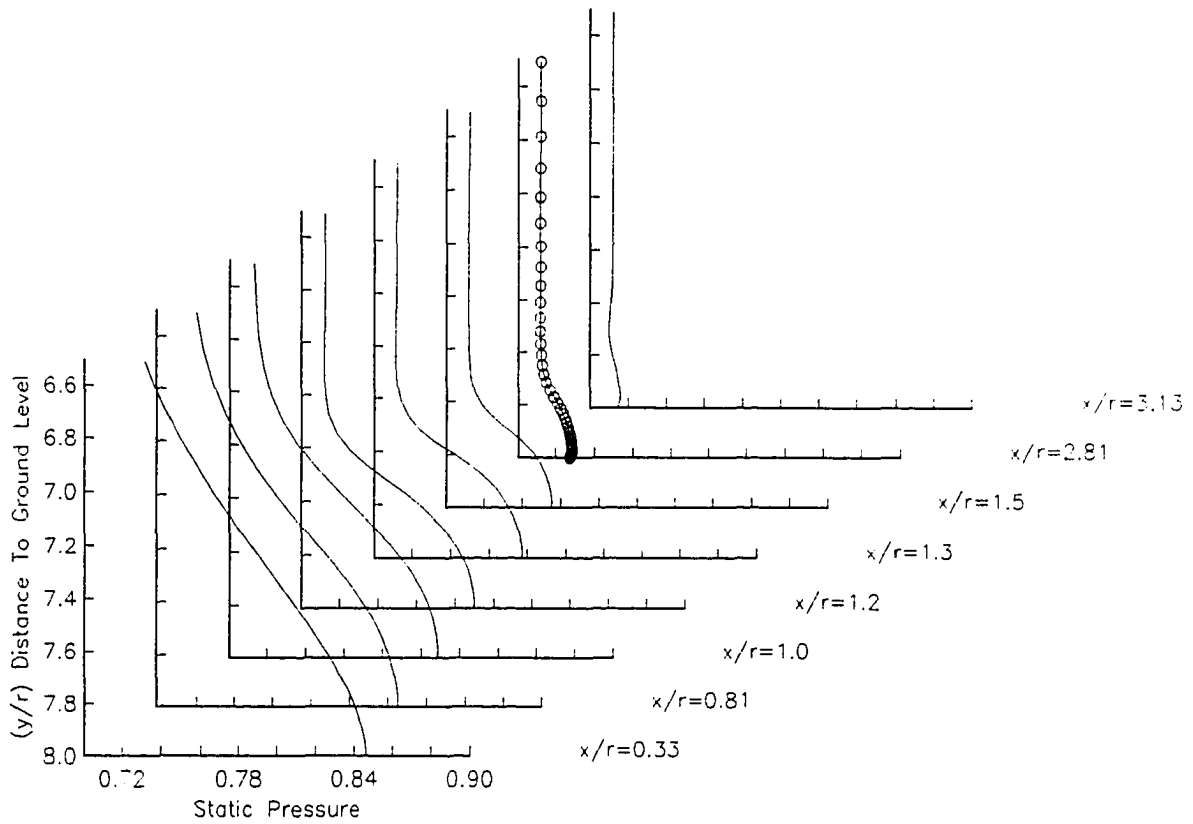


Figure 7.39 Wall Jet Time-Averaged-Static-Pressure Profiles, at Various Radial Positions, Symbol ° Represents the Spacing of the Grid in the Normal Direction to the Ground Plane.

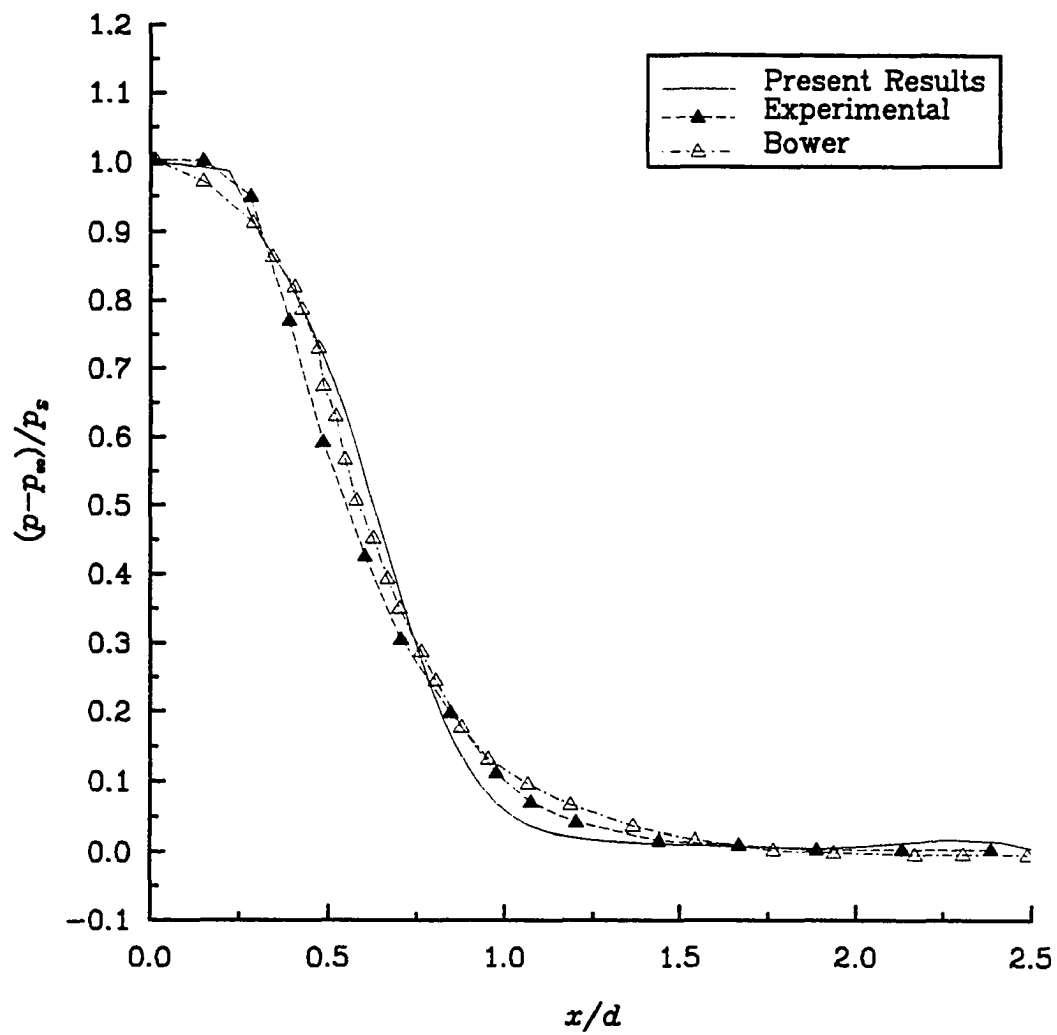


Figure 7.40 Comparison of Ground-Plane Pressure
Variation for Jet Impinging on a Ground Plane

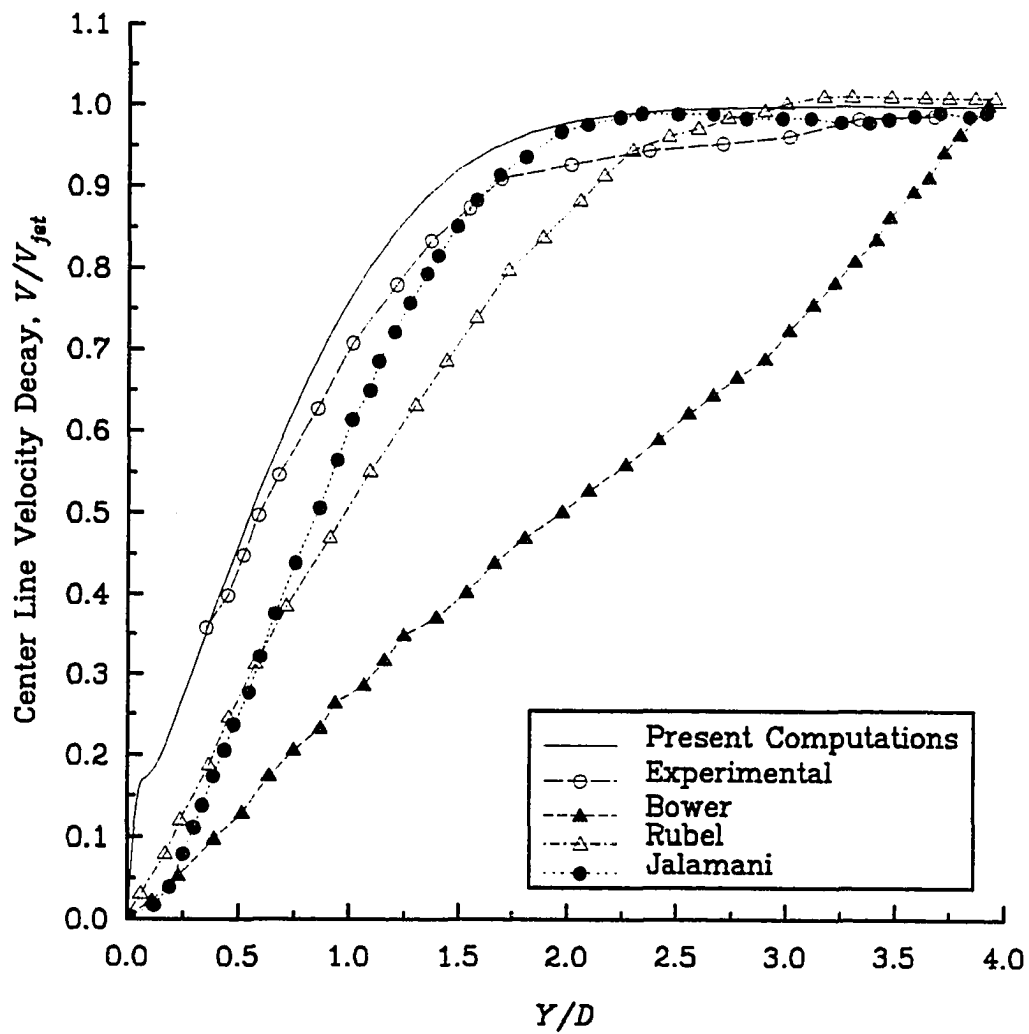


Figure 7.41 Comparison of Jet Centerline Velocity Decay

Jet Impinging on a Ground Plane in Presence of Crossflow When a jet impinges on a ground plane a wall jet is formed. The wall jet is symmetric and flows radially outward from the zone of impingement, as discussed in the previous section. In the presence of a crossflow, the wall jet flowing radially outward is opposed by the crossflow (free stream) and rolls up into a horseshoe-ground vortex as shown in Fig. 7.4. The flow field of the impinging jet in a cross flow has been investigated extensively in recent years, because of its relevance to the flow field around V/STOL aircraft.

In the present study, the flow of a jet impinging on a ground plane in the presence of a cross flow was simulated numerically. A partial view of the grid used in the computations is shown in Fig. 7.42. The grid dimensions were $66 \times 66 \times 33$ in the normal, radial, and circumferential directions, respectively. The jet Mach number was 0.5 and the Reynolds number was 100,000, based on the jet diameter. The jet exit was 3 jet diameters away from the ground plane, and the plane extended radially outwards 7.5 jet diameters. The ratio of the crossflow velocity to the jet velocity was 0.1 ($V_\infty/V_{jet} = 0.1$). The flow properties were normalized with respect to the jet inlet conditions.

The numerical simulation started by allowing the crossflow to propagate in the computational domain until a steady-state solution was obtained. Thus, allowing a boundary layer to form on the ground plane. To enhance the rate of convergence of

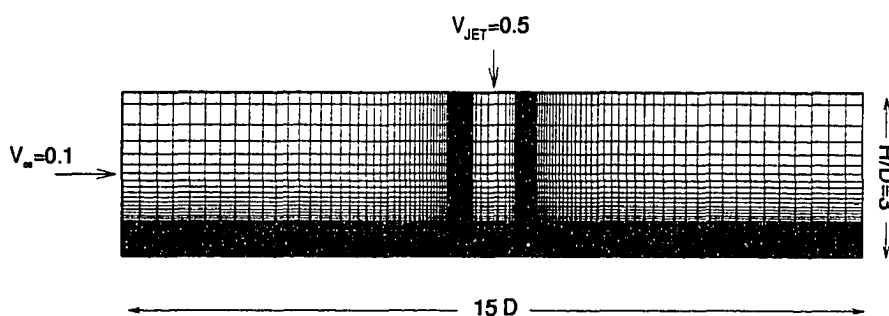


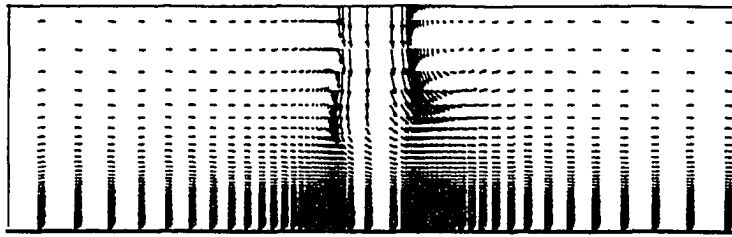
Figure 7.42 Partial View of the Grid Utilized to Compute
Jet Impinging on a Ground Plane in Presence of Crossflow.

the cross flow to steady-state, local time-stepping, residual smoothing and multigrid were employed. The next step in the simulation was to release the jet flow into the computational domain. Once the jet was initiated into the computational domain, it was allowed to interact immediately with the crossflow, and the accelerating techniques; local time-stepping, residual smoothing and multigrid were turned off for this unsteady flow. The solution was advanced in time using a non-dimensional global time step. The time step is based on the stability criteria of the scheme. Snap shots, at different time steps, of the velocity vectors and the corresponding streamlines are shown in Figs. 7.43 and 7.44.

