

Cranfield University

Tingrong Ren

**Multiple and Moving Bodies:
CFD Research in an Adaptive Environment**

College of Aeronautics

Ph.D. Thesis

ProQuest Number: 10820983

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10820983

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by Cranfield University.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346



**Cranfield University
College of Aeronautics**

Ph.D. Thesis

Academic Year 2000-01

Tingrong Ren

**Multiple and Moving Bodies:
CFD Research In an Adaptive Environment**

Supervisor: Prof. John A Edwards

November 2000

This thesis is submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy



To my parents and my family.

ABSTRACT

During this study a new solution adaptive scheme was developed which applied the structured Adaptive Mesh Refinement (AMR) algorithm directly into a Chimera grid system. Two new Computational Fluid Dynamics (CFD) methodologies, the Chimera grid method and the AMR algorithm, have been successfully integrated. The originality of this study is outlined below.

- 1). This new scheme integrated the Chimera grid method with the AMR algorithm. By using the Chimera grid method to mesh complicated configurations and then applying the AMR algorithm in different Chimera sub-domains to resolve complicated flow features the advantages of these two methods are combined.
- 2). Both Graph data structure and orthogonal list storage are employed in this new scheme to describing the Chimera sub-domain relationship and improve the region and point searching efficiency. These methods make this new scheme more flexible and more efficient than existing Chimera grid schemes and AMR schemes.
- 3). This study extended the application of the AMR algorithm. The main limitation of the Quirk's AMR code, poor geometric packing ability, has been replaced by a versatile Chimera grid scheme.
- 4). This study improved the accuracy of the Chimera inter-grid communication when simulating supersonic or hypersonic flows with strong discontinuities due to applying the AMR algorithm.
- 5). The combination of the versatile geometric packing ability of the Chimera grid method and the efficient computation ability of the AMR algorithm makes the current scheme capable of simulating multiple and moving body flow problem with strong discontinuities using moderate computational resources.

The material in this thesis documents this development of the Chimera grid scheme with an existing AMR code, and the validation and applications of this scheme.

Firstly, a Chimera grid scheme has been developed. Most of the tasks associated with the Chimera grid scheme are performed in a fully automated mode. The graph data

structure and orthogonal list storage are used to describe the domain relationships. This allows the sub-domains to be arranged in an arbitrary manner and increases efficiency.

Secondly, in order to properly integrate the Chimera grid scheme with the AMR, some modification work on the original AMR code had to be done. This work allowed the Chimera grid scheme to be successfully incorporated into the AMR code.

Thirdly, this scheme has been employed to solve time-dependent and steady state shock hydrodynamics problems at supersonic and hypersonic speed. Various validation and application cases, such as, shock reflection, shock diffraction, shock/boundary interaction and multiple moving body flows at supersonic speeds are presented, analysed and compared with experimental results or numerical solutions either from the literature or obtained by using another CFD code. The simulations demonstrate the flexibility of the grid generation and the high efficiency and capability to resolve complex flow features.

Finally, future research work arising from the present study has also been discussed and highlighted.

ACKNOWLEDGEMENTS

I would first like to express my deep sense of gratitude to my supervisor Prof. John A Edwards for suggesting this research work and for the guidance, encouragement, constructive criticism and support he has provided throughout this project.

Heartfelt thanks are also due to my academic supervisor Prof. N. Qin for his valuable advice and helpful discussions regarding the project.

I gratefully acknowledge the financial support and help from DERA. I feel very privileged to have worked on such a challenging research project, which extended my understanding of both computational fluid dynamics and aerodynamics.

Many thanks are also due to Dr. Jeff Roper and all my friends for their help throughout the period of this research. I also must especially thank Angus McDonald for his critically reading and correcting most of the text of this thesis.

Indeed, special thanks must go to my father Haiqing Ren and my mother Shulan Wang for their continuous encouragement, and to my wife Zehua Chen for her continuous support and encouragement.

Finally, I want to thank my daughter Wendy Ren for being so understanding when I had to work. I feel very sorry for having left her such little spare time.

Table of Contents

Abstract	i
Acknowledgements	iii
Table of Contents	iv
List of Figures	ix
List of Tables	xiii
Notation	xvi
1 Introduction	1
1.1 Background to the Research Project	1
1.1.1 Motivations	1
1.1.2 Research Objectives and Scope	3
1.1.3 Thesis Outline	5
1.2 Contributions of This Study	6
1.3 Literature Review	8
1.3.1 On the Chimera Grid Method	9
1.3.2 On the AMR Algorithm	14
2 A Brief Introduction to the AMR Algorithm	23
2.1 Introduction	23
2.2 Basic Concepts Behind the AMR Algorithm	23
2.2.1 Hierarchical Grid Structure	24
2.2.2 Local Refinement Strategy	25
2.2.3 Refinement Criterion	26
2.2.4 Clustering Algorithm	28
2.2.5 Flow integration	29
2.2.6 Summarisation of the Properties of the AMR Algorithm	30
2.3 Brief Introduction to the Quirk's AMR Code	31
2.3.1 Definition and Description of Grid System	32

2.3.2	Main Procedures	35
2.3.3	Code Structure	38
2.3.4	Flow Solver	40
2.4	Modification of the AMR Code	41
2.4.1	Extension to Multi-Domain	42
2.4.2	Replacement of the Bitwise Operation	43
2.5	Summary	44
3	Fully Automated 2D Chimera Grid Scheme	51
3.1	Introduction	51
3.2	Overview	51
3.2.1	Basic Concepts and Main Procedures	52
3.2.2	Desirable Properties	53
3.2.3	Code Structure Chart	54
3.3	Data Structure and Storage	54
3.4	Description of the Domain Relationship	55
3.4.1	Graph Data Structure	56
3.4.2	Orthogonal List Storage	57
3.5	Automated Hole-cutting Procedure	58
3.5.1	Implementation Sequence	58
3.5.2	Defining Hole-Cutting	59
3.5.3	Hole Cell Flagging Algorithm	60
3.5.4	Fringe Cell Flagging Algorithm	61
3.6	Inter-grid Communication	62
3.6.1	Recipient–Donor Stencil	62
3.6.2	Flow Variable Transfer	65
3.7	Moving Body Module	67
3.7.1	Dimensionless Motion Parameters	67

3.7.2	Prescribed Movement	69
3.7.3	Free Movement	69
3.7.4	Domain Movement	70
3.8	Summary	71
4	Integration of the Chimera Grid Scheme with the AMR Algorithm	79
4.1	Introduction	79
4.2	Combination Method	79
4.3	Modification of the AMR Code	80
4.3.1	Replacement of the Pre- and Post-Processors	81
4.3.2	Modification of the Flow Solver	83
4.3.3	Mesh Refinement Treatment	84
4.3.4	Time Step Treatment	85
4.3.5	Correction of the AMR Code	86
4.4	The Complete AMRC Scheme	88
4.4.1	Code Frame Structure	88
4.4.2	Code Implementation Sequence	89
4.5	Validation Case Study	90
4.5.1	Shock Reflection and Diffraction over a Cylinder	90
4.5.2	Computational Details	92
4.5.3	Simulations of Transition Events	94
4.5.4	Comparison with Experiment	95
4.5.5	Computational Efficiency	96
4.6	Investigation of the Chimera Boundary	98
4.6.1	Interpolation Error Analysis	98
4.6.2	Shock Wave/Domain Boundary Interaction	101
4.6.3	Influence of the Overlap Depth	102
4.7	Summary	103

5	Complex Shock Reflections from Aerofoil at High Angle of Attack	116
5.1	Introduction	116
5.2	Specification of the Simulation Problem	116
5.3	Numerical Simulation by Other Researchers	118
5.4	Current Simulation Details	119
5.5	Numerical Solution and Comparison	120
5.5.1	Shock Diffraction on a 90° Convex Corner	121
5.5.2	Oblique Shock Interaction with an Aerofoil	122
5.5.3	Vortex Interaction with an Aerofoil	125
5.6	Summary	127
6	Laminar Viscous Flow Computations	143
6.1	Introduction	143
6.2	Shock Wave/Laminar Boundary Layer Interaction	144
6.2.1	Specification of the Problem	144
6.2.2	Computational Details	146
6.2.3	Comparison with Experimental Results	149
6.2.4	Comparison with Flap Ramp Model	152
6.3	Shock Wave Interference on a Cylindrical Leading Edge	153
6.3.1	Specification of the Problem	154
6.3.2	Computational Details	154
6.3.3	Comparison with Experimental Results	157
6.4	Summary	161
7	Moving Body Flow Computations	180
7.1	Introduction	180
7.2	Multiple and Moving Body Flow Computations	180
7.2.1	Specification of the Problem	181
7.2.2	Steady State Solution	182

Table of Contents

7.2.3	Prescribed Motion	184
7.2.4	Impacting Moving Body flow	185
7.3	Freely Oscillating Aerofoil Flow Simulation	186
7.3.1	Specification of the Problem	186
7.3.2	Computational Details	187
7.3.3	Steady State Simulation	188
7.3.4	Unsteady State Simulation	190
7.4	Summary	191
8	Closure	206
8.1	Conclusions	206
8.2	Suggestions for Future Research Work	208
References		213
Appendix A Bilinear Interpolation		225
Appendix B Moving Shock Relationships		228

List of Figures

Figure 1-1 Grid adaption by the AMR and the Chimera method	19
Figure 1-2. Grid and solution for tear-drop geometry by the ASPaR method	20
Figure 1-3. Integration of the Chimera scheme and AMR algorithm	20
Figure 1-4. Composite structured grid methods	21
Figure 1-5. Finer overlap grids for a 15° ramp channel flow	21
Figure 1-6. Finer moving overlap grid for vortex tracking	21
Figure 1-7. Various adaptive mesh schemes	22
Figure 2-1. Hierarchical grid structure	46
Figure 2-2. Parts of a mesh	46
Figure 2-3. Abutting meshes with partially absorbed buffer regions	47
Figure 2-4. Local mesh refinement	47
Figure 2-5. Example of Berger's clustering algorithm	48
Figure 2-6. Transferring solution to the new finer meshes	49
Figure 2-7. Mesh nesting restriction	49
Figure 2-8. Mesh index	50
Figure 2-9. Different types of mesh boundary	50
Figure 2-10. Dummy cells surround each mesh	50
Figure 3-1 Chimera grid system	72
Figure 3-2, An example of a Chimera grid system	73
Figure 3-3. Chimera overset domain and Adaptive mesh	73
Figure 3-4. Chimera grid scheme structure chart	74
Figure 3-5. Restricted hierarchy by tree data structure	75
Figure 3-6. Description of domain relationships by Graph data structure	75

Figure 3-7. Orthogonal list storage	76
Figure 3-8. Define hole-cutting curve	76
Figure 3-9. Sketch of the ‘Non-Exterior Winding Role’	77
Figure 3-10. Nine-point stencil	77
Figure 3-11. Interpolation quadrilateral formed by neighbour cell	77
Figure 3-12. Interpolation quadrilateral formed by surrounding cell	78
Figure 3-13. Co-ordinate system	78
Figure 4-1. Integration of the Chimera scheme and AMR algorithm	104
Figure 4-2. Sub-domain type and arrangement	104
Figure 4-3. Mesh refinement in valid region	105
Figure 4-4. Solid boundary type	105
Figure 4-5. AMRC Code Structure	106
Figure 4-6. Chimera grid system and boundary conditions	106
Figure 4-7. Simulation results (density plots)	107
Figure 4-8. Transition point of RR to MR (enlarged density plot)	108
Figure 4-9. Transition from MR to DMR (enlarged)	108
Figure 4-10. Density plot comparison with experimental data	109
Figure 4-11. Comparison of the triple point locus	110
Figure 4-12. Enlarged grid system near Chimera hole and overset boundary	111
Figure 4-13. Enlarged density contour near Chimera hole and overset boundary	112
Figure 4-14. Enlarged density plot near Chimera hole and overset boundary	113
Figure 4-15. Enlarged density plot over a double wedge body	114
Figure 4-16. Chimera grid system (smaller overset domain 2)	114
Figure 4-17. Reflecting shock wave crosses hole boundary	115
Figure 4-18. Reflecting shock wave crosses overset boundary	115
Figure 5-1. Schematic of the experiment	128
Figure 5-2. Schematic of the computational domain in reference	128

Figure 5-3. Chimera grid system and boundary conditions	129
Figure 5-4. Simulation results around the convex corner	130
Figure 5-5. Comparison of density contours with the experimental Interferograms ... and simulation result from the literature	131
Figure 5-6. Comparison of density contours with the experimental Interferograms ... and simulation result from the literature	132
Figure 5-7. Comparison of the resolution	133
Figure 5-8. Pressure plot just before impinging the aerofoil	133
Figure 5-9. Line prove along the symmetric plane	134
Figure 5-10. Density plot of the oblique shock interaction with the aerofoil at 30° angle of attack	134
Figure 5-11. Comparison with the experimental Interferograms and simulation result from the literature	135
Figure 5-12. Vortex interaction with the target aerofoil	136
Figure 5-13. Vortex traveling to downstream (enlarged)	137
Figure 5-14. Enlarged numerical solution around NACA0018 aerofoil	138
Figure 5-15. Adaptive Meshes (with density) around the NACA0018 aerofoil	139
Figure 5-16. Flow variable profiles across vortex	140
Figure 5-17. Vortex interaction with the target aerofoil nose	142
Figure 6-1. Schematic of model of the experiment	160
Figure 6-2. Schematic of shock wave/laminar boundary layer interaction	160
Figure 6-3. Chimera grid system and boundary conditions	161
Figure 6-4. Computational flow structure (density plot)	161
Figure 6-5. Enlarged U velocity plot at x=75mm	161
Figure 6-6. Velocity vector inside boundary layer	162
Figure 6-7. Comparison of numerical solution resolution (density plots)	163
Figure 6-8. Enlarged grid system	164

Figure 6-9. Computational pressure distributions 165

Figure 6-10. Comparison of pressure distributions with Experimental results 165

Figure 6-11. Comparison of pressure distributions with Experimental results 167

Figure 6-12. Effects of free stream Reynolds number on pressure distributions 168

Figure 6-13. Comparison of pressure distributions with flat ramp model 169

Figure 6-14. Simulation results of convex ramp model 170

Figure 6-15. Simulation results of flat ramp model 171

Figure 6-16. Six types of shock wave interference patterns 172

Figure 6-17. Chimera grid system and boundary conditions 173

Figure 6-18. Grid system for simulating undisturbed bow wave 173

Figure 6-19. Comparison of the undisturbed flow 174

Figure 6-20. Computational Type IV Interference pattern 175

Figure 6-21. Comparison of the Type IV Interference pattern 176

Figure 6-22. Comparison of the pressure distributions of Type IV pattern 177

Figure 6-23. Computational Type III interference pattern 178

Figure 7-1. Double wedge aerofoil geometry dimensions 193

Figure 7-2. The Chimera grid system and boundary conditions 193

Figure 7-3. Steady state simulation results 194

Figure 7-4 Pressure coefficient distributions on #2 aerofoil 195

Figure 7-5. Comparison of streamlines, shock wave and expansions 196

Figure 7-6 Pressure coefficient distribution on #1 aerofoil 196

Figure 7-7 Moving body flow numerical solutions, Mach plots 197

Figure 7-8 Moving body flow numerical solutions, density plots 198

Figure 7-9. The Chimera grid system and boundary conditions 199

Figure 7-10. Steady state simulation results (impact flow) 199

Figure 7-11. Impact Moving body flow numerical solutions (Mach plots)..... 200

Figure 7-12. Chimera holes and adaptive meshes 201

Figure 7-13 Chimera grid system and boundary conditions 201

Figure 7-14. Computational Mach plots from the AMRC code 202

Figure 7-15. Computational Mach plots from the CFD-FASTRAN 203

Figure 7-16. Comparison of the pressure coefficients 204

Figure 7-17. Comparison of pitching moment coefficients 205

Figure 7-18. Aerofoil oscillation incidence vs time 205

List of Tables

Table 2-1. Quasi-code fragment to orchestrate the AMR algorithm	38
Table 2-2. Quasi-code fragment to co-ordinate the adaptation process	39
Table 2-3. Quasi-code fragment to the main program	39
Table 3-1. Quasi-code fragment to the Chimera Hole-cutting Procedure	59
Table 3-2. Quasi-code fragment to the Recipient–Donor Stencil Procedure-1	64
Table 3-3. Quasi-code fragment to the Recipient–Donor Stencil Procedure-2	65
Table 4-1. Wind tunnel test conditions (shock jump conditions)	91
Table 4-2. Grid adaptation parameters	93
Table 4-3 Run time information (one iteration)	97
Table 5-1. Grid adaptation parameters	119
Table 6-1. Wind tunnel test conditions	146
Table 6-2. Error bounds in the measurements	146
Table 6-3. Free stream conditions for ramp angle of 7.5° case	147
Table 6-4. Free stream conditions for ramp angle of 10° case	147
Table 6-5. Grid adaptation parameters	148
Table 6-6. Wind tunnel test conditions	154
Table 6-7. Flow physics and adaptive calculation setting	155
Table 6-8. Grid adaptation parameters for domain D ₁	156
Table 6-9. Free stream conditions	156
Table 7-1. Free stream conditions	181
Table 7-2. Grid adaptation parameters	182
Table 7-3. Comparison of pressure coefficients and Mach numbers	183
Table 7-4. Wind tunnel test conditions	187

Table 7-5. Free stream conditions 187

Table 7-6. Grid adaptation parameters 188

Notation

Acronyms

AMR	Adaptive Mesh Refinement
CFD	Computational Fluid Dynamics
CFL	Courant—Friedrichs—Lewy condition
MR	Mach Reflection
RR	Regular Reflection
CMR	Complex Mach Reflection
DMR	Double Mach Reflection
SMR	Simple Mach Reflection

Chimera Grid System

D	Overset sub-domain
G	Computational grid
Nd	Chimera sub-domain index
$Ndmax$	Total domain number of a Chimera grid system
$Icells$	Total cell number in a domain

AMR

G_l	Grid at level l
G_{pl}	Index for 1 st mesh at level l
$G_{l,k}$	k^{th} mesh at level l
GRD	Mesh index

i, j	Cell co-ordinate
IJ	Cell index in linked list.
l	Grid level
L_{max}	Highest grid level
rI	Refinement factor in I direction
rJ	Refinement factor in J direction

Interface Fluxes and Riemann Solver

A	Jacobian matrix
F	Conservative flux vector
M	Mach number
(n, t)	Local co-ordinate system normal and tangential to a cell interface
W	Flow solution

Physical Variables

$()_{\infty}$	Free stream reference conditions
a	Speed of sound (m/s)
E_t	Total Energy (J)
H	Total Enthalpy (J)
P	Static pressure (N/m ²)
Pr	Prandtl number
T	Temperature (°K)
Re	Reynolds number
(u, v)	Flow velocity in the x and y direction (m/s)
(x, y)	Physical co-ordinates (m)
α	Angle of attack. (°)
ρ	Static density (kg/m ³)

γ	Ratio of specific heats
λ	Coefficient of heat conduction (W/(m·K))
μ	Coefficient of viscosity (N·s/m ²)
τ	Time (s)

Body Movement

A	Surface area (m ²)
a	Acceleration (m/s ²)
d	Force arm length (m)
F_p	Force caused by surface pressure (N)
F_g	Force caused by gravity (N)
I	Moment of inertia (kg·m ²)
M	Moment (N·m)
m	Mass (kg)
U	Translation velocity in x direction (m/s)
V	Translation velocity in y direction (m/s)
α	Angular (rad)
ω	Angular velocity (rad/s)
$\dot{\omega}$	Angular acceleration (rad/s ²)

Moving Shock Relationships

$()_1$	Quiescent fluid
$()_2$	Post-shock fluid
M_s	Shock Mach number
U_s	Shock speed

Some notation, which is used infrequently, is labelled within the main body of the text.

Chapter 1

Introduction

This chapter introduces the background to the research project and the objectives of the present study, and presents a review of the Chimera grid method and the AMR algorithm. The contributions of this study are also highlighted and the structure of the thesis is summarised.

1.1 Background to the Research Project

1.1.1 Motivations

Computational Fluid Dynamics (CFD) is maturing and being widely used for both research and practical engineering problems. The ability to predict accurately and efficiently a steady or unsteady flow field about geometrically complex or moving component configurations or complicated flow structures is becoming more and more important for many applications. In simulations of supersonic and hypersonic flows, accurate predictions of viscous effects, moving shocks and aerodynamic interference between moving and stationary body components are particularly important. For example, the high fidelity analysis of the flow fields around modern aerospace vehicles often requires the accurate simulation of the unsteady, viscous flow fields. The task is further complicated when the relative motion of one or more components with respect to the others induces an unsteady flow field. This challenges the capability of a numerical scheme and physical models.

Clearly, practical configurations and complex flow features and body movement complicate the task considerably. These issues greatly increase the requirement for a numerical scheme with the following capabilities, besides the fundamental requirement of robustness.

- More efficient and versatile grid methods that can handle complex geometric configurations and multiple moving bodies.
- A high resolution flow solver that can lead to more accurate simulation results for viscous flows.
- Less computational expense.

To achieve each of these capabilities, various versatile grid methodologies and high-resolution schemes as well as adaptive mesh refinement strategies have been developed and considerable progress has been made by many previous researchers; these include efficient numerical schemes. The Chimera grid method, second-order Roe's scheme and the AMR algorithm have drawn wide attention.

The Chimera grid method is basically a domain de-composition technique which uses two or more simple overset sub-domains to mesh complex configurations¹⁻⁶. This grid method uses multiple overset domains to map the computational domain. It can yield easier grid generation, easier boundary condition application, better grid resolution. And most importantly this grid method can easily handle multiple and moving body problem.

The second-order Roe's scheme belongs to a class of characteristic based schemes that have been developed from the ideas of Godunov. It has proved to be very successful in calculating flows with strong shocks.

The Structured AMR algorithm is an automatic, adaptive mesh refinement strategy for solving hyperbolic conservation laws. It was proposed by Berger⁷ in the early 1980's and further developed by herself, many of her co-workers⁷⁻¹⁷ and Quirk¹⁸. This algorithm can greatly enhance local grid resolution and resolve complicated flow features, such as shock waves and boundary layers. Since the AMR algorithm performs high-resolution

computations only in regions that require it, about only 5~10 percent of computational resources compared to a non-adaptive calculation are required for a given resolution.

However, most research efforts focus on individual numerical approaches and deal with a specific flow problem. Relatively little research work has been done on the integration of these advanced CFD methodologies. To develop a powerful tool with the above capabilities has become more and more necessary for shock hydrodynamics and multiple moving body flow analysis. Based on this background, the current research project was proposed. The general motivation of this study is to integrate a flexible grid method (the Chimera grid method) with an efficient high resolution scheme (second-order Roe's scheme) as well as the AMR algorithm.

1.1.2 Research Objectives and Scope

The primary objectives of this study are to develop a fully automated Chimera grid scheme and then to integrate it with an existing AMR code¹⁸ (James Quirk's AMR code, available for application from Professor John A Edwards in the College of Aeronautics, Cranfield University). This AMR code employs the second order Roe's scheme (in both Operator Splitting and Finite-Volume modes) as the flow solver. At the completion of this project, it is aimed to have a solution adaptive Chimera scheme that not only enhances the resolution by using finer meshes in high gradient regions, but is also capable of handling complex geometry and moving body flows.

The development of this scheme is driven largely by expecting a powerful tool for multiple and moving body flow simulations at high speed. It is also desired that this code to be a general-purpose code, namely, the code itself is fully de-coupled from any simulation problem. The following is a list of the desirable properties, which will be included in our scheme.

- The scheme must have versatile geometric packing capabilities.
- The scheme must have a high resolution flow solver.

- A mesh refinement strategy must be used.
- The process must have a lower computational cost than existing methods.
- The scheme must be well suited to transient simulations.
- The scheme must be general purpose.

The present work is twofold. The first part is to develop and integrate the Chimera grid scheme into the structured AMR algorithm. In this way, it is desired to provide a versatile, high resolution, efficient tool for modelling practical engineering problems, typically, applications which are geometrically complex or with complicated flow features. The second part is to investigate steady and unsteady shock hydrodynamic flows and multiple moving body flows at supersonic and hypersonic speed using the current scheme.

The final goal of this thesis is to produce a solution adaptive Chimera grid scheme for modelling multiple moving body supersonic and hypersonic flows. To achieve this goal, the following research tasks need to be completed during this study.

- To carry out a literature review of the Chimera grid method and the AMR algorithm. To gain and attempt to employ new techniques and new ideas into the current research project.
- To improve and modify the original AMR code, to enable it to be applied into a Chimera grid system.
- To develop a fully automated Chimera grid scheme. This research in particular focuses on the problem of moving body flow simulations. Therefore this fully automated feature is one of the main considerations. In addition, to enable it to extend to three dimensions is also taken into account.
- To combine the Chimera grid scheme and the Roe's Flux Difference Split (FDS) scheme as well as the AMR code.
- To carry out code verification and validation.
- To investigate steady and unsteady shock hydrodynamic flows and multiple moving body flows at supersonic and hypersonic speed using the new scheme.

To verify the feasibility of using the Chimera grid approach to accurately simulate unsteady flows about multiple moving bodies.

1.1.3 Thesis Outline

This study can be naturally subdivided into three major parts. The first part concerns the modification and improvement of Quirk's original AMR code and the flow solver included in that code. The second part concerns the development of a fully automated Chimera grid scheme and the integration of the Chimera grid method and the AMR algorithm. The third part is to carry out code validation and investigate some applications, i.e. use this scheme to carry out simulating for complicated configurations and moving body flows at supersonic and hypersonic speed.

This thesis consists of eight chapters. Chapter 1 introduces the subject area, the research motivation behind the work and the objectives, the contributions of this study and the literature review. The purpose of this chapter is to lay down a clear set of motivations and objectives of this research project. Chapters 2, 3 and 4 will present the numerical methodologies employed in this research project and the modification of the AMR code and the development of a solution adaptive Chimera grid scheme. These three chapters are the technical core of this thesis. Chapter 2 provides an overview of the AMR algorithm and a brief introduction to Quirk's AMR code. It not only explains what the AMR algorithm is and how it works, but also presents the modifications and improvements in the current version of the AMR code. Chapter 3 describes the development of a fully automated Chimera grid scheme. Specifically, it covers the details of all the numerical methodologies and algorithms employed in this scheme. Chapter 4 focuses upon the integration of the Chimera grid scheme and the AMR algorithm, and the code validation. The applications are presented in Chapters 5, 6 and 7, in which I investigate shock hydrodynamics problems and multiple moving body flows. Chapter 8 summarises the conclusions from the research and outlines suggestions for future research work arising from this project.

1.2 Contributions of This Study

From the literature, we can find a couple of methods to fulfil the solution adaptive mesh refinement by using Chimera overset finer domain¹⁹⁻²⁰ and using the Adaptive Spatial Partitioning and Refinement (ASPaR) method²¹⁻²². Together with several adaptive mesh schemes, these have drawn wide attention from CFD analysts. In this section, I will discuss the difference between our scheme with the ASPaR method and some other schemes presented in literature.

Figure 1-1a shows a simple grid and the flagged region for refinement. Using the AMR scheme, a finer mesh is to be found to completely cover and overlay the flagged region, see Figure 1-1b. This method can well resolve complicated flow features in a flow field by using moderate computational resources. Moreover, under normal circumstances the AMR algorithm ensures that strong discontinuities are wholly encased within a single embedded grid which shadows their every movement. Thus a strong discontinuity is never given the chance to cross a fine-coarse mesh boundary (see §2.3.1 for mesh boundary definition). In this way, the conservation problem is circumvented. However, the relatively poor geometric packing capability of quasi-rectangular, body-fitted meshes is a major limitation in some problem areas. (This is one of the main reasons why we are interested in integrating the AMR algorithm with the Chimera grid method.)

Using the overset fine domain method, a sub-domain with finer cells is to be formed to cover the flagged region, see Figure 1-1c. This is a direct extension of the Chimera method of domain decomposition used as a means of mesh refinement. For a complicated configuration or a complicated flow, it is likely to be necessary to construct a number of overset finer domains. At least two problems will arise. The first one is that it is not easy to construct so many overset finer sub-domains automatically during a calculation. The second problem is that the inter-grid communication will be very expensive, since there are many finer domains overset on and/or intersecting with each other.

Using the ASPaR scheme, the physical domain is partitioned into several sub-domains, see Figure 1-1d. In this example, one finer sub-domain is used to cover the flagged region. The ASPaR scheme uses finer sub-domains to singly cover the high-resolution region, except for the minimum overlap required for inter-grid communication. A domain covered by a single mesh level should result in fewer points overall and, hence, lower computational cost. However, for a complicated flow feature, it will certainly require too many sub-domains to get a high resolution, and hence, the donor searching and inter-grid communication will be very expensive. Furthermore, the flow solution will suffer more serious errors caused by the interpolation on the Chimera overset boundary, see Figure 1-2, or spend more computational time on the overset boundary treatment. For a complicated flow feature, the ASPaR method obviously needs many sub-domains to get a high resolution. This will require more expensive computation for inter-grid communication and introduce more errors into the flow solution due to the interpolation on the Chimera overset boundary.

After taking all these things into consideration, a new scheme which applying the AMR algorithm within a Chimera grid system directly was proposed by the author. Figure 1-3 shows the idea behind this new scheme (it is called as AMRC for short, where C stands for Chimera). That is, using several Chimera sub-domains to mesh complicated configurations and provide a high quality initial grid system, and then applying the AMR algorithm in each domain to resolve complicated flow features. It is believed that the AMRC scheme is more efficient than the ASPaR method and other existing solution adaptive Chimera schemes reported in the literature.

Firstly, the combination of versatile geometric packing ability of the Chimera grid method and efficient computation ability of the AMR algorithm make the AMRC scheme can simulate multiple and moving body flow problem with strong discontinuities by using moderate computational resources.

Secondly, both Graph data structure and orthogonal list storage are employed in the AMRC scheme to describing the Chimera sub-domain relationship and improve the region and point searching efficient.

Thirdly, the accuracy of the inter-grid communication can be improved significantly when simulating a supersonic or hypersonic with strong discontinuities due to applying the AMR algorithm.

So far, little work has been done in applying the AMR algorithm within a Chimera structured grid system directly. To the author's knowledge, this new scheme to be presented in this thesis is the first to apply the AMR algorithm directly within a Chimera grid system. Comparing with the methods demonstrated in Figure 1-1, it is believed that the scheme, demonstrated in Figure 1-3, is more reasonable for modelling multiple moving body flows at high speed.

As mentioned above, currently the major limitation of the AMR algorithm is the relatively poor geometric packing capabilities offered by its grid. The present research is also a contribution to effectively and efficiently solving this problem.

In the Chimera grid method, properly and precisely describing the domain relationship is a key issue for a fully automated Chimera grid scheme. It is opinion of the author that this issue has not been sufficiently addressed in literature and that the present research is also a contribution to efficiently solve this problem.