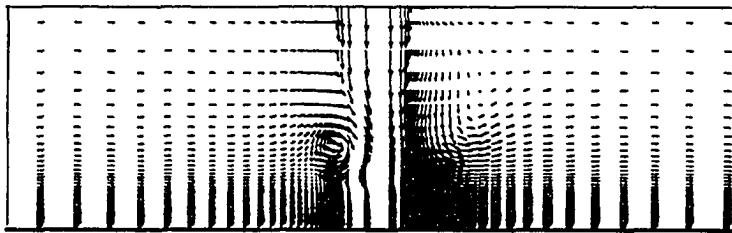
The jet propagates into the computational domain and interacts immediately with the crossflow, as shown in Figs. 7.43a and 7.44a. The crossflow momentum attempts to divert the incoming jet, but due to a lack of sufficient momentum, it fails ($\frac{V_{\infty}}{V_{jet}} = 0.1$) to cause a significant diversion. At the zone of interaction between the free jet and the crossflow, a primary vortex starts to form. The size of the vortex increases as it is convected toward the ground plane with the jet, as shown in Figs. 7.43b and 7.44b. The free jet impinges on the ground plane and is deflected as a wall jet as shown in Figs. 7.43c and 7.44c. Subsequently, the stagnation region creates a favorable pressure gradient which causes the wall jet to depart from the stagnation region. The wall jet, flowing radially outward, is opposed on one side by the crossflow (free stream), and rolls up into a horseshoe-ground vortex as shown in Figs. 7.43d and 7.44d. The size of the ground vortex increases as it moves radially outward, Figs. 7.43e-h and 7.44e-h. The ground vortex forms at a radial location where a momentum balance is reached between the wall and crossflow. Fluid mass, from the wall jet (impinging zone) and crossflow, accumulates inside the vortex until it cannot sustain itself, and a vortex breakdown occurs. A new vortex forms and the process repeats itself. The visualization study conducted by Cimbala et al. [147] supports the above description of the flow field of a jet impinging

on a ground plane in presence of cross flow.

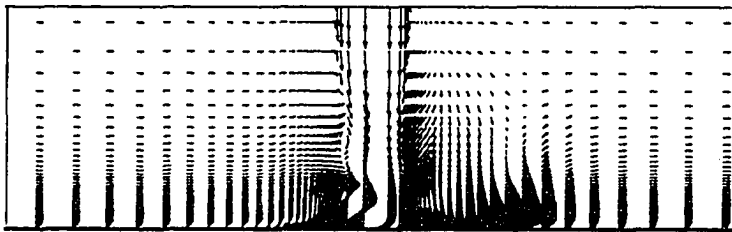
A comparison between the computed pressure coefficient distributions, measured along the jet centerline, and the numerical results (laminar and turbulent) of van Dalsem [142] and the experimental results of Stewart [141] is shown in Fig. 7.45. The present results compare fairly well with the experimental data [141]. The developed algorithm computed the location of the ground vortex accurately. Both the laminar and turbulent results of van Dalsem predicted that the ground-vortex was located further upstream than the present numerical results, and experimental results. The core of the ground vortex corresponds to the point with the minimum C_p value. The present numerical simulation captured the large scale unsteady phenomena of a jet impinging on a ground plane with crossflow, and predicted the location of the ground vortex accurately. Thus the predictive capability of the developed algorithm has been demonstrated by accurately resolving the complicated fluid dynamics phenomena associated with such a basic, but complex flow field.



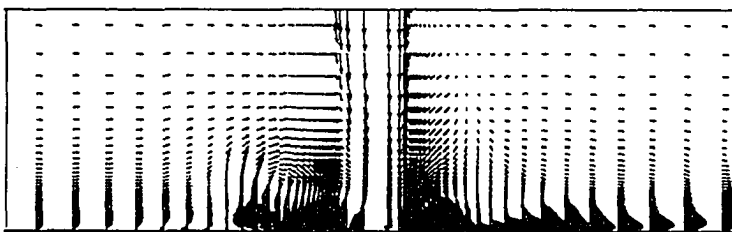
(a)



(b)



(c)

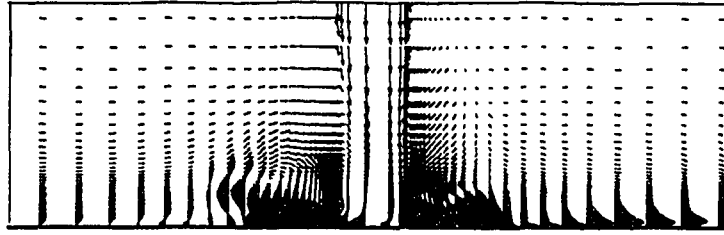


(d)

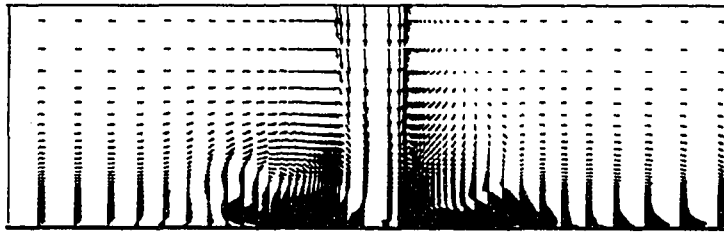
Figure 7.43 Velocity Vectors of the Developing Jet Flow Field in a Crossflow;

$$M_{jet} = 0.5, \quad M_{\infty} = 0.1, \quad P_{jet}/P_{\infty} = 1, \quad P_{jet}/P_{\infty} = 1,$$

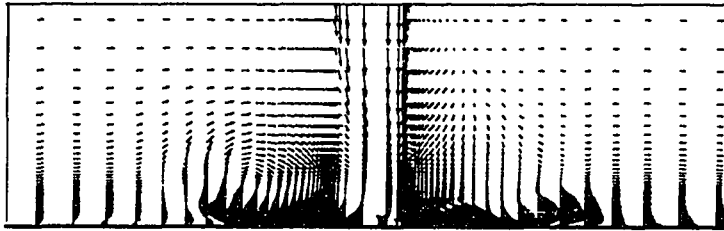
$$\frac{T_{jet}}{T_{\infty}} = 1, \quad \frac{H}{D} = 3, \quad Re = 1 \times 10^5 \quad (\text{Continued } \dots)$$



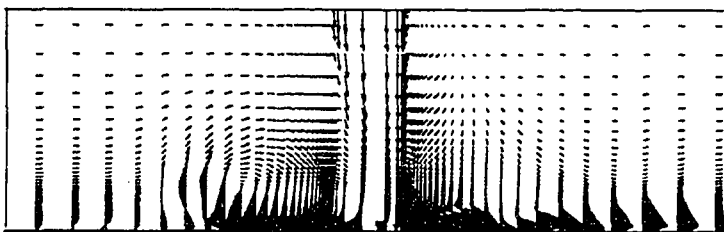
(e)



(f)



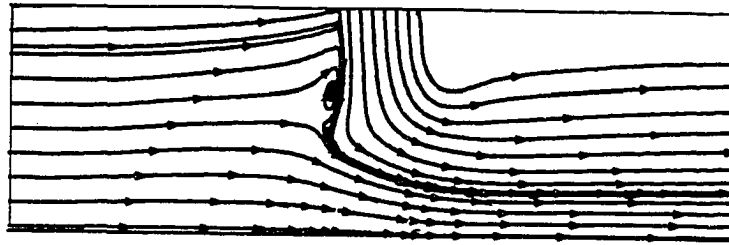
(g)



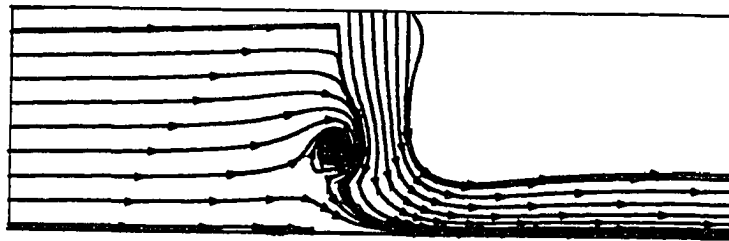
(h)

Figure 7.43 Velocity Vectors of the Developing Jet Flow Field in a Crossflow;

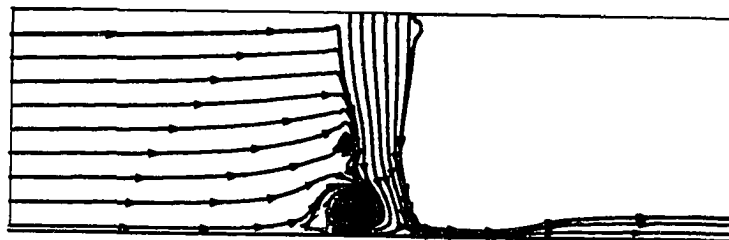
$$M_{jet} = 0.5, \quad M_{\infty} = 0.1, \quad P_{jet}/P_{\infty} = 1, \\ P_{jet}/P_{\infty} = 1, \quad \frac{T_{jet}}{T_{\infty}} = 1, \quad \frac{H}{D} = 3, \quad Re = 1 \times 10^5$$



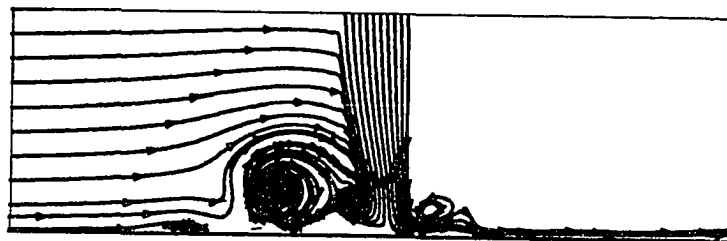
(a)



(b)



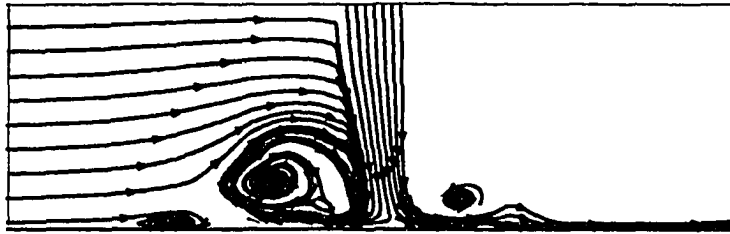
(c)



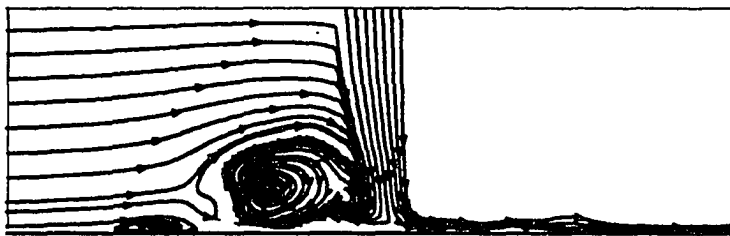
(d)

Figure 7.44 Streamlines of the Developing Jet Flow Field in a Crossflow $M_{jet} = 0.5$,

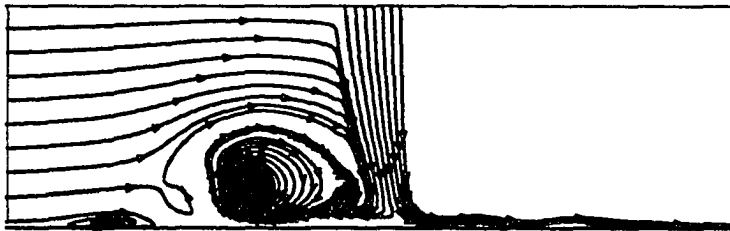
$M_{\infty} = 0.1$, $P_{jet}/P_{\infty} = 1$, $\frac{T_{jet}}{T_{\infty}} = 1$, $\frac{H}{D} = 3$, $Re = 1 \times 10^5$ (Continued . . .)