Finally, it is very clear that, for such a scheme, the robustness, accuracy and efficiency are achieved by employing the second-order Roe's FDS scheme and by using the Chimera grid method to simplify grid generation and to provide a high quality (with minimum skewness) grid system and by employing the AMR algorithm to further enhance the resolution. It is believed that this scheme will be a powerful tool for simulating shock hydrodynamic problems and multiple moving body flows at high speed. Throughout this thesis, I will present and prove this proposal.

1.3 Literature Review

Since a specific objective of the present work is to integrate the Chimera grid method and the AMR algorithm, the literature review certainly includes these two methodologies. It is naturally divided into two parts. The first part will focus on the Chimera grid method,

and the second part will focus on the AMR algorithm. It should be pointed out that this review is not an attempt to collect all available information, but instead to briefly describe the most important developments and applications carried out on the Chimera grid method and the AMR algorithm by previous researchers.

1.3.1 On The Chimera Grid Method

Normally, for simple body shapes, the use of a single body-fitted curvilinear grid is the most efficient solution procedure. However, for more complex configurations or multiple moving body flows, it is quite difficult to generate a single body-fitted well-clustered curvilinear grid that is not overly skewed and has smooth variation. This is especially true for complex 3D configurations and it is generally impractical to build a single grid. Moreover, relative motion between components would require a new grid to be generated after each time step. Re-gridding a configuration at each time step is often too costly and not always reliable. As the configurations become more complex, grid generation and grid definition become the major obstacles to computational simulations.

In order to reduce the difficulty of grid generation to manageable levels for complex configurations, unstructured grid and composite structured grid schemes have been widely used. Compared with unstructured grid methods, structured grid methods are generally considered to require less computer memory, and can handle viscous flow problems with less difficulty. Therefore, the requirement for a somewhat structured grid, whilst maintaining the need for geometric flexibility, points towards the composite structured grid methods for a large-scale practical viscous flow simulation.

Composite structured grid schemes use more than one grid to mesh an overall configuration. The physical domain is subdivided into individual sub-domains that can accommodate easily generated grids. Each sub-domain grid of the system still could be a body-fitted grid. Different sub-domain grids might be either patched or overset, see Figure1-4. In a patched grid system, different sub-domains are joined together at certain common interface grid lines. Difficulties arise with patched grids because various interface

grid lines have to be defined and grids have to be generated with both inner and outer defined boundary surfaces.

Unlike the patched grids, the Chimera overset grids do not require a common grid line. The different sub-domains can overlap each other in an arbitrary fashion. This method can greatly mitigate the difficulties associated with the whole grid system generation for a complicated configuration. Furthermore, by removing the common grid line, each sub-domain grid generation is also greatly simplified since hyperbolic grid generation may be used. This is becoming the most efficient and versatile grid method for handling complex geometric configurations and multiple moving bodies.

The Chimera grid method is based on the early work of Steger, Benek, Dougherty et al¹⁻⁶. It is an immensely flexible technique which provides a conceptually simple method for domain decomposition. Since it was first introduced into the CFD community in the early 1980's, the Chimera grid method has undergone significant development, and has been improved considerably by many previous researchers.

The initial research work on the Chimera grid method by Steger et al¹ mainly focused on the fulfilment of this new idea, and illustrated the main concepts as well as the procedures behind the Chimera grid method. Afterwards, their studies²⁻⁶ were concerned with exploring the flexibility and accuracy of the Chimera grid method and developing techniques associated with this method, such as hole-cutting, inter-grid communicating, describing the domain relationship etc. A tree structured grid hierarchical system was proposed so as to simplify describing the sub-domain relationship². First-order, second-order Taylor series approximations and tri-linear interpolation were examined for inter-grid communication³. In particular, a stencil walk algorithm was proposed for improving point searching efficiency⁴, a normal vector search procedure was used for locating hole points³. In reference⁴, this grid method was extended into three dimensions. In reference⁶, a graph structured grid hierarchical system was used for the generalisation of the sub-domain interaction, and the Chimera grid method was extended with application to a thin-layer Navier-Stokes flow model. The flexibility of the Chimera grid method has been greatly

exploited not only for simplifying grid generation, but also for viscous flow simulation. In different domains, the flow physics can be specified differently.

Since it has a number of advantages over other grid methods, soon after it was introduced into the CFD community, it drew wide attention from many other researchers. Many different techniques associated with the Chimera grid method were proposed and developed²⁵⁻³⁶. The accuracy and other properties of the Chimera grid were examined widely in references³⁷⁻³⁹ and many others, which will be discussed in detail below.

In particular, since the multi-domain feature of the Chimera grid method is suitable for mesh refinement, a couple of different mesh refinement schemes by using a Chimera grid or patched grid were developed^{19-20, 40-41}. Typically, Kao et al¹⁹ used relatively finer overlapping meshes in high gradient regions, see Figure1-5, to get solution adaptive enhancement for shock reflections and shock wave/boundary layer interaction. Chawla et al²⁰ employed moving finer overset meshes to track dynamic flow features, such as moving shock and vortices, see Figure1-6. Wang et al⁴⁰⁻⁴¹ developed a moving grid (patched grid) and a hybrid moving grid/mesh enrichment approach to simulate an unsteady flow involving multiple objects moving relative to each other. The latter approach takes advantage of both structured and unstructured grid. Most importantly, Meakin et al²¹⁻²² developed an Adaptive Spatial Partitioning and Refinement (ASPaR) method, see Figures 1-1d and 1-2, which uses one level but finer sub-domains to singly cover the high-resolution region.

The independent multi-domain feature also makes the Chimera grid method can take advantage of advanced parallel computational technology. Some research work in this area has also been reported in literature⁴³⁻⁴⁸. However, if the sub-domain sizes are disparate, it will be not necessarily an efficient parallel strategy.

In most Chimera grid approaches, inter-grid communication between individual domains is achieved by the use of interpolation. Problems may occur when strong discontinuities pass through domain interfaces. Therefore, a debate on this issue has arisen. Some researchers deem that the treatment of the domain interface needs to be conservative³⁸⁻⁵¹. Some improvements on the conservation properties of Chimera grid

schemes were suggested by several researchers^{49-51, 58-62}. Typically, Wang et al⁴⁹⁻⁵⁰ used a common boundary between zones to divide the overlapped zones into patched zones. In this approach, the problem of enforcing conservation for overlapped zones is transformed to the problem of enforcing conservation for patched zones. Hence the conservation condition becomes flux conservation along this newly generated boundary. Rai⁶⁰⁻⁶¹ developed a boundary conservation scheme for the Euler equations when using patched grids. Using this method, global conservation is maintained in the grid by calculating the fluxes along a boundary and interpolating this information for the matching grid. Berger⁶² generalised this technique for a set of hyperbolic equations when using two-dimensional embedded grids with overlapping boundaries.

On the contrary, the other side claims that as long as the computational mesh is fine enough, a physically correct numerical solution should always be obtained. Meakin⁶³ showed that the formal order of accuracy was maintained without conservative treatment for a number of steady and unsteady transonic aerofoil cases using the Euler equations. In the work of Holst⁶⁴⁻⁶⁵, the error associated with a Chimera scheme was studied for a range of two-dimensional and three-dimensional potential flow problems. Results from these studies demonstrated that the error generated from single-zone and multiple-zone Chimera grid schemes is almost identical across a broad range of grid refinement. Wang et al⁵⁵ have demonstrated that their Chimera scheme (with conservative boundary treatment) can give significantly better computational results than an original Chimera scheme on coarse to medium grids for smooth flows with all other factors the same. However, they also pointed out that on a sufficiently fine mesh, a Chimera scheme with conservative boundary treatment and a Chimera scheme without conservative boundary treatment converge to the same solution⁵⁵. Meakin²¹ argues that grid resolution is the primary issue and that:

"An effective adaptive grid technique appropriate for system of overset grids should be viewed as the primary remedy for issues relating to conservation at grid interfaces."

The Chimera grid method provides a very powerful means of handling complex geometries and multiple body problems. During the last decade, the majority of research work has focused on exploiting its applications. It has been applied with considerable success to many different flow situations. There are various wide ranging examples of applications of this method in the literature, including computing the flow through an artificial heart⁶⁶, computing flow fields around complicated configurations and multi-bodies⁶⁷⁻⁸², computing flow fields⁸³⁻⁸⁹ even the aeroacoustics⁹⁰ around rotors, and computing flow over a complex space shuttle launch vehicle configuration etc⁹¹⁻⁹³.

In recent years, the time-accurate properties and applications of the Chimera grid method have been examined^{37, 84}. The impact of time-step, size of the overlap region, interpolation error, and the use of sub-iteration were analysed³⁷. It is recommended that sub-iteration be routinely applied to Chimera-type applications³⁷. At the same time, many investigations have been made in the area of unsteady flow⁸⁵⁻⁸⁷ and analyses of multiple moving bodies^{88-89, 95-99}. The presence of moving bodies continually changes the relative orientation of the overset domains during the flow simulation. In such cases, it is difficult for the user to provide apriori all the input necessary for hole-cutting and stencil identification. A most important development has been made recently by Wang et al²⁸⁻²⁹. A fully automated Chimera methodology has been developed. Most of the tasks associated with a Chimera scheme such as hole-cutting, stencil identification, and interpolation coefficients are performed in a fully automated mode. An Alternating Digital Tree (ADT) algorithm is used for point searching²⁸ and the major-minor grid relationship that is conventionally used in earlier Chimera grid schemes is completely removed. Instead, each domain is considered an independent entity which can intersect the others in an arbitrary manner²⁸. This makes the present Chimera methodology a powerful tool for flow simulations with moving bodies. It has been proven to be an efficient and versatile approach for movement in that grids for different parts of the configuration can be moved relative to one another to simulate the unsteady body dynamics. By using the automated Chimera grid scheme, almost any relative motion between sub-domains is allowed because domain boundaries can intersect each other in an arbitrary manner²⁸. Unsteady flow

problems such as the separation of a solid rocket booster from an integrated space shuttle vehicle and the unsteady flow around a complete helicopter can be solved. More recently the Chimera grid scheme was applied to complicated three-dimensional unsteady problems with relative motion, such as store separation events⁹⁹⁻¹⁰⁰, aircraft escape systems^{31, 39}, sabot discards¹⁰³ and the separation of a submunition from a missile¹⁰⁴, which is the real purpose and advantage of the methodology.

1.3.2 On The AMR Algorithm

In the CFD community, it is well known that the accuracy of the numerical solution is strongly influenced by the order of accuracy of the numerical solution procedure and the discretization of the space in which the solution is sought. Normally, high-order schemes require far fewer mesh cells to achieve a given accuracy of solution than do low-order schemes. However, to resolve a strong discontinuity, a numerical solution scheme, no matter how high the order, can only yield accurate results for length scales that are larger than a few mesh cells. Consequently, for many types of problem, the accuracy of simulation is limited by the resolution of the computational grid. It is no longer advantageous to improve the order of accuracy of the numerical solution procedure; further improvements can only be gained by improving the resolution of the computational grid¹⁸. This is also true for temporal resolution.

In general, the introduction of a highly dense distribution of grid points throughout a computational domain will yield a more accurate solution than a coarse distribution. This requires enormous computer resources both in memory and computational cost if a flow field involves complicated flow features, such as a shock wave, boundary layer, wake flow etc. However, these resources are often largely wasted on regions where the solutions are relatively smooth. This is particularly true for a flow field with unsteady complicated flow features.

Adaptive mesh schemes are a class of strategies that address this problem by performing high-resolution computations only in regions that require it. Very fine mesh

cells are restricted to those regions where they are needed, and elsewhere the cells may be quite coarse. Such a strategy can significantly reduce the computational resources required to simulate problems that contain disparate physical length scales¹⁸.

The literature for adaptive mesh schemes is also extensive, dating back over more than twenty years and continuing today as a rich field of research. The many approaches vary considerably in both philosophy and implementation. Besides those schemes based on the adaptive unstructured grid approach (which is beyond the current research project) and mesh refinement by using overset meshes (which is been discussed in §1.3.1 and illustrated in §1.2), there are a number of adaptive mesh schemes for structured grids, such as moving grids, adaptive refined Cartesian grids, tree based embedded grids and (of course) Berger's AMR scheme, see Figure 1-7d.

Among the existing structured adaptive mesh schemes, the moving grid adaptation technique improves the resolution of a simulation by redistributing grid points¹⁰⁵⁻¹⁰⁸ of an existing structured grid into the regions of interest, see Figure 1-7a, rather than by creating new grids at different resolutions. This method employs a given number of grids, with the grid nodes being allowed to gravitate towards regions deemed to be of special importance. Such methods try to get the most accurate solution for a fixed cost. The primary advantage of this kind of method is its relative ease of implementation into existing flow solvers since no modification to the data structure is required. It has been demonstrated to be quite effective when only a limited number of features need resolving¹⁰⁸. Unfortunately, since the number of grid points is fixed, increasing the resolution in one part of the computational domain must pay the penalty of decreasing the resolution in other parts. This often results in highly stretched, skewed or distorted grid lines and can leave some regions of the flow field with inadequate resolution. Moreover, most shock-capturing schemes only maintain their accuracy on smoothly varying, well-behaved grids. Therefore, this kind of scheme is not well suited to complicated flow problems.

The adaptive embedding grid scheme¹⁰⁹⁻¹¹⁷ is another class of structured adaptive mesh schemes, which refines the meshes by creating new grids in specific regions of an existing grid based on some reasonable criteria such as high flow gradients or high local

errors. This allows improved resolution of important features with a relatively small increase in grid points and without adversely affecting other portions of the grid.

The design of an efficient data structure is one of the keys to the efficiency and flexibility of such schemes. Some of the adaptive embedding grid schemes employed a tree-based data structure in which only the flagged cells are refined. The coarse mesh cells are split into identical subcells, see Figure 1-7b. The description of the mesh is stored in a quadtree-like structure¹¹⁴⁻¹¹⁵ (for 2-D problems), or octree-like structure¹¹⁶ (for 3-D problems). The storage overhead of these data structures usually exceeds the storage overhead associated with the block structured approach (like the AMR algorithm). Moreover, the constraint on the refinement ratio, in which cells are exactly halved in each direction, is insufficiently flexible. If a simulation problem has a small number of widely disparate length and time scales, then these scales must be implemented by many intervening levels, requiring considerable storage space and computation time. Davis et al¹⁰¹ developed a tuneable data structure (semi-structured) and directional embedding adaptive approach. With this approach, the computational grid only sub-divides in the necessary directions, see Figure 1-7c. This directional flexibility in grid-embedding adaptation is essential for 3-D flows in order to efficiently resolve critical flow features, such as boundary layers (normally streamwise) and shock front (normally transverse).

The basic idea behind the multigrid strategies is to accelerate the solution generated on a fine grid system by computing corrections on a coarser grid. These are not adaptive mesh approaches. However, the multigrid methods are an instructive example of the ways in which meshes of varying resolutions can interact with one another. They have been naturally used by some researchers^{113-114, 118-119} to provide grid level communication and to accelerate the convergence of the tree based adaptive mesh scheme.

Since the early 1980's, in a series of papers⁷⁻¹², Berger, Oliger and Colella et al have been developing a particular Adaptive Mesh Refinement (AMR) strategy for structured meshes. (Throughout this thesis, the term AMR is used to indicate Berger's Adaptive Mesh Refinement concept and Quirk's Adaptive Mesh Refinement code). This algorithm is based on the notion of multiple, independently solvable grids, all of an identical type but each of

arbitrary size and shape. It is a kind of grid embedding scheme, but it uses a hierarchical grid structure (block structured approach, differing from a tree based data structure), see Figure 1-7d. The grid of coarsest resolution is referred to as the level 0 grid. This grid by definition covers the entire problem domain. There are also one or more additional higher (and finer) levels of grid. These finer levels of grid do not cover the whole domain, but only those regions where a higher resolution is needed (determined by a refinement criterion).

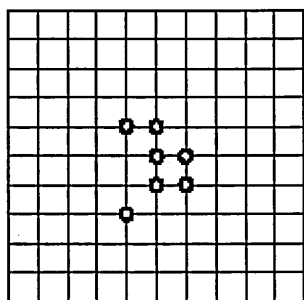
Berger, with various co-workers, has published more than ten papers on various aspects of her AMR strategy, such as the AMR strategy itself⁷⁻¹⁰, AMR data structures¹², conservation issues and the parallelisation¹⁶⁻¹⁷ of the AMR algorithm. Some of Berger's co-workers, including Oliger¹²⁰, Colella¹²¹ et al, have also published AMR studies separately from Berger.

Berger's AMR algorithm has found little favour for it is perceived as being overly complicated. However, the basic idea is excellent. (Chapter 2 will give an introduction to the principle of the AMR algorithm). The AMR algorithm has a number of advantages over than the other adaptive mesh schemes. Firstly, it is a general-purpose numerical scheme for simulating shock hydrodynamic flows. Secondly, it places no special constraints on the basic numerical method used to integrate the flow solution. Thirdly, this scheme refines in both space as well as time. More, but smaller time steps are taken on fine grids than on coarse grids. It is particularly suited to unsteady flows with complicated flow features. This scheme has been successfully used in solving time dependent and steady state shock hydrodynamic problems^{7-11, 13-15, 18, 121}. It has also been shown to produce very high resolution results.

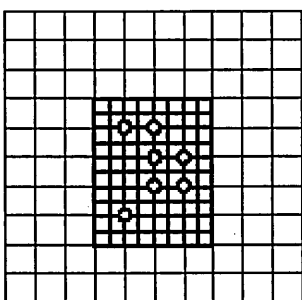
James Quirk¹⁸ further developed the AMR algorithm. In particular, his scheme removes much of the complexity of Berger's scheme, without compromising the quality of the resolution simulations¹⁸. Quirk¹²⁴ also developed various parallel methods for the structured AMR algorithm.

Another class of adaptive mesh schemes is the adaptive Cartesian grid method, see Figure 1-7e. This is also a very attractive research topic. Some of the adaptive Cartesian grid methods¹²⁵⁻¹³³ actually are extensions of Berger's AMR algorithm. In particular, Roper

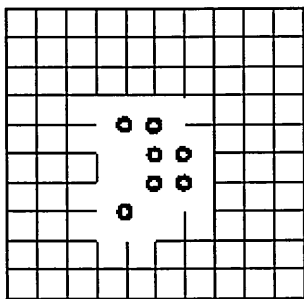
and Edwards¹³¹⁻¹³³ developed an adaptive body recovery technique for the efficient computation of dynamically moving and/or deforming body flows. This technique was based on Quirk's AMR algorithm coupled with the automatic generation of body surfaces, in which mesh adaptation was used to capture the body geometry to within a specified accuracy, see Figure 1-7f.



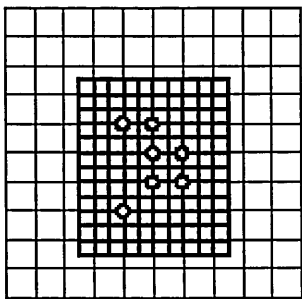
(a). Refinement Region



(b). AMR Overlap mesh



(c). Chimera finer overlap domain



(d). Single covered Chimera grid by ASPaR Method

Figure 1-1 Grid adaption by the AMR and the Chimera method

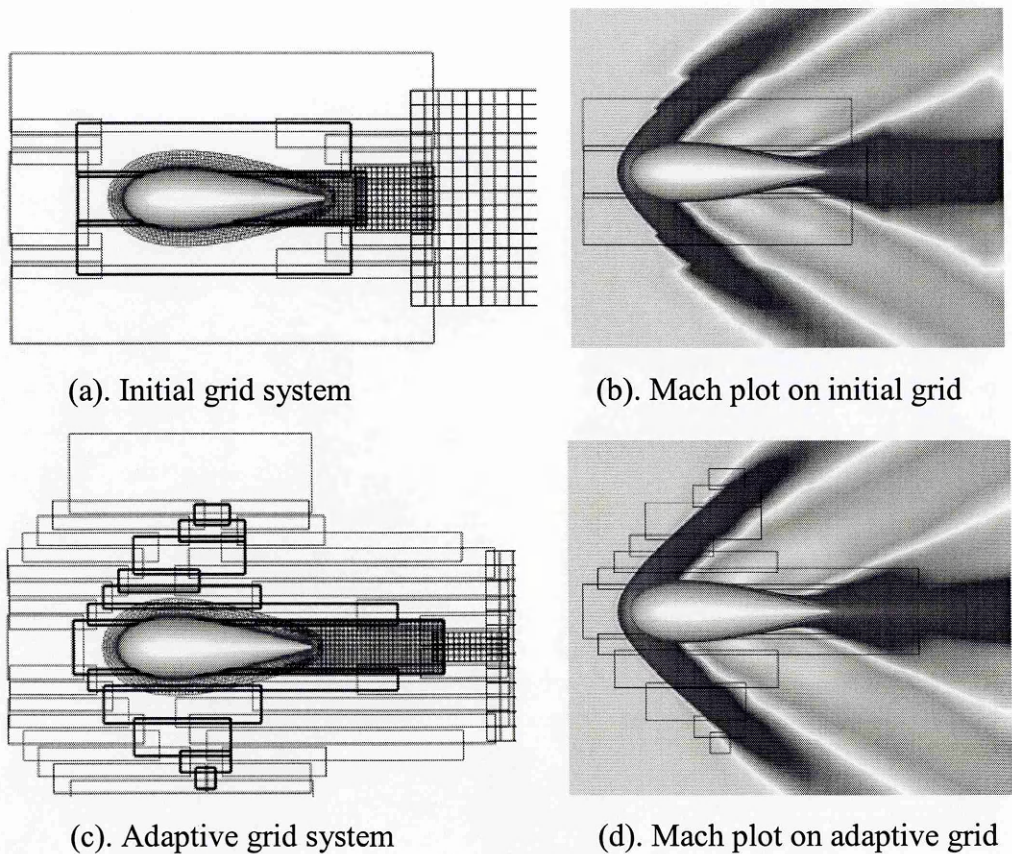


Figure 1-2. Grid and solution for tear-drop geometry by the ASPaR method [22]

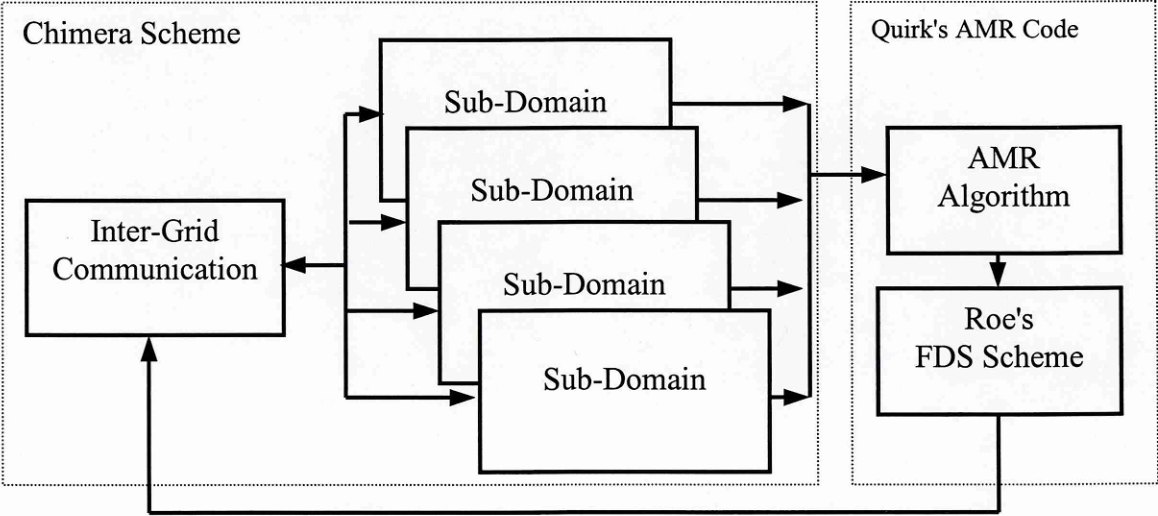
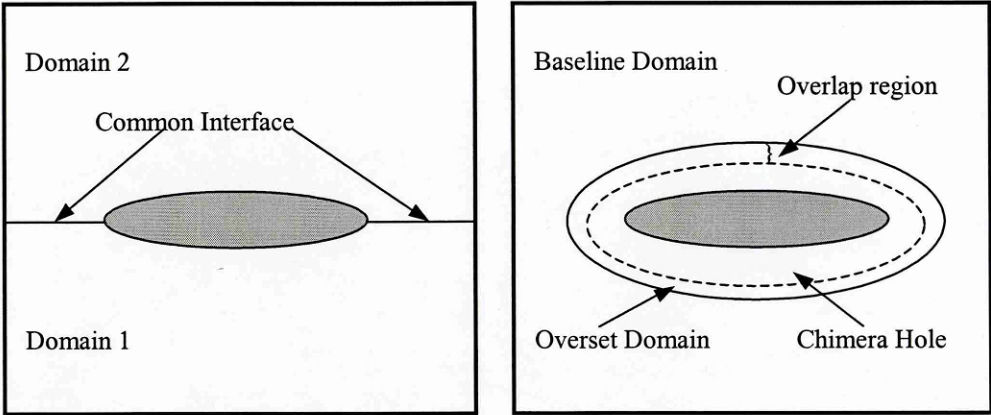


Figure 1-3. Integration of the Chimera scheme and AMR algorithm



a. Patched grid b. Chimera grid

Figure 1-4. Composite structured grid methods

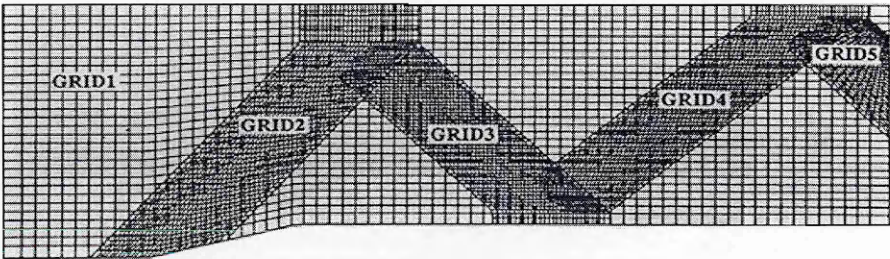


Figure 1-5. Finer overlap grids for a 15° ramp channel flow [19]

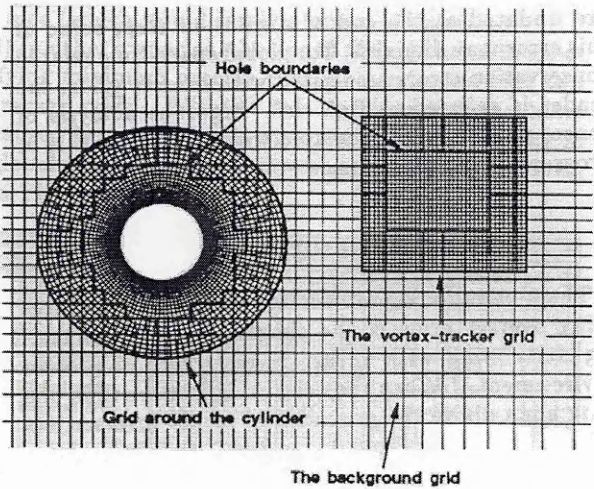
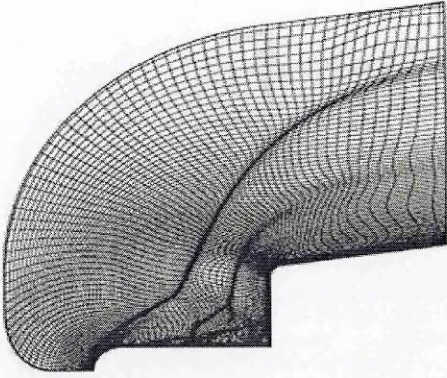
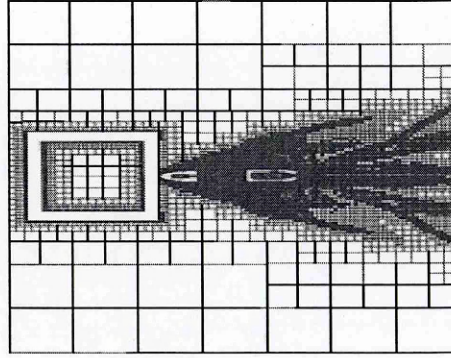


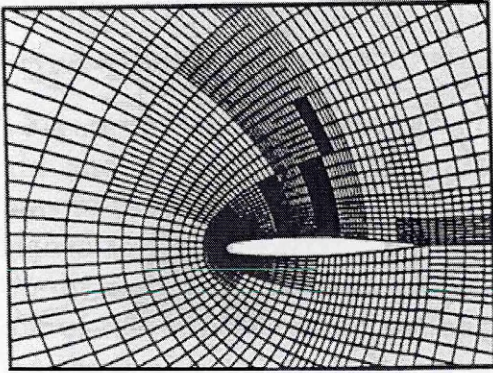
Figure 1-6. Finer moving overlap grid for vortex tracking[20]



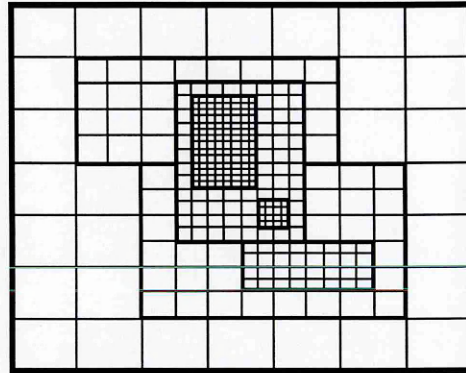
(a). Adaptive moving grid[134]



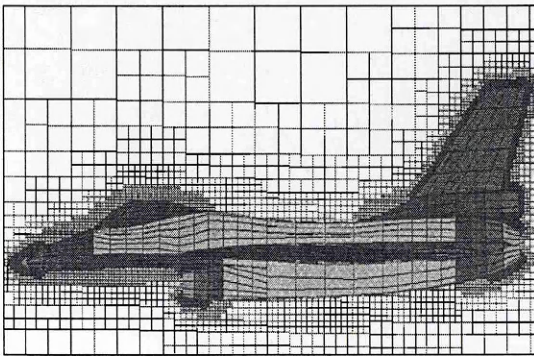
(b). Tree based adaptive embedding grid[134]



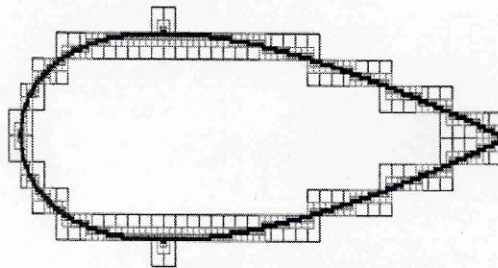
(c). Directional adaptive embedding grid[112]



(d). Berger's AMR scheme



(e). Cartesian adaptive grid[134]



(f). Cartesian adaptive body recovery[131]

Figure 1-7. Various adaptive mesh schemes

Chapter 2

A Brief Introduction to the AMR Algorithm

2.1 Introduction

This chapter is solely concerned with the structured AMR algorithm. The AMR algorithm is well known, not only for its high resolution but also for its complexity to code and implement. Moreover, it has changed in several significant ways since Berger proposed it in the earlier 1980's; some aspects of the strategy have been improved, some altered and some completely eliminated. It is not intended to introduce the algorithm in depth in this thesis but instead to give a brief introduction to some basic ideas of the AMR algorithm and an existing AMR code (Quirk's AMR code) for completeness. For more detailed information on the AMR algorithm, readers are referred to references⁷⁻¹⁸.