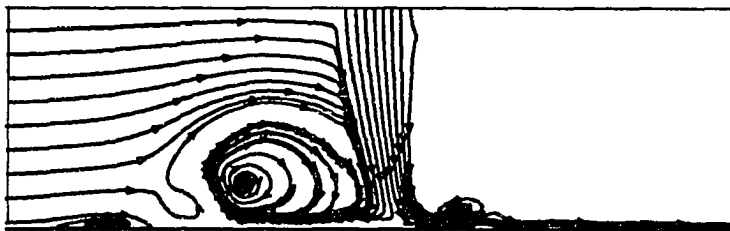
(e)



(f)



(g)



(h)

Figure 7.44 Streamlines of the Developing Jet Flow Field in a Crossflow

$$M_{jet} = 0.5, \quad M_{\infty} = 0.1, \quad P_{jet}/P_{\infty} = 1, \quad \frac{T_{jet}}{T_{\infty}} = 1, \quad \frac{H}{D} = 3, \quad Re = 1 \times 10^5$$

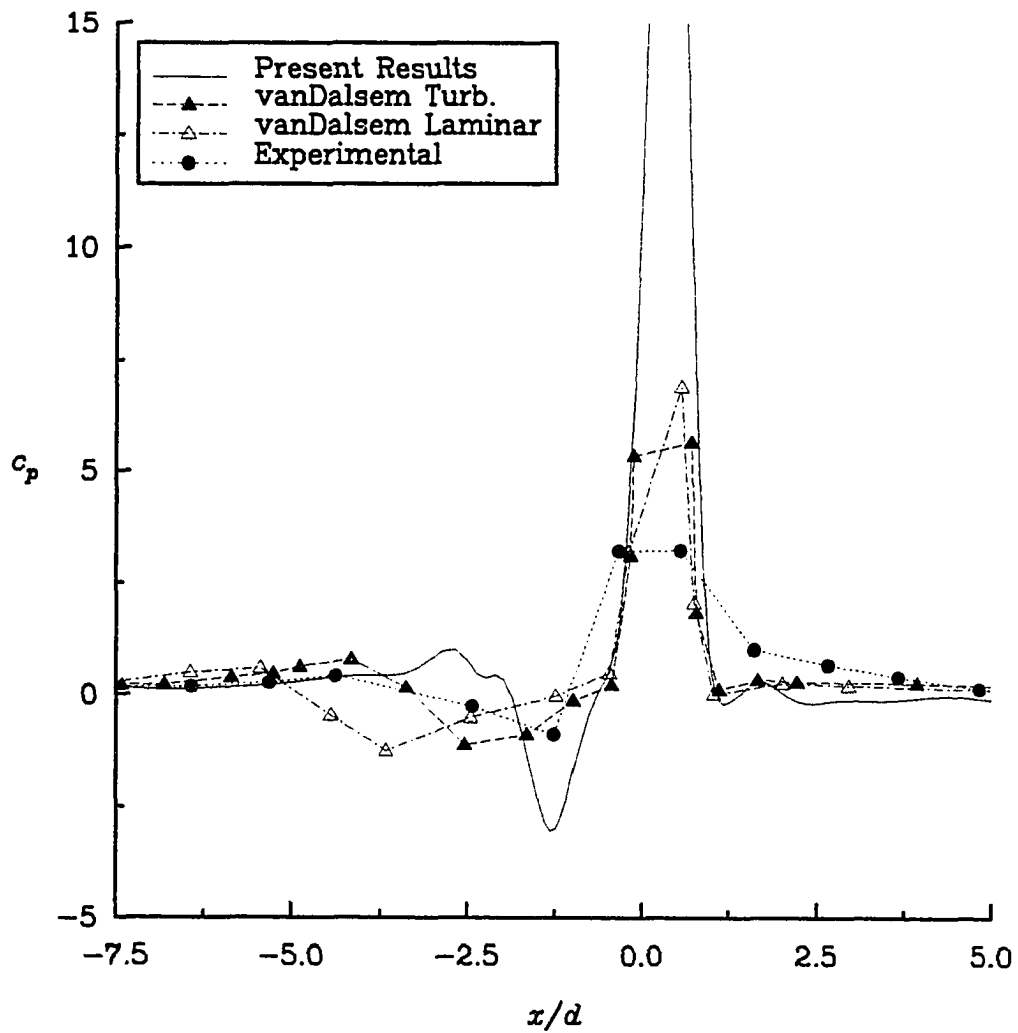


Figure 7.45 Comparison of Centerline C_p Distributions for a Jet
 Impinging on a Ground Plane with Crossflow; $M_{jet} = 0.5$,
 $M_\infty = 0.1$, $P_{jet}/P_\infty = 1$, $\frac{T_{jet}}{T_\infty} = 1$, $\frac{H}{D} = 3$, $Re = 1 \times 10^5$.

CHAPTER 8

CONCLUSIONS

The main objective of the present work was to develop a high-resolution-explicit-multi-block algorithm, suitable for efficient computation of three-dimensional, time-dependent Euler and Navier-Stokes equations. The developed algorithm employed a finite volume approach, and used MUSCL-type differencing to obtain the state variables at the cell interface. The variable interpolations were written in the κ -scheme formulation. Two state-of-the-art techniques were available to construct the inviscid fluxes: Roe's flux-difference splitting, and van Leer's flux-vector splitting. The viscous terms were discretized using a second-order central-difference operator.

The present study investigated two classes of explicit time integration (two-stage predictor-corrector schemes, and multistage time-stepping schemes) for solving the compressible inviscid/viscous flows. The 1-2 Predictor-Corrector Scheme was very effective for predicting inviscid flows, but did not perform well with multigrid acceleration techniques, due to insufficient damping of the high frequency components of the error, over a wide range of CFL numbers. The 2-2 Predictor-Corrector Scheme provided much better damping for the high frequency components of the error, but the maximum allowable CFL number was only one. However, the 2-2 Predictor-Corrector Scheme combined well with multigrid acceleration technique and the implicit residual smoothing.

The standard coefficients of the modified Runge-Kutta method have been modified successfully to achieve better performance with upwind differencing. A technique has been developed to optimize the coefficients for good high frequency damping at relatively

high CFL numbers. The coefficients were optimized using the stability and damping factor analysis of the linear wave equation. The optimization was carried out for two-, three-, and four-stage schemes. For each scheme the coefficients were optimized for four spatial upwind operators: first-order, second-order upwind ($\kappa = -1$), Fromm second-order upwind biased ($\kappa = 0$), and third-order upwind biased ($\kappa = 1/3$), operators. The coefficients for a total of twelve schemes were optimized. The multistage schemes with the optimized set of coefficients were tested on a number of three-dimensional inviscid/viscous flows. In general, the optimum CFL number agreed surprisingly well with the stability analysis study, especially for the inviscid cases.

The explicit upwind schemes, especially for viscous flows, were not effective in solving the compressible flow equations, when no additional convergence acceleration techniques were incorporated. The main draw back was the restriction on the time step. Coupling the explicit schemes with convergence accelerating techniques is essential. The accelerating techniques: local time-stepping, implicit residual smoothing, and multigrid procedure, proved to be effective in accelerating the rate of convergence to steady-state.

The implicit residual smoothing extended the stability range of the explicit scheme, which simultaneously allowed the use of higher CFL numbers. The smoothing operator was applied after every stage of the multistage stepping scheme. Not only did the smoothing operator extend the stability range of the explicit scheme, but also provided better damping of the high frequency component of the error, which is necessary for multigrid acceleration techniques to be effective. The adaptive (variable coefficient) implicit residual smoother proved to be successful in accelerating the rate of convergence on highly stretched grids with high aspect ratios.

When using multigrid, the improved damping properties are more important than a slight increase in the maximum allowable CFL number. Good damping of the high

frequency component of the error (close to $\beta = \pi$), usually occurred at CFL numbers lower than the maximum allowable. Multigrid acceleration techniques enhanced the rate of convergence, but true multigrid convergence was not achieved (to the best of the author's knowledge, because true multigrid performance has not been achieved in any compressible code). Multigrid performed well with the multistage time-stepping techniques and the 2-2 Predictor-Corrector Scheme. It did not perform well with the 1-2 Predictor-Corrector Scheme, due to insufficient damping of the high frequency component of error over a wide range of CFL numbers.

The developed algorithm was implemented successfully in a multi-block code. The multi-block structure provides complete topological and geometric flexibility. The only requirement is C^0 continuity of the grid across the block interface. The solution domain was divided into multiple blocks, and the grid for each block was then generated. The type of boundary conditions used on each of the blocks was provided through an input file at run time, rather than the traditional method of hard-coding the different boundary conditions in the source code. Non-homogenous boundary conditions per block face allowed for more geometric flexibility, and simplified handling the boundaries for complex three-dimensional configurations. The application of the developed explicit upwind schemes to realistic three-dimensional configurations of significant geometric complexity is virtually impossible without the use of multi-block capability.

The developed algorithm was validated on a number of diverse three-dimensional test cases with increasingly complex flow characteristics: (1) supersonic corner flow, (2) supersonic plume flow, (3) laminar and turbulent flow over a flat plate, (4) ONERA M6 wing, and (5) the unsteady flow of a jet impinging on a ground plane (with and without cross flow). The test cases were selected to validate certain aspects of the developed

algorithm. All of the test cases presented in this study have been computed with the same source code; only inputs to the program have been changed.

Supersonic inviscid corner flow was calculated to verify the multi-block structure. It was also used as the bench mark test case for all the explicit time integration schemes. Detailed examination of a quasi-two-dimensional jet exhaust plume demonstrated that van Leer's flux-vector splitting, and Roe's flux-difference splitting gave solutions of comparable quality on grids not aligned with shocks and contact discontinuities. Extrapolating the conservative variables across slip lines resulted in large over/under shoots for Roe's scheme. The primitive variable extrapolation was found to render a smoother solution.

Laminar and turbulent flow over a flat plate was computed to verify the correct implementation of the viscous terms and the Baldwin-Lomax algebraic turbulent model in the developed algorithm. The computations demonstrated the effectiveness of the adaptive implicit residual smoothing and multigrid procedure in accelerating convergence to steady-state.

The ONERA M6 wing was computed for three different cases: a subcritical test case ($M_\infty = 0.699$, $\alpha = 3.06^\circ$), a supercritical case with attached flow ($M_\infty = 0.84$, $\alpha = 3.06^\circ$), and a supercritical case with separated flow ($M_\infty = 0.84$, $\alpha = 6.06^\circ$). For all three cases, the Reynolds number was $11.7 \times 10^6/\text{unit}$ based on the free stream conditions and the mean aerodynamic dimension. Good agreement between the computed results and other numerical and experimental results was achieved. The developed algorithm cannot resolve separated turbulent flows accurately, because of the inadequacy of the Baldwin-Lomax algebraic-turbulence model to resolve separated flows. A better turbulence model needs to be incorporated if turbulent separated flows are to be investigated accurately. It should be noted that turbulence modeling was not the emphasis in this study.

The unsteady viscous flow resulting from a jet impinging on a ground plane, with and without cross flow, was successfully computed. The different flow fields that identify this type of flow, namely the free jet, the stagnation region, the wall jet, the ground vortex, and the moving separation zone, were predicted accurately. The jet impingement problem demonstrated the capability of the developed scheme to perform time-dependent calculations. For unsteady flow computations, if the maximum allowable time step is of the same order as the time scale of the physical problem, then explicit schemes will best suit the calculations.

A state-of-the-art computational tool has been developed that is capable of computing the flow field around geometrically complex three-dimensional configurations. The method has been tested on a number of diverse three-dimensional test cases with increasingly complex flow characteristics. The developed algorithm is capable of accurately resolving steady and unsteady compressible flow fields. To further enhance the performance of the developed algorithm the following recommendations are suggested:

The damping of the explicit time-stepping schemes on highly stretched grids with cells that have extremely high aspect ratios should be investigated. A stability analysis study should be conducted on the two-dimensional, scalar wave equation and the diffusion equation to investigate the effect of the grid aspect ratio on the damping characteristics of the multistage schemes. This study should take into account the effect of implicit residual smoothing and multigrid procedure.

Multigrid acceleration techniques performed well for inviscid non-stretched grids, but the convergence rate deteriorated on viscous grids. One possible remedy lies in using the technique of semi-coarsening in the direction normal to the wall, to improve the non-uniformity on the coarse-grids. Other approaches have to be combined with the multigrid method to damp the high frequency-component of error at all grid levels. An

attempt to develop a mathematical operator to annihilate the high frequency component of the error should be investigated.

A joint analytical and numerical study should be conducted to develop higher order accurate, non-oscillatory schemes. A stability study should be performed to investigate the damping characteristics of the higher order schemes.

Explicit schemes are amenable for implementation on massively parallel supercomputers. Work is already in progress to implement the developed algorithm on massively data-parallel architecture (Connection Machine CM-2).

A higher order turbulence model needs to be employed in the developed algorithm, to be capable of resolving turbulent separated flows.

References

- [1] Rizzi, A., "Selected Topics in the Theory and Practice of Computational Fluid Dynamics," *Journal of Computational Physics*, vol. 72, 1987, pp. 1-69.
- [2] Courant, R., Friedrichs, K. O., and Lewy, H., "On the partial differential equations of mathematical physics," *IBM, Journal of Research and Development*, vol. 11, 1967, pp. 215-234.
- [3] Jameson, A., "Successes and Challenges in Computational Aerodynamics.," AIAA Paper 87-1184, 1987.
- [4] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*. McGraw-Hill Book Company, Inc., 1984.
- [5] Roe, P. L., "Characteristic Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, vol. 18, 1986, pp. 337-365.
- [6] Hirsch, C., *Numerical Computation of Internal and External Flows*, vol. 1 & 2. John Wiley & Sons, 1990. Computational Methods for Inviscid and Viscous Flows.
- [7] Hess, J. L. and Smith, A. M. O., "Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies," tech. rep., Douglas Aircraft Report ES40622, 1962.
- [8] Rubbert, P. E. and Saaris, G. R., "A General Three-Dimensional Potential Flow Method Applied to V/STOL Aerodynamics," SAE Paper 680304, 1968.
- [9] Woodward, F. A., "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic And Supersonic Flow; Part 1 Theory and Application," tech. rep., NASA CR 2228 Part 1, May 1973.
- [10] Kandil, O. A. and Yates Jr., E. C., "Transonic Vortex Flows Past Delta Wings: Integral Equation Approach," *AIAA Journal*, vol. 24, no. 11, 1986, pp. 1729-1736.
- [11] Kandil, O. A. and Hong, H., "Full-Potential Integral Solution for Transonic Flows with and without Embedded Euler Domains," AIAA Paper 87-1461, 1987.
- [12] Murman, E. M. and Cole, J. D., "Calculation of Plane Steady Transonic Flows," *AIAA Journal*, vol. 9, no. 1, 1970, pp. 114-121.
- [13] Jameson, A., "Numerical calculation of the three dimensional transonic flow over a yawed wing," in *Proceedings, AIAA Computational Fluid Dynamics Conference, Palm Springs, California*, July 19-20, 1973.
- [14] Hall, M. G., "Computational Fluid Dynamics — A Revolutionary Force in Aerodynamics.," AIAA Paper 81-1014, 1981.