2.2 Basic Concepts Behind the AMR Algorithm

The Structured AMR algorithm is an automatic, adaptive mesh refinement strategy for solving hyperbolic conservation laws. It was proposed by Berger⁷ in the early 1980's and further developed by herself, many of her co-workers⁸⁻¹⁷ and Quirk¹⁸. This strategy is not easy to code and implement since it requires a recursive process, but the basic idea of Berger's AMR algorithm is excellent and straightforward to understand. Since the AMR algorithm performs high-resolution computations only in regions that require it, about only 5~10 percent of computational resources compared to a non-adaptive calculation are

community, especially in modelling unsteady viscous flow or complex flow features with strong discontinuities. In the following sub-sections, we shall introduce the basic ideas and concepts behind Berger's AMR algorithm and summarise the properties of the AMR algorithm.

Usually the terms "mesh" and "grid" are synonymous in the CFD community, but it should be noted that throughout this thesis we use the term "mesh" to indicate a single local quasi-rectangular patch and the term "grid" to indicate a collection of multiple patches at a given level.

2.2.1 Hierarchical Grid Structure

The AMR algorithm uses a block, hierarchical grid structure, higher level finer meshes overlay the lower level coarse meshes. Each level consists of many simple, quasi-rectangular meshes. Each mesh in the hierarchy represents a structured block of refined cells. Using this method, the grid of coarsest resolution is referred to as the level 0 grid. This grid by definition covers the entire problem domain. The flow domain may be refined locally by overlaying finer mesh patches onto the coarse grid at level 0 to form the next grid level within the hierarchy, namely level 1. The patches are formed by linearly subdividing rectangular groups of coarse cells. These finer levels of grid do not cover the whole domain but only those regions where a higher resolution is needed (as determined by a refinement criterion).

The grids at different levels within the hierarchy co-exist, for underlying a finer mesh there is a complete coarse mesh and complete coarse field solution, as shown in Figure 2-1. The finer meshes on each level directly represent the resolution on the sub-regions that they cover, while the coarser meshes without overlaying finer meshes represent the resolution on the rest of the domain. Ultimately, the purpose of this approach is to represent every portion of the domain with the minimal resolution required to satisfy some criteria.

At the root level, level 0, each mesh consists of a computational interior and a ghost boundary region. At all finer levels, each mesh consists of: a region of interest that has been

refined from the immediately coarser level; a buffer region, which allows the existing meshes to continue to cover fully the phenomena of interest until the next regridding; and a ghost boundary region, see Figure 2-2. In some cases, the buffer region may be partially absorbed if the mesh abuts another mesh at the same level, see Figure 2-3. In fact, it is conceptually more appropriate to consider the buffer as surrounding the region of interest, which may be distributed among several meshes, rather than as surrounding the sub-region of interest that a specific mesh contains.

Berger's AMR presents a significant data management challenge. It requires the management not only of the solution vectors for a large, dynamically changing collection of meshes, but also of the relationships between the meshes, the relationships between the solution vectors and various other data items, and the relationships of these data items to the methods that operate on them.

It also should be pointed out that in Berger's AMR scheme, all meshes must be "properly nested", i.e. the boundaries of nested higher-resolution meshes must not coincide. The new fine meshes must be checked to ensure that they are properly contained within their lower level meshes. If they are not, the new mesh is repeatedly subdivided until each piece does fit. This constraint avoids sudden large transitions in resolution, which can lead to spurious numerical errors.

2.2.2 Local Refinement Strategy

The AMR algorithm employs local mesh refinement and adaptation strategies. The refinement covers a subset of the domain and uses smaller logically rectangular mesh patches. These finer meshes can be recursively nested until a given level of accuracy is attained. Figure 2-4 is an example which clearly illustrates the local mesh refinement strategies. Much finer mesh cells are restricted to those regions where they are needed, and elsewhere cells may be quite coarse. The user does not need to guess at what will be an adequate number of cells to resolve features in the solution. With local mesh refinement, cells are added or removed as necessary during a computation.

In particular, the AMR algorithm allows refinement in time as well as space, by the same mesh refinement factor r , where $r = \Delta x_{l-1} / \Delta x_l$, l refers to grid level. In Berger's AMR scheme, the refinement factor r is an arbitrary even integer number (typically 2 or 4). Thus,

$$\frac{\Delta t_l}{\Delta x_l} = \frac{\Delta t_{l-1}}{\Delta x_{l-1}} = \dots = \frac{\Delta t_1}{\Delta x_1} \quad (2-1)$$

So the same numerical difference scheme is stable on all meshes. This means more time steps are taken on the finer meshes than on the coarser ones. This is a reasonable requirement from the point of view of accuracy, since for many numerical difference schemes, the leading terms in the spatial and temporal error are of the same order. In addition, smaller time steps for fine meshes are not imposed globally.

2.2.3 Refinement Criterion

The flagging of cells that need to be refined in regions of interest (according to refinement criteria) is one of the main procedures in the AMR algorithm. This process is probably the most algorithmically complex operation in Berger's AMR strategy. It is implemented recursively: a refinement at level l first refines level $l+1$, and so on recursively to the finest level, level L_{max} . In this way, proper nesting of the refined regions at finer levels is ensured.

Selection of cells to be refined is normally dependent on the nature of the application to which the AMR algorithm is applied; a variety of selection criteria can be applied. In Berger's AMR scheme, an error estimation procedure based on Richardson extrapolation⁷ was used to determine the regions where the resolution of the solution is insufficient, i.e., the local truncation error was used as the refinement criterion to flag those cells on a given level with unacceptable large errors. If the flow solution $u(x, t)$ is smooth enough, the local truncation error $u(x, t+k) - Qu(x, t)$ on a mesh with spatial step h and time step k satisfies

$$\begin{aligned} u(x, t+k) - Qu(x, t) &= k(c_1(x, t)k^q + c_2(x, t)h^q + kO(k^{q+1} + h^{q+1})) \\ &\equiv \tau(x, t) + kO(k^{q+1} + h^{q+1}) \end{aligned} \quad (2-2)$$

where the leading term is denoted by τ . It is assumed that the difference method Q has an order of accuracy q in both time and space. If u is smooth enough, then for two time steps, the following equation can be given,

$$u(x, t+2k) - Q^2u(x, t) = 2\tau + kO(k^{q+1} + h^{q+1}) \quad (2-3)$$

Let Q_{2h} denote the same difference method as Q but based on mesh widths of $2h$ and $2k$, then

$$u(x, t+2k) - Q_{2h}u(x, t) = 2^{q+1}\tau + O(h^{q+2}) \quad (2-4)$$

By taking two steps with the regular integration scheme, and one "giant" step using every other grid point, the difference

$$\frac{Q^2u(x, t) - Q_{2h}u(x, t)}{2^{q+1} - 2} = \tau + O(h^{q+2}) \quad (2-5)$$

gives an estimate of the local truncation error at time t .

In other words, the truncation error estimation is obtained by advancing the solution one time step of interval $2k$ on a mesh space of $2h$, advancing the solution two time steps of interval k on a mesh space of h , and comparing the results. Cells where the error is greater than a given tolerance are flagged for refinement.

In principle, this error estimator is ideal for the AMR algorithm since it uses the same solver as is used for evolving the solution. In practice, however, it can be somewhat unwieldy. The difficulty is in the calculation of $Q_{2h}u$. Computing this solution is trivial: it is simply another call to the solver. In practice, however, setting the grid with mesh space of $2h$ can be considerably more complicated. Moreover, near flow discontinuities, the very types of flow feature we are most interested in, Richardson extrapolation is formally

invalid, although to be fair, the truncation error calculated by this method will be large and so discontinuities will still be refined.

Of course, other criteria might be used in addition to, or instead of, this error estimate, e.g. identifying steep gradients in some variable. As we shall see later in Quirk's AMR algorithm, any gradient of any scalar of the flow solution can be used as a monitor function for flagging those mesh cells that require refinement. Usually density gradient is used for simplicity.

2.2.4 Clustering Algorithm

The clustering process in Berger's AMR algorithm is used for packing a higher level grid $\tilde{G}_l$ when given a set of flagged cells contained by the grid G_{l-1} . The term "clustering" was adopted in Berger's AMR algorithm. In Quirk's AMR algorithm, he adopted the term "Grouping". The clustering algorithm of Berger is based on a method used in computer vision and pattern recognition. The algorithm is as follows¹²³, as illustrated in Figure 2-5.

- First, a minimal bounding box is placed around the cells that have been flagged to be refined.
- Next, a signature list is computed along each grid axis. A signature Σ_i is the number of flagged entries along the i^{th} grid surface; e.g. a grid line in 2D, a grid plane in 3D,

$$\Sigma_i = \sum_j \sum_k f_{ijk}, \quad \text{where} \quad f_{ijk} = \begin{cases} 1 & \text{if cell } ijk \text{ is flagged} \\ 0 & \text{otherwise} \end{cases} \quad (2-6)$$

- Then, any zero entry in the signatures list of any axis is a grid surface perpendicular to that axis that contains no flagged cells, and thus this zero entry is a potential cutting index. The best of these zero entries, the most central along some axis, is used as the cutting index to subdivide the domain.

- In many cases, the signatures are all nonzero, yet these signatures are often obtained from arrangements of flagged cells that clearly can be further decomposed. In this case, the Laplacian second derivative of the signatures

$$\Delta = \Sigma_{i+1} - 2\Sigma_i + \Sigma_{i-1} \quad (2-7)$$

is used. A *zero crossing* is an index where Δ changes sign, and such a *zero crossing* is a potential cutting index. The best *zero crossing* is the one whose magnitude

$$Z_{i+\frac{1}{2}} = |\Delta_{i+1} - \Delta_i| \quad (2-8)$$

is largest, and this index is chosen as the cutting index.

This process clusters the flagged cells in order to get finer meshes in a new higher level (creates new finer meshes) so as to cover the selected area of refinement at the immediately lower level. This method of placing finer meshes over coarser meshes can be repeated recursively, either to some predefined maximum resolution, or until the refinement criteria no longer exceed the threshold at some level. Thus, the algorithm traverses a $k-d$ tree with specially defined cutting criteria. Recursion continues until some halting criteria are satisfied.

2.2.5 Flow Integration

Berger's AMR algorithm is based on the notion of multiple, independently solvable meshes, all of an identical type but each of arbitrary size and shape. The underlying premise of Berger's AMR algorithm is that all meshes of any given resolution that cover a problem domain are equivalent in the sense that, given proper boundary information, they can be solved independently by identical means. The following are the main processes of the solution procedure.

- **Transfer the solution to the new finer meshes.** This process collects values for new finer meshes, including ghost boundary cells surrounding each mesh's active region.

Typically most of the values on the new meshes are actually copied directly from the old meshes, or injected from a coarser level, as illustrated in Figure 2-6. Values for ghost cells are collected from three sources as appropriate: by injecting from parents on the coarser level; by copying from siblings at the same level, and by extrapolating from the exterior of the computational domain.

- **Integration for every mesh.** This process evolves the solution by advancing it forward by a specified time interval. Like refinement, it is also implemented recursively. The details of a flow solver are not directly relevant to the AMR algorithm, because the AMR abstracts it into a "black box" operation, i.e. given the correct input, the solver is expected to produce the correct output. All meshes at different levels can be solved independently by an identical solver.
- **Flux Correction for Conservation.** This process maintains the conservation on the interface between a fine grid and a coarse grid.
- **Projection from fine to coarse meshes.** This process updates a coarser level, using the more accurate solution of the cells at the finer level to replace the covering cells on the immediately coarser level.

2.2.6 Summarisation of the Properties of the AMR Algorithm

The AMR algorithm is a general-purpose numerical scheme for simulating shock hydrodynamic flows. It is particularly suited to transient flows. It places no special constraints on the basic numerical method used to integrate the discretised flow solution.

The AMR algorithm contains machinery which allows a mesh patch to be integrated independently of any other mesh patch; the process which integrates a mesh patch never sees a mesh boundary. In principle, any cell-centred scheme developed for a single quasi-rectangular mesh may form the basis of the flow integration process.

The AMR algorithm can automatically restrict the extra grid levels to those regions where the flow activity is of particular interest. The computational grid can be made to adapt to the evolving flow solution, i.e. each extra grid level the effective mesh spacing becomes progressively finer thereby increasing the resolution of the simulation. Thus it is

possible to achieve the numerical resolution associated with very fine meshes without incurring the computational penalties of having to employ a fine mesh throughout the flow domain.

The AMR algorithm refines both in time and space. More, but smaller time steps are taken on fine grids than on coarse grids. The hierarchical nature of the grid system allows the different sized time steps to be interleaved such that the simulation remains time accurate. Therefore, in contrast to other mesh refinement schemes, the presence of a few extremely fine mesh cells in one part of the flow domain does not have an adverse affect on the rate at which the rest of the flow solution may be advanced. It is this combination of spatial and temporal refinement strategies that accounts for the efficiency of the AMR algorithm. It requires only modest computing power to resolve complex shock on shock interactions.

In general, it is possible to utilise a dynamic regridding process on either regular or irregular grids. However, due to the inherent lack of symmetry of irregular grids, grid-based errors are more difficult to suppress than in regular grid schemes. Since many time-dependent problems are unstable to small perturbations, these grid-based errors can rapidly grow to dramatically impact the outcome of a calculation.

Berger's AMR is difficult to code for three primary reasons. First, this strategy presents a very complicated data management problem, not only because of the dynamic nature of memory usage but also because of the relationships and interactions between the various parts. Second, the AMR algorithm itself is not only elaborate but also is very sensitive to even the smallest incongruities. At every step in the development process, new problems are discovered that must be addressed in a manner consistent with the rest of the AMR implementation. Third, the main procedures, such as refinement, regridding and integration must be implemented recursively.

2.3 Brief Introduction to Quirk's AMR Code

The foundations of Quirk's¹⁸ AMR code lie with Berger's AMR algorithm. This code has proved to be an efficient means of producing very detailed simulations for complex

shock hydrodynamics problems. Starting from this section, we shall give a brief introduction to the important features and the structure of Quirk's AMR code, and then some code modifications so that it can be integrated into the Chimera grid method.

2.3.1 Definition and Description of Grid System

Quirk's AMR code has a very precise definition and description of the grid system, mesh index, mesh boundary and so on. Following on from this descriptive overview, it is likely to be easier to understand such an intricate algorithm and to grasp the main points of the AMR algorithm. The nomenclature and concepts introduced in this section will be used extensively throughout the remainder of this thesis.

1. Grid hierarchy and mesh index system

Quirk's AMR code employs a multiple level hierarchical structure describing the grid system, as sketched in Figure 2-1. An arbitrary grid structure G is formed from a set of $l_{max}+1$ grids, one grid G_l for each level within the hierarchy. Thus,

$$G = \{ G_l : l \in N, 0 \leq l \leq l_{max} \} \quad (2-9)$$

Each grid G_l is composed by one or more mesh patches at the same level l . All meshes are required to be properly nested, that is, a mesh at level l must be wholly contained within one or more coarse meshes at level $l-1$. Thus,

$$G_l \subseteq G_{l-1} \quad (2-10)$$

As an example, the meshes shown in Figure 2-7a satisfy this definition of proper nesting, while those in Figure 2-7b do not. Note that this definition of proper nesting is less restrictive than that of the original Berger's AMR strategy.

If there are nG_l mesh patches at level l , the k^{th} mesh is denoted by $G_{l,k}$, then,

$$G_l = \bigcup_{k=1}^{nG_l} G_{l,k} \quad (2-11)$$

And the k^{th} mesh $G_{l,k}$ can be identified by a unique mesh index system,

$$\begin{aligned} GRD &= G_{pl} + k \\ G_{pl+1} &= G_{pl} + nG_l \end{aligned} \quad (2-12)$$

Where GRD denotes the mesh index (following Quirk's nomenclature).

G_{pl} denotes the first mesh index at level l , and $G_{p0} = 0$.

According to equation (2-12), the mesh index system of the mesh patches shown in Figure 2-7a will be denoted as Figure 2-8.

In Quirk's AMR code, the refinement factors, rI or rJ , (integer numbers, typically with values between 2 and 10) are required to be the same for every coarse cell that is refined at the same grid level. This restriction enables every mesh cell contained at level l to be referenced by its own (i, j) co-ordinate system. And a simple relationship exists between the co-ordinates at different levels. A cell (i_l, j_l) at level l would be formed from the sub-division of a coarse cell at level $l-1$ whose co-ordinate (i_{l-1}, j_{l-1}) can be given by,

$$i_{l-1} = \frac{(i_l - 1)}{rI_l} + 1; \quad j_{l-1} = \frac{(j_l - 1)}{rJ_l} + 1 \quad (2-13)$$

It should be noted that the refinement factors, rI and rJ , can be specified differently and they do not need to be **even integers** as required in Berger's AMR algorithm.

2. Mesh boundary definition

In Quirk's AMR code, a cell interface that forms part of the boundary of a mesh $G_{l,a}$ may be categorised as one of following three types.

- **External**, if it borders the edge of the computational domain;

- **Fine-Fine**, if it borders a cell from a mesh, $G_{l,b}$, which belongs to the same grid;
- **Fine-Coarse**, if it borders a cell from the underlying coarse grid G_{l-1} .

These different types of boundary are shown in Figure 2-9. Strictly, a single interface could be tagged as more than one type, as for the east boundary of mesh $G_{l,2}$ in Figure 2-9. However, external boundaries determine the problem to be solved. Consequently, an external tag should take precedence over both Fine-Fine and Fine-Coarse tags. While from accuracy considerations, a Fine-Fine tag should take precedence over a Fine-Coarse tag.

3. Dummy cells

One of the strategies underpinning the AMR algorithm is that the basic mesh integrator should not need to know about the mesh connectivity; this burden is offloaded onto a separate process. The same numerical scheme that is used to compute the fluxes across cell interfaces which are internal to a mesh is also used to compute fluxes across cell interfaces which form mesh boundaries. Adopting this approach, to integrate a mesh of $IMAX$ by $JMAX$ cells using an arbitrary second order cell-centred scheme requires data on a mesh of $(IMAX+4)$ by $(JMAX+4)$ cells, see Figure 2-10. Those cells that lie outside the basic mesh of $IMAX$ by $JMAX$ cells are called **dummy cells**. Prior to the integration of a mesh the dummy cells for that mesh are primed with data. This data is chosen such that the resultant numerical fluxes across the mesh boundary are consistent with the various boundary conditions that are to be met.

4. Linked list storage

In opposition to most CFD software packages, the AMR code requires storage data for more than one structured mesh and the mesh sizes for different meshes may vary significantly. Precisely organising and managing the data from different grid levels is one of the key issues.

In Quirk's AMR code, for memory efficiency, all flow variables and the geometric data are stored in a series of linked lists. Each list consists of a continuous set of storage

elements that may be viewed as a one-dimensional array. The index pointer of cell (i, j) of a mesh GRD in the linked list is given by,

$$IJ = H2PTR(GRD) + (I+2) + (J+1) \times (IMX(GRD) + 4) \quad (2-14)$$

where the array element $H2PTR(GRD)$ contains a pointer to the start of the data block for the mesh with index GRD .

For example, the x co-ordinates of all grid points in different meshes at different levels are stored in array $X(IJ)$. Here IJ can be calculated by equation (2-14).

2.3.2 Main Procedures

As we have already mentioned, Quirk's AMR code was based on Berger's AMR algorithm. But Quirk further developed the AMR algorithm. In particular, his scheme removes much of the complexity of Berger's scheme, without compromising the quality of the resolution of simulations. Comparing the following main procedures with those of Berger's, we can clearly see the differences.

1. Flagging for refinement

Quirk's AMR code is primarily concerned with computing solutions to unsteady shock hydrodynamic flow problems; the resolution of numerical solutions to such flows can be improved if the computational grid can be locally refined in the vicinity of shock waves and contact discontinuities. If these basic features can be flagged for refinement, preferably by some economical means, then more complicated flow features such as triple points and roll up vortices will also be flagged for refinement.

One obvious refinement criterion (monitor function) with which to flag shock fronts and contact surfaces is based on the density gradient of the flow solution. Checking the change in density between two neighbouring mesh cells, if this change is larger than some fraction, F_{tol} , of the maximum density jump between any two neighbouring cells then both cells are flagged as requiring refinement.

$$F_{tol} = \frac{\Delta\rho}{\Delta\rho_{\max}} \quad (2-15)$$

Typical values for F_{tol} are in the range 0.05-0.15. This somewhat rough and ready approach generally works very well¹⁸.

Actually, in Quirk's AMR code, any gradient of any scalar of the flow solution can be used as a refinement criterion for flagging those mesh cells that require refinement, such as density gradient, shear stress, and other criteria might also be used. Usually density gradient is used for inviscid shock hydrodynamic flows and one might well use shear stresses for viscous flows. By keeping the flagging process separate from the rest of the algorithm it is then relatively simple to incorporate the relevant physics for a particular problem into the adaptation process. This should allow an expedient monitor function to be found whatever the problem to be solved.

2. Grouping/Clustering

A simple area sub-divide algorithm (different from Berger's clustering algorithm) is used for clustering flagged cells to produce a set of patches for each mesh of G_{l-1} in turn. The area sub-divide algorithm proceeds as follows:

- (1) Find the smallest rectangular patch that covers all the flagged cells. This patch is a candidate for forming one of the meshes of the adapted grid $\tilde{G}_l$.
- (2) A simple packing criterion is used to see if the mesh is acceptable.

$$P_p = \frac{N_f}{N_p} \quad (2-16)$$

If the number of flagged cells, N_f , divided by the total number of cells, N_p , contained by the prospective mesh is greater than some value, P_{tol} , then the mesh is deemed acceptable. Typical values for P_{tol} are in the range of 0.5-0.75.

- (3) If the value of P_p is smaller than P_{tol} , the patch is not acceptable. It is necessary to sub-divide it into two regions along the longest side.

- (4) Every time the mesh is sub-divided, the right-hand/top sub-division is stored on a LIFO (Last In First Out) stack and the left-hand/bottom sub-division is further processed following steps (2) to (4), until an extent of flagged cells meets the packing criterion, $P_p < P_{tol}$.
- (5) If the extent of flagged cells meets the packing criterion, this extent is then saved as a mesh patch. An area is taken from the LIFO stack for further processing.
- (6) This general procedure continues until it is no longer necessary to sub-divide an area and there are no more areas left on the LIFO stack.

Obviously, this area sub-divide algorithm requires a recursive process. And every mesh $\tilde{G}_{l,k}$ produced by this process will be wholly contained by at least one mesh of G_{l-1} , that is, $\tilde{G}_l \subseteq G_{l-1}$.

It should be noted that this property is not sufficient to ensure that a grid structure remains properly nested after two or more grid levels have been adapted. For $\tilde{G}_l \subseteq G_{l-1}$ and $\tilde{G}_{l-1} \subseteq G_{l-2}$ do not guarantee that $\tilde{G}_l \subseteq \tilde{G}_{l-1}$. This problem can be overcome by adapting G_l before adapting G_{l-1} . By performing the adaptation in this reverse order, the $\tilde{G}_l$ is already known when it comes to flag those cells that will eventually be clustered to form $\tilde{G}_{l-1}$. This enables us to enforce proper nesting by simply flagging all those cells contained by G_{l-2} that are covered by $\tilde{G}_{l-1}$. This is why in Quirk's AMR code the adaptation process, *Adapt(l)*, is performed in reverse, see §2.3.3 pseudo-code.

3. Temporal refinement and integration sequence

The method of temporal refinement employed in Quirk's AMR code is simple but effective. Different time steps are used for different levels within the grid hierarchy. The time step used for the integration of the grid G_l , Δt_l , is related to that used for the coarser grid G_{l-1} by,

$$\Delta t_l = \frac{\Delta t_{l-1}}{\max(rI_l, rJ_l)} \quad (2-17)$$

Choosing the time step in this manner ensures that the CFL number associated with the grid G_l is comparable to that associated with the grid G_{l-1} . The advance of the solutions at different grid levels is synchronised by recursively interleaving the integration at different grid levels¹⁸.

2.3.3 Code Structure

Quirk's AMR code was coded in Fortran 77. This code inherited the nature of Berger's AMR strategy. The main procedures, such as refinement, adaptation and integration must be implemented recursively.

It is not easy to explain the intricacies of the AMR algorithm, especially when it is coded in Fortran 77, since it contains many elements which require careful co-ordination, and the recursive process is not explicitly supported in Fortran-77. Probably, using pseudo-code is a simple way to accomplish this. The main structure of Quirk's AMR code and the main procedure, adaptation process *Adapt(l)*, can be illustrated in a pseudo-code fragment as below:

Table 2-1. Quasi-code fragment to orchestrate the AMR algorithm

```

Procedure AMR(l, Nt)
  Integrate-Grid(l, Nt)
  If l < lmax {
    Initialise_Conservative_Fixup(l+1)
    For Nft=1 to rtl+1 {
      AMR(l+1, Nft)
    }
    Apply_Conservative_Fixup(l+1)
    Project_Solution(l+1)
    Set_bc(l, Nt)
    If Nt < rtl Adapt(l)
  }
End Procedure

```

Table 2-2. Quasi-code fragment to co-ordinate the adaptation process

```

Procedure Adapt( $l_p$ )
  Initialise  $\tilde{G}$ 
  For  $l=l_{max}-1$  down to  $l_p$  {
    Set_Refine_Flags( $l$ )
    Ensure_Proper_Nesting( $l$ )
    Cluster( $l$ )
  }
  Transfer_Solution
  Transfer_Data_Structure
  Set_Bdy_Types( $l_p+1$ )
  Calculate_Geometry( $l_p+1$ )
End Procedure

```

The call AMR(0,0) will perform one complete sequence of the grid integrations and adaptations. So, only a very simple code fragment is required to form a main program procedure for marching a flow solution over some specified period of time.

Table 2-3. Quasi-code fragment to the main program

```

Main program
... ..
repeat{
  Find_Stable_Time_Step
  Call AMR(0, 0)
} until The_Calculation_Is_Finished
... ..
End Program

```

The procedure **Find_Stable_Time_Step** should be able to calculate the set of time steps $\{\Delta t_l : l \in N, 0 \leq l \leq l_{max}\}$ such that the calculation remains stable. While the function **The_Calculation_Is_Finished** simply checks to see if the calculation should be stopped or not.

In order to prevent a discontinuity from escaping an overlaying finer mesh, the order of grid adaptation follow the order of grid integration. Therefore the grid G_{l+1} is adapted, by calling *Adapt(l)*, after its sequence of rt_{l+1} integration is complete and conservation with the grid G_l has been enforced. Note that the call *Adapt(l)* will in fact adapt all grid levels from $l+1$ to l_{max} . Consequently there is no need to explicitly adapt the grid G_{l+1} if this is to be closely followed by an adaptation of the grid G_l , hence there is a logical test to see if it is strictly necessary to call the adaptation procedure.

In order to calculate the correct gradients along mesh boundaries the ghost cells bordering these boundaries should be primed with the most up-to-date values possible. Therefore, before adapting the grid at level $l+1$ the ghost cells contained by the grid G_l are primed by calling *Set_bc(l, Nt)*.

2.3.4 Flow Solver

Since the AMR algorithm is a general purpose numerical scheme for simulating shock hydrodynamic flows, any cell-centred scheme that integrates the equations of fluid motion using a quasi-rectangular mesh could be routinely incorporated into the AMR algorithm. In Quirk's AMR code, several different schemes have been incorporated to integrate the Euler equations and N-S equations, and one scheme for detonation flows. However, for the majority of our calculations (which will be presented in Chapters 4 to 7) we have chosen to use second order Roe's FDS scheme because of its inherent robustness and the balance it strikes between resolution and computational expense.

Here we do not intend to introduce the details of flow solvers for a couple of reasons: Firstly, these schemes are common flow solvers, they can be found very easily from the literature. Secondly, the purpose of the current research work is to integrate the Chimera grid method into the AMR algorithm. There is not much modification work needed on the flow solver. The only small modification to account for the Chimera holes will be presented in §4.3.2. Thirdly, Quirk has presented Roe's FDS scheme in depth in his PhD thesis¹⁸.

However, it should be pointed out that throughout Quirk's AMR code, the following non-dimensionalisation is employed to provide enhanced numerical accuracy.

$$\begin{aligned}
 \bar{x} &= \frac{x}{L}; & \bar{y} &= \frac{y}{L}; & \bar{t} &= \frac{t}{L/U_{\infty}}; \\
 \bar{u} &= \frac{u}{U_{\infty}}; & \bar{v} &= \frac{v}{U_{\infty}}; & \bar{\rho} &= \frac{\rho}{\rho_{\infty}}; \\
 \bar{E}_t &= \frac{E_t}{\rho_{\infty} U_{\infty}^2}; & \bar{P} &= \frac{P}{\rho_{\infty} U_{\infty}^2}; & \bar{T} &= \frac{T}{T_{\infty}}.
 \end{aligned} \tag{2-18}$$

where L is the reference length used in the definition of the Reynolds number, i.e. by setting $L=1\text{m}$ the normalised lengths are physical lengths and the Reynolds number is the unit Reynolds number.

The equation of state has become,

$$\bar{P} = \frac{\bar{\rho} \bar{T}}{\gamma M_{\infty}^2} \tag{2-19}$$

And Sutherland's law,

$$\bar{\mu} = \bar{T}^{\frac{3}{2}} \left(\frac{1 + 110/T_{\infty}}{\bar{T} + 110/T_{\infty}} \right) \tag{2-20}$$

These definitions and equations directly related to the development (e.g. body motion) and application (input parameters) of our solution adaptive Chimera grid scheme.

2.4 Modification of the AMR Code

As the first step of this project, we start with the modification and improvement of Quirk's original AMR code, to grasp how the AMR algorithm is implemented and find the

best way to make it applicable in a Chimera grid system. As we have seen, Quirk's AMR code inherited the nature of Berger's AMR strategies. The refinement, adaptation and integration processes must be implemented recursively. So it is very subtle to implement but not easy to support, modify or improve. However, in order to properly incorporate the Chimera grid scheme into the AMR code, the following main modification work on Quirk's AMR code has to be done.