- [15] Magnus, R. and Yoshihara, H., "Inviscid Transonic Flow Over Airfoils," *AIAA Journal*, vol. 8, 1970, pp. 2157-2162.
- [16] MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-352, 1969.
- [17] MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flows.," AIAA Paper 81-0110, 1981.
- [18] Warming, R. F. and Beam, R. M., "Upwind second-order difference schemes and applications in unsteady aerodynamic flows," in *AIAA 2nd Computational Fluid Dynamics Conference, Hartford, Connecticut*, 1975.
- [19] Briley, W. R. and McDonald, H., "Solution of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique," in *Proceedings of the 4th International Conference on Numerical Methods in Fluid Dynamics*, 1974.
- [20] Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," *Journal of Computational Physics*, vol. 22, 1976, pp. 87-110.
- [21] Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA*, vol. 16, April, 1978, pp. 393-402.
- [22] Peaceman, D. W. and Rachford, H. H., "The Solution of Parabolic and Elliptic Differential Equations," *Journal of the Society of Industrial and Applied Mathematics*, vol. 3, pp. 28-41.
- [23] Douglas, J., "On the Numerical Intergration of $u_{xx} + u_{yy} = u_t$ By Implicit Methods," *Journal of the Society of Industrial and Applied Mathematics*, vol. 3, 1955, pp. 42-65.
- [24] Douglas, J. and Gunn, J. E., "A General Formulation of Alternating Direction Methods, Part 1. Parabolic and Hyperbolic Problems," *Numerische Mathematik*, vol. 6, 1964, pp. 428-453.
- [25] Steger, J. L., "Implicit Finite Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries," *Journal of Computational Physics*, vol. 16, 1978, pp. 679-686.
- [26] Pulliam, T. H. and Steger, J. L., "On Implicit Finite-Difference Simulations of Three-Dimensional Flow," *AIAA Journal*, vol. 18, 1980, pp. 159-167.
- [27] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite-Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June, 1981.

- [28] Vatsa, V. N. and Wedan, B. W., "Development of a Multigrid Code for 3-D Navier-Stokes Equations and its Application to a Grid-Refinement Study," *Computers & Fluids*, vol. 18, no. 4, 1990, pp. 391-403.
- [29] Jameson, A. and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper 83-1929, 1983.
- [30] Jameson, A., "Solution of the Euler Equations by a Multigrid Method," *Applied Mathematics and Computation*, vol. 13, 1983, pp. 327-356.
- [31] Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," Tech. Rep. MAE Report 1743, Princeton University, Department of Mechanical and Aerospace Engineering, October, 4, 1985. Text of lecture given at 2nd European Conference on Multigrid Methods, Cologne.
- [32] Swanson, R. C. and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," tech. rep., ICASE Report 84-62, February, 1985.
- [33] Swanson, R. C., Turkel, E., and White, J. A., "An Effective Multigrid Method for High-Speed Flows," tech. rep., ICASE Report 91-56, July 1991.
- [34] von Lavante, E., Elmiligui, A., Cannizzaro, F. E., and Warda, H., "Simple Explicit Upwind Schemes for Solving Compressible Flows," in *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics, NNFM*, vol. 29, 1989, pp. 291-301.
- [35] van Leer, B., Tai, C. H., and Powell, K., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," AIAA Paper 89-1933-CP, 1989.
- [36] Jespersen, D. and Levit, C., "A Computational Fluid Dynamics Algorithm on a Massively Parallel Computer," AIAA Paper 89-1936, 1989. AIAA 9th Computational Fluid Dynamics Conference.
- [37] Saati, A. A., *Numerical Simulation of Compressible Couette Flow Stability: Non-Linear and Three-Dimensional Effects*. PhD thesis, Department of Aerospace Engineering Sciences, University of Colorado, 1991.
- [38] Lerat, A., "Une classe de schemas aux differences implicites pour les systemes hyperboliques de lois de conservation." *Comptes Rendus Academie Sciences Paris.*, vol. 228A, June 1979, pp. 1033-1036.
- [39] Lax, P. D. and Wendroff, B., "Systems of Conservation Laws," *Communications of Pure and Applied Mathematics*, vol. 13, 1960, pp. 217-237.
- [40] Martinelli, L., *Calculations of Viscous Flow with a Multigrid Method*. PhD thesis, MAE Dept., Princeton University, 1987.

- [41] Radespiel, R., Rossow, C., and Swanson, R. C., "Efficient Cell-Vertex Multigrid Scheme for the Three Dimensional Navier-Stokes Equations," *AIAA Journal*, vol. 28, no. 8, August 1990, pp. 1464-1472.
- [42] Elmiligui, A., Cannizzaro, F. E., Melson, N. D., and von Lavante, E., "A Three Dimensional Multigrid Multiblock Multistage Time Stepping Scheme for the Navier-Stokes Equations.," in *Proceedings of the Ninth GAMM-Conference on Numerical Methods in Fluid Mechanics, NNFM*, 1991.
- [43] von Lavante, E. and Gronner, J., "Semi-Implicit Schemes for Solving the Navier-Stokes Equations.," in *Proceedings of the Ninth GAMM-Conference on Numerical Methods in Fluid Mechanics, NNFM*, 1991.
- [44] Blazek, J., Kroll, J., Radespiel, R., and Rossow, C. C., "Upwind implicit residual smoothing method for multi-stage schemes," *AIAA Paper 91-1533-CP*, 1991.
- [45] Fedorenko, R. P., "A Relaxation Method for Solving Elliptic Difference Equations," *USSR Comp. Math. and Math. Physics*, vol. 1, 1961, pp. 922-927. (In Russian).
- [46] Fedorenko, R. P., "The Speed of Convergence of One Iteration Process," *USSR Comp. Math. and Math. Physics*, vol. 4, 1964, pp. 227-235. (In Russian).
- [47] Brandt, A., "Multi-Level Adaptive Solutions to Boundary Value Problems," *Mathematics of Computations*, vol. 31, no. 138, 1977, pp. 333-390.
- [48] Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," *AIAA Paper 79-1458*, July, 1979.
- [49] Sankar, N. L., "A Multigrid Strongly Implicit Procedure for Two-Dimensional Transonic Potential Flow Problems," *AIAA Paper 82-0931*, 1982.
- [50] Thames, F. C., "Multigrid Applications to Three-Dimensional Elliptic Coordinate Generation," *Applied Mathematics and Computation*, vol. 15, no. 4, 1984, pp. 325-342.
- [51] Swanson, R. C. and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," *AIAA Paper 87-1107-CP*, June 9-11, 1987. *AIAA 8th Computational Fluid Dynamics Conference*, Honolulu, Hawaii.
- [52] Melson, N. D., "Vectorizable multigrid algorithms for transonic flow calculations," Master's thesis, The George Washington University, 1985.
- [53] Anderson, W. K., *Implicit Multigrid Algorithms for the Three-Dimensional Flux Split Euler Equations*. PhD thesis, Mississippi State, 1986.

- [54] Cannizzaro, F. E., *Runge-Kutta Upwind Multigrid Multi-Block Three-Dimensional Thin Layer Navier-Stokes Solver*. PhD thesis, MEM Department, Old Dominion University, 1992.
- [55] 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, no. 11, 1989, pp. 986-993.
- [56] Swanson, R. C. and Turkel, E., "On Central-Difference and Upwind Schemes.," tech. rep., ICASE Report 90-44, June, 1990.
- [57] Steger, J. L. and Warming, R. F., "Flux-Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite-Difference Methods," *Journal of Computational Physics*, vol. 40, 1981, pp. 263-293.
- [58] Anderson, W. K., Thomas, J. L., and van Leer, B., "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," AIAA Paper 85-0122, January 14-17, 1985. AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada.
- [59] van Leer, B., "Flux-Vector Splitting for the Euler Equations," tech. rep., ICASE Report 82-30, 1982.
- [60] Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "Split-Coefficient Matrix Method for Hyperbolic Systems of Gas Dynamic Equations," AIAA Paper 80-0268, January 1980.
- [61] Moretti, G., "The λ -Scheme," *Computers and Fluids*, vol. 7, 1979, pp. 191-205.
- [62] Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation," *Communications of Pure and Applied Mathematics*, vol. 7, 1954, pp. 159-193.
- [63] Lax, P. D. and Richtmyer, R. E., "Survey of the Stability of Linear Finite Difference Equations," *Communications of Pure and Applied Mathematics*, vol. 9, 1956, pp. 267-293.
- [64] Godunov, S. K., "A Difference Method for the Numerical Calculation of Discontinuous Solutions of Hydrodynamic Equations," *Mat. Sbornik*, 1959, pp. 271-306. Translated as JPRS 7225 by U.S. Dept. of Commerce, 1960.
- [65] Roe, P. L., "The Use of the Riemann Problem in Finite Difference Schemes," in *Proceedings of the 7th International Conference on Numerical Methods in Fluid Dynamics* (Reynolds, W. C. and MacCormack, R. W., eds.), Springer, 1981, pp. 354-359. Stanford, 1980.
- [66] Roe, P. L., "Numerical Modeling of Shock Waves and Other Discontinuities," in *Proceedings of the IMA Conference on Numerical Methods in Aeronautical Fluid*

Dynamics (Roe, P. L., ed.), Academic Press, New York, 1982, pp. 211–243. Reading, 1981.

- [67] Lombard, G. K., Oliger, J., and Yang, J. Y., "A Natural Conservative Flux Difference Splitting for the Hyperbolic Equations of Gas Dynamics," AIAA Paper 82-0976, 1982.
- [68] Engquist, B. and Osher, S., "One-Sided Difference Approximations for Nonlinear Conservation Laws," *Mathematics of Computation*, vol. 36, 1981, pp. 321–351.
- [69] Osher, S. and Chakravarthy, S., "High Resolution Schemes and the Entropy Condition," *SIAM Journal of Numerical Analysis*, vol. 21, no. 5, October 1984, pp. 955–984.
- [70] Harten, A., Lax, P. D., and van Leer, B., "On upstream difference and godunov-type schemes for hyperbolic conservative laws.," *SIAM Review*, vol. 25, 1983, pp. 35–61.
- [71] Yee, H. C., "Upwind and Symmetric Shock-Capturing Schemes," tech. rep., NASA TM-89464, 1987.
- [72] Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," tech. rep., NASA TM 101088, 1989.
- [73] Thomas, J. L., Krist, S. T., and Anderson, W. K., "Navier-Stokes Computations of Vortical Flows Over Low-Aspect-Ratio Wings," *AIAA Journal*, vol. 28, no. 2, February 1990, pp. 205–212.
- [74] Morrison, J., "Flux Difference Split Scheme for Turbulent Transport Equations," AIAA Paper 90-5251, October 29-31, 1990. AIAA Second International Aerospace Planes Conference, Orlando, FL.
- [75] S., A.-H. K., "A Multiblock/Multizone Code (PAB 3D-v2) for the Three-Dimensional Navier-Stokes Equations: Preliminary Applications.," tech. rep., NASA CR-182032, September 1990.
- [76] Kandil, O. A., Wong, T. C., and Kandil, H., "Thin-Layer and Full Navier-Stokes, Locally-Conical and Three-Dimensional Asymmetric Solutions.," AIAA Paper 91-0547, January, 1991.
- [77] Kandil, O. A. and Chaung, H. A., "Influence of Numerical Dissipation on Computational Euler Equations for Vortex Dominated Flows.," *AIAA Journal*, vol. 25, no. 11, 1987, pp. 1426–1434.
- [78] "Spurious Entropy Production and very Accurate Solutions to the Euler Equations.," AIAA Paper 84-1644, 1984. AIAA 17th Fluid Dynamics, Plasma Dynamics, and Laser Conference.

- [79] Pulliam, T. H., "Artificial Dissipation Models for the Euler Equations," *AIAA Journal*, vol. 24, no. 12, December, 1986, pp. 1931-1940.
- [80] Thompson, J. F., *Numerical Grid Generation: Foundations and Applications*. North-Holland, New York, NY., 1985.
- [81] Biedron, R. T. and Thomas, J. L., "A Generalized Patched-Grid Algorithm with Application to the F-18 Forebody with Actuated Control Strake," *Computing Systems in Engineering*, vol. 1, no. 2-4, 1990, pp. 563-576.
- [82] Bayasal, O., Fouladi, K., and Lessard, V. R., "A Multigrid Method to solve 3-D Viscous Equations on Overlapped and Embedded Grids," AIAA Paper 89-0464, 1989. AIAA 27th Aerospace Sciences Meeting, Reno, Nevada.
- [83] Benek, J. A., Buning, P. G., and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper 85-1523-CP, July, 1985.
- [84] Rai, M. M., Chakravarthy, S. R., and Hessenius, K. A., "Zonal Grid Calculations Using the Osher Scheme," *International Journal of Computers and Fluids*, vol. 12, no. 3, 1984, pp. 161-173.
- [85] Cannizzaro, F. E., Elmilgui, A., Melson, N. D., and von Lavante, E., "A Multiblock Multigrid Method for the Solution of the Three-Dimensional Euler Equations," AIAA paper number 90-0105, Presented at the AIAA 28th Aerospace Sciences Meeting, 1990.
- [86] Batina, J. T., "Three dimensional flux-split euler schemes involving unstructured dynamic meshes," AIAA 90-1649, June, 1990.
- [87] Frink, N. T., *Three-Dimensional Upwind Scheme for Solving The Euler Equations on Unstructured Tetrahedral Grids*. PhD thesis, Virginia Polytechnic Institute and State University, September 1991.
- [88] Mavriplis, D. J., "Three Dimensional Unstructured Multigrid for the Euler Equations," AIAA 91-1549, June 1991.
- [89] Anderson, W. K., "Grid generation and flow solution method for euler equations on unstructured grids," tech. rep., NASA TM 4295, April 1992.
- [90] von Lavante, E., "Accuracy of Upwind Schemes Applied to the Navier-Stokes Equations," *AIAA Journal*, vol. 28, no. 7, July 1990, pp. 1312-1314.
- [91] Malvern, L. E., *Introduction to the Mechanics of a Continuous Medium*. Englewood Cliffs, N.J., Prentice-Hall, 1969.
- [92] Hoffmann, K. L., *Computational Fluid Dynamics For Engineers*. Engineering Education System, Austin Texas, 1990.