- (1) It is required that the AMR code be suitable to run a multi-domain Chimera grid system.
- (2) Bitwise arrays and operations in Quirk's AMR code have to be replaced so that more boundary types, such as overset and interface boundaries, can be applied. It is necessary to modify the AMR code so that it can handle large grids. Grid size is currently limited to 255×255 cells.
- (3) We hope to develop a general-purpose code, so some problem set-up subroutines and data format transfer procedures need to be modified or developed.
- (4) The flow solver has to be modified to account for the Chimera holes.
- (5) The mesh refinement and adaptation must be available only in valid regions. In a Chimera hole, it would not make sense to employ the AMR algorithm.
- (6) Finally, inter-grid communication has to be carried out at the same time level.

As the AMR code is not easy to modify or improve, great care has to be taken at every step. In the following three sub-sections, the first two parts of the modification work will be discussed. The remaining four parts will be discussed in detail in §4.3.

2.4.1 Extension to Multi-Domain

Before incorporating the Chimera grid scheme into Quirk's AMR code, The AMR code itself must be able to run a multi-domain Chimera grid system and manage the data in different sub-domains. Therefore, a lot of modification work on variables, arrays, loops, subroutines and even code structure has to be done with great care.

For the sake of minimising the modification work on Quirk's AMR code, and to integrate the Chimera grid scheme with the AMR algorithm in a simple way, the following methodologies have been employed.

- Extend most of the arrays with one extra dimension of size $Ndmax$. For example, the geometrical data $x(IJ)$ will be extended to $x(Nd, IJ)$, and the flow solution $W(Eqn, IJ)$ will be extended to $W(Nd, Eqn, IJ)$. Therefore, the geometrical data and flow variables in different domains are stored in a linked list fashion but separately.
- Extend some variables to one dimensional arrays of size $Ndmax$. For example, the logical variable *Adapting* (switch for adaptive or non-adaptive calculation) is replaced by *Adapting(Nd)*, *Euler* (switch for viscous or inviscid calculation) is replaced by *Euler(Nd)*. The integer variable $Lmax$ is replaced by $Lmax(Nd)$. This treatment allows the different sub-domain parameters to be specified separately. Therefore, some sub-domains might be adapted, $Adapting(Nd)=True.$, while others are not. In different sub-domains, adaptation levels $Lmax(Nd)$ and flow physics $Euler(Nd)$ can be also set differently. This capability can provide good flexibility and high spatial resolution for a viscous flow around complex configurations.
- Add a parameter Nd for most of the Subroutines and Functions. For example, the Function $INDEXgij(GRD, I, J)$ will be extended to $INDEXgij(Nd, GRD, I, J)$ so that the cell index pointer IJ for different domain can be calculated by the same Function.
- A loop is added into the main program (Table 2-4) for the different domains to apply the AMR algorithm and to integrate the flow solution separately.

2.4.2 Replacement of the Bitwise Operation

In order to save memory, in Quirk's original AMR code there are two bitwise arrays, $BDYTYPE(Iside, IJ)$ and $CNRTYPE(Iside, IJ)$, which are used for storing each mesh side boundary conditions. Using the bitwise operation, four shifted parameters, boundary type, mesh index, dimensions i and j are put together and stored in a single word (32bit) as follows:

$$\underbrace{iiiiiii}_{8} \quad \underbrace{jjjjjj}_{8} \quad \underbrace{gggggggggggg}_{12} \quad \underbrace{ttt}_{4} \quad (2-21)$$

where t = boundary type
 g = mesh index GRD
 i and j are dimensions.

To extract each of these parameters from *BDYTYPE* or *CNRTYPE* logical *AND'ing* and *SHIFT'ing* are used.

According to this treatment, t has 4 bits and both i and j only have 8 bits, so the number of boundary types cannot exceed 16, and the cell number in the i and j dimensions must be less than 256. This treatment made the code more difficult to understand and limited the application of more boundary conditions. Furthermore, it may cause gross errors if the dimension i or j exceeds 256, i.e. bitwise overlap. For a large computational problem, this is inevitable.

Therefore, it is necessary to replace the bitwise variables and bitwise operation before combining the Chimera grid scheme and the AMR code. The following is the method:

- Extend *BDYTYPE* and *CNRTYPE* with two extra dimensions of sizes $N=4$ and $Nd=Ndmax$. Here the Nd refers to the Chimera sub-domain index.
- Boundary type to be stored in *BDYTYPE*($Nd, Iside, IJ, 1$).
 Mesh index *GRD* to be stored in *BDYTYPE*($Nd, Iside, IJ, 2$).
 j to be stored in *BDYTYPE*($Nd, Iside, IJ, 3$).
 i to be stored in *BDYTYPE*($Nd, Iside, IJ, 4$).
- Therefore, *BDYTYPE*($Iside, IJ$) is replaced by *BDYTYPE*($Nd, Iside, IJ, N$). Similarly, *CNRTYPE*($Iside, IJ$) is replaced by *CNRTYPE*($Nd, Iside, IJ, N$).
- Replace every occurrence in the code of the bitwise operation.

2.5 Summary

In this Chapter we have introduced Berger's AMR algorithm and Quirk's AMR code. At the end of the chapter some modification details of Quirk's AMR code are reported.

Chapter 2 A Brief Introduction to the AMR Algorithm

These modifications speed up the whole AMRC code development and helped determine how to integrate the AMR with a Chimera grid system. A substantial amount of modification work still needed to be done before it could be incorporated into a Chimera grid system. This will be further discussed in Chapter 4.

In writing some parts of this Chapter, the author has drawn on the excellent text of James Quirk's PhD thesis¹⁸ and Henry Neeman's PhD thesis¹²³ which can be consulted for a more detailed presentation.

Before ending this chapter, it should be noted that the current scheme is to integrate the Chimera grid method into Quirk's AMR code. Therefore, from this point on, I shall use the Acronym AMR solely to indicate Quirk's AMR code, unless otherwise stated.

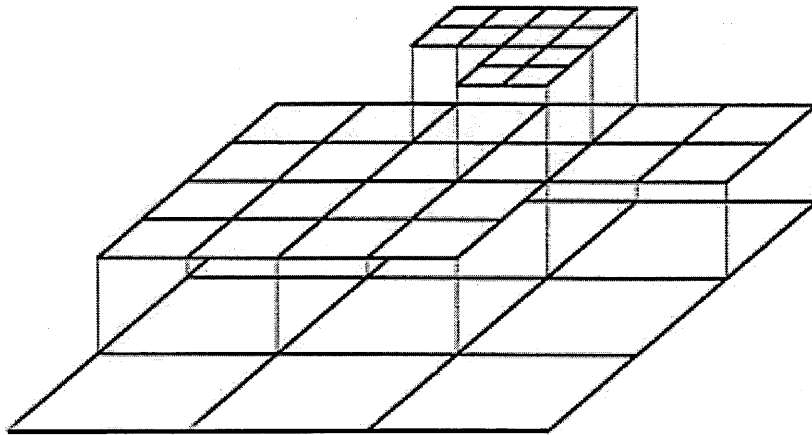


Figure 2-1. Hierarchical grid structure

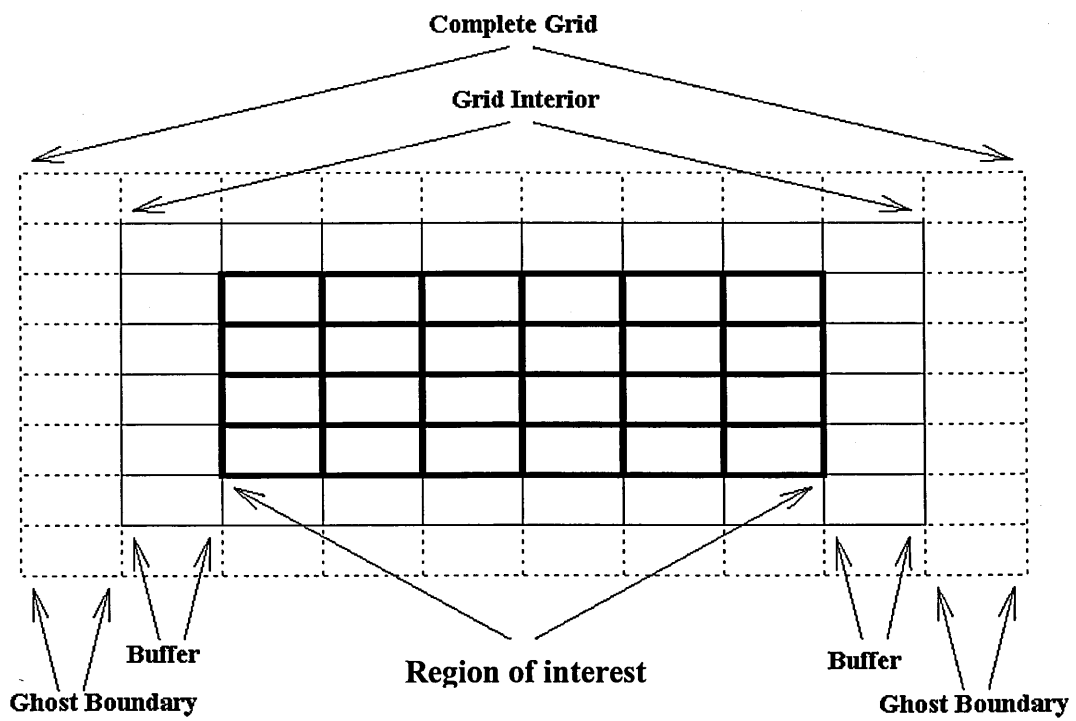


Figure 2-2. Parts of a mesh

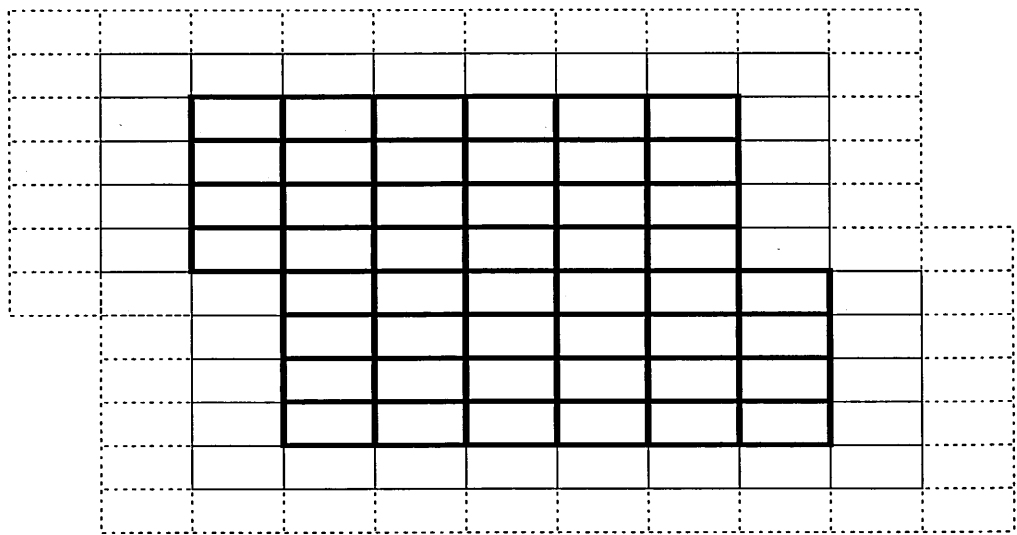


Figure 2-3. Abutting meshes with partially absorbed buffer regions

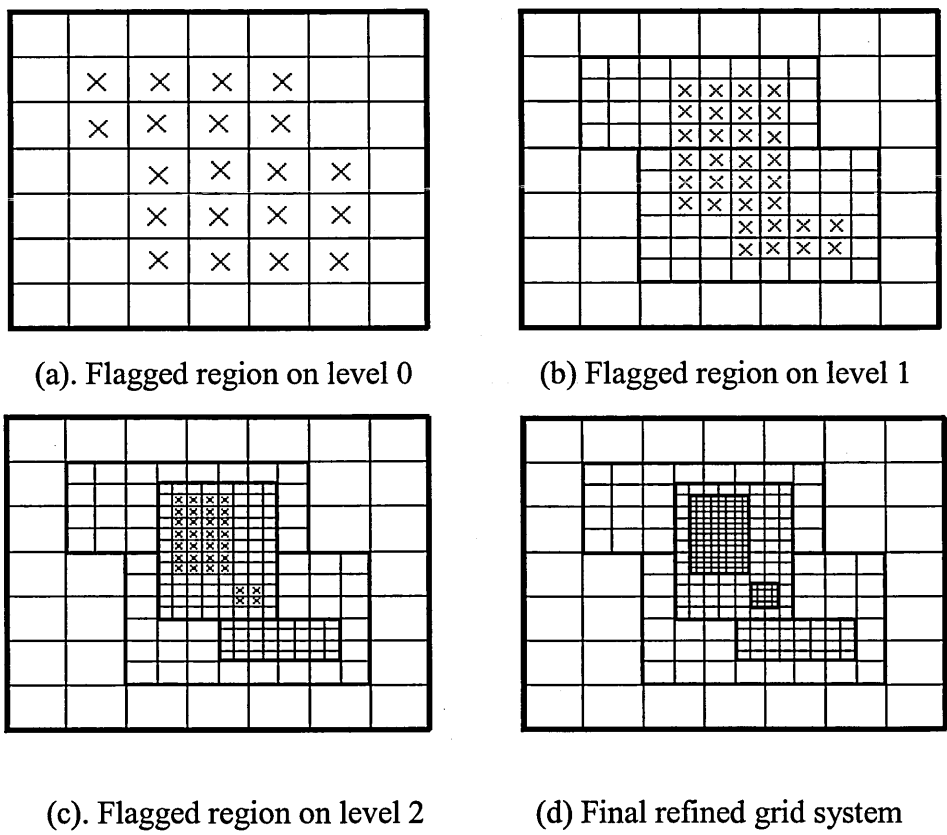


Figure 2-4. Local mesh refinement

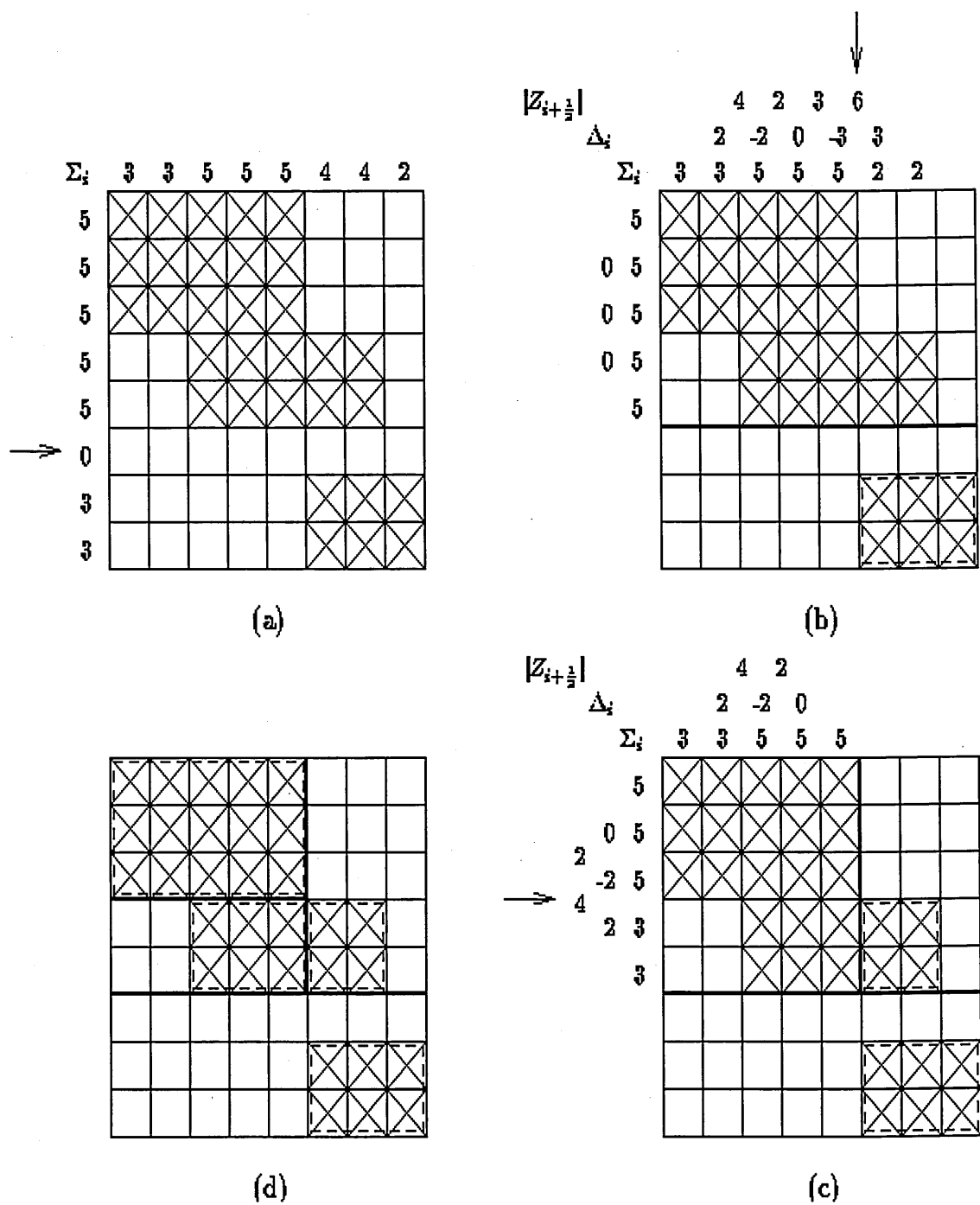


Figure 2-5. Example of Berger's clustering algorithm (clockwise from upper left)[123]

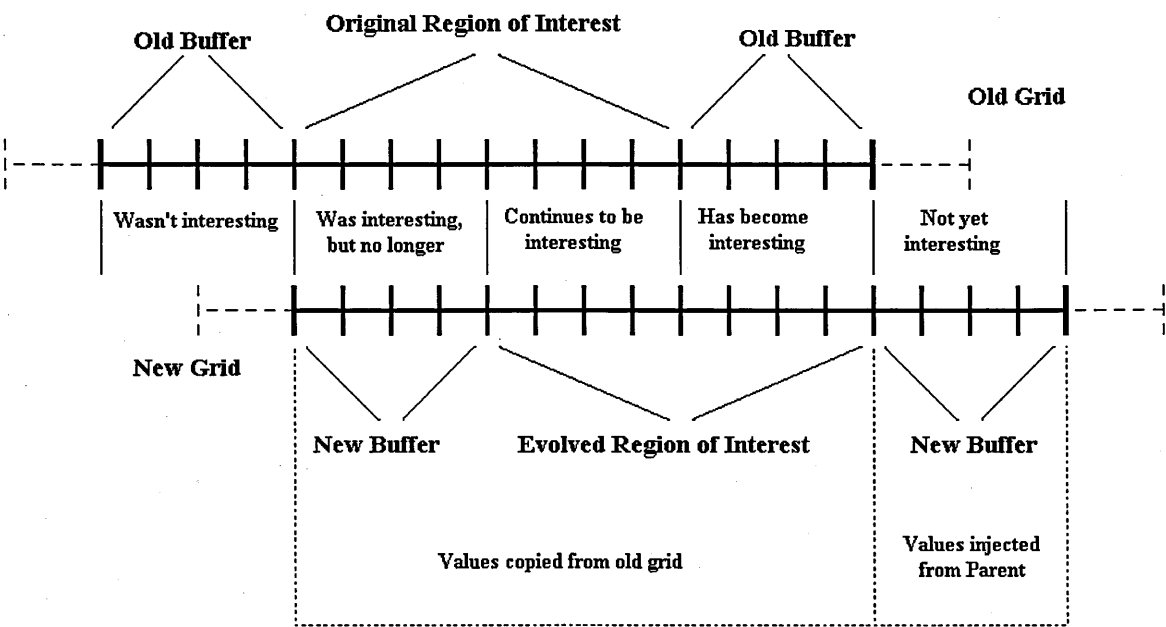


Figure 2-6. Transferring solution to the new finer meshes [123]

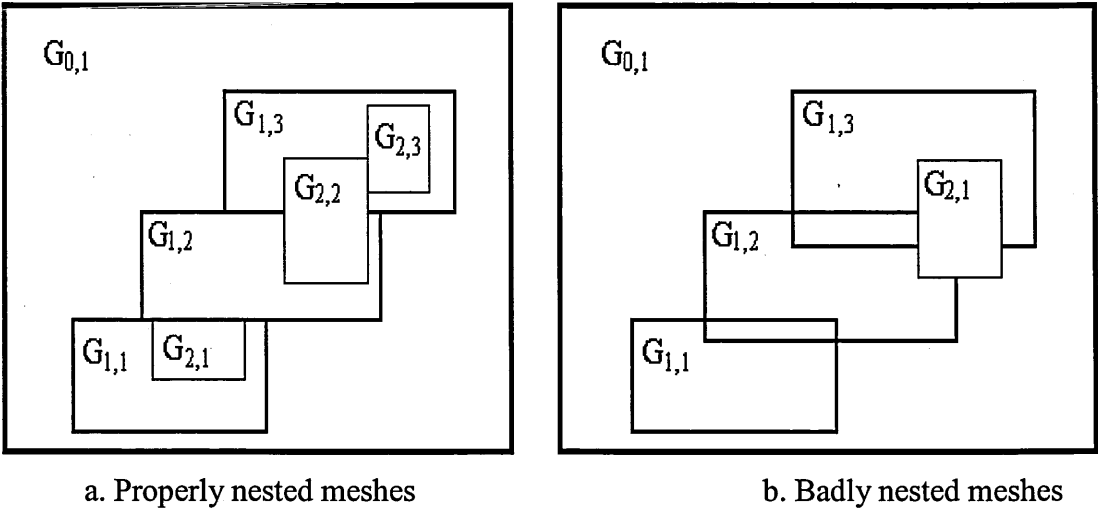


Figure 2-7. Mesh nesting restriction

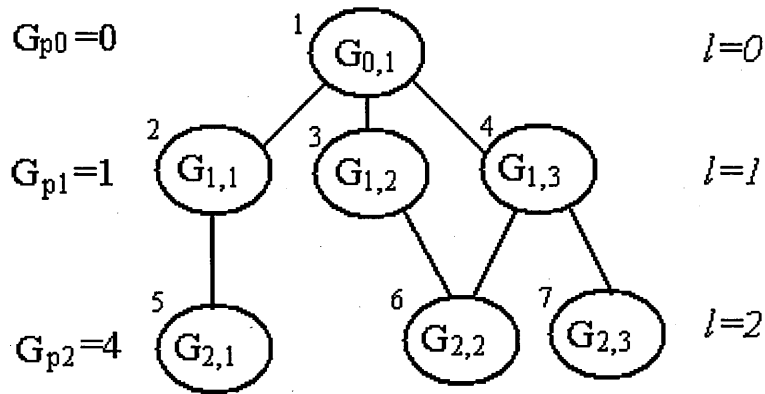


Figure 2-8. Mesh index

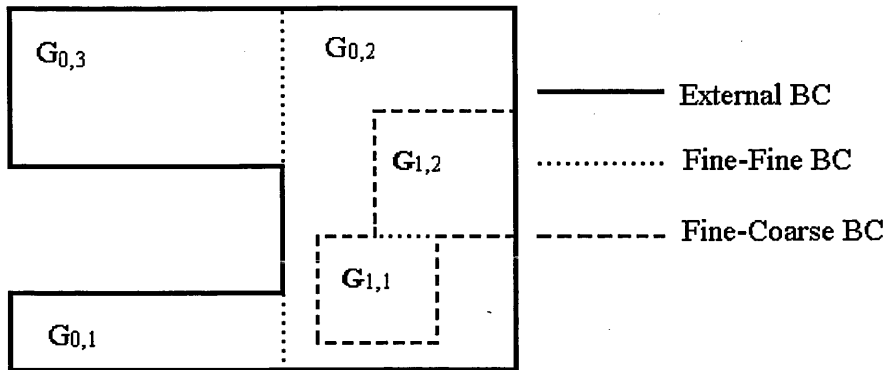


Figure 2-9. Different types of mesh boundary

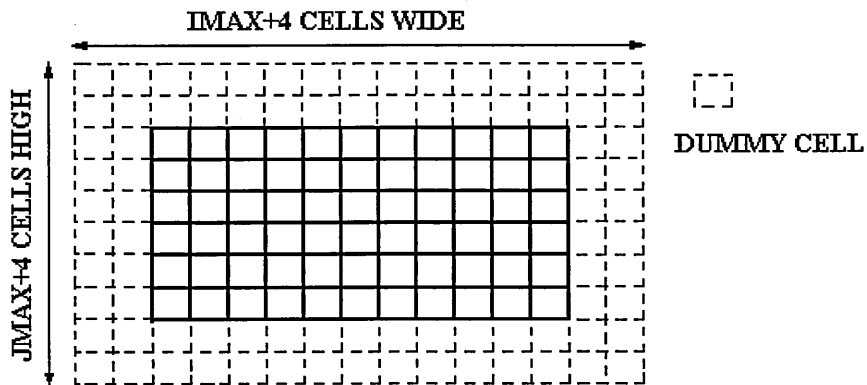


Figure 2-10. Dummy cells surround each mesh

Chapter 3

Fully Automated 2D Chimera Grid Scheme

3.1 Introduction

This chapter covers the development of a fully automated 2D Chimera grid scheme. It provides a full description of all the numerical methodologies and algorithms we have employed in this scheme. In particular it deals with the domain relationship description, hole cutting, stencil identification, interpolation coefficients, grid and body moving. The fully automated feature is one of the major considerations. This makes the present Chimera scheme a powerful tool for flow simulations with moving bodies. In the following sections, both the basic concepts and major components of this Chimera grid scheme will be introduced in detail.

3.2 Overview

The Chimera grid method is based on the early work of Sterger, Benek, Dougherty et al¹⁻⁶. It is an immensely flexible technique, which provides a conceptually simple method for grid generation. In the case of complex problems that involve multiple moving bodies, great flexibility and very good grid quality can be achieved by using this methodology. In this section, I start with a brief introduction on the basic concepts and main procedures of the Chimera grid methodology, followed by a summary of the desirable properties related

to our Chimera grid scheme and an overview of the structure chart of our Chimera grid scheme.

3.2.1 Basic Concepts and Main Procedures

The Chimera grid method is basically a domain de-composition technique which uses two or more simple overset sub-domains to mesh complex configurations. This method has a number of advantages over other grid methods, in particular,

- It can easily mesh complex geometries and multiple bodies.
- Different flow physics can be defined in different domains.
- Different solution algorithms can be applied in different domains.
- It can easily simulate moving body problems.

The basic idea of this grid method is also easy to understand. Figure 3-1 illustrates the Chimera grid method and shows how it meshes either complex geometries or multiple bodies. In this example, a store is supposed to be separating from an airfoil.

Using the Chimera grid method, the above problem can be solved as below.

- Firstly, the computational domain can be simply mapped by D_1 ignoring the existence of the airfoil and store, and then domains D_2 and D_3 (or more for simplicity in grid generation) used to mesh the airfoil and store separately. This procedure is known as **domain-decomposition**. Domains D_2 and D_3 might intersect each other at the beginning of separation and they overset D_1 . During the separation, D_3 will move together with the store, and it is not necessary to re-generate the grid system. This is a very important feature for modelling moving body problems.
- The second step is, inside D_2 and D_3 to define two closed curves C_2 and C_3 , and then cut two holes in D_1 , that is, flag and blank out all cells which fall in the closed curves C_2 and C_3 . Next cut a hole inside D_2 , since D_3 overlaps D_2 as well. This is known as **hole cutting**.

- The third step is to locate the hole boundary cells, normally called fringe cells, in D_1 , and to find their nearest donor cells inside D_2 and D_3 . Searching for donor cells inside D_1 (or D_2 or D_3) for the outer boundary cells of D_2 and D_3 is then required. This is known as **point searching**.
- Then solve each of the three domains individually.
- Finally carry out the **inter-grid communication** at every time step. Transfer the flow solution on the Chimera common boundary.

These are the basic strategies and procedures associated with the Chimera grid methodology. Some basic definitions associated with the Chimera grid method, e.g. hole boundary and outer boundary are illustrated in Figure 3-2.

3.2.2 Desirable Properties

In this study, the Chimera grid scheme will be integrated with the AMR algorithm to simulate multiple moving body flow. Therefore, we expect our Chimera grid scheme to be a versatile, efficient and fully automated scheme. The following is a list of the desirable properties which are the major considerations during the development of this scheme.

- This Chimera grid scheme has to be easily integrated with Quirk's AMR code. Since Quirk's AMR code was coded in Fortran 77, it is not easy to do much code modification. For the sake of minimum modification work on the AMR code, the Chimera grid scheme has to be designed taking the AMR code into account.
- Most of the tasks associated with this Chimera grid scheme, such as hole cutting and stencil identification, must be performed in a fully automated mode. This allows our scheme to simulate moving body flow easily.
- There is no unreasonable restriction placed on the domain arrangement. The sub-domain could be arranged in any arbitrary manner. This allows more complicated configurations, such as multiple body separation, multiple impact, moving body flows etc, to be solved by our Chimera scheme.

- Range and point searching must be efficient. For such a scheme, the inter-grid communication procedure is likely to be the most computationally expensive procedure. Searching efficiency has to be improved by all means possible.
- The data management and storage must be efficient and compatible with Quirk's AMR code. This property minimises the modification work on the AMR code and ensures that the integration of the Chimera grid scheme with the AMR code does not compromise the advantages of the AMR algorithm.
- This Chimera grid scheme can be extended into three dimensions very easily. The real world of fluid dynamics is mainly a three-dimensional world and it is in three-dimensional flow problems that the great flexibility of the Chimera grid scheme can be fully shown. This scheme certainly needs to be extended into three dimensions in the future.
- The scheme must be general purpose. It can be applied to a variety of problems.

3.2.3 Code Structure Chart

Figure 3-4 shows the construction and the main parts of our Chimera grid scheme. There are six main modules, divided according to their functions. The first module, is used to input multiple domain grid data files, specify boundary conditions and initial conditions, and to build up the Chimera grid system. The last module is used to obtain the flow solution and to employ the AMR algorithm inside a Chimera grid system. These two modules will be discussed in detail in Chapter 4. In the rest of this chapter, we shall present the other four modules as well as the data structure and storage we have used for this Chimera grid scheme.

3.3 Data Structure and Storage

One of the most important stages in the development of this Chimera grid scheme is to determine the data structure and storage, because both the Chimera grid scheme and the

AMR algorithm require precise organisation and management the data in different domains or in different meshes at different levels. For such a scheme, the data management must store data and transfer flow solutions among the domains (in the Chimera grid system) and grids as well as meshes (in the AMR hierarchical system), as sketched in Figure 3-3. It must be reasonable, efficient and compatible with Quirk's AMR code.

In Quirk's AMR code, for memory efficiency, all flow variables and the geometric data are stored in a series of linked lists. Each list consists of a continuous set of storage elements that may be viewed as a one-dimensional array. For example, the x co-ordinates of all grid points in different meshes at different levels are stored in array $X(IJ)$. Here the index pointer, IJ , of a cell (i, j) of a mesh GRD in the linked list is given by equation (2-14).

During the development of our Chimera grid scheme, this data structure is retained to save memory and to minimise the amount of modification to the original AMR code. One more dimension is added so that the different sub-domain data can be stored in a linked list fashion but separately. For example, the x co-ordinates of a sub-domain Nd are stored in the following way,

$$X(Nd, IJ) \tag{3-1}$$

where Nd indicates the Chimera sub-domain index.

This allows that for a given sub-domain, the data structure and storage is exactly the same as that in Quirk's AMR code. Inside the AMR code's main program, adding a loop ($Nd=1$ to $Ndmax$) and adding one more element (domain index Nd) into most of the Subroutines and Functions easily extends the AMR code to run multiple domain problems (see Chapter 4 for a more detailed description).

This treatment allows the Chimera grid scheme to be incorporated into Quirk's AMR code in a simple way and makes the different domain data easy to manage.

3.4 Description of the Domain Relationship

In the Chimera grid method, properly and precisely describing the domain relationship is another key issue, especially for a fully automated Chimera scheme. Tree

data structure is a mature technique widely used in traditional Chimera grid schemes. However, this method places an unreasonable limitation on the domain arrangement. It uses a hierarchical structure to describe the sub-domain relationship. By using this method, the sub-domain is constrained to overlay its parent domain, as shown in Figure 3-5. Thus it is not suitable for our scheme.

In this study, we expect that each domain in a Chimera grid system can be arranged in an arbitrary way as necessary. The different domains can overlap, connect or intersect with each other, and even can intersect with a solid body, see Figure 3-6. For this reason, a graph data structure is introduced to describe the domain relationship and an orthogonal list is used to store that relationship.

3.4.1 Graph Data Structure

The graph data structure is becoming a popular method in computer science. Here, it fits very well the requirement of describing, linking and managing the different Chimera sub-domains. Using the graph data structure, the Chimera domain relationship can be precisely described, and allows the sub-domains to intersect with each other in an arbitrary way.

Figure 3-6 is an example which illustrates and shows how this methodology is used to describe the domain relationship of the Chimera grid system. The domain index number is treated as the vertex. The arc in this digraph describes the relationships between the different domains. In this example there are four domains. Sub-domains D_2 , D_3 and D_4 overset D_1 . Sub-domain D_4 intersects with D_2 and D_3 . Using the graph data structure, the domain relationship can be described as Figure3-6b. Since sub-domains D_2 and D_3 neither overlap nor intersect each other, there is no *arc* between the vertex D_2 and D_3 .

This method allows the sub-domains can be intersect with each other in an arbitrary way, like domains 2, 3 and 4. The restriction sketched in Figure 3-5 is completely removed. Instead, each domain is considered an independent entity which can have hole boundary cells (or fringe boundary cells) and/or interpolation boundary cells. Each domain communicates with one or more domains in the Chimera grid system.

3.4.2 Orthogonal List Storage

Another methodology used in describing the domain relationship is orthogonal list storage. Two 2-dimensional arrays, $DMA(N, 4)$ and $DMV(N, 3)$, are used to store the information and relationship of the arc and vertex respectively, as sketched in Figure 3-6a.

In the DMV array, the first element, *vertex*, indicates the vertex index, i.e. the domain index number of a Chimera grid system. The second one, *firstin*, refers to the first *arc* index which points to this vertex (from it we can trace the first sub-domain index number on which the current sub-domain overlays or interacts with). And the third element, *firstout*, refers to the first *arc* index which originates from this vertex (from it we can trace the first sub-domain number which overlays or interacts with the current sub-domain). A zero value of the *firstin* or *first out* element indicates that there is no *arc* points to, or originating from, this vertex. Using this information, the code can easily recognise from which domain to search for a donor cell.

In the DMA array, the first element refers to the vertex from which the arc originates (*arc* tail), and the second one refers to the vertex to which the arc points (*arc* head). The third element refers to the next arc index, which points to (*arc* head) the same vertex, and the last element refers to the next arc index, which originates from (*arc* tail) the same vertex. It clearly describes how many domains exist and which domain overlies a given domain.

As an example, the final relationships of the Chimera grid system shown in Figure 3-6a are stored into DMA and DMV arrays, see Figure 3-7b. It looks very complicated, but if we precisely design the code, it becomes very simple to build up and link the domain relationships. For this example, using a loop starting from 1 to 3, we only need to input domain indexes 2, 3 and 4 which overset D_1 , a domain index 4, which oversets D_2 , and input domain index 4 again, which oversets D_3 as well. The AMRC code automatically sorts out this relationship.

This method allows the sub-domains to intersect each other, as for domains D_2 , D_3 and D_4 , and to intersect a solid body, such as D_4 .

This method can also greatly improve the hole-cutting and point searching efficiency. From the *DMA* and *DMV* arrays, we can easily sort out from which domain to locate a donor cell for a given recipient cell (a fringe cell or an outer boundary cell). It is not necessary to sweep out all domains in a Chimera grid system.

3.5 Automated Hole-cutting Procedure

In a Chimera grid system, Chimera-holes need to be cut in regions that overlap solid bodies or any other non-flow regions that belong to the other domains of the grid system. Hole cells must be blanked or excluded from the flow field solution. The presence of moving bodies continually changes the relative position of the overset domain during the flow simulation. In such cases, it is difficult for the user to provide apriori all the input data necessary for the hole-cutting or carry out the hole-cutting manually. Therefore, a reasonable yet very important requirement for the hole-cutting procedure is that all the tasks associated with hole-cutting have to be performed in a fully automated mode.

3.5.1 Implementation Sequence

Since we use two 2-dimensional arrays, $DMA(N, 4)$ and $DMV(N, 3)$, to describe and store the information and relationship of different domains in a Chimera grid system, a fully automated Chimera hole cutting procedure can be easily fulfilled. A relatively simple code fragment, see Table 3-1, shows the implementation sequence.

From this code fragment, we can see that the arrays, *DMV* and *DMA*, have stored sufficient information so that the Chimera hole cutting procedure can be implemented automatically and efficiently. It can directly locate the overlap domain, $Nd2$, (for defining the hole cutting curve) and the underlay domain, Nd , (for flagging hole cells).

The three sub-procedures which appear in the hole cutting procedure will be discussed in the following three sub-sections individually.

Table 3-1. Quasi-code fragment to the Chimera Hole-cutting Procedure

```

Procedure Hole_cutting
  For  $Nd=1$  to  $Ndmax$  {
     $IE=DMV(Nd, 3)$ 
    Do While ( $IE \neq 0$ ) {
       $Nd2=DMA(IE, 2)$ 
      Define_hole_curve( $Nd2$ )
      Flag_hole_cell( $Nd, Nd2$ )
       $IE=DMA(Nd, 4)$ 
    }
  }
  For  $Nd=1$  to  $Ndmax$  {
    Flag_fringe_cell( $Nd$ )
  }
End Procedure

```

3.5.2 Defining Hole-Cutting

The primary function of the hole cutting is to blank out a region of a domain which may fall within other domains. As shown in Figure 3-8, inside D_1 , it is necessary to blank out the region covered by domain D_{2-1} and D_{2-2} . This is one kind of hole in our Chimera grid scheme. For simplicity, we define it as D type hole (D-hole), since it is introduced by other domains.

The D-hole cells can be defined through a creation boundary, which consists of a curve or a group of curves. Since a common or overlap region is required to match the solutions across the different domains, typically, an internal closed hole-cutting curve is defined inside the overset domain, as illustrated in Figure 3-8. In our scheme, for simplicity, the overlap depth is defined by a buffer cell number. When constructing the domain relationship, it is also necessary to specify the buffer cell number, $Nbuf(Nd)$, for each overset domain. This is the only parameter associated with hole-cutting that needs to be input. This treatment also makes the overlap depth adjustable.

Figure 3-8 is an example which illustrates how to automatically define a closed hole-cutting curve. For an overset boundary, the curve will shrink by $N_{buf}(Nd)$ cells from the boundary and form an overlap region. Then the code automatically defines a closed hole-cutting curve.

Of course, if an overset domain covers solid bodies contained in other domains, as for D_4 shown in Figure 3-6, then a hole must be blanked out from the overset domain as well. This is another kind of hole in our Chimera grid scheme. In order to distinguish from a D-hole, we define this kind of hole as an S type hole (S-hole), which is introduced by solid bodies. Certainly, the solid boundaries can be used as the hole cutting curves for blanking out the S-hole cells.

3.5.3 Hole Cell Flagging Algorithm

The purpose of the closed hole-cutting curve is to identify points that are within this curve, i.e. to blank out all hole cells. Once the closed hole-cutting curve has been specified and constructed, a search must be made to determine and flag those points within it. The difficulty associated with hole cutting is searching and assessing the interior points.

There are various techniques that may be employed to assess whether a point is within or outside a closed curve. For more details of the different methods, refer to references^{2-5, 131}. In our scheme, we chose the “non-exterior winding rule”¹³¹ because this method is most applicable to our class of problem. (Note that it is not suitable for three-dimensional problems). This method uses an imaginary marker placed at the point of interest attached to an imaginary length of string, see Figure 3-9. The free end of the string is used to trace a closed curve, the hole-cutting curve, and if the string has not wound around the marker then the point is external, otherwise, the point is internal. This rule can be significantly optimised by avoiding trigonometric inversions by examining the sign of Δx and Δy (where $\{\Delta x, \Delta y\}$ represents the vector from the point of interest to a curve vertex) and determining which quadrant the vector is in. Assessing winding on a quadrant

basis is vastly less expensive than on an angular basis although care must be taken when on inter-quadrant boundaries.

In order to improve searching efficiency, the searching region is further restricted. While determining the closed hole-cutting curve, a bounding box $\{(x, y)_{min}, (x, y)_{max}\}$ is generated automatically. Therefore, only those points inside this bounding box need to be inspected using the above method.

Once a point is determined as an interior point, it will be flagged as a hole point and will be blanked out from the flow field solution. A mature and popular method for storing the hole information in the Chimera grid system is to establish an array, IBLANK, for each domain. The elements of IBLANK are set to 0 if the point lies within a hole or 1 if it does not. i.e.

$$IBLANK = \begin{cases} 1 & \text{if a cell is not blanked,} \\ 0 & \text{if a cell is blanked.} \end{cases} \quad (3-2)$$

3.5.4 Fringe Cell Flagging Algorithm

After blanking out all the hole points inside the bounding box $\{(x, y)_{min}, (x, y)_{max}\}$, a “nine-point stencil” algorithm is applied to flag the Chimera hole boundary cells, i.e. fringe cells. Figure 3-10 illustrates this process. If the centre grid point ⑤ is not blanked, we have to check its eight surrounding neighbour points. If there is at least one blanked point, then point ⑤ will be flagged as a fringe point. The corresponding cell is called a fringe cell. Its flow variables are provided by another domain. The IBLANK element of the fringe cell will be set as $-Nd$ for easy recognition and further reference. Consequently, the information stored in the IBLANK array will become as:

$$IBLANK = \begin{cases} 1 & \text{if a cell is not blanked} \\ 0 & \text{if a cell is blanked} \\ -Nd & \text{if a cell is a fringe cell} \end{cases} \quad (3-3)$$

3.6 Inter-grid Communication

Inter-grid communication is likely to be the most important procedure in a Chimera grid scheme. The accuracy of a flow solution largely depends on whether or not the flow information can be properly and correctly transferred between the different sub-domains. The computational cost also largely depends on the efficiency of identifying recipient–donor stencils. This is especially true for our scheme. Since it is to be used for simulating shock hydrodynamic flows and multiple moving body supersonic or hypersonic flows, strong discontinuities have to be able to pass through the Chimera overset boundaries. Since the AMR algorithm will be applied in each sub-domain for enhancing resolution, the point searching efficiency will be a key issue in such a scheme.

In this section, we will present the main procedures of inter-grid communication and the strategies we have employed to overcome the issues mentioned above.

3.6.1 Recipient–Donor Stencil

In a Chimera grid system, see Figure 3-2, the flow variables of the fringe cells are provided from another domain. In addition, since the separate domains are to be treated as independent entities, boundary conditions must be supplied to each domain. Thus the outer boundary conditions of the Chimera overset boundaries that are not coincident with those of the computational domain (this kind of boundary is called an "overset boundary" in our scheme) have to be provided from the other domain as well. These two kinds of cells (fringe cells and overset boundary cells) are defined as recipient cells.

Before carrying out the inter-grid communication, it is necessary to sort out the recipient-donor stencil.

Identification of recipient cells is quite straightforward. The first kind of recipient cell, fringe cells, can be identified when they are flagged, see Table 3-1 and §3.5.4. The

second kind of recipient cell, overset boundary cells, can be identified more easily, since they are boundary cells.

Searching the nearest donor cells for each recipient cell is quite time consuming. In particular, in our scheme, after applying the AMR algorithm, the meshes are refined dynamically at every time step. Hence, it is necessary first to check every mesh at each level in which the nearest donor cell might be located and to carry out the donor searching at every time step as well. Sweeping through all domains and all meshes definitely would find the nearest donor cell but it is expensive. The “stencil walk” algorithm presented in reference⁵ is an efficient donor cell searching algorithm. Unfortunately, it is not suitable for our adaptive meshes. During the development of the AMRC code, the following strategies were employed for improving the searching efficiency.

- (1) The first step is to locate the donor domain to which the donor cell might belong. It is necessary to judge if a domain is a possible donor domain. Again, the information stored in the arrays, *DMV* and *DMA*, can be used to directly point to the possible donor domains.
- (2) For some cases, there are more than one candidate donor domains. For example, in Figure 3-6a, for a given fringe cell (recipient cell) in domain D_1 , domains D_2 , D_3 , D_4 are all possible donor domains. So it is necessary to judge whether the recipient cell falls in a domain and to judge whether it is necessary to carry out further donor cell searching from this domain. A similar procedure to hole cell flagging, see §3.5.3, is used with the domain boundary being used as the closed curve. This method excludes all other domains in which the recipient cell does not fall.
- (3) After locating the donor domain, the third step is to locate the donor mesh at high levels. Starting from the highest level, L_{max} , the same procedure as step two is used to find the finest donor mesh.
- (4) Once the finest donor mesh is found, sweep through all mesh cells to find the nearest donor cell. Normally, the adaptive finer mesh size is very small. Therefore it is not necessary to apply a more complicated searching algorithm.

Tables 3-2 and 3-3 show the code fragments of the identification procedure of the Recipient–Donor Stencil. Where Ndr , Ndd and $GRDd$ indicate the recipient domain, donor domain and donor mesh indexes respectively.

Table 3-2. Quasi-code fragment to the Recipient–Donor Stencil Procedure-1

```

Procedure Searching_Donor_Cell      /* for fringe cell */
  For  $Ndr=1$  to  $Ndmax$  {
     $IE=DMV(Ndr, 3)$ 
    If ( $IE \neq 0$ ) {
      Sort_Fringe_Cell( $Ndr$ )      /* if necessary */
      Do While ( $IE \neq 0$ ) {
         $Ndd=DMA(IE, 2)$ 
        Find_Donor_Domain( $Ndr, Ndd$ )
         $IE=DMA(IE, 4)$ 
      }
      Find_Donor_Mesh( $Ndr, Ndd, GRDd$ )
      Find_Donor_Cell( $Ndr, Ndd, GRDd$ )
    }
  }
End Procedure

```

Table 3-3. Quasi-code fragment to the Recipient–Donor Stencil Procedure-2

```

Procedure Searching_Donor_Cell      /* for overset cell */
  For  $Ndr=1$  to  $Ndmax$  {
     $IE=DMV(Ndr, 2)$ 
    If ( $IE \neq 0$ ) {
      Sort_Overset_Cell( $Ndr$ )      /* if necessary */
      Do While ( $IE \neq 0$ ) {
         $Ndd=DMA(IE, 1)$ 
        Find_Donor_Domain( $Ndr, Ndd$ )
         $IE=DMA(IE, 3)$ 
      }
      Find_Donor_Mesh( $Ndr, Ndd, GRDd$ )
      Find_Donor_Cell( $Ndr, Ndd, GRDd$ )
    }
  }
End Procedure

```

This method excludes all other domains in which the recipient cell does not fall. The searching regions are largely restricted. Therefore it can greatly improve the searching efficiency, especially when many sub-domains are used to map a complex configuration or multiple bodies.

3.6.2 Flow Variable Transfer

The inter-grid communication between recipient-donor cells is achieved by interpolation of the flow variables from the donor cell and its neighbour cells. A few different interpolation methods exist. The bilinear interpolation method is the most popular for a two-dimensional problem. For the majority of calculations, we chose this method to transfer the flow solution from donor cells to recipient cells. The detailed formulae of the bilinear interpolation are presented in **Appendix A** of this thesis.

Using this method, a cell containing a point at which a function value is to be interpolated must first be transformed into a square, see Figure A-1, and the values at four vertexes must be known. Therefore, after sorting out the recipient-donor stencil, a quadrilateral has to be formed by using the donor cell and/or its neighbour cells.

It should be noted that Quirk's AMR code is a cell centre scheme. The flow variables are stored at the cell centre, while the geometric data is stored at the cell vertex, i.e. grid data (x, y) indicates the grid node co-ordinate. Whilst searching the donor cell, if we select a cell centre co-ordinate (for both recipient and possible donor cells) to judge if a cell is the nearest donor cell (indeed, in Quirk's AMR code there is a Function which can be used to calculate a cell centre co-ordinates), a problem may arise. In order to make sure that the recipient cell falls in the closest interpolation quadrilateral, we should only use four neighbour cells to form the interpolation quadrilateral, as shown in Figure 3-11. An alternative method is that for a given recipient cell, knowing its cell centre co-ordinate we search for the nearest donor cell vertex co-ordinate. In this way, we can use the four surrounding cells to form the interpolation quadrilateral, as shown in Figure 3-12. Compared with Figure 3-11, we can see that the interpolation quadrilateral shown in Figure

3-12 is much closer to the recipient cell and the nearest donor cell is one of the four surrounding cells. Therefore, when coding the donor cell searching procedure, the latter method mentioned above should be applied.

After finding the interpolation coefficients ξ and η for each interpolation cell, see **Appendix A** of this thesis, the flow solution for the recipient cell can be achieved by the following bilinear interpolation equation.

$$W_R = \sum_{i=1}^4 (-1)^{\xi_i + \eta_i} (\xi + \xi_i - 1)(\eta + \eta_i - 1) W_{Di} \quad (3-4)$$

where W_R indicates the interpolated value for the recipient cell,

W_{Di} indicates the values of interpolation cells.

During the development of our Chimera scheme, two other methods have also been applied to transfer flow variables from donor cells to recipient cells. One method is to directly shovel the value of the nearest donor cell to the recipient cell. This method is less accurate mathematically but very simple and more efficient because it is not necessary to calculate the interpolation coefficients ξ and η for each interpolation cell. In addition, since our scheme employs the AMR algorithm, in regions with a high gradient the cells will be refined by a couple of levels. Due to this, the nearest donor cell might be very close to the recipient cell. It is a reasonable option for transferring flow information. According to our experience using this method, a strong discontinuity can also properly cross through the Chimera overset boundary.

The other interpolation method is the first-order Taylor series approximation. It is also simple and has less computational cost than bilinear interpolation.

$$W_R = W_D + \dot{W}_X \Delta x + \dot{W}_Y \Delta y \quad (3-5)$$

where W_R is the required Recipient cell value,

W_D is Donor cell value.

It is quite easy to calculate the local $\dot{W}_x$, $\dot{W}_y$ and Δx , Δy ,

3.7 Moving Body Module

There are many advantages in using the Chimera grid method for modelling moving body problems. A major effort in the development of this scheme is the ability to handle moving bodies. Currently, the moving body capability requires that the body be rigid. The motion of a body can be determined by a three-degree-of-freedom (3DOF free movement for 2D problems, two translations and one rotation) calculation or prescribed by the user.

Basically, in the motion module, there are two processes that must be performed for each time step. First, for free movement, the loads on the moving body must be calculated and then used to determine the location and orientation of the body. This process requires computer code that integrates the 3DOF equations of motion. For prescribed movement, this process is straightforward. Second, the new locations of all moving domains must be calculated according to the body motion.

In this last section of this Chapter, we begin by addressing the normalised motion parameter, then we present the two motion modules (prescribed and free movement), and domain movement.

3.7.1 Dimensionless Motion Parameters

In §2.3.4 we presented the dimensionless parameters employed in Quirk's AMR code. Here is a good place for us to present the dimensionless parameters associated with the body motion calculations.

Currently, we only consider two major kinds of force which act on a moving body. One results from the surface pressure (for viscous flow, the force due to shear stresses should also be taken into account, but is not yet) and the other is due to body gravity. The moments acting on a moving body result from these two kinds of force.

According to the definition in (2-18), for the force due to surface pressure we have,

$$F_p = PA = \bar{P} \rho_\infty u_\infty^2 \bar{A} L^2 \quad (3-6)$$

therefore,

$$\bar{F}_p = \frac{F_p}{\rho_\infty u_\infty^2 L^2} \quad (3-7)$$

and the normalised gravity becomes,

$$\bar{F}_g = \frac{F_g}{\rho_\infty u_\infty^2 L^2} \quad (3-8)$$

The normalised acceleration can be given as,

$$\bar{a} = \frac{aL}{u_\infty^2} \quad (3-9)$$

The normalised mass becomes,

$$\bar{m} = \frac{m}{L^3 \rho_\infty} \quad (3-10)$$

Similarly, for the moment and inertia, we have,

$$\bar{M} = \frac{M}{L^3 \rho_\infty u_\infty^2} \quad (3-11)$$

$$\bar{I} = \frac{I}{L^3 \rho_\infty u_\infty^2} \quad (3-12)$$

3.7.2 Prescribed Movement

The first motion module is prescribed movement. In this module, the translation and rotation motions can be specified separately (but movement parameters must be specified as dimensionless parameters). Currently, a body can be specified to move (translate, rotate or both) under a constant velocity. The calculations of the body movement increment, Δx , Δy or $\Delta \alpha$ is quite straightforward.

3.7.3 Free Movement

For the free movement module, currently, we use a fixed-axis Cartesian co-ordinate system to describe the motion of a free moving body. For a free moving body, six degrees of freedom are required to adequately model real behaviour, that is three translations along the axes and three rotations about the axes. For a two-dimensional movement, only three degrees of freedom are required, two translations along the x and y axes and one rotation about the z axis, see Figure 3-13. Therefore, two force equations and one moment equation need to be given.

Each of the force equations obeys Newton's Second Law,

$$Mass \times Acceleration = \sum Forces \quad (3-13)$$

and applied to our scheme it will be (for simplicity, here we present original formulae of the force and moment equations),

$$\begin{aligned} ma_x &= F_{px} - F_{gx} \\ ma_y &= F_{py} - F_{gy} \end{aligned} \quad (3-14)$$

The terms on the right hand side are:

- (1) force due to surface pressure
- (2) force due to gravity

Similarly, the moment equation is the rotational version of Newton's Second Law,

$$Inertia \times Angular \text{ Acceleration} = \sum Moments \quad (3-15)$$

and thus in our code it will be:

$$I_z \dot{\omega} = F_p d + F_g d \quad (3-16)$$

where d indicate force arm length.

The inertial properties (such as mass and moment of inertia) and initial velocities (U_0 and V_0) of the moving body must be specified before simulation.

Forces due to the surface pressure are integrated from the simulation solution. Moments, acceleration and/or angular acceleration can be calculated according to the above equation.

$$\begin{aligned} U_2 &= U_1 + a_x \Delta\tau & \Delta X &= \frac{1}{2}(U_1 + U_2) \Delta\tau \\ V_2 &= V_1 + a_y \Delta\tau & \Delta Y &= \frac{1}{2}(V_1 + V_2) \Delta\tau \\ \omega_2 &= \omega_1 + \dot{\omega} \Delta\tau & \Delta\alpha &= \frac{1}{2}(\omega_1 + \omega_2) \Delta\tau \end{aligned} \quad (3-17)$$

Then equation (3-17) can be used to determine the new location and orientation of the moving body at the next time step.

3.7.4 Domain Movement

For a moving body problem, some domains will move together with the moving body/bodies. The body movement model might be either prescribed or free motion under the action of aerodynamic forces or moments as mentioned in the above two sub-section. The calculations of the body movement increment, ΔX , ΔY and $\Delta\alpha$ after a time step $\Delta\tau$, for these two motion models are different. However, after obtaining ΔX , ΔY and $\Delta\alpha$, the calculations for the new co-ordinates $(x, y)_{new}$ of a moving domain grid are the same and are

quite straightforward. The co-ordinate translation equation (3-18) and rotation equation (3-19) can be directly used to carry out this calculation.

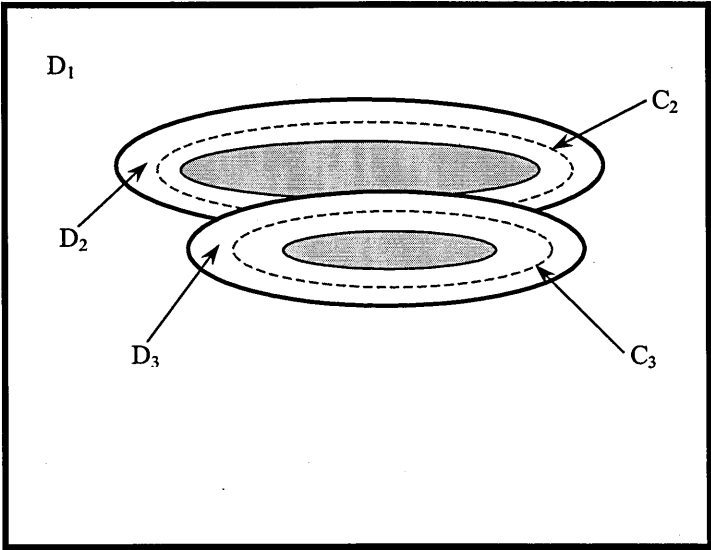
$$\begin{cases} X = x + \Delta X \\ Y = y + \Delta Y \end{cases} \quad (3-18)$$

$$\begin{cases} X = x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \\ Y = -x \cdot \sin(\alpha) + y \cdot \cos(\alpha) \end{cases} \quad (3-19)$$

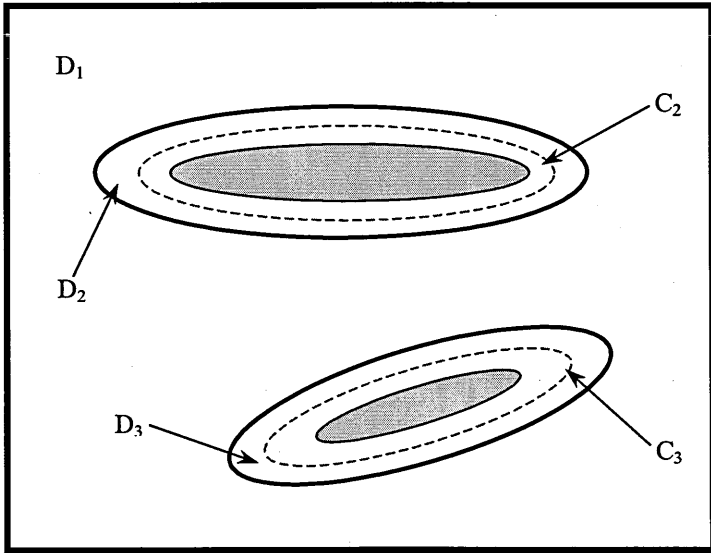
3.8 Summary

A fully automated 2D Chimera grid scheme has been developed and presented in this Chapter. The graph data structure and the orthogonal list storage method have been successfully introduced into the Chimera grid method to simplify the description of the Chimera grid system domain relationship. This treatment not only means that the Chimera sub-domains can be arranged in an arbitrary manner, either overlap or intersect each other, or even can intersect a solid body, but also largely improves the donor cell searching efficiency.

The development of the motion module allows our scheme to greatly reflect the flexibility of the Chimera grid method, and also allows the AMR algorithm to be used in simulating moving body flow.