- [93] Schlichting, H., *Boundary-Layer Theory*. McGRAW-Hill Book Company, 1979.
- [94] White, F. M., *Viscous Fluid Flow*. McGRAW-Hill, 1974.
- [95] "Equations, Tables, and Charts for Compressible Flow," tech. rep., NASA TR-1135, 1953.
- [96] Ghaffari, F., Bates, B., Luckring, J., and Thomas, J., "Navier-Stokes Solutions about the F/A-18 Forebody-LEX Configuration," AIAA 89-0338, January 1989. 27th Aerospace Science Meeting.
- [97] Moin, P., "The Computation of Turbulence," *Aerospace America*, January 1992, pp. 42-46.
- [98] Hinze, J. O., *Turbulence*. McGraw-Hill Book Company, Inc., 1987.
- [99] Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, January 16-18, 1978. AIAA 16th Aerospace Sciences Meeting.
- [100] Kuruwila, G. and Salas, M. D., "Three-Dimensional Simulation of Vortex Break-down," tech. rep., NASA TM-102664, June 1990.
- [101] Chakravarthy, S. R., "Development of Upwind Schemes for the Euler Equations," tech. rep., NASA CR-4043, January 1987.
- [102] Roe, P. L., "Some Contributions to the Modelling of Discontinuous Flow," *Lectures in Applied Mathematics*, vol. 22, 1985, pp. 163-193.
- [103] Sweby, P. K., "High resolution TVD Schemes Using Flux Limiters," *Lectures in Applied Mathematics*, vol. 22, 1985, pp. 289-309.
- [104] van Leer, B., "Toward the Ultimate Conservative Difference Scheme, II, Monotonicity and Conservation Combined in a Second-Order Scheme," *Lectures Notes in Physics*, vol. 14, 1982, pp. 3361-376.
- [105] Rumsey, C. L., *Development of a Grid-Independent Approximate Riemann Solver*. PhD thesis, University of Michigan, 1991.
- [106] Oksuzoglu, H., "State Vector Splitting For the Euler Equations of Gas dynamics," AIAA 92-0326, January 1992. 30th Aerospace Science Meeting & Exhibit.
- [107] 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 87-1104, June 1987.
- [108] von Lavante, E. and Haertl, A., "Numerical Solutions of Euler Equations Using Simplified Flux Vector Splitting," AIAA Paper 85-1333, 1985.

- [109] Melson, N. D. and von Lavante, E., "Multigrid Acceleration of the Isenthalpic Form of the Compressible Flow Equations," Third Copper Mountain Conference on Multigrid Methods, April 6-10, 1987.
- [110] Cannizzaro, F. E., von Lavante, E., and Melson, N. D., "Calculations of Three-Dimensional Flows Using the Isenthalpic Euler Equations With Implicit Flux-Vector Splitting," in *Proceedings of the AIAA 6th Applied Aerodynamics Conference*, AIAA Paper 88-2516, 1988, pp. 593-614.
- [111] Roe, P. L., "A Survey of Upwind Differencing Techniques," in *Lecture Notes in Physics*, vol. 323, Springer-Verlag, June 1988, pp. 69-78. 11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, Virginia.
- [112] Gaffney, J. R. L., *An abbreviated Reynolds Stress Turbulence Model for Airfoil Flows*. PhD thesis, Department of Mechanical and Aerospace Engineering, North Carolina State University., 1990.
- [113] Vatsa, V. N., Turkel, E., and Abolhassani, J. S., "Extension of Multigrid Methodology to Supersonic/Hypersonic 3-D Viscous Flows," Fifth Copper Mountain Conference on Multigrid Methods, Copper Mountain, Colorado, March 31-April 5, 1991.
- [114] Turkel, E., Swanson, R. C., Vatsa, V. N., and White, J. A., "Multigrid for Hypersonic Viscous Two- and Three-Dimensional Flows," AIAA Paper 91-1572, June 24-26, 1991. AIAA 10th Computational Fluid Dynamics Conference.
- [115] Briggs, W. L., *A Multigrid Tutorial*. SIAM, 1987.
- [116] South Jr., J. C. and Brandt, A., "Application of a Multilevel Grid Method to Transonic Flow Calculations," in *Transonic Flow Calculations in Turbomachinery* (Adamson, T. C. and Platzler, M. C., eds.), Hemisphere Publications, 1977.
- [117] Melson, N. D., Cannizzaro, F. E., and Elmiligui, A., "A Multigrid Multiblock Programming Strategy," Fifth Copper Mountain Conference on Multigrid Methods, Copper Mountain, Colorado, March 31 - April 5, 1991.
- [118] Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equations Calculations," *Journal of Computational Physics*, vol. 62, February, 1986, pp. 472-503. 2.
- [119] Thomas, J. L., Walters, R. W., Reu, T., and Ghaffari, F., "A Patched-Grid Algorithm for Complex Configurations Directed Towards the F/A-18 Aircraft," AIAA Paper 89-0121, January 9-12, 1989. 27th Aerospace Sciences Meeting.
- [120] Yadlin, Y. and Caughey, D. A., "Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow," *AIAA Journal*, vol. 29, no. 5, March, 1990, pp. 712-719.

- [121] Kallinderis, Y., "Numerical Treatment of grid Interface for Viscous Flows," *Journal of Computational Physics*, vol. 98, 1992, pp. 129-144.
- [122] MacCormack, R. W. and Paullay, A. J., "The Influence of the Computational Mesh on Accuracy for the Initial Value Problems with Discontinuous or Nonunique Solutions.," *Computers and Fluids*, vol. 2, 1974, pp. 339-361.
- [123] Thomas, J. L. and Salas, M. D., "Far-field Boundary Conditions for transonic lifting Solutions to the Euler Equations.," AIAA 85-0020, January 1985. 23rd Aerospace Sciences Meeting, Reno, Nevada.
- [124] Charwat, A. F. and Redekeopp, L. G., "Supersonic Interference Flow along the Corner of Intersecting Wedges," *AIAA Journal*, vol. 5, no. 3, March 1967, pp. 480-488.
- [125] West, J. E. and Korkegi, R. H., "Supersonic Interaction in the Corner of Intersecting Wedges at High Reynolds Numbers," *AIAA Journal*, vol. 10, no. 5, May 1972, pp. 652-656.
- [126] Kutler, P., "Numerical Solution for the Inviscid Supersonic Flow in the Corner Formed by Two Intersecting Wedges," AIAA Paper 73-675, July 16-18, 1973. AIAA 6 Fluid and Plasma Dynamics Conference, Palm Springs, California.
- [127] Marconi, F., "Internal Corner Flow Fields," AIAA paper number 79-0014, Presented at the AIAA 17th Aerospace Sciences Meeting, 1979.
- [128] Barger, R. L. and Melson, N. D., "Comparison of Jet Plume Shape Predictions and Plume Influence on Sonic Boom Signature.," tech. rep., NASA TP-3172, March 1992.
- [129] Re, R. J. and Leavitt, L. D., "Static Internal Performance of Single-Expansion-Ramp Nozzles with Various Combinations of Internal Geometric Parameters.," tech. rep., NASA TM-86270, December 1984.
- [130] Vatsa, V. N., "Accurate Numerical Solutions for transonic Viscous Flow over Finite Wings," *Journal of Aircraft*, vol. 24, no. 6, June 1987, pp. 377-385.
- [131] Bonhaus, D. L. and Wornom, S. F., "Comparison of Two Navier-Stokes Codes for Attached Transonic Wing Flows," *Journal of Aircraft*, vol. 29, no. 1, January-February 1992, pp. 101-107.
- [132] Marx, Y. P., "Numerical Solution of 3D Navier-Stokes Equations with Upwind Implicit Schemes," tech. rep., NASA TM-101656, January 1990.
- [133] Radespiel, R., "A Cell-Vertex Multigrid Method for the Navier-Stokes Equations.," tech. rep., NASA TM-101557, January 1990.
- [134] Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers," No. 138, AGARD, May 1979, pp. B2-1-B2-61.

- [135] NASA Conference Publication 2462, *Ground-Effects Workshop*, August 20, 1985. Workshop held at NASA Ames Research Center, Moffet Field, California.
- [136] NASA Conference Publication 10008, *Ground Vortex Workshop*, April 22-23, 1987. Workshop held at NASA Ames Research Center, Moffet Field, California.
- [137] Jalamani, Z. A., van Dalsem, W. R., and Nakamura, S., "A Computational Study of the Impinging Region of an Unsteady Subsonic Jet.," AIAA Paper 90-1657, June, 1990.
- [138] Didden, N. and Ho, C., "Unsteady separation in a boundary layer produced by an impinging jet.," *Journal of Fluid Mechanics*, vol. 160, November, 1985, pp. 235-256.
- [139] Stewart, V. R., "An experimental investigation of the ground vortex created by a moving jet.," Tech. Rep. NASA CR-181841, NASA, Langley Research Center, July 1989.
- [140] Bower, W. W., "Computations of Three-Dimensional Impinging Jets Based on Reynolds Equations," AIAA 82-1024, June 1982. AIAA/ASME 3rd Joint Thermodynamics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri.
- [141] Stewart, V. R., Kuhn, R. E., and Walters, M. M., "Characteristics of the Ground Vortex developed by Various V/STOL Jets at Forward Speed.," AIAA Paper 83-2494, October, 1983. AIAA Aircraft Design, Systems and Technology Meeting, Fort Worth, Texas, October 1983.
- [142] van Dalsem, W. R., "Study of Jet in Ground Effect with CrossFlow Using the Fortified Navier-Stokes Scheme.," AIAA Paper 87-2279, August, 1987. AIAA 5th Applied Aerodynamics Conference, Monterey, California.
- [143] Abeloff, P., van Dalsem, W. R., and Dougherty, F. C., "Thermal Interaction Between an Impinging Hot Jet and a Conducting Solid Surface.," AIAA Paper 90-3010, August 20-22, 1990. AIAA 8th Applied Aerodynamics Conference, Portland, Oregon.
- [144] Bray, D. and K., K., "A Review of Impinging Jets in Crossflows — Experimental and Computation," AIAA paper number 92-0633, Presented at the AIAA 30th Aerospace Sciences Meeting, January 6-9, 1992.
- [145] Childs, R. E. and Nixon, D., "Unsteady Three-Dimensional Simulations of a VTOL Upwash Fountain.," AIAA paper number 86-0212, Presented at the AIAA 24th Aerospace Sciences Meeting, January 6-9, 1986.
- [146] Barata, M.J. M., Durao, D., and McGuirk, J., "Numerical Study of a Single Impinging Jets Through a Cross-Flow.," *Journal of Aircraft*, vol. 26, no. 11, November, 1989.
- [147] Cimbal, J. M., Billet, M. L., and P., G. D., "Experiments on the unsteady ground vortex.," Tech. Rep. NASA CR177566, NASA Ames Research Center, August, 1990.

- [148] Salas, M. D., "The Numerical Calculation of Inviscid Plume Flow Fields," AIAA Paper 74-523, June 17-19, 1974.
- [149] Wedan, B. W. Private Communication.
- [150] Abid, R., Vatsa, V. N., Johnson, D. A., and Wedan, B. W., "Prediction of Separated Transonic Wing Flows with a Non-Equilibrium Algebraic Model," AIAA 89-0558, January 1989. 27th Aerospace Sciences Meeting.
- [151] Childs, R., "Turbulence Modeling for Impinging Jets.," AIAA paper number 90-0022, Presented at the AIAA 28th Aerospace Sciences Meeting, January 8-11, 1990.
- [152] Bradbury, L. J. S., "The Impact of an Axisymmetric Jet onto a Normal Ground.," *Aeronautical Quart*, vol. 23, 1972, pp. 141-147.
- [153] Bower, W. W., Kotansky, D. R., and H., H. G., "Computations and Measurements of two Dimensional Turbulent Jet Impinging Flow Fields," in *Proceedings of the 1st International Symposium on Turbulent Shear Flows, Pennsylvania State University, University Park, Pennsylvania, 1977*, pp. 3.1-3.8.
- [154] Rubel, A., "Computations of Jet Impingement on a Flat Surface.," *AIAA Journal*, vol. 18, no. 2, February 1978, pp. 168-175.