(a). Before separating



(b). Separating

Figure 3-1 Chimera grid system

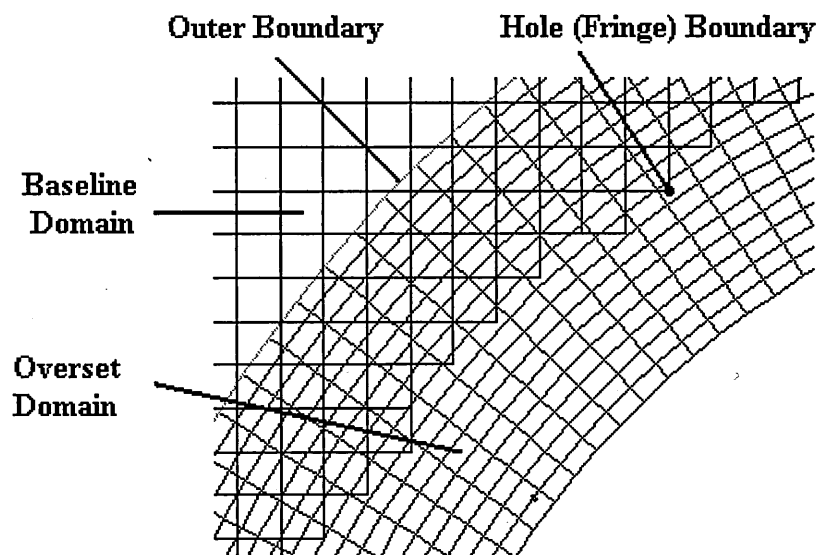


Figure 3-2, An example of a Chimera grid system

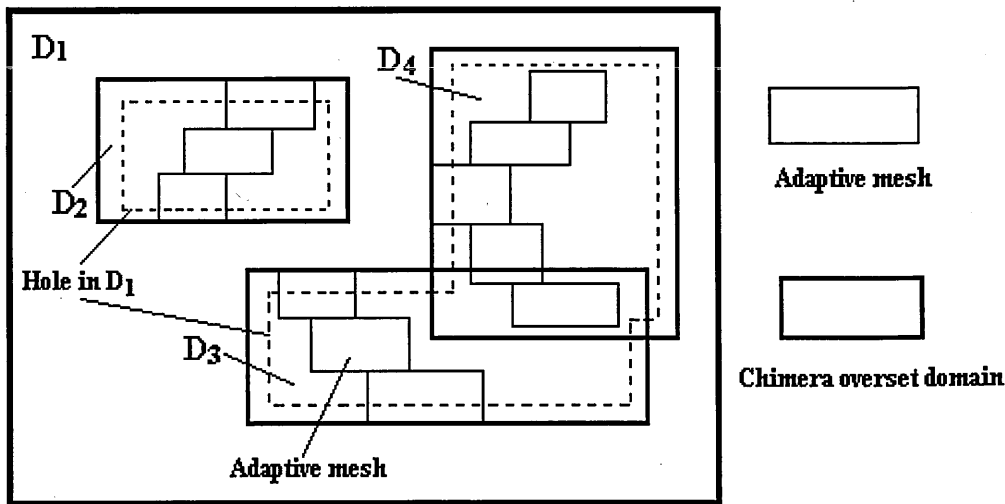


Figure 3-3. Chimera overset domain and Adaptive mesh

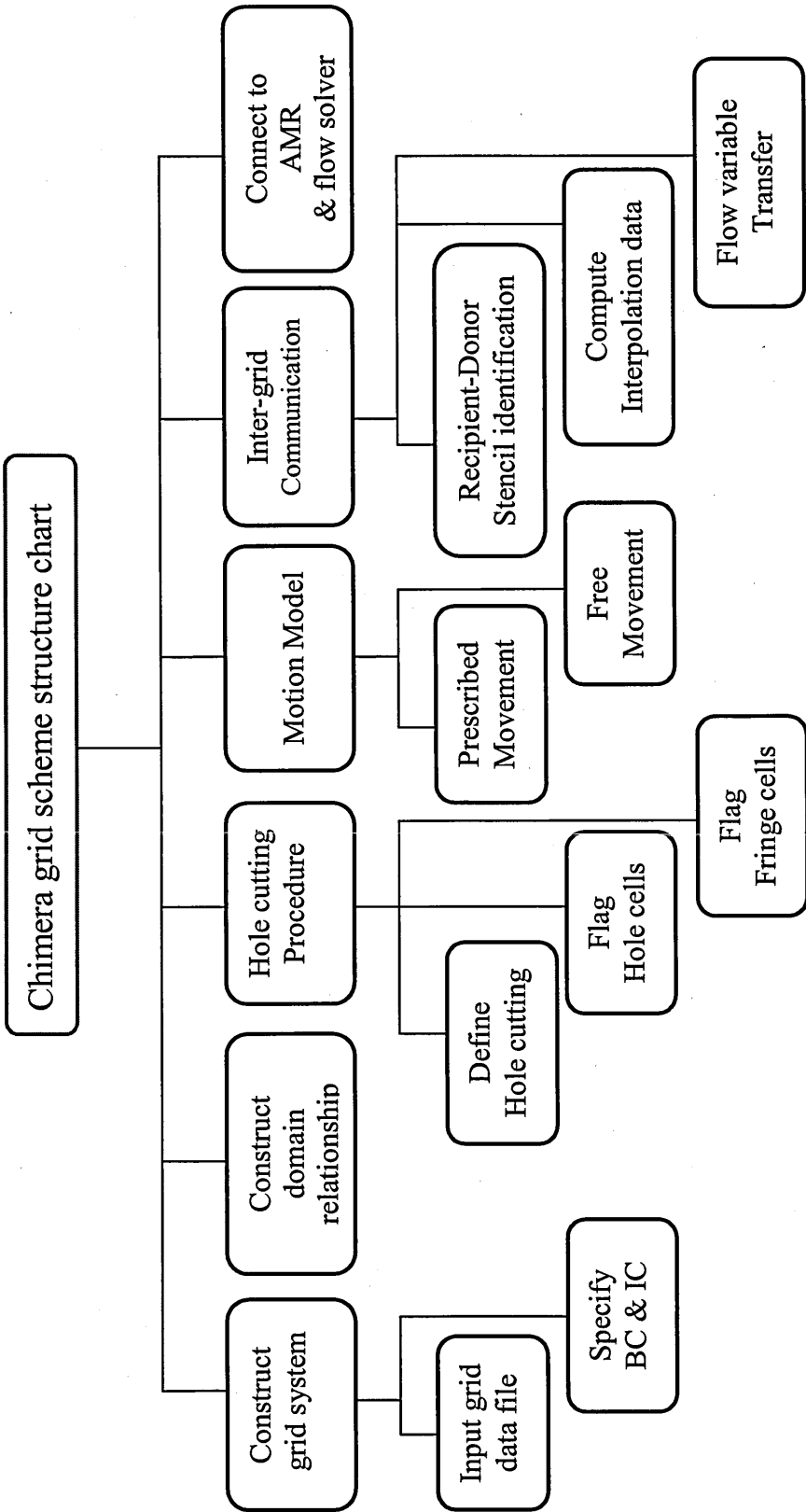


Figure 3-4. Chimera grid scheme structure chart

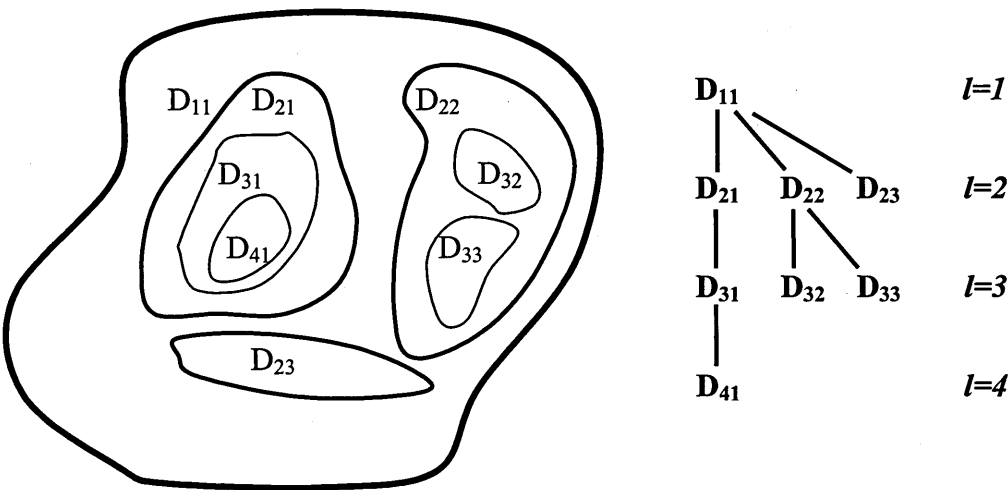
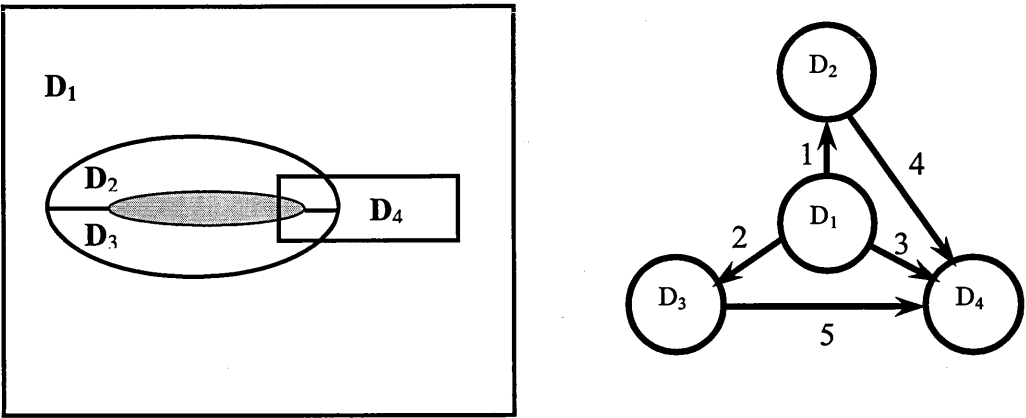


Figure 3-5. Restricted hierarchy by tree data structure



a. Overset domain

b. Digraph

Figure 3-6. Description of domain relationships by Graph data structure

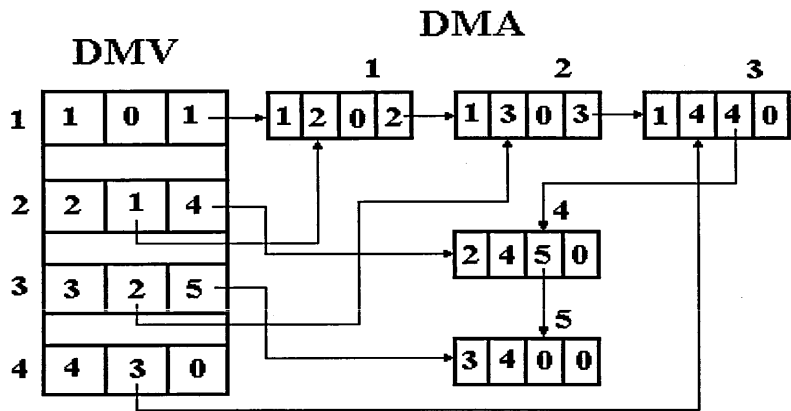
DMA(N,4)

taivex	headvex	hlink	tlink
--------	---------	-------	-------

DMV(N,3)

vertex	firstin	firstout
--------	---------	----------

a. Vertex and Arc Array



b. Storage

Figure 3-7. Orthogonal List Storage

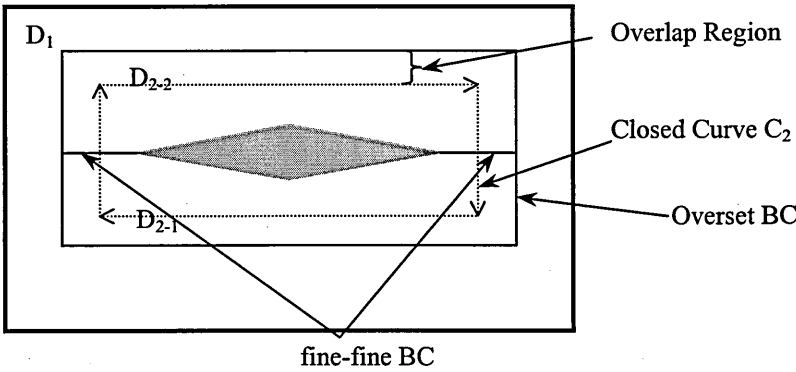


Figure 3-8. Define hole-cutting curve

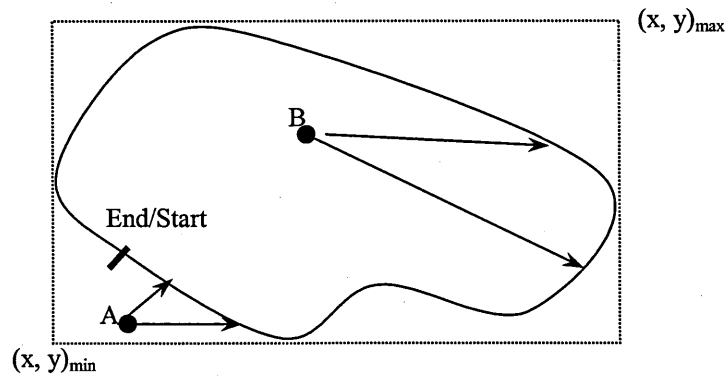


Figure 3-9. Sketch of the 'Non-Exterior Winding Rule'

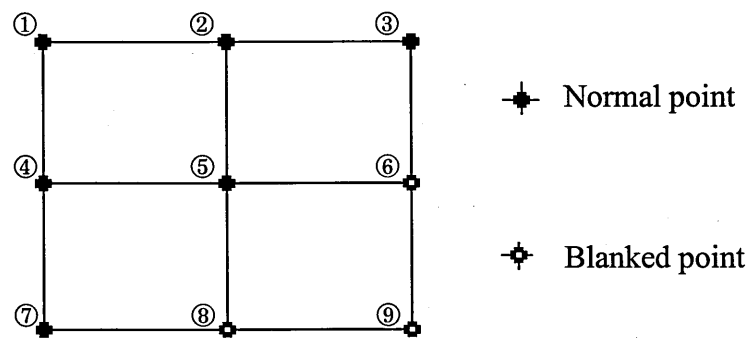


Figure 3-10. Nine-point stencil

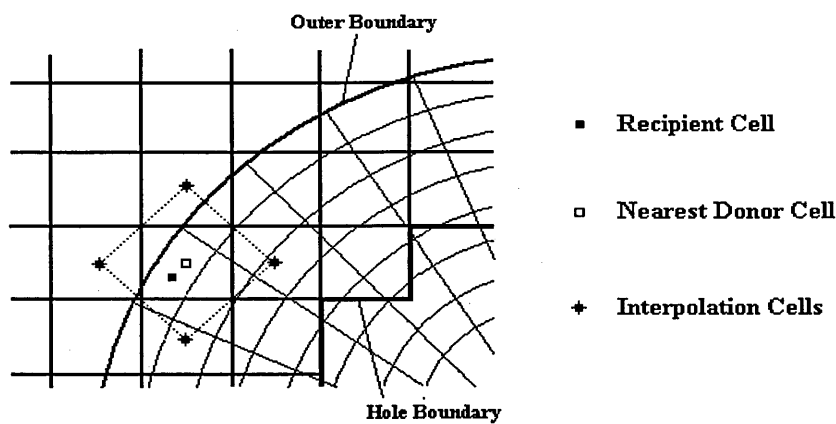


Figure 3-11. Interpolation quadrilateral formed by neighbour cell

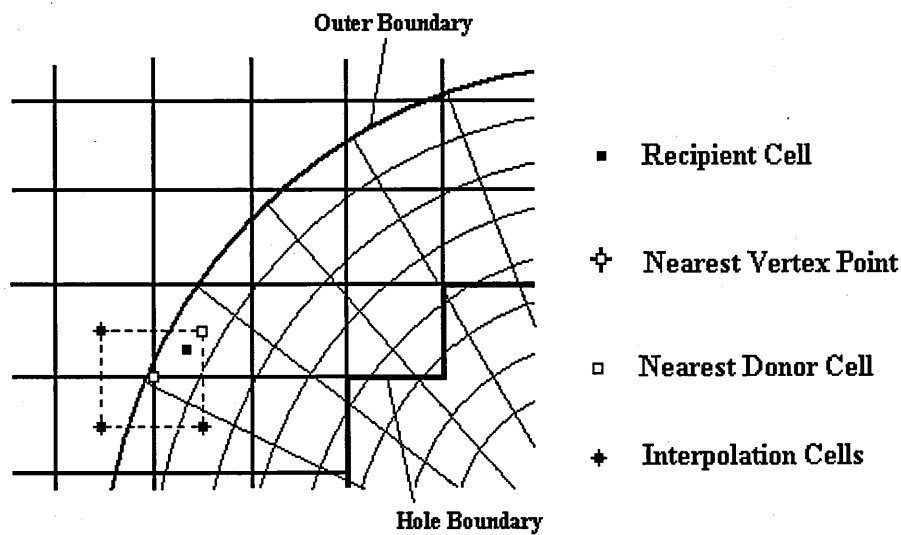


Figure 3-12. Interpolation quadrilateral formed by surrounding cell

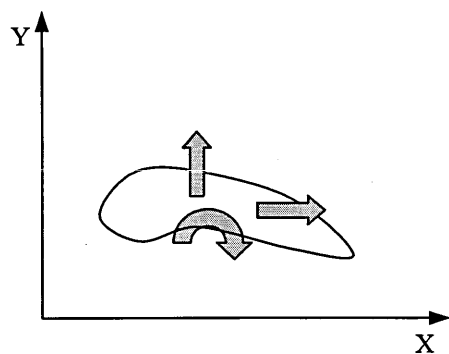


Figure 3-13. Co-ordinate system

Chapter 4

Integration of the Chimera Grid Scheme with the AMR algorithm

4.1 Introduction

Thus far, we have given separate descriptions for the basic concepts of the AMR algorithm and the main procedures of a fully automated 2D Chimera grid scheme. The objective of this study is to develop a solution adaptive Chimera grid scheme that is not only capable of handling flows around complex geometries, and moving body flows, but also enhances the resolution by using finer meshes in high gradient regions. This chapter deals with the integration of the Chimera grid scheme and the AMR algorithm, i.e. it will illustrate how to apply the AMR algorithm inside a Chimera grid system. A special section is dedicated to the further modification of the original AMR code so as to make it more compatible with a Chimera grid system. Critical code verification and validation will be also presented in this chapter.

4.2 Combination Method

One of the most key issues in the development of this solution adaptive Chimera grid scheme is that of how the AMR algorithm is to be applied inside a Chimera grid system. It requires that the advantages of two CFD methods be retained and complemented with each other and the code structure has to be compatible with the original AMR code in order to

minimise the modification work.

The basic method of combination of the Chimera grid scheme and the AMR algorithm is shown in Figure 4-1. The data structure and storage have been designed to be compatible with Quirk's AMR code (see §3.3) and Quirk's AMR code has been extended to be able to run multiple domain problems at the same time (see §2.4.1). Properly organising data and carrying out inter-grid communication at the same time level, the AMR algorithm can be applied in a Chimera grid system. In this way, we can use more than one overset sub-domain to mesh complex geometries or multiple bodies or flow features. Then for each sub-domain we can apply the AMR algorithm separately. Thus, the initial coarse Chimera grid system can be adapted according to the flow solution. This is a simple way to combine the Chimera grid scheme and the AMR algorithm.

This combination method does not compromise, but largely retains and demonstrates the advantages of both the Chimera grid method and the AMR algorithm. For example, since using the Chimera grid method, not only can complicated configurations be simply mapped, but also sub-domain grid quality can be easily guaranteed; the requirement of quasi-rectangular grids by the AMR algorithm can be easily satisfied.

4.3 Modification of the AMR Code

There are still three major problems in integrating Quirk's AMR code with the Chimera grid scheme. First, the flow solver has to be modified to account for the existence of the Chimera holes. The second problem is that the mesh refinement and adaptation must be available only in valid regions. In a Chimera hole, it would not make sense to employ the AMR algorithm. The third problem is that inter-grid communication has to be carried out at the same time level.

In fact, as mentioned in §2.4, six main changes and modifications must be made to Quirk's original AMR code and flow solver before the Chimera grid package can be incorporated. The first two changes have been described in §2.4.1 and §2.4.2 in detail. The

remaining four changes and modifications will be described in this section. Some improvement made to Quirk's original AMR code will also be presented in this section.

4.3.1 Replacement of the Pre- and Post-Processors

In order to simplify and speed up the coding of the AMRC code, the beginning of this project was to develop the pre-processor and post-processor.

The pre-processor is used to simplify the generation of an initial coarse Chimera grid system, set-up boundary conditions and initial conditions. During the current research work, we simplified the grid generation and problem description. A complicated configuration, or flow feature, or moving body can be mapped by several separate simple overset or intersected sub-domains. For example, in Figure 4-2, D_2 , D_3 and D_4 overset D_1 , D_3 and D_4 intersect with each other. In order to further simplify the grid generation, each sub-domain might be a single mesh, such as D_3 and D_4 , or composed of multiple properly nested interface meshes, (see §2.3.1 for the definition), such as D_{2-1} and D_{2-2} . The interface meshes can be also generated separately.

A single mesh is the minimum unit in the whole Chimera grid system. It can be generated by any kind of grid generation code. The only requirement is that the grid data file must be Plot3D data format. The details of Plot3D data format can be found in references¹³⁵⁻¹³⁶. Currently, the grid data file must be Plot3D, Formatted, Grid file, (i.e. *.PFG file, or Double precision, *.PFGD file). Of course, it is straightforward to extend to other grid data file formats. Currently, we use the CFD-GEOM software (a commercial grid generation code developed by the CFD Research Corporation) to generate the initial single meshes which will form the sub-domains and finally the whole Chimera grid system.

After generating every single mesh, the next step is to construct the Chimera grid system. As we have designed a special subroutine, the Chimera grid system can be automatically constructed when the grid data is input. What the user needs to do is to input every single mesh file (*.PFG or *.PFGD file) one by one from the first domain to last domain. The code will automatically construct and store the Chimera grid system, including

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

sorting out the total number of domains, $Nzones$; and for each domain, Nd , the mesh number at level 0, $NGA(Nd, 0)$; the X and Y direction cell number of each mesh, $IMAX(Nd, GRD)$ and $JMX(Nd, GRD)$ etc. Some of these contain the information required to run the AMR code, for example, $NGA(Nd, 0)$ is used to construct the mesh index system, $IMAX(Nd, GRD)$ and $JMX(Nd, GRD)$ are used to construct the cell index system. Most importantly, the domain relationship can also be easily sorted out when the grid data files are input.

In the AMRC code, all flow variables and geometric data are stored in a series of linked lists. Therefore, it is necessary to transfer the grid data format as required by Quirk's AMR code (see §2.3.1 and §3.3). It is also necessary to add dummy cells for each initial coarse sub-domain (each initial coarse mesh at level 0 exactly)

The post-processor is mainly used to decode the flow variables, transfer data formats, view the grid system and flow field, and analyse the simulation results. Quirk's original AMR code had a simple post-processor, and could be used to view single domain adaptive simulation results but it is not powerful enough to view a Chimera grid system and to carry out analysis of the simulation results. It would have taken too much effort to modify these graphics procedures. At the very beginning of this project, a new post-processor was coded according to the data structure of the AMRC code, Chimera grid scheme and the output format of the results. The post-processor had to be able to view the simulation results in different sub-domains of the Chimera grid system and at the different overlapped grid levels of the AMR grid hierarchy at the same time.

In order to properly view the simulation results on the different levels (AMR) of different domains (Chimera), we chose to use commercially available CFD post-processor software package. In the AMRC, we write a data format transfer code. By running this code the simulation results are transferred into Plot3D format¹³⁵. Therefore we can simply use the CFD-VIEW or FIELDVIEW software or other commercial packages to view or analyse the simulation results.

This procedure is very simple, yet very important, as well as being very useful in the development of the AMRC code. It has made huge contributions to the whole AMRC

coding. The coding work is much faster, since the results can thus be checked clearly and promptly.

4.3.2 Modification of the Flow Solver

In a Chimera grid system, as is inevitable, Chimera holes exist in regions that overlap with other domains or any other non-flow regions that belong to the other domains of the grid system. Hole cells must be excluded from the flow field solution. The flow solver must be modified to account for the use of ‘Chimera holes’ in the grid system. A popular method is to store the hole information in an integer array, *IBLANK*, for each domain. The elements of this array are set to 0 if the cell falls in a hole or 1 if it does not, refer to equation (3-3).

Since for a fringe cell its flow variables are provided from another domain, the flow solver must be able to account for these cells as well. For this sake, (and for viewing the grid system and flow solution), a fringe cell should be treated exactly the same as a hole cell.

$$IB = \begin{cases} 1 & \text{if } IBLANK = 1 \\ 0 & \text{if } IBLANK \leq 0 \end{cases} \quad (4-1)$$

In a time marching flow solver, each element in the correction step is multiplied by the *IB* value before the solutions are updated:

$$W^{n+1} = W^n + IB \cdot (W^{n+1} - W^n) \quad (4-2)$$

With this approach, no special routines are required to exclude and treat the blanked cells from the field solution. Of course, inter-grid communication and update routines are needed for the transfer of information across the various common boundaries.

4.3.3 Mesh Refinement Treatment

For the same reason as above, with the existence of a Chimera hole in the computational domain, it is required that the refinement and adaptation should be carried out only in the valid region. Inside a Chimera hole, it wouldn't make sense to refine meshes. However, this requirement is not easy to satisfy due to the recursive refinement and adaptation.

In Quirk's AMR code, no matter which monitor function is used, it is obvious that the Chimera hole cells and fringe cells are likely to be flagged for refinement. Therefore, it is necessary to judge if a Chimera hole cell or fringe cell is flagged for refinement. In the AMR code, the refinement flagging process starts from the highest mesh level and is implemented recursively. However, the hole-cutting process is only carried out at level 0. Therefore, only those hole cells and fringe cells at level 0, require their relevant entries in $IBLANK(IJ)$ to be blanked out, either 0 or $-Nd$. It cannot be judged directly whether a cell (i_l, j_l) at level l falls inside a Chimera hole.

Fortunately, in Quirk's AMR code, a simple relationship exists between the co-ordinate systems of a coarse mesh cell and its refined high level mesh cells, refer to equation (2-13). Based on this relationship, we can calculate the cell co-ordinates (i_0, j_0) at level 0 on which the cell (i_l, j_l) overlays. If the cell (i_0, j_0) corresponding to the cell (i_l, j_l) is blanked out as a hole cell, $IBLANK(IJ_0)=0$, or as a fringe cell, $IBLANK(IJ_0)=-Nd$, then do not flag it for refinement(i.e. $Refine(IJ_l)=0$).

In the Grouping/Clustering process, a set of logical rectangular patches is to be found to completely cover the flagged cells. A simple packing criterion, P_{tol} , is used to judge if a mesh is acceptable. That is, if the number of flagged cells divided by the number of cells contained by the prospective mesh is greater than a value, e.g. 0.6, then the mesh is deemed acceptable. Otherwise, sub-divide the mesh into two regions and judge if they are acceptable. This procedure continues until all meshes are acceptable.

Obviously, according to this packing criterion, the hole cells or fringe cells are still likely to be packed for refinement, although they are not flagged for refinement. It is

necessary to modify the Grouping/Clustering procedure to account for the existence of hole cells and fringe cells inside a Chimera grid system. A simple method is to add an additional criterion, i.e. to check the prospective mesh boundary cells. If there are not any mesh boundary cell overlays on a hole cell or a fringe cell at level 0, then the mesh is acceptable. Otherwise, it is not.

Figure 4-3 is an example which shows a grid system with a Chimera hole and three level adaptive meshes. Comparing Figure 4-3(a) and (b), we can clearly see that mesh refinement and adaptation are only performed outside the Chimera hole according to the above treatment.

4.3.4 Time Step Treatment

In the AMRC code, different domains are solved separately. As a result, the time steps for different domains are different, as is inevitable for a time accurate method. However, the inter-grid communications must be carried out at the same time level. Especially for an unsteady problem with body motion, the timeliness of boundary update is very important. Extra care must be taken to update the common boundaries in a time-accurate fashion.

A reasonable, yet a simple method has been employed to handle this problem.

- (1) Finding a maximum time step $\Delta\tau_{max}=MAX(\Delta\tau_{max}, \Delta\tau(Nd))$ from all domains defines the next inter-grid communication time level as $T2_{max}=T1+\Delta\tau_{max}$.
- (2) Solving each domain separately. For those domains with smaller time steps, iterate until all domains reach the same time level $T2_{max}$, i.e. $T2(Nd)=T2_{max}$. It should be noted that $n \times \Delta\tau(Nd)$ might be greater than $\Delta\tau_{max}$. If this is the case, the last iteration time step $\Delta\tau(Nd)$ needs to be modified as $\Delta\tau(Nd)=T2_{max}-T1(Nd)$.

This treatment guarantee that the inter-grid communication is carried out at the same time level after every time step $\Delta\tau_{max}$.

4.3.5 Correction of the AMR Code

Besides the above modifications and that described in §2.4, some improvements and corrections on Quirk's AMR code have been completed as well. Here we present the improvement of the calculation of the viscosity coefficient and the fixes for just one major bug within the AMR code.

1. Viscosity Calculation

Quirk's original AMR code has considered the simulation of a viscous flow problem. For the N-S governing equations, splitting the flux vectors into their viscous and inviscid components, it may be written in the form,

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x} [F^I + F^V] + \frac{\partial}{\partial y} [G^I + G^V] = 0 \quad (4-3)$$

where the dimensionless vectors W , F^I , G^I , F^V and G^V are given by,

$$W = \begin{pmatrix} \bar{\rho} \\ \bar{\rho} \bar{u} \\ \bar{\rho} \bar{v} \\ E_t \end{pmatrix} \quad (4-4)$$

$$F^I = \begin{pmatrix} \bar{\rho} \bar{u} \\ \bar{\rho} \bar{u}^2 \\ \bar{\rho} \bar{u} \bar{v} \\ (\bar{E}_t + \bar{P}) \bar{u} \end{pmatrix} \quad G^I = \begin{pmatrix} \bar{\rho} \bar{v} \\ \bar{\rho} \bar{v} \bar{u} \\ \bar{\rho} \bar{v}^2 \\ (\bar{E}_t + \bar{P}) \bar{v} \end{pmatrix}$$

$$F^V = \begin{pmatrix} 0 \\ -\bar{\tau}_{xx} \\ -\bar{\tau}_{xy} \\ -\bar{u} \bar{\tau}_{xx} - \bar{v} \bar{\tau}_{xy} + \bar{q}_x \end{pmatrix} \quad G^V = \begin{pmatrix} 0 \\ -\bar{\tau}_{yx} \\ -\bar{\tau}_{yy} \\ -\bar{u} \bar{\tau}_{yx} - \bar{v} \bar{\tau}_{yy} + \bar{q}_y \end{pmatrix} \quad (4-5)$$

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

The components of the shear-stress tensor and the heat-flux vector are now given by,

$$\begin{aligned}
 \bar{\tau}_{xx} &= \frac{2\bar{\mu}}{3\text{Re}} \left[2 \frac{\partial \bar{u}}{\partial x} - \frac{\partial \bar{v}}{\partial y} \right] & \bar{\tau}_{yy} &= \frac{2\bar{\mu}}{3\text{Re}} \left[2 \frac{\partial \bar{v}}{\partial y} - \frac{\partial \bar{u}}{\partial x} \right] \\
 \bar{\tau}_{xy} &= \frac{\bar{\mu}}{\text{Re}} \left[\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} \right] & \bar{\tau}_{yx} &= \frac{\bar{\mu}}{\text{Re}} \left[\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} \right] \\
 \bar{q}_x &= \frac{-\bar{\mu}}{(\gamma-1)M_\infty^2 \text{RePr}} \frac{\partial \bar{T}}{\partial x} & \bar{q}_y &= \frac{-\bar{\mu}}{(\gamma-1)M_\infty^2 \text{RePr}} \frac{\partial \bar{T}}{\partial y}
 \end{aligned} \tag{4-6}$$

After correcting the calculation of viscous coefficient, the AMR code is possible to simulate laminar viscous flow problems. In our calculations, the physical property of dynamic viscosity varies with temperature according to the following formulation of Sutherland's Law.

$$\bar{\mu} = \bar{T}^{-\frac{3}{2}} \left(\frac{1 + 110 / T_\infty}{\bar{T} + 110 / T_\infty} \right) \tag{4-7}$$

That is, in the dimensionless calculations, use the free stream temperature, T_∞ , as the reference temperature, T_{ref} , so that, at $T=T_\infty$, $\mu=\mu_\infty$, i.e. at $\bar{T} = 1$, $\bar{\mu} = 1$.

2. Fine-Fine Boundary Flagging

In Quirk's AMR code, the fine-fine boundary can be flagged automatically during the implementation of refinement and adaptation. For example, in Figure 4-4, $G_{0,1}$ and $G_{0,2}$ are two properly nested meshes at level 0. In Figure 4-4a, B-C and B-D are solid boundaries. The A-B mesh boundary and all refined high-level mesh boundaries, which exactly overlay those cells along both sides of A-B, will be flagged as the fine-fine boundary or fine-coarse boundary. All cells along B-C and B-D will be always correctly flagged as an external solid boundary.

However, we have noted that for some cases this flagging procedure may result in a

serious error. For example, in Figure 4-4b, both B-C-D and B-C'-D are solid boundaries, but their adaptive high-level mesh boundaries will be flagged as a fine-fine boundary according to the original flagging criterion. This is obviously not correct and will definitely result in a wrong solution, if there is such a boundary inside a grid system.

To solve this problem, the fine-fine flagging procedure has been modified. A similar method as mentioned in §4.3.3 was used. For a mesh boundary cell, if it is flagged as a fine-fine boundary type, it is necessary to determine its relevant cell co-ordinate (i_0, j_0) at level 0 according to the relationship (4-4), then check its original boundary type. If it is not specified as the fine-fine or fine-coarse boundary condition, then restrain its boundary condition to that which was specified in the pre-processor. In this way, the boundary conditions of B-C-D and B-C'-D in Figure 4-4b will be always kept as solid surface boundary conditions.

4.4 The Complete AMRC Scheme

In this section, we present the complete AMRC code frame structure and a brief introduction to each part of the code, and illustrate how this scheme is used to simulate a practical problem.

4.4.1 Code Frame Structure

Figure 4-5 shows the complete AMRC code structure. There are four main modules, each of which corresponds to one of the major components of the AMRC scheme: ①.pre-processor; ②.fully automated Chimera grid scheme; ③.AMR solver; and ④.post-processor. Modules ①, ② and ④ are newly developed, and module ③ is a modification of Quirk's AMR code.

The pre-processor is used to simplify the generation of an initial coarse Chimera grid system, and to set-up boundary conditions and initial conditions for the simulation. This module de-couples the problem to be simulated from the code. In this way the AMRC code

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

is less problem-dependent but can solve many problems. The user can set up a simulation very quickly and easily. The intention is that it will not require expert knowledge of this code to run it.

The Chimera grid method module is mainly used to mesh complicated configurations and handle multiple moving body flows. The fully automated feature employed here was one of the main considerations during the development of the procedures associated with the Chimera grid method.

The third module, the AMR flow solver, is used to perform high-resolution simulations in regions that require it. That is, to provide local refinement and adaptation according to the flow solution, and to resolve complex flow features in each domain of the Chimera grid system.

The final module, the post-processor, is used to decode the flow variables, transfer data format and view the grid system and simulation results.

4.4.2 Code Implementation Sequence

The AMRC code is not problem-dependent but general-purpose code. And it allows most of the procedures associated with the Chimera grid method to be fully automated. The user can set-up a simulation problem very quickly and easily. It does not require an expert knowledge of this code to run it. To simulate a practical problem, the following steps are involved in the current implementation:

- (1) Generate the sub-domain grid, construct the Chimera grid system, and specify the boundary and initial conditions.
- (2) Construct domain relationships.
- (3) Carry out hole cutting in each domain, if necessary.
- (4) Compute a time step for each domain.
- (5) Apply the AMR algorithm in each domain.
- (6) Repeat steps (4) and (5), until all domains reach the same time level.

- (7) Locate recipient and donor cells and generate interpolation stencils.
- (8) Carry out inter-grid communication.
- (9) If any domain is moving, calculate forces, moments and domain movement for the new time level.
- (10) Go to step (3), if not finished.

With this implementation sequence, the multiple domain Chimera grid system can be solved and adapted at the same time, and the inter-grid communications are guaranteed to be carried out at the same time level.

4.5 Validation Case Study

So far, we have integrated a fully automated Chimera grid scheme coded by the author with Quirk's AMR code. Although the AMR code (including the AMR algorithm and flow solver) was validated by Quirk and its main parts and implementation sequence have not been changed during the integration with the Chimera grid scheme, the validation of the AMRC code is a necessary step towards trusting its results. The purpose of this validation is to mainly check if the AMR algorithm can be properly applied in a Chimera grid system and to see if strong discontinuities can cross through a Chimera overset boundary. This is achieved by simulating a shock wave reflection and diffraction over a cylinder and comparing the numerical solutions with experimental results. In this section, we will present a validation case study. More code validation and application cases will be reported in the following three chapters.

4.5.1 Shock Reflection and Diffraction over a Cylinder

The reflection of oblique shock waves is a non-linear problem which has been investigated analytically and experimentally by many researchers¹³⁸⁻¹⁴⁰. The experimental investigation of the diffraction of a strong shock by a cylinder presented in reference¹³⁸ is a

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

suitable case to test our simulation. Actually, Quirk used the AMR algorithm to simulate this experiment as a validation case in his PhD thesis. We chose this case so that the numerical solutions can also be compared with those by the sole AMR algorithm using a single polar grid.

The experimental investigations were performed in the Harvard University 4in.×12in.×40ft. shock tube in air. An incident moving shock wave from upstream impinges on a cylinder of diameter $D = \frac{1}{2}$ in. within a quiescent down stream. A schlieren optical system using a spark light source of approximately $0.2\mu sec$ duration was used to photograph the diffraction patterns. Two series of experiments were performed at the same mean shock Mach number of 2.82 but at two different pressures P_2 behind the undisturbed shock and, therefore, at two different Reynolds numbers, $R_e = \frac{\rho u D}{\mu}$ (u being the velocity of the undisturbed shock, and μ the viscosity behind the undisturbed shock afterflow). We chose the series 1 experiment to test our simulations mainly because the schlieren photographs are available. The wind tunnel test conditions are as below.

$Ms=2.82, \quad P_2=6.85\times10^4 \text{ N/m}^2, \quad Re=7.79\times10^4$

From these known values, we can determine the other test conditions as below.

Table 4-1. Wind tunnel test conditions (shock jump conditions)

		Upstream conditions (undisturbed)	Down stream conditions (quiescent)	Unit
Pressure	P	6.85×10^4	7.52×10^3	Pa
Density	ρ	0.349	0.134	kg/m^3
Temperature	T	683.7	195.7	K
Velocity	u	576.3	0	m/s
Viscosity	μ	3.281×10^{-5}	1.299×10^{-5}	$\text{kg/m}\cdot\text{s}$

4.5.2 Computational Details

For this problem, the shock initially meets the cylinder head on, this is analogous to the reflection from a 90° wedge, and a Regular Reflection (RR) ensues. But as the incident shock propagates up the cylinder so the reflecting angle of the surface decreases until there is a transition from RR to MR (Mach Reflection). The type of MR depends on the strength of the incident shock. Moreover, further transitions can occur and it is possible to have the sequence of events RR→DMR (Double Mach Reflection) →CMR (Complex Mach Reflection) →SMR (Single Mach Reflection). The same sequence as Quirk used; our simulation was also done in two parts. The first part captured the transition events. Unfortunately, the experimental investigations for this part were not done. The second part continued the simulation so as to be able to compare the numerical shock pattern with the schlieren photographs taken from the experiment.

Since it is a symmetric problem, only the top half of the physical domain was considered. Figure 4-6 shows the Chimera grid system and the boundary conditions. We used two domains to mesh the computational domain and the cylinder. Compared with Quirk's simulation which used one polar grid system, we can certainly state that our grid system is more reasonable for such a simulation problem. The far field boundary condition can be properly specified, while in Quirk's simulation, parts of the far field boundary were either under or over specified.

In the Chimera grid system, the baseline domain is a uniform Cartesian grid of 160×80 cells at level 0. The cell size is 0.05×0.05 of the radius of the cylinder. The overset domain is a body fitted grid (polar grid) of 180×45 cells at level 0. In the angular direction, 181 rays with a uniform angular spacing of one degree were used. In the radial direction, 45 uniform grid cells were distributed, and the cell size near to the cylinder surface is 0.044×0.017 of the radius.

The refinement monitor functions chosen for this simulation are density gradient and contact surface. The grid adaptation parameters for this simulation are given in table 4-2.

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

The refinement factors rI , rJ are all specified as 2, and the maximum refinement level $Lmax(Nd)$ is specified as 3. Thus the grids at level 3 have the same resolution as uniformly spaced meshes of 1280×640 (for domain 1) and 1440×360 (for domain 2). The finest cell size will be $1/8$ of that of the coarsest cell size.

Table 4-2. Grid adaptation parameters

l	rI	rJ	F_{tol}	P_{tol}
0	1	1	0.05	0.6
1	2	2	0.1	0.6
2	2	2	0.1	0.6
3	2	2	N/A	N/A

This example is a time-dependent problem. The whole flow field is set as an inviscid flow. The normalised initial conditions for the quiescent flow are set as $\bar{\rho}=1$ and $\bar{p}=1$. The simulation was started with an exact shock jump condition which could be defined using the moving shock relationships given in appendix B. A Mach 2.82 normal shock wave was initially specified just before the cylinder, see Figure 4-7a. The upstream was set as post-shock condition and downstream was set as pre-shock condition. For our calculation we chose to integrate these equations using the second order Roe's FDS scheme combined with the SUPERBEE limiter function,

$$SuperBee(r_k) = \max(0, \min(1, 2r_k), \min(2, r_k)) \quad (4-8)$$

A suitable choice for the parameter r_k is the ratio of the upwind to local wave strengths,

$$r_k = \frac{(\alpha_k)_{upwind}}{(\alpha_k)_{local}} \quad (4-9)$$

this allows each wave of the approximate Riemann solution to be limited separately. Note in smooth regions of the flow $r_k \rightarrow 1$ and $B(r_k) \rightarrow 1$, while near extrema, $r_k < 0$ and so $B(r_k) = 0$.

Harten's entropy fix is employed,

$$\begin{cases} v' = \frac{v}{2} \left[\frac{v^2}{\delta} + \delta \right] & \text{for } |v| < 2\delta \\ v' = v & \text{for } |v| \geq 2\delta \end{cases} \quad (4-10)$$

This modification to the wave speeds is only applied to the acoustic waves,

4.5.3 Simulations of Transition Events

Figure 4-7 presents a sequence of numerical simulation results (density plots). Figure 4-7a shows the initial condition. The incident shock wave moves towards and impinges the cylinder, a Regular Reflection ensues. As the incident shock propagates up the cylinder so the reflecting angle of the surface decreases until there is a transition from RR to MR, see Figure 4-7b. As the shock wave continues to evolve so the Mach stem grows in length and the salient features of DMR appear, see Figure 4-7c. After the incident shock passes the peak of the cylinder the Mach stem starts to diffract around the cylinder, see Figure 4-7d. Shortly afterwards there is a transition from DMR to CMR. The salient features of CMR are shown clearly in Figure 4-7e. The simulation was terminated when the Mach stem had reached a point about 5° from the rear stagnation point, see Figure 4-7f. At later times there is considerable separation of the flow from the surface of the cylinder, and since our calculation is inviscid this separation would not be modelled properly.

Figure 4-8 is an enlarged density plot which shows the transition point from RR to MR. This transition event occurs at approximately 48° from the forward stagnation point according to our calculation.

This point is about 3.2° later than the simulation result (44.8°) obtained by Quirk. The difference is mainly due to the fact that the cell size in our simulation (2.18×10^{-3} of the radius of the cylinder) is about 3 times larger than that in Quirk's simulation (6.98×10^{-4} of the radius of the cylinder). According to Quirk¹⁸, the coarser the mesh the larger the transition angle. So it is no surprise that our simulated transition angle is slightly higher

than that of Quirk's simulation.

In Quirk's simulation, only two adaptive levels were used, but the refinement factors, rI , rJ were all specified as 5. Therefore, the finest cell size was 1/25 of that of the coarsest cell size. The other reason is that in our simulation, we chose the shock Mach number as 2.82 (in Quirk's simulation it was set to 2.81), the same as the mean shock Mach number of the experiment and at which an experimental schlieren photograph was available, see Figure 4-10c.

It is difficult to determine the accuracy of our numerical result. In Bryson & Gross' report, they merely stated that the transition from RR to MR occurs somewhere between 40° and 50° of the forward stagnation point.

Figure 4-9 shows a sequence of the numerical simulation results of the transition from MR to DMR and the developing of the DMR. The salient features of DMR are well resolved.

4.5.4 Comparison with Experiment

Figure 4-10 shows the comparison between the numerical solution and experimental data. Figure 4-10a is an enlarged density plot. Figure 4-10b is an enlarged density contour plot. Figure 4-10c is an experimental schlieren photograph presented in Bryson & Gross' report¹³⁸. Comparing the numerical result with the experimental result, we can see that reasonable agreement has been achieved. The shock reflection, diffraction and triple point are all resolved. The simulation results show that a fairly detailed shock structure was obtained as well, especially in the rear region. The vortex and the contact discontinuities emanating from the triple point were clearly captured. The contact discontinuities close to the cylinder surface were also captured. Overall, the simulation result closely duplicated the experimental schlieren photograph.

Closely examining the simulated density plot, it can be seen that in front of the stagnation point the so-called 'carbuncle' phenomenon appears. This is a known problem

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

associated with Roe's solver near the symmetry line for 2D blunt noses in Euler solutions¹⁴⁰⁻¹⁴¹.

Figure 4-11a shows the simulated triple point locus and shock reflection, diffraction pattern (resulting from density contour by wiping out undesirable contour lines but leaving the reflection shock, incident shock, Mach stem and part of contact discontinuity). From this Figure we can clearly see that the reflection shock wave moves away from the cylinder surface. The DMR (see point b) occurs after the RR to MR transition point (point a) and the transition to CMR occurs before the Mach stem reaches the cylinder peak. The Mach stem is exactly normal to the cylinder surface.

Figure 4-11b shows part of the experimental triple point locus and vortex locus present in Bryson & Gross' report. For the upper triple point locus, the experimental results are slightly lower than the position of the triple point as given by Whitham's approximate theory¹³⁸. Comparing the triple point "d" in both the simulation results and the experimental results (approximately the same position), the distance from the triple point to cylinder surface is $1.2R$ according to the simulation results, while $1.18R$ according to the experimental results. Comparing the trends of the triple point locus, we can claim that the simulation closely duplicated the experiment.

4.5.5 Computational Efficiency

One of the most important properties of the AMR algorithm is its high computational efficiency. The AMR algorithm refines both in time and space. It is this combination of spatial and temporal refinement strategies that accounts for the efficiency of the AMR algorithm. It requires only modest computing power to resolve complex shock hydrodynamic flow problems. Quirk, using a shock reflection over a wedge validation case, demonstrated that for a given resolution the non-adaptive calculation takes almost 10 to 30 times longer (varies as the solution progresses) to run than the adaptive calculation does. This figure is remarkably low.

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

During the development of the Chimera grid scheme, one of the main considerations is to retain computational efficiency. For such a solution adaptive Chimera grid scheme, the inter-grid communication computational cost is a key issue. Much effort has been made on improving the point searching and interpolation efficiency. For our validation case, Table 4-3 gives the run time profile information for one typical iteration (CMR has occurred) by using the *etime()* function. The hole-cutting procedure only need be called once in the first iteration for this case. The amount of run time on this procedure is then added to the total run time for a later iteration.

Table 4-3 Run time information (one iteration)

Procedure	Run time (sec)	% of total run time
Total	31.95	100
Hole-cutting	0.395	1.24
AMR & solver	16.97	53.11
Donor searching	14.58	45.63
Interpolation	0.005	0.02

From this run time profile, we can see that the hole-cutting procedure and flow variable interpolation procedure (include calculating interpolation coefficients) are less computational cost. The donor searching procedure accounted for 45.63% of the total run time, while the AMR and flow solver accounted for 53.11% of the total run time. This means that after combining the Chimera grid scheme with the AMR code, the run time is only about twice that by merely using AMR. Therefore, the total computational efficiency of our scheme is still much higher than a non-adaptive calculation. In addition, for a non-adaptive Chimera grid scheme, as the cell number increases (for high resolution), the amount of time spent on donor searching will also increase significantly.

Actually, for a viscous flow problem, using our solution adaptive Chimera scheme, the computational efficiency can be largely enhanced. Inside those near body overset domains, the flow physics is set as viscous flow. Far away from the solid boundary, the

viscous effects are very small and can be neglected. So inside other domains, the flow physics can be set as inviscid flow. By this arrangement, the viscous flow calculation is restricted in those domains where it is necessary. This is another important feature of the Chimera grid method.

4.6 Investigation of the Chimera Boundary

Examination of the density contours of the numerical solutions (Figure. 4-10b) reveals a mismatch in the overlap region near the shock/domain-boundary intersection (near the triple point). This is no surprise for a Chimera grid method. Several factors could contribute—the error caused by flow variable interpolation; the reflecting boundary conditions imposed at the overset boundary; the overlap depth; the cell size difference between the different domains. Among these factors, the error caused by interpolation is a key issue and cannot easily be eliminated.

This section is specially designed to evaluate the performances of the proposed AMRC scheme at the Chimera overset boundary and hole boundary.

4.6.1 Interpolation Error Analysis

There is much disagreement about the proper interpolation method to employ in a Chimera grid scheme, especially for cases in which shock waves or other regions of high gradients cross domain boundaries. In our scheme, we employed two simple but popular interpolation methods, Bilinear interpolation and Taylor series approximation. It is believed that for our high-resolution scheme, such interpolation methods can properly transfer the flow solution on Chimera common boundaries. This subsection gives the interpolation error analysis that demonstrates the reasonable accuracy of our scheme.

An interpolation method assumes that the variable to be interpolated is continuous. For simplicity, here we assume that a function, $y = f(x)$, is analytical and its high order derivatives exist. According to the Taylor expansion,

$$f(x + \Delta x) = f(x) + \frac{\partial f(x)}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} \Delta x^2 + \dots + \frac{1}{n!} \frac{\partial^n f(x)}{\partial x^n} \Delta x^n + \dots \quad (4-11)$$

we have

$$\left(\frac{\partial f(x)}{\partial x} \right) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x) \quad (4-12)$$

Let x_0 and x_1 be distinct and $x_0 \leq x \leq x_1$. Using the notation $\Delta x_0 = x - x_0$ and $\Delta x_1 = x_1 - x$ we can get,

$$\frac{f(x) - f(x_0)}{\Delta x_0} + O(\Delta x_0) = \frac{f(x_1) - f(x)}{\Delta x_1} + O(\Delta x_1) \quad (4-13)$$

Rearranging equation (4-13) gives,

$$f(x) = f(x_1) \frac{x - x_0}{x_1 - x_0} + f(x_0) \frac{x_1 - x}{x_1 - x_0} + O[(x_1 - x)(x - x_0)] \quad (4-14)$$

So we have the linear interpolation equation,

$$L_1(x) = f(x_1) \frac{x - x_0}{x_1 - x_0} + f(x_0) \frac{x_1 - x}{x_1 - x_0} \quad (4-15)$$

and the truncation error,

$$R_1(x) = \frac{1}{2} f''(\xi)(x - x_0)(x - x_1), \quad \xi = \xi(x) \in (x_0, x_1) \quad (4-16)$$

On $[x_0, x_1]$, the maximum value of $|(x - x_0)(x - x_1)|$ will be $\frac{1}{4}(x_1 - x_0)^2$ at the point $x = \frac{1}{2}(x_0 + x_1)$. Therefore, for any $x \in [x_0, x_1]$, we have,

$$|R_1(x)| \leq \frac{(x_1 - x_0)^2}{8} \max_{x_0 \leq x \leq x_1} |f''(x)| \quad (4-17)$$

This means that the truncation error of linear interpolation depends on two factors, the interval Δx and the maximum second derivative $f''(x)$ inside this interval. In a smooth region, since $f''(x)$ is small, using linear interpolation one can get a sufficient accurate solution. In regions where $f''(x)$ is large, the error caused by linear interpolation would be large. However, for a given $f(x)$, as the interpolation interval Δx decreases, the truncation error decreases.

If there are discontinuities such as strong shock waves in a flow field, strictly speaking, its derivatives f' and f'' do not exist at the discontinuous point. But for a given shock-capturing scheme, the numerical solution normally is smeared by several cells. So it can be viewed as a solution with large $f''(x)$. Using linear interpolation in such a region, it is true that the conservation can not be guaranteed. However, equation (4-17) implies that decreasing the interval Δx can positively increase the accuracy.

Using the AMRC scheme, normally inside high gradient regions the cell size can be easily refined by one order, see Figure 4-3. So reasonable accuracy can be easily achieved by interpolation (bilinear or first order Taylor series approximation) or even by directly shovelling the value of the nearest donor cell to the recipient cell. There is no visible difference in the flow solution from these three methods. In addition, since simple interpolation methods are employed, the computational cost on interpolation is less, (see Table 4-3).

Figures 4-12 to 4-14 show the comparison between the numerical solutions on the initial coarse grid system (adapting off) and those presented in §4.5 by applying the AMR algorithm (adapting on). These are enlarged grids and density contour plots near the triple point, where the reflection shock and Mach stem cross the Chimera hole boundary and the triple point is crossing the overset boundary. Comparing both solutions, we can clearly see that by applying the AMR algorithm, not only is the solution resolution enhanced, but also

the error caused by interpolation is decreased significantly, especially at the overset boundary (point A). From the adaptive solution, we can see that the shock wave crosses through the common boundary properly. There is no reflection at the hole boundary and overset boundary. In Figures 4-13b, near the points C and D, we can still see slightly mismatched density contours. This is mainly due to the nature of the AMR code, i.e. co-existence of the coarse solution and finer solution. In the post-processor, we have blanked out the coarser cells if they overlap with finer meshes. However, some coarser cells are not properly blanked out yet if they overlap with a finer mesh which oversets more than one coarser mesh (see Figure 2-7a, mesh $G_{2,2}$ for example).

Figure 4-15 is another example presented in one of the author's report¹³⁷. This test case is to simulate shock-shock interaction due to two double wedge bodies. This Figure shows the enlarged density plot near the lower body upper surface. Figures 4-15a is a non-adaptive simulation result from the AMRC code. Figure 4-15b shows the initial coarse meshes. We can clearly see the errors caused by interpolation on the Chimera overset boundary. However, the adaptive simulation result, shown in Figure 4-15c, is much better on the overset boundary, especially in the high gradient region indicated by A. This is obviously because the coarse mesh cells in these regions are refined, see Figure 4-15d. In region B, the baseline domain mesh is not refined, see Figure 4-15d. That is why in region B, the resolution is not improved very much.

4.6.2 Shock Wave/Domain Boundary Interaction

In the literature, some researchers have mentioned that if a shock wave coincides with an overlapped interface boundary, the computation may diverge due to large local errors (caused by interpolation) around the shock wave. This is especially true for a non-adaptation Chimera scheme, since high resolution cells cannot be guaranteed to be placed in regions with high gradient. For this validation case, we also have experience that non-adaptation calculations might crash on a coarser initial grid, if a strong shock wave crosses a hole boundary or an overset boundary.

In order to further critically test the performances of the AMRC scheme at the Chimera overset boundary and hole boundary, a smaller overset domain was used (20×180 uniform cells), see Figure 4-16. In this case, the normal shock wave was initially specified just before the overset domain (all the other settings are the same as for the calculation presented in §4.5). During propagation the incident shock wave first crosses the overset domain boundary, then the hole boundary and moves towards to the cylinder. After impinging the cylinder, the shock reflection occurs and the reflecting shock wave moves away from the cylinder surface, crosses the hole boundary and then the overset boundary. In this case, the curved strong reflecting shock wave almost crosses through the hole boundary at the same time (not just a point, but along the hole boundary), see Figure 4-17. This is a very challenging test case for a Chimera scheme. Since the AMRC scheme can automatically refine cell sizes in regions where it is necessary, the reflecting shock wave can properly cross through the hole boundary (see Figure 4-17) and the overset boundary (see Figure 4-18).

4.6.3 Influence of the Overlap Depth

In a Chimera grid scheme, since the separate domains are to be treated as independent entities, overlap regions between different domains must match the flow solutions in different domains. The overlap depth definitely has an influence on the whole flow solution. Generally, the thicker the overlap depth, the more accuracy can be achieved. However, the price is to increase domain size or cell numbers, hence to increase the memory requirement and computational cost.

In fact, the influence of the overlap region depends on whether the flow information has been accurately transferred from donor cells to recipient cells. Again, since the AMRC scheme can automatically refine cells in a high gradient region, sufficient accuracy can be achieved in inter-grid communication, only 3 to 4 coarsest cell overlap depth is enough to properly match the flow solutions. As long as a cell in level 0 is not the donor cell and recipient cell at the same time, the flow information can then be properly transferred

Chapter 4 Integration of the Chimera Grid Scheme with the AMR Algorithm

through the Chimera common boundaries. Figures 4-17 and 4-18 have shown the reflecting shock wave crosses through the hole boundary and overset boundary, only 4 cell depth is given in this critical case.

4.7 Summary

This chapter documents the integration of the Chimera grid scheme and Quirk's AMR code. A special section is dedicated to the further modification of the original AMR code so as to make it compatible with a Chimera grid system. The code validation demonstrates the computational accuracy, efficiency and the performances of the AMRC scheme at the Chimera overset boundary and hole boundary