Appendix A

Full Navier-Stokes Equations in Body-Fitted Coordinates

The time dependent compressible three-dimensional Navier-Stokes equations in general curvilinear coordinates, written in strong conservation form (neglecting the body forces and the external heat sources are [6]

$$\frac{\partial Q}{\partial t} + \frac{\partial \{F - F_v\}}{\partial \xi} + \frac{\partial \{G - G_v\}}{\partial \eta} + \frac{\partial \{H - H_v\}}{\partial \zeta} = 0 \quad (\text{A.1})$$

where Q is the vector of dependent variables and is given by ;

$$Q = \frac{1}{J} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} \quad (\text{A.2})$$

The inviscid fluxes F , G , and H are function from the state vector Q . They are given by :

$$F = \begin{pmatrix} \rho U \\ \rho u U + p \xi_x \\ \rho v U + p \xi_y \\ \rho w U + p \xi_z \\ (E + p)U \end{pmatrix}, \quad G = \begin{pmatrix} \rho V \\ \rho u V + p \eta_x \\ \rho v V + p \eta_y \\ \rho w V + p \eta_z \\ (E + p)V \end{pmatrix}, \quad H = \begin{pmatrix} \rho W \\ \rho u W + p \zeta_x \\ \rho v W + p \zeta_y \\ \rho w W + p \zeta_z \\ (E + p)W \end{pmatrix} \quad (\text{A.3})$$

where U , V , and W are the contravariant velocity components in the ξ , η and ζ . They are given by ,

$$\begin{aligned} U &= u \xi_x + v \xi_y + w \xi_z \\ V &= u \eta_x + v \eta_y + w \eta_z \\ W &= u \zeta_x + v \zeta_y + w \zeta_z \end{aligned} \quad (\text{A.4})$$

The viscous flux, F_v is given by $F_v = \{F_{v_1}, F_{v_2}, F_{v_3}, F_{v_4}, F_{v_5}\}$, where,

$$F_{v_1} = [0] \quad (\text{A.5})$$

$$F_{v_2} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi(\frac{4}{3}\xi_x^2 + \xi_y^2 + \xi_z^2) + u_\eta(\frac{4}{3}\eta_x\xi_x + \eta_y\xi_y + \eta_z\xi_z) + \\ u_\zeta(\frac{4}{3}\zeta_x\xi_x + \zeta_y\xi_y + \zeta_z\xi_z) + \\ v_\xi\frac{1}{3}\xi_y\xi_x + v_\eta(\eta_x\xi_y - \frac{2}{3}\eta_y\xi_x) + v_\zeta(\zeta_x\xi_y - \frac{2}{3}\zeta_y\xi_x) + \\ w_\xi\frac{1}{3}\xi_z\xi_x + w_\eta(\eta_x\xi_z - \frac{2}{3}\eta_z\xi_x) + w_\zeta(\zeta_x\xi_z - \frac{2}{3}\zeta_z\xi_x) \end{bmatrix} \quad (\text{A.6})$$

$$F_{v_3} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi\frac{1}{3}\xi_x\xi_y + u_\eta(\eta_y\xi_x - \frac{2}{3}\eta_x\xi_y) + u_\zeta(\zeta_y\xi_x - \frac{2}{3}\zeta_x\xi_y) + \\ v_\xi(\xi_x^2 + \frac{4}{3}\xi_y^2 + \xi_z^2) + v_\eta(\eta_x\xi_x + \frac{4}{3}\eta_y\xi_y + \eta_z\xi_z) + \\ v_\zeta(\zeta_x\xi_x + \frac{4}{3}\zeta_y\xi_y + \zeta_z\xi_z) + \\ w_\xi\frac{1}{3}\xi_z\xi_y + w_\eta(\eta_y\xi_z - \frac{2}{3}\eta_z\xi_y) + w_\zeta(\zeta_y\xi_z - \frac{2}{3}\zeta_z\xi_y) \end{bmatrix} \quad (\text{A.7})$$

$$F_{v_4} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi\frac{1}{3}\xi_x\xi_z + u_\eta(\eta_z\xi_x - \frac{2}{3}\eta_x\xi_z) + u_\zeta(\zeta_z\xi_x - \frac{2}{3}\zeta_x\xi_z) + \\ v_\xi\frac{1}{3}\xi_y\xi_z + v_\eta(\eta_z\xi_y - \frac{2}{3}\eta_y\xi_z) + v_\zeta(\zeta_z\xi_y - \frac{2}{3}\zeta_y\xi_z) + \\ w_\xi(\xi_x^2 + \xi_y^2 + \frac{4}{3}\xi_z^2) + w_\eta(\eta_x\xi_x + \eta_y\xi_y + \frac{4}{3}\eta_z\xi_z) + \\ w_\zeta(\zeta_x\xi_x + \zeta_y\xi_y + \frac{4}{3}\zeta_z\xi_z) \end{bmatrix} \quad (\text{A.8})$$

$$F_{v_5} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{array}{l} u_\xi \left[u \left(\frac{1}{3}\xi_x^2 + \phi^2 \right) + \frac{1}{3}v\xi_y\xi_x + \frac{1}{3}w\xi_z\xi_x \right] + \\ u_\eta \left[\begin{array}{l} u \left(\frac{4}{3}\eta_x\xi_x + \eta_y\xi_y + \eta_z\xi_z \right) + v \left(\eta_y\xi_x - \frac{2}{3}\eta_x\xi_y \right) + \\ w \left(\eta_z\xi_x - \frac{2}{3}\eta_x\xi_z \right) \end{array} \right] + \\ u_\zeta \left[\begin{array}{l} u \left(\frac{4}{3}\zeta_x\xi_x + \zeta_y\xi_y + \zeta_z\xi_z \right) + v \left(\zeta_y\xi_x - \frac{2}{3}\zeta_x\xi_y \right) + \\ w \left(\zeta_z\xi_x - \frac{2}{3}\zeta_x\xi_z \right) \end{array} \right] + \\ v_\xi \left[\frac{1}{3}u\xi_x\xi_y + v \left(\frac{1}{3}\xi_y^2 + \phi^2 \right) + \frac{1}{3}w\xi_z\xi_y \right] + \\ v_\eta \left[\begin{array}{l} u \left(\eta_x\xi_y - \frac{2}{3}\eta_y\xi_x \right) + v \left(\eta_x\xi_x + \frac{4}{3}\eta_y\xi_y + \eta_z\xi_z \right) + \\ w \left(\eta_z\xi_y - \frac{2}{3}\eta_y\xi_z \right) \end{array} \right] + \\ v_\zeta \left[\begin{array}{l} u \left(\zeta_x\xi_y - \frac{2}{3}\zeta_y\xi_x \right) + v \left(\zeta_x\xi_x + \frac{4}{3}\zeta_y\xi_y + \zeta_z\xi_z \right) + \\ w \left(\zeta_z\xi_y - \frac{2}{3}\zeta_y\xi_z \right) \end{array} \right] + \\ w_\xi \left[\frac{1}{3}u\xi_x\xi_z + \frac{1}{3}v\xi_y\xi_z + w \left(\frac{1}{3}\xi_z^2 + \phi^2 \right) \right] + \\ w_\eta \left[\begin{array}{l} u \left(\eta_x\xi_z - \frac{2}{3}\eta_z\xi_x \right) + v \left(\eta_y\xi_z - \frac{2}{3}\eta_z\xi_y \right) + \\ w \left(\eta_x\xi_x + \eta_y\xi_y + \frac{4}{3}\eta_z\xi_z \right) \end{array} \right] + \\ w_\zeta \left[\begin{array}{l} u \left(\zeta_x\xi_z - \frac{2}{3}\zeta_z\xi_x \right) + v \left(\zeta_y\xi_z - \frac{2}{3}\zeta_z\xi_y \right) + \\ w \left(\zeta_x\xi_x + \zeta_y\xi_y + \frac{4}{3}\zeta_z\xi_z \right) \end{array} \right] + \\ \sigma \left[T_\xi\phi^2 + T_\eta(\eta_x\xi_x + \eta_y\xi_y + \eta_z\xi_z) + T_\zeta(\zeta_x\xi_x + \zeta_y\xi_y + \zeta_z\xi_z) \right] \end{array} \right] \quad (A.9)$$

The viscous flux, G_v is given by $G_v = \{G_{v_1}, G_{v_2}, G_{v_3}, G_{v_4}, G_{v_5}\}$ where,

$$G_{v_1} = [0] \quad (A.10)$$

$$G_{v_2} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{array}{l} u_\xi \left(\frac{4}{3}\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z \right) + u_\eta \left(\frac{4}{3}\eta_x^2 + \eta_y^2 + \eta_z^2 \right) + \\ u_\zeta \left(\frac{4}{3}\zeta_x\eta_x + \zeta_y\eta_y + \zeta_z\eta_z \right) + \\ v_\xi \left(\xi_x\eta_y - \frac{2}{3}\xi_y\eta_x \right) + v_\eta \frac{1}{3}\eta_x\eta_y + v_\zeta \left(\xi_x\eta_y - \frac{2}{3}\xi_y\eta_x \right) + \\ w_\xi \left(\xi_x\eta_z - \frac{2}{3}\xi_z\eta_x \right) + w_\eta \frac{1}{3}\eta_x\eta_z + w_\zeta \left(\xi_x\eta_z - \frac{2}{3}\xi_z\eta_x \right) \end{array} \right] \quad (A.11)$$

$$G_{v_3} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi(\xi_y\eta_x - \frac{2}{3}\xi_x\eta_y) + u_\eta\frac{1}{3}\eta_y\eta_x + u_\zeta(\zeta_y\eta_x - \frac{2}{3}\zeta_x\eta_y) + \\ v_\xi(\xi_x\eta_x + \frac{4}{3}\xi_y\eta_y + \xi_z\eta_z) + v_\eta(\eta_x^2 + \frac{4}{3}\eta_y^2 + \eta_z^2) + \\ v_\zeta(\zeta_x\eta_x + \frac{4}{3}\zeta_y\eta_y + \zeta_z\eta_z) + \\ w_\xi(\xi_y\eta_z - \frac{2}{3}\xi_z\eta_y) + w_\eta\frac{1}{3}\eta_y\eta_z + w_\zeta(\zeta_y\eta_z - \frac{2}{3}\zeta_z\eta_y) \end{bmatrix} \quad (A.12)$$

$$G_{v_4} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi(\xi_z\eta_x - \frac{2}{3}\xi_x\eta_z) + u_\eta\frac{1}{3}\eta_z\eta_x + u_\zeta(\zeta_z\eta_x - \frac{2}{3}\zeta_x\eta_z) + \\ v_\xi(\xi_z\eta_y - \frac{2}{3}\xi_y\eta_z) + v_\eta\frac{1}{3}\eta_z\eta_y + v_\zeta(\zeta_z\eta_y - \frac{2}{3}\zeta_y\eta_z) + \\ w_\xi(\xi_x\eta_x + \xi_y\eta_y + \frac{4}{3}\xi_z\eta_z) + w_\eta(\eta_x^2 + \eta_y^2 + \frac{4}{3}\eta_z^2) + \\ w_\zeta(\zeta_x\eta_x + \zeta_y\eta_y + \frac{4}{3}\zeta_z\eta_z) \end{bmatrix} \quad (A.13)$$

$$G_{v_5} = \frac{M_{ref}\mu}{R_{ref}} \begin{bmatrix} u_\xi \left[u(\frac{4}{3}\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z) + v(\xi_y\eta_x - \frac{2}{3}\xi_x\eta_y) + w(\xi_z\eta_x - \frac{2}{3}\xi_x\eta_z) \right] + \\ u_\eta \left[u(\frac{1}{3}\eta_x^2 + \theta^2) + \frac{1}{3}v\eta_y\eta_x + \frac{1}{3}w\eta_z\eta_x \right] + \\ u_\zeta \left[u(\frac{4}{3}\zeta_x\eta_x + \zeta_y\eta_y + \zeta_z\eta_z) + v(\zeta_y\eta_x - \frac{2}{3}\zeta_x\eta_y) + w(\zeta_z\eta_x - \frac{2}{3}\zeta_x\eta_z) \right] + \\ v_\xi \left[u(\xi_x\eta_y - \frac{2}{3}\xi_y\eta_x) + v(\xi_x\eta_x + \frac{4}{3}\xi_y\eta_y + \xi_z\eta_z) + w(\xi_z\eta_y - \frac{2}{3}\xi_y\eta_z) \right] + \\ v_\eta \left[\frac{1}{3}u\eta_y\eta_x + v(\frac{1}{3}\eta_y^2 + \theta^2) + \frac{1}{3}w\eta_y\eta_z \right] + \\ v_\zeta \left[u(\zeta_x\eta_y - \frac{2}{3}\zeta_y\eta_x) + v(\zeta_x\eta_x + \frac{4}{3}\zeta_y\eta_y + \zeta_z\eta_z) + w(\zeta_z\eta_y - \frac{2}{3}\zeta_y\eta_z) \right] + \\ w_\xi \left[u(\xi_x\eta_z - \frac{2}{3}\xi_z\eta_x) + v(\xi_y\eta_z - \frac{2}{3}\xi_z\eta_y) + w(\xi_x\eta_x + \xi_y\eta_y + \frac{4}{3}\xi_z\eta_z) \right] + \\ w_\eta \left[\frac{1}{3}u\eta_z\eta_x + \frac{1}{3}v\eta_z\eta_y + w(\frac{1}{3}\eta_z^2 + \theta^2) \right] + \\ w_\zeta \left[u(\zeta_x\eta_z - \frac{2}{3}\zeta_z\eta_x) + v(\zeta_y\eta_z - \frac{2}{3}\zeta_z\eta_y) + w(\zeta_x\eta_x + \zeta_y\eta_y + \frac{4}{3}\zeta_z\eta_z) \right] + \\ \sigma [T_\xi(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z) + T_\eta\theta^2 + T_\zeta(\zeta_x\eta_x + \zeta_y\eta_y + \zeta_z\eta_z)] \end{bmatrix} \quad (A.14)$$

The viscous flux H_v is given by, $H_v = \{H_{v_1}, H_{v_2}, H_{v_3}, H_{v_4}, H_{v_5}\}$ where,

$$H_{v_1} = [0] \quad (\text{A.15})$$

$$H_{v_2} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{array}{l} u_\xi \left(\frac{4}{3}\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z \right) + u_\eta \left(\frac{4}{3}\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z \right) + \\ u_\zeta \left(\frac{4}{3}\zeta_x^2 + \zeta_y^2 + \zeta_z^2 \right) + \\ v_\xi \left(\xi_x\zeta_y - \frac{2}{3}\xi_y\zeta_x \right) + v_\eta \left(\eta_x\zeta_y - \frac{2}{3}\eta_y\zeta_x \right) + v_\zeta \frac{1}{3}\zeta_y\zeta_x + \\ w_\xi \left(\xi_x\zeta_z - \frac{2}{3}\xi_z\zeta_x \right) + w_\eta \left(\eta_x\zeta_z - \frac{2}{3}\eta_z\zeta_x \right) + w_\zeta \frac{1}{3}\zeta_z\zeta_x \end{array} \right] \quad (\text{A.16})$$

$$H_{v_3} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{array}{l} u_\xi \left(\xi_y\zeta_x - \frac{2}{3}\xi_x\zeta_y \right) + u_\eta \left(\eta_y\zeta_x - \frac{2}{3}\eta_x\zeta_y \right) + u_\zeta \frac{1}{3}\zeta_x\zeta_y + \\ v_\xi \left(\xi_x\zeta_x + \frac{4}{3}\xi_y\zeta_y + \xi_z\zeta_z \right) + v_\eta \left(\eta_x\zeta_x + \frac{4}{3}\eta_y\zeta_y + \eta_z\zeta_z \right) + \\ v_\zeta \left(\zeta_x^2 + \frac{4}{3}\zeta_y^2 + \zeta_z^2 \right) + \\ w_\xi \left(\xi_y\zeta_z - \frac{2}{3}\xi_z\zeta_y \right) + w_\eta \left(\eta_y\zeta_z - \frac{2}{3}\eta_z\zeta_y \right) + w_\zeta \frac{1}{3}\zeta_z\zeta_y \end{array} \right] \quad (\text{A.17})$$

$$H_{v_4} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{array}{l} u_\xi \left(\xi_z\zeta_x - \frac{2}{3}\xi_x\zeta_z \right) + u_\eta \left(\eta_z\zeta_x - \frac{2}{3}\eta_x\zeta_z \right) + u_\zeta \frac{1}{3}\zeta_z\zeta_x + \\ v_\xi \left(\xi_z\zeta_y - \frac{2}{3}\xi_y\zeta_z \right) + v_\eta \left(\eta_z\zeta_y - \frac{2}{3}\eta_y\zeta_z \right) + v_\zeta \frac{1}{3}\zeta_z\zeta_y + \\ w_\xi \left(\xi_x\zeta_x + \xi_y\zeta_y + \frac{4}{3}\xi_z\zeta_z \right) + w_\eta \left(\eta_x\zeta_x + \eta_y\zeta_y + \frac{4}{3}\eta_z\zeta_z \right) + \\ w_\zeta \left(\zeta_x^2 + \zeta_y^2 + \frac{4}{3}\zeta_z^2 \right) \end{array} \right] \quad (\text{A.18})$$

$$H_{v_5} = \frac{M_{ref}\mu}{R_{ref}} \left[\begin{aligned} & u_\xi \left[\frac{u(\frac{4}{3}\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z) + v(\xi_y\zeta_x - \frac{2}{3}\xi_x\zeta_y) +}{w(\xi_z\zeta_x - \frac{2}{3}\xi_x\zeta_z)} \right] + \\ & u_\eta \left[\frac{u(\frac{4}{3}\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z) + v(\eta_y\zeta_x - \frac{2}{3}\eta_x\zeta_y) +}{w(\eta_z\zeta_x - \frac{2}{3}\eta_x\zeta_z)} \right] + \\ & u_\zeta \left[u(\frac{1}{3}\zeta_x^2 + \varphi^2) + \frac{1}{3}v\zeta_y\zeta_x + \frac{1}{3}w\zeta_z\zeta_x \right] + \\ & v_\xi \left[\frac{u(\xi_x\zeta_y - \frac{2}{3}\xi_y\zeta_x) + v(\xi_x\zeta_x + \frac{4}{3}\xi_y\zeta_y + \xi_z\zeta_z) +}{w(\xi_z\zeta_y - \frac{2}{3}\xi_y\zeta_z)} \right] + \\ & v_\eta \left[\frac{u(\eta_x\zeta_y - \frac{2}{3}\eta_y\zeta_x) + v(\eta_x\zeta_x + \frac{4}{3}\eta_y\zeta_y + \eta_z\zeta_z) +}{w(\eta_z\zeta_y - \frac{2}{3}\eta_y\zeta_z)} \right] + \\ & v_\zeta \left[\frac{1}{3}u\zeta_y\zeta_x + v(\frac{1}{3}\zeta_y^2 + \varphi^2) + \frac{1}{3}w\zeta_z\zeta_y \right] + \\ & w_\eta \left[\frac{u(\eta_x\zeta_z - \frac{2}{3}\eta_z\zeta_x) + v(\eta_y\zeta_z - \frac{2}{3}\eta_z\zeta_x) +}{w(\eta_x\zeta_x + \eta_y\zeta_y + \frac{4}{3}\eta_z\zeta_z)} \right] + \\ & w_\zeta \left[\frac{1}{3}u\zeta_z\zeta_x + \frac{1}{3}v\zeta_z\zeta_y + w(\frac{1}{3}\zeta_z^2 + \varphi^2) \right] + \\ & \sigma [T_\xi(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z) + T_\eta(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z) + T_\zeta\varphi^2] \end{aligned} \right] \quad (A.19)$$

Appendix B

Derivation of the Reynolds-Averaged Navier-Stokes Equations

The Reynolds-Averaged Navier-Stokes equations are derived from the Navier-Stokes equations by decomposing the randomly changing flow variables into a mean and a fluctuating components. For compressible flow triple correlations involving density appears in the equation. Favré [98] suggested a mass weighted decomposition for compressible flows to avoid the triple correlation involving density. The following formulations were used to decompose the flow variables in the Navier-Stokes equations 2.1.

$$\begin{aligned}\tilde{f} &= \frac{\rho f}{\bar{\rho}}, \quad \tilde{u} = \frac{\rho u}{\bar{\rho}}, \quad \tilde{T} = \frac{\rho T}{\bar{\rho}}, \quad \tilde{H} = \frac{\rho H}{\bar{\rho}} \\ \text{where } u &= \tilde{u} + \hat{u}, \quad T = \tilde{T} + \hat{T}, \quad H = \tilde{H} + \hat{H} \\ \text{note } p &= \bar{p} + \hat{p}, \quad \rho = \bar{\rho} + \hat{\rho} \\ \text{where } \overline{(\bar{\rho} + \hat{\rho})\hat{f}} &= 0 \\ \text{but } \overline{\hat{f}} &\neq 0\end{aligned}\tag{B.1}$$

By substituting these quantities into the governing equations and averaging in time , we get the continuity equation

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial x_i} = 0.0\tag{B.2}$$

the momentum equations

$$\frac{\partial \{\bar{\rho} \tilde{u}_i\}}{\partial t} + \frac{\partial \{\bar{\rho} \tilde{u}_i \tilde{u}_j\}}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \{\bar{\tau}_{ij} - \bar{\rho} \hat{u}_i \hat{u}_j\}}{\partial x_j} + \bar{\rho} \frac{\partial \tilde{f}}{\partial x_i}\tag{B.3}$$

the energy equation

$$\frac{\partial \{ \tilde{E} \}}{\partial t} + \frac{\partial \{ \bar{\rho} \tilde{u}_i \tilde{H} + \overline{\rho \hat{u}_i \hat{H}} - \tilde{u}_j \bar{\tau}_{ij} - \overline{\tau_{ij} \hat{u}_i} - k \frac{\partial \bar{T}}{\partial x_i} \}}{\partial x_i} = 0.0 \quad (\text{B.4})$$

it can be shown that

$$\tilde{H} = \tilde{h} + \tilde{K} + K \quad (\text{B.5})$$

and

$$\hat{H} = \hat{h} + \tilde{u}_i \hat{u}_i + \hat{K} - K \quad (\text{B.6})$$

with

$$\bar{\rho} K = \frac{\overline{\rho \hat{u}_i \hat{u}_i}}{2} = \overline{\rho \hat{K}}, \quad \text{and} \quad \hat{K} \equiv \frac{\hat{u}_i \hat{u}_i}{2} \quad (\text{B.7})$$

where K is the average of the turbulent Kinetic Energy of the Turbulent fluctuations which by definition is \hat{K} . Hence

$$\overline{\rho \hat{u}_i \hat{H}} = \overline{\rho \hat{u}_i \hat{h}} - \tau_{im}^R \tilde{u}_m + \overline{\rho \hat{u}_i \hat{K}} \quad (\text{B.8})$$

where

$$\tau_{im}^R \equiv -\overline{\rho \hat{u}_i \hat{u}_m} \quad (\text{B.9})$$

and is defined as the Reynolds stress tensor. Substituting into the energy equation gives

$$\frac{\partial \{ \tilde{E} \}}{\partial t} + \frac{\partial}{\partial x_i} \left\{ \bar{\rho} \tilde{u}_i \tilde{H} + \overline{\rho \hat{u}_i \hat{h}} - k \frac{\partial \bar{T}}{\partial x_i} - \bar{\tau}_{ij} \tilde{u}_j - \tau_{im}^R \tilde{u}_m - \overline{\tau_{ij} \hat{u}_j} + \overline{\rho \hat{u}_i \hat{K}} \right\} = 0.0 \quad (\text{B.10})$$

A host of terms involving the fluctuations evolved and need to be modeled in order to solve the governing equations. These terms are

$$\overline{\rho \hat{u}_i \hat{h}}, \quad \tau_{ij}^R \tilde{u}_j, \quad \overline{\tau_{ij} \hat{u}_i}, \quad \overline{\rho \hat{u}_i \hat{K}} \quad (\text{B.11})$$

$\overline{\rho \hat{u}_i \hat{h}}$ is generally considered the turbulent heat flux, and is modeled as

$$\overline{\rho \hat{u}_i \hat{h}} = -k_T \nabla \bar{T} \quad (\text{B.12})$$

where k_T is the coefficient of turbulent thermal conductivity. τ_{ij}^R will be modeled using Boussinesq's approximation [98]:

$$\tau_{ij}^R = -\overline{\rho u_i u_j} = \mu_T \left[\frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j} - \frac{2}{3} \frac{\partial \tilde{u}_m}{\partial x_m} \delta_{ij} \right] - \frac{2}{3} \bar{\rho} K \delta_{ij} \quad (\text{B.13})$$

μ_T is the turbulent eddy viscosity and is related to k_T by

$$k_T = \frac{c_p \mu_T}{P_{rT}} \quad (\text{B.14})$$

P_{rT} is the turbulent Prandtl number and is equal to 0.9. τ_{ij}^R and $\tilde{\tau}_{ij}$ are combined together to give

$$\tilde{\tau}_{ij}^T = (\mu + \mu_T) \left[\frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j} - \frac{2}{3} \frac{\partial \tilde{u}_m}{\partial x_m} \delta_{ij} \right] - \frac{2}{3} \bar{\rho} K \delta_{ij} \quad (\text{B.15})$$

In the present work $\overline{\tau_{ij} u_j}$ and $\overline{\rho u_i K}$ are assumed to be small and are neglected. They are calculated in a higher order scheme.

Applying the same procedure to the equation of state we get

$$\bar{p} = (\gamma - 1) \left[\tilde{E} - \frac{\bar{\rho} \tilde{u}_i \tilde{u}_i}{2} - \bar{\rho} K \right] \quad (\text{B.16})$$

Thus, the Reynolds-Averaged Navier-Stokes equations in dimensional Cartesian form become;

$$\frac{\partial q}{\partial t} + \frac{\partial(f - f_v)}{\partial x} + \frac{\partial(g - g_v)}{\partial y} + \frac{\partial(h - h_v)}{\partial z} = 0 \quad (\text{B.17})$$

where

$$q = \begin{Bmatrix} \bar{\rho} \\ \bar{\rho} \tilde{u} \\ \bar{\rho} \tilde{v} \\ \bar{\rho} \tilde{w} \\ \tilde{E} \end{Bmatrix} \quad (\text{B.18})$$

$$f = \begin{Bmatrix} \bar{\rho} \tilde{u} \\ \bar{\rho} \tilde{u}^2 + \bar{p} \\ \bar{\rho} \tilde{u} \tilde{v} \\ \bar{\rho} \tilde{u} \tilde{w} \\ (\tilde{E} + \bar{p}) \tilde{u} \end{Bmatrix}, \quad f_v = (\mu + \mu_{\mathbf{T}}) \begin{Bmatrix} 0.0 \\ \frac{2}{3} \left[2 \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{v}}{\partial y} - \frac{\partial \tilde{w}}{\partial z} \right] + \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} \\ \frac{\partial \tilde{v}}{\partial x} + \frac{\partial \tilde{u}}{\partial y} \\ \frac{\partial \tilde{w}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} \\ \tilde{u} \frac{2}{3} \left[2 \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{v}}{\partial y} - \frac{\partial \tilde{w}}{\partial z} \right] + \tilde{v} \left(\frac{\partial \tilde{v}}{\partial x} + \frac{\partial \tilde{u}}{\partial y} \right) + \\ \tilde{w} \left(\frac{\partial \tilde{w}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} \right) + \tilde{u} \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} + \frac{(k + k_{\mathbf{T}})}{(\mu + \mu_{\mathbf{T}})} \frac{\partial \bar{T}}{\partial x} \end{Bmatrix} \quad (\text{B.19})$$

$$g = \begin{Bmatrix} \bar{\rho} \tilde{v} \\ \bar{\rho} \tilde{u} \tilde{v} \\ \bar{\rho} \tilde{v}^2 + \bar{p} \\ \bar{\rho} \tilde{v} \tilde{w} \\ (\tilde{E} + \bar{p}) \tilde{v} \end{Bmatrix}, \quad g_v = (\mu + \mu_{\mathbf{T}}) \begin{Bmatrix} 0.0 \\ \frac{\partial \tilde{u}}{\partial y} + \frac{\partial \tilde{v}}{\partial x} \\ \frac{2}{3} \left[2 \frac{\partial \tilde{v}}{\partial y} - \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{w}}{\partial z} \right] + \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} \\ \frac{\partial \tilde{v}}{\partial z} + \frac{\partial \tilde{w}}{\partial y} \\ \tilde{u} \left(\frac{\partial \tilde{u}}{\partial y} + \frac{\partial \tilde{v}}{\partial x} \right) + \tilde{v} \frac{2}{3} \left[2 \frac{\partial \tilde{v}}{\partial y} - \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{w}}{\partial z} \right] + \\ \tilde{w} \left(\frac{\partial \tilde{v}}{\partial z} + \frac{\partial \tilde{w}}{\partial y} \right) + \tilde{v} \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} + \frac{(k + k_{\mathbf{T}})}{(\mu + \mu_{\mathbf{T}})} \frac{\partial \bar{T}}{\partial y} \end{Bmatrix} \quad (\text{B.20})$$

$$h = \begin{Bmatrix} \bar{\rho} \tilde{w} \\ \bar{\rho} \tilde{u} \tilde{w} \\ \bar{\rho} \tilde{v} \tilde{w} \\ \bar{\rho} \tilde{w}^2 + \bar{p} \\ (\tilde{E} + \bar{p}) \tilde{w} \end{Bmatrix}, \quad h_v = (\mu + \mu_{\mathbf{T}}) \begin{Bmatrix} 0.0 \\ \frac{\partial \tilde{u}}{\partial z} + \frac{\partial \tilde{w}}{\partial x} \\ \frac{\partial \tilde{v}}{\partial z} + \frac{\partial \tilde{w}}{\partial y} \\ \frac{2}{3} \left[2 \frac{\partial \tilde{w}}{\partial z} - \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{v}}{\partial y} \right] + \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} \\ \tilde{u} \left(\frac{\partial \tilde{u}}{\partial z} + \frac{\partial \tilde{w}}{\partial x} \right) + \tilde{v} \left(\frac{\partial \tilde{v}}{\partial z} + \frac{\partial \tilde{w}}{\partial y} \right) + \\ \tilde{w} \frac{2}{3} \left[2 \frac{\partial \tilde{w}}{\partial z} - \frac{\partial \tilde{u}}{\partial x} - \frac{\partial \tilde{v}}{\partial y} \right] + \tilde{w} \frac{2\bar{\rho}K}{3(\mu + \mu_{\mathbf{T}})} + \frac{(k + k_{\mathbf{T}})}{(\mu + \mu_{\mathbf{T}})} \frac{\partial \bar{T}}{\partial z} \end{Bmatrix} \quad (\text{B.21})$$

the superscript “ \sim ” has been removed for clarity, and the lower case letters q , f , f_v , g , g_v , h , and h_v are used to identify these vectors are functions from the averaged variables.

In this study, the Baldwin-Lomax, algebraic-turbulence model was selected which model the eddy viscosity. Baldwin-Lomax model implements a simple algebraic expression for the eddy viscosity. The model is easy to implement and provide reasonable results for attached flows. For separated flows or for the purpose of investigating the turbulent behavior of fluid motion, it is recommended that a higher order models, such as the two equation models $K - \varepsilon$, $K - \omega$, and $K - \tau$ [155] or solve for the Reynolds stresses [6]. The Baldwin-Lomax model is implemented with the understanding of the limitations and restrictions of the model. Higher order models should be included in the code to give the generality and accuracy in solving turbulent problems.

There is no mechanism in an algebraic-turbulence model that can account for the turbulent kinetic energy, K . Thus it was dropped from the governing set of equations as well as from the equation of state. The Reynolds-Averaged Navier-Stokes equations in non-dimensional Cartesian form takes the following form ;

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial (\tilde{F} - \tilde{F}_v)}{\partial x} + \frac{\partial (\tilde{G} - \tilde{G}_v)}{\partial y} + \frac{\partial (\tilde{H} - \tilde{H}_v)}{\partial z} = 0.0 \quad (\text{B.22})$$

$$\tilde{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix},$$

$$\tilde{F} = \begin{Bmatrix} \rho \tilde{u} \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{Bmatrix}, \quad F_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_T) \left\{ \begin{array}{c} 0.0 \\ \frac{2}{3} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ u \frac{2}{3} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) + v \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \\ w \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \sigma \frac{\partial T}{\partial x} \end{array} \right\} \quad (\text{B.23})$$

$$\tilde{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E + p)v \end{Bmatrix}, \quad \tilde{G}_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_{\mathbf{T}}) \begin{Bmatrix} 0.0 \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{2}{3} \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ u \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + v \frac{2}{3} \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) + \\ w \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \sigma \frac{\partial T}{\partial y} \end{Bmatrix} \quad (\text{B.24})$$

$$\tilde{H} = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E + p)w \end{Bmatrix}, \quad \tilde{H}_v = \frac{M_{ref}}{R_{ref}}(\mu + \mu_{\mathbf{T}}) \begin{Bmatrix} 0.0 \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{2}{3} \left(2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\ u \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + v \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \\ w \frac{2}{3} \left(2 \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) + \sigma \frac{\partial T}{\partial z} \end{Bmatrix} \quad (\text{B.25})$$

where

$$\sigma = \frac{1}{\beta(\mu + \mu_{\mathbf{T}})} \left(\frac{\mu}{P_r} + \frac{\mu_{\mathbf{T}}}{P_{r\mathbf{T}}} \right) \quad (\text{B.26})$$

Appendix C Limiters

C.1 Minum-Modulus Limiter

The min-mod limiter is implemented in the present algorithm by rewriting the κ -scheme formulation, eq. (19), as follows

$$\begin{aligned} Q_{i+\frac{1}{2}}^L &= Q_i + \frac{\phi}{4} [(1 - \kappa) \hat{\nabla}_i + (1 + \kappa) \hat{\Delta}_i] \\ Q_{i+\frac{1}{2}}^R &= Q_{i+1} - \frac{\phi}{4} [(1 - \kappa) \hat{\nabla}_{i+1} + (1 + \kappa) \hat{\Delta}_{i+1}] \end{aligned} \quad (\text{C.1})$$

where $\hat{\nabla}_i$ and $\hat{\Delta}_i$, are defined as

$$\begin{aligned} \hat{\Delta}_i &= \minmod(\Delta_i, \beta \nabla_i) \\ \hat{\nabla}_i &= \minmod(\nabla_i, \beta \Delta_i) \\ \Delta_i &= Q_{i+1} - Q_i \\ \nabla_i &= Q_i - Q_{i-1} \end{aligned} \quad (\text{C.2})$$

where, Δ_i and ∇_i are the backward and forward differences, and β is the compression parameter given by

$$\beta = \frac{3 - \kappa}{1 - \kappa} \quad (\text{C.3})$$

The min-mod operator is

$$\minmod(x, y) = \text{sign}(x) \max\{0, \min|x \text{ sign}\{y\}, y \text{ sign}\{x|\}\} \quad (\text{C.4})$$

C.2 Vanalbda Limiter

The Vanalbda differentiable limiter modifies the κ —scheme formulation, eq. (19), as follows

$$\begin{aligned} Q_{i+\frac{1}{2}}^L &= Q_i + \frac{\phi S_i}{4} [(1 - \kappa S_i) \nabla_i + (1 + \kappa S_i) \Delta_i] \\ Q_{i+\frac{1}{2}}^R &= Q_{i+1} - \frac{\phi S_{i+1}}{4} [(1 - \kappa S_{i+1}) \nabla_{i+1} + (1 + \kappa S_{i+1}) \Delta_{i+1}] \end{aligned} \quad (\text{C.5})$$

where

$$S_i = \frac{2 \Delta_i \nabla_i + \epsilon}{\nabla_i^2 + \Delta_i^2 + \epsilon} \quad (\text{C.6})$$

and ϵ is a small number to prevent the division by zero in the region of zero gradients.

Appendix D

Multigrid Cycles

Three types of cycles have been employed in this study, V-Cycle, W-Cycle, and the Full Multigrid Cycle (FMG). Basically, a standard V-cycle can be broken into halves. The first half is the restriction part of the cycle going from the fine-grid through the coarser grids down to the coarsest grid. The second half is the prolongation part of the cycle going from the coarsest grid up to the finest grid. An example is shown in Fig. D.1 for a four level multigrid. The circles indicate when iterations are performed on the given

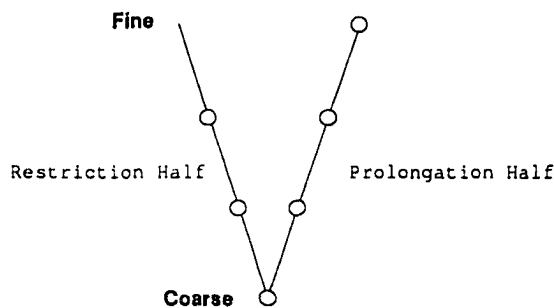


Figure D.1 V-Cycle.

grid level, and the lines between grid levels indicate either a restriction or prolongation operation. Notice that the circle for the fine-grid at the beginning of the cycle is omitted since the iterations on the fine-grid are performed at the end of the prolongation section. This ensures that the last operations in a multigrid cycle include updates on the fine-grid. It is often necessary to perform more than one iteration on a given grid to get the required smoothness in the error for multigrid to work.

A W-cycle can be thought of as consisting of several components which are similar to V-cycles but with different varying 'coarsest' and 'finest' grids. This idea is shown

in Fig. D.2 where a W-cycle is graphically expanded to show its 'legs'. This allows a simple coding modification to the V-cycle program to allow W-cycles.

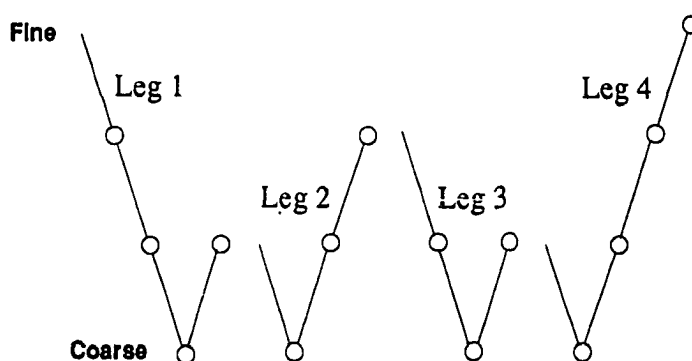


Figure D.2 W-Cycle.

The Full Multigrid Cycles (FMG) are used to get a good initial approximation on the fine-grid, which can be used as the starting solution for a V- or W-cycle. Figure D.3 shows the schematic of a FMG V-cycle, while the schematic of a FMG W-cycle is shown in Fig. D.4. The basic idea of a FMG cycle is to iterate first on the coarsest levels, assuming the finer level to be the solution level. The solution is then prolonged to the next finer level and the problem is solved again on these three levels. The process is repeated until the finest level is reached. The FMG cycles are inexpensive since we are solving the problem on a coarse-grid. By the time we include the finest grid in the FMG procedure and start to apply the regular V- or W-cycle, the global features of the solution has already been developed.

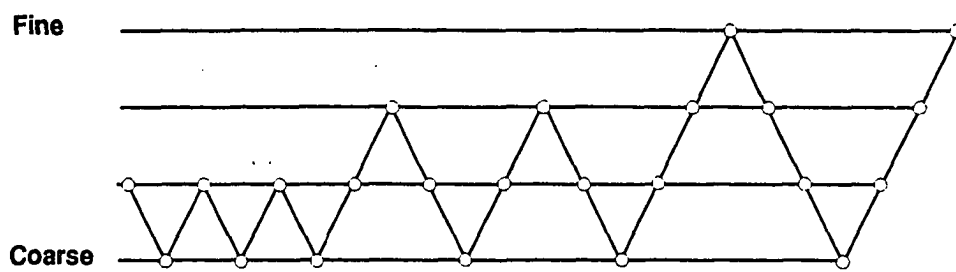


Figure D.3 Full Multigrid V-Cycle

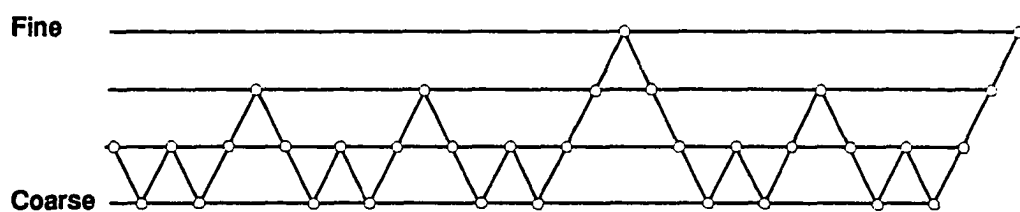


Figure D.4 Full Multigrid W-cycle