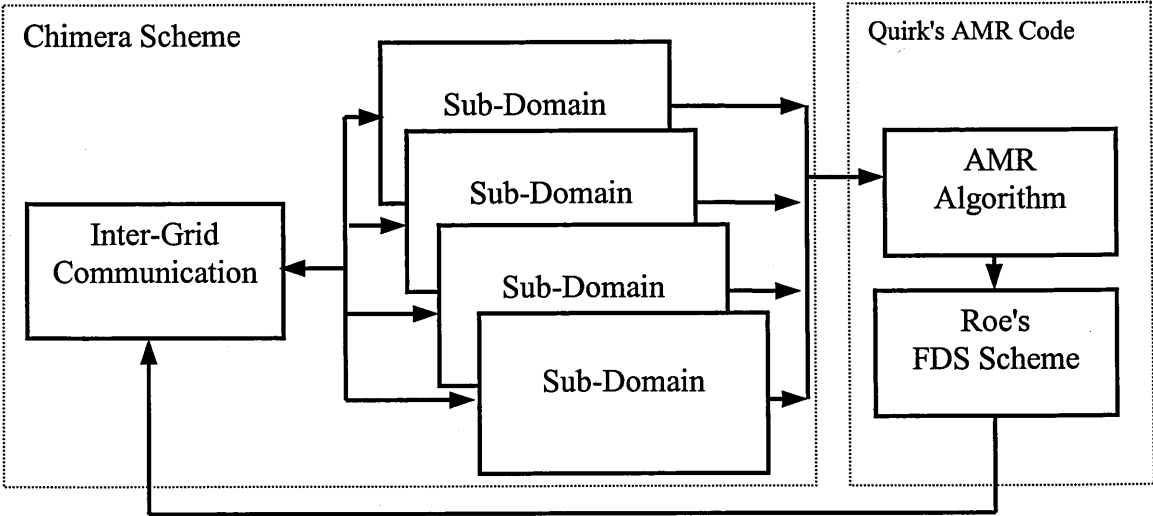


Figure 4-1. Integration of the Chimera scheme and AMR algorithm

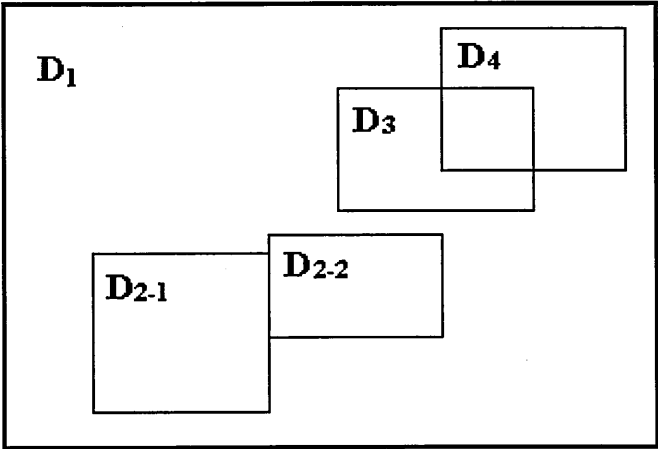


Figure 4-2. Sub-domain type and arrangement

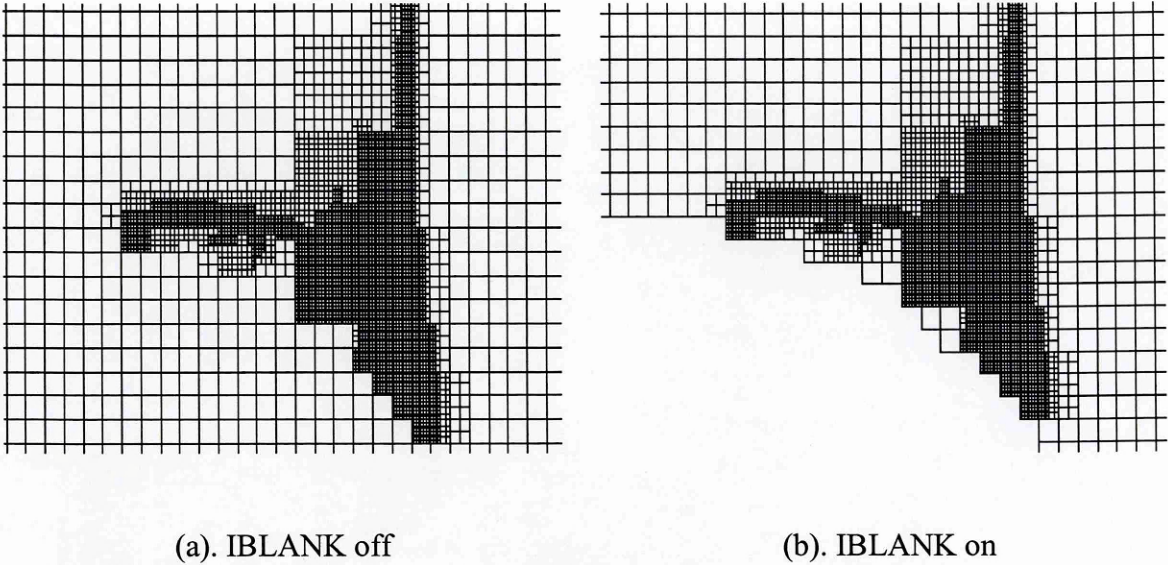


Figure 4-3. Mesh refinement in valid region

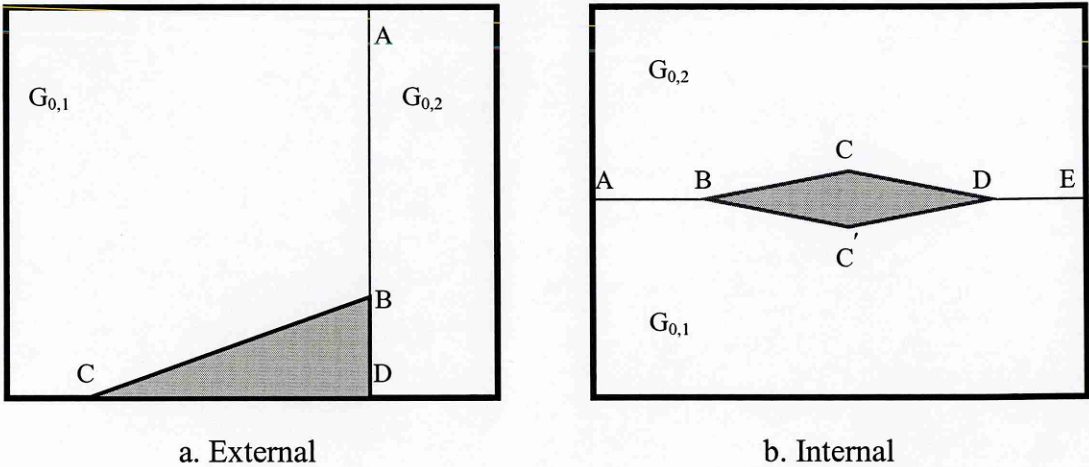


Figure 4-4. Solid boundary type

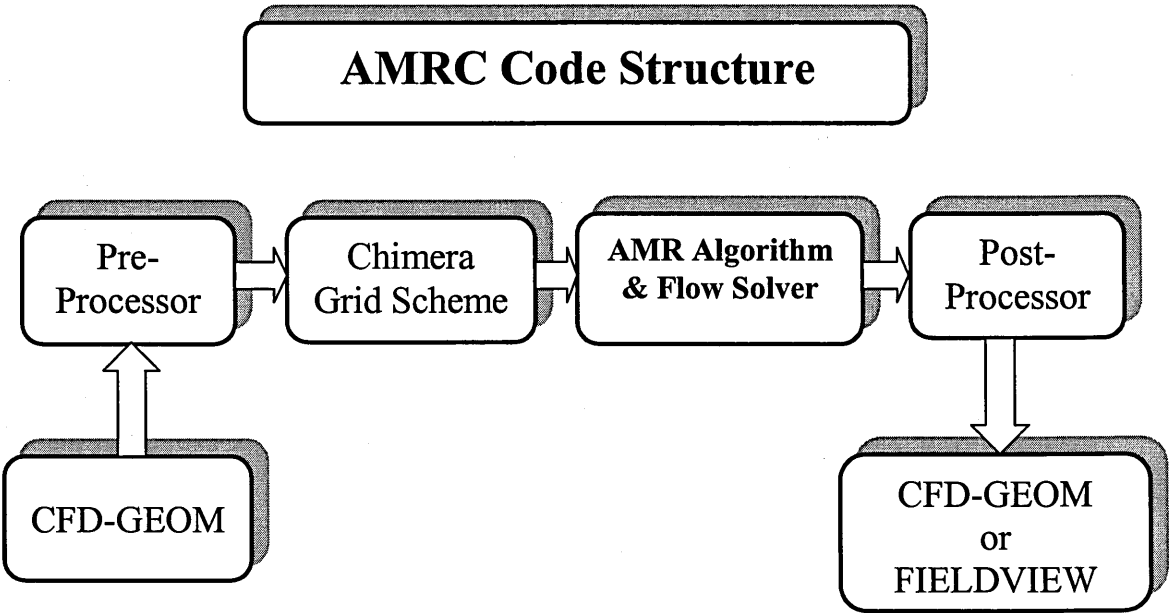


Figure 4-5. AMRC Code Structure

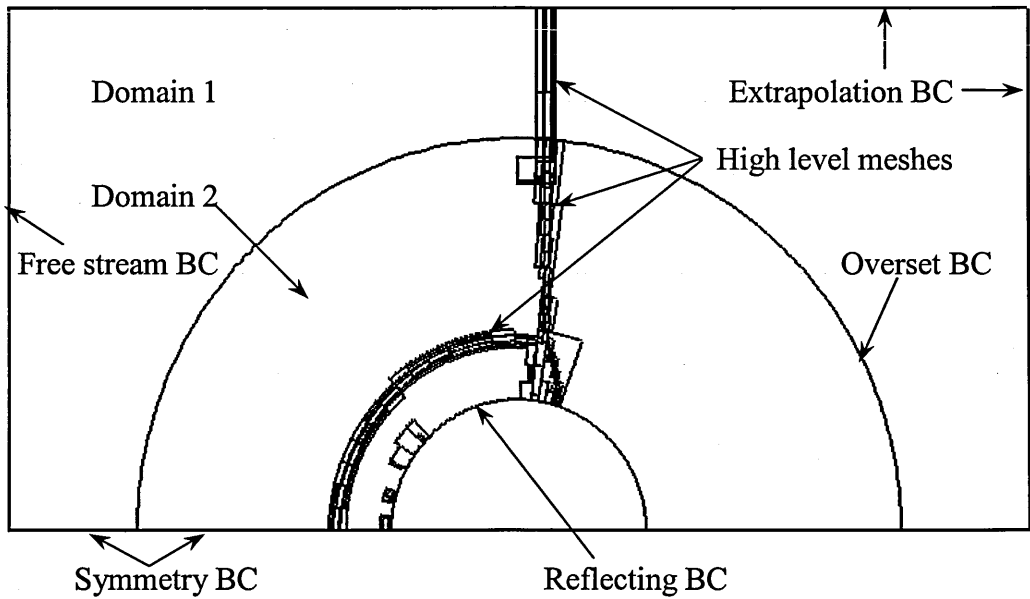


Figure 4-6. Chimera grid system and boundary conditions

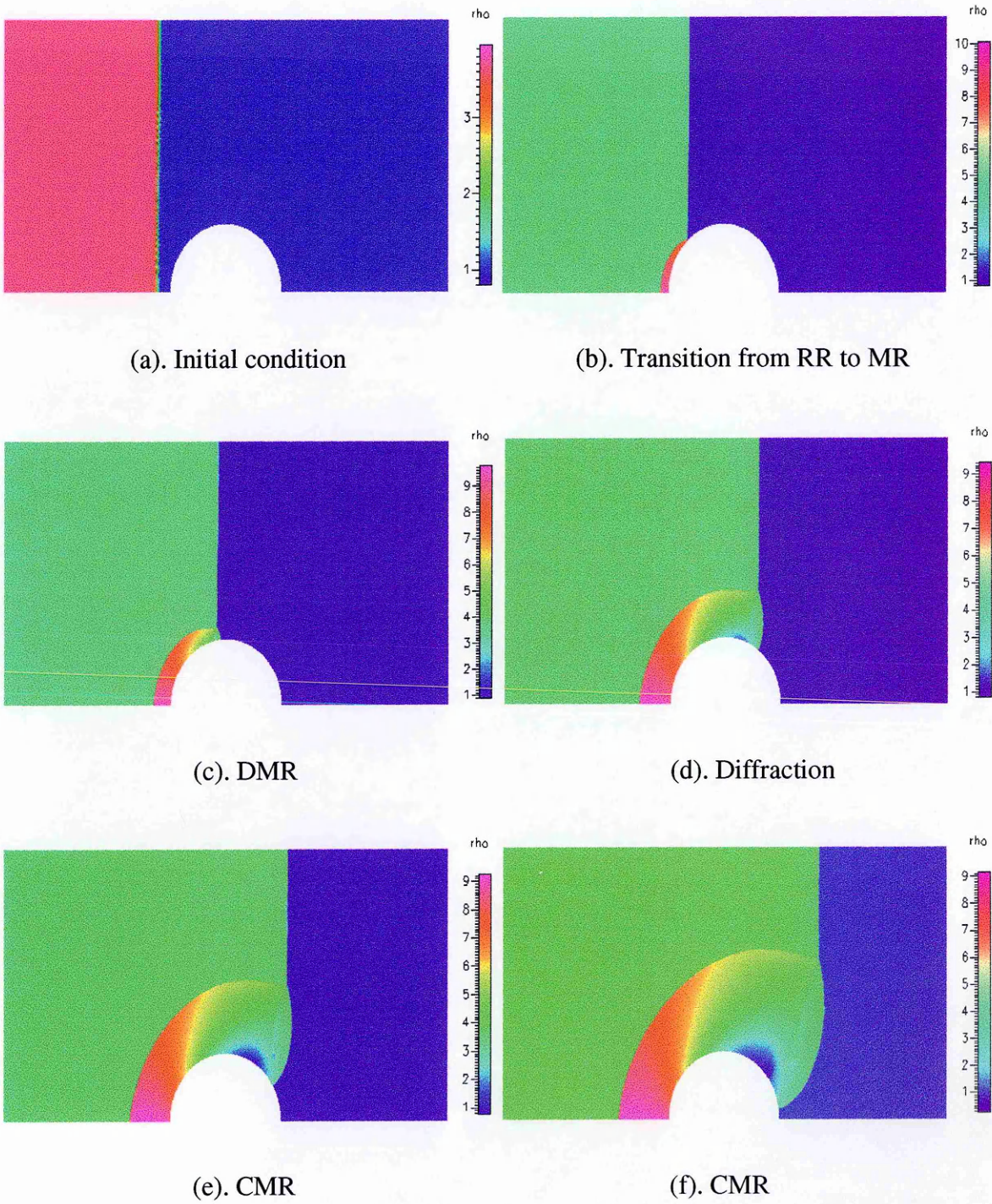


Figure 4-7. Simulation results (density plots)

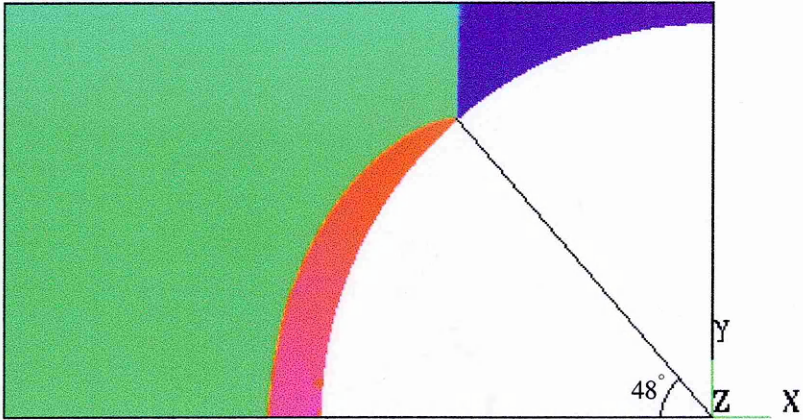
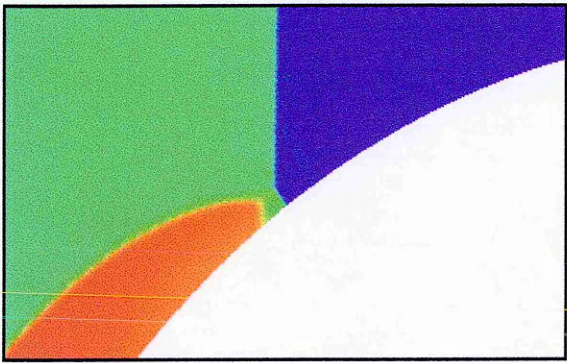


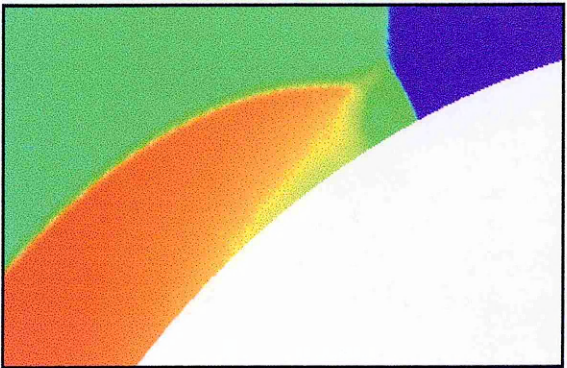
Figure 4-8. Transition point of RR to MR (enlarged density plot)



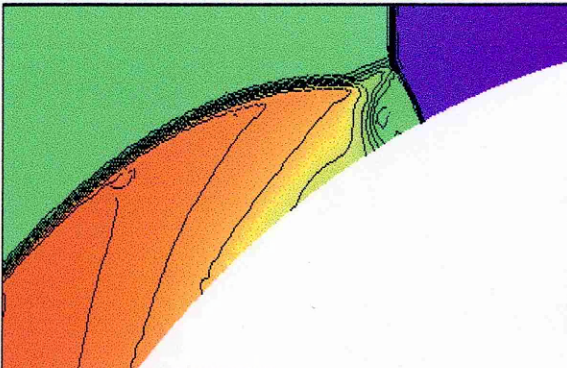
(a). Density



(b). Density contour

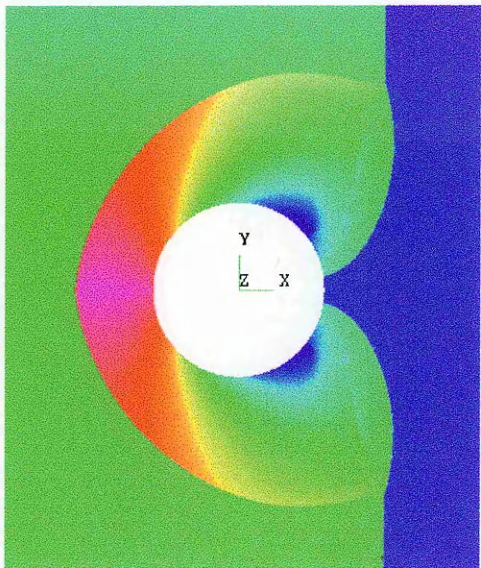


(c). Density

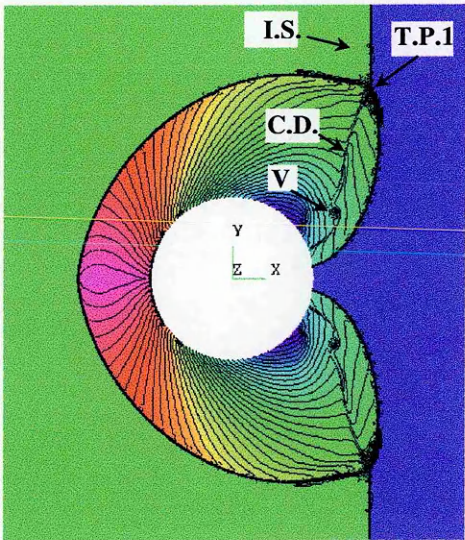


(d). Density contour

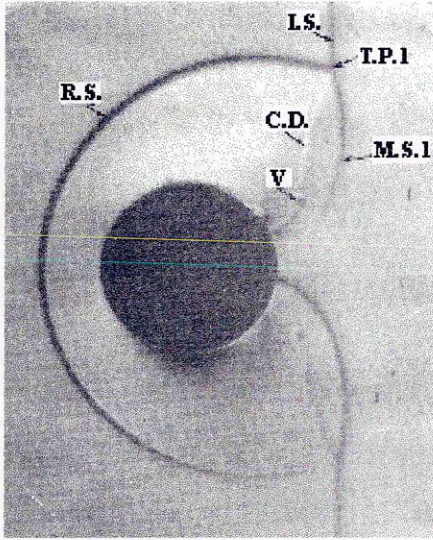
Figure 4-9. Transition from MR to DMR (enlarged)



(a). Simulation result



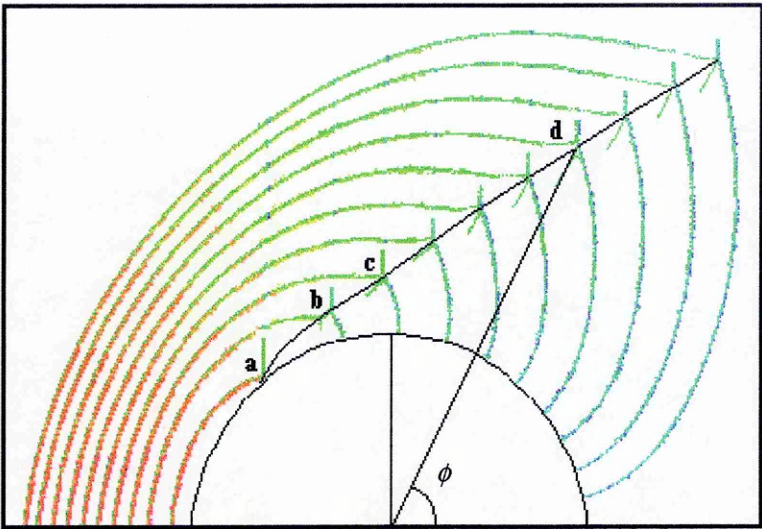
(b). Simulation result (density contour)



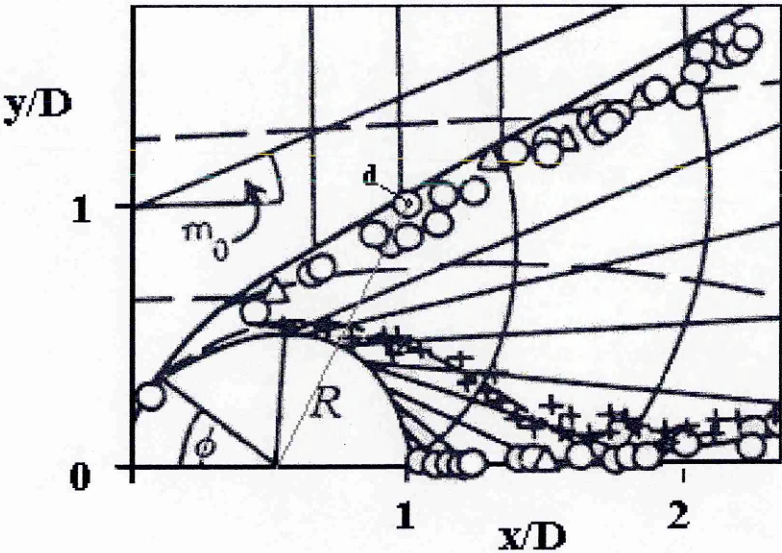
(c). Experimental Schlieren photograph

Key: I.S.—Incident Shock; R.S.—Reflected Shock; M.S.—Mach Stem;
C.D.—Contact Discontinuity; V—Vortex; T.P.—Triple Point.

Figure 4-10, Density plot comparison with experimental data

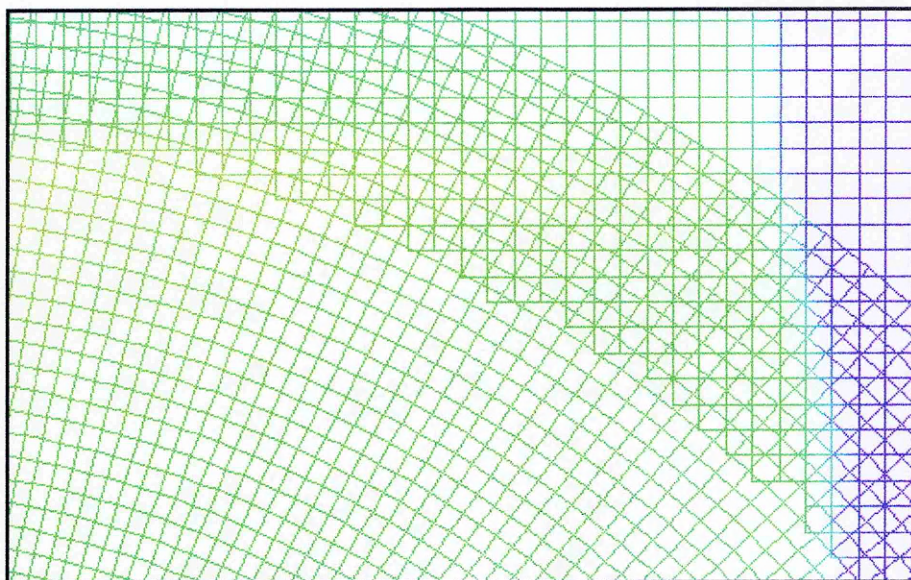


(a). Simulated triple point locus and shock reflection, diffraction pattern

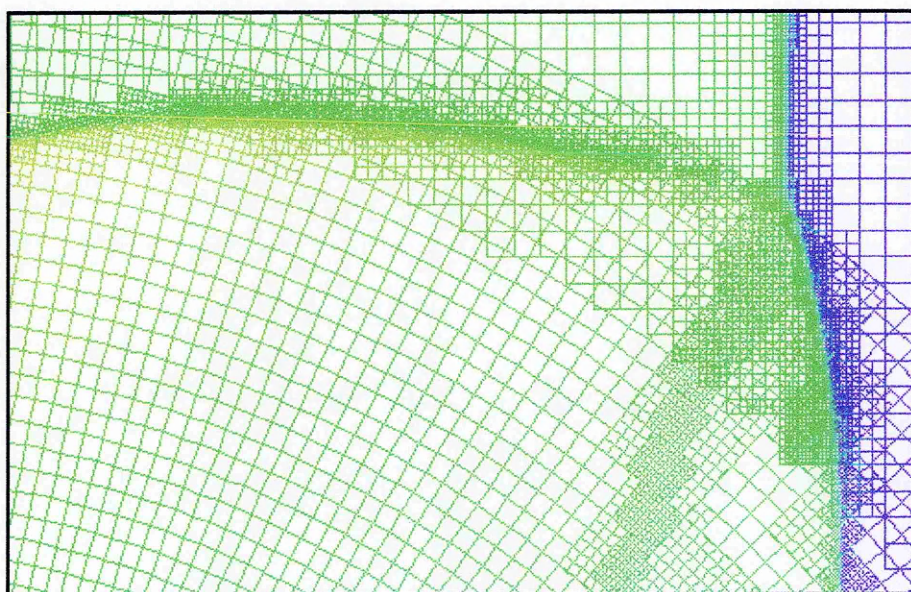


(b). Experimental triple point locus "o" and vortex locus "+"

Figure 4-11. Comparison of the triple point locus



(a). Coarse grids (no adapting)

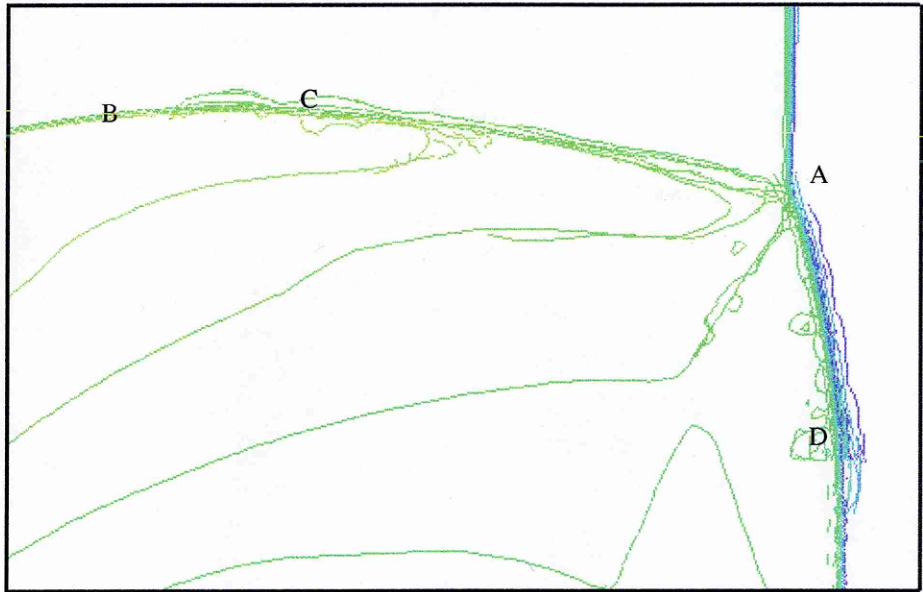


(b). Adaptive meshes

Figure 4-12. Enlarged grid system near Chimera hole and overset boundary

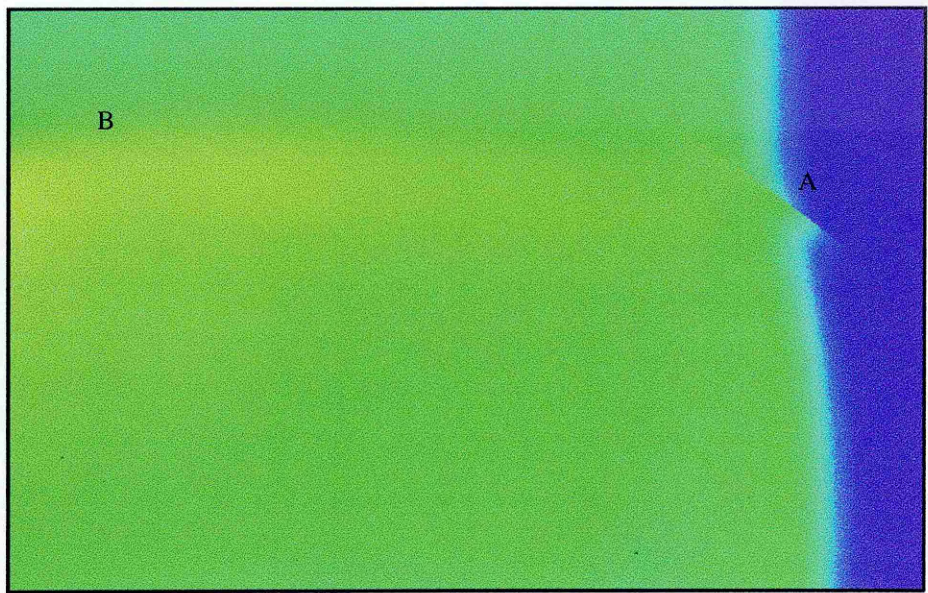


(a). Numerical solution on coarse grids

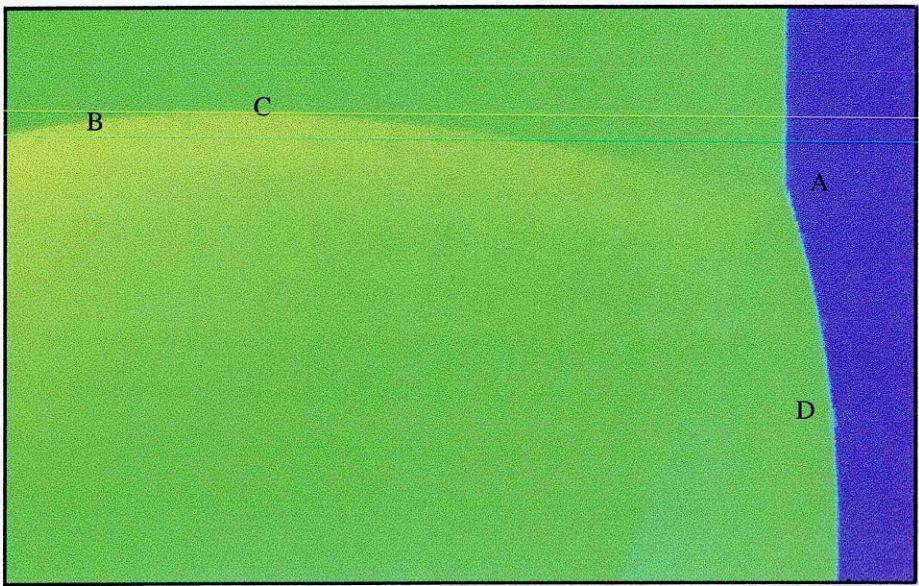


(b). Numerical solution on adaptive grids and meshes

Figure 4-13. Enlarged density contour near Chimera hole and overset boundary



(a). Numerical solution on coarse grids



(b). Numerical solution on adaptive grids and meshes

Figure 4-14. Enlarged density plot near Chimera hole and overset boundary

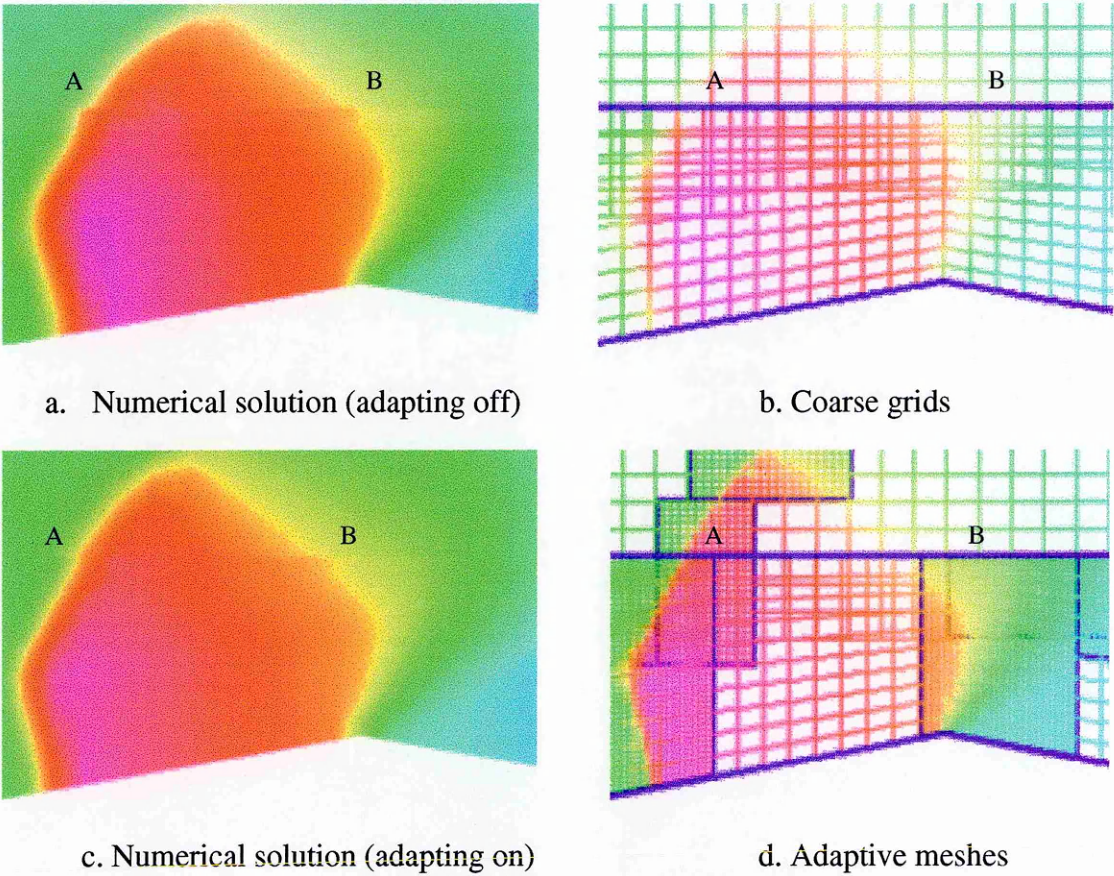


Figure 4-15. Enlarged density plot over a double wedge body[143]

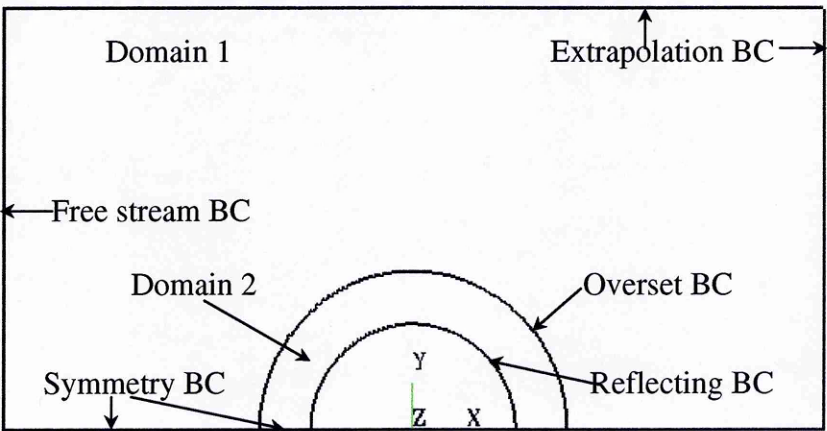


Figure 4-16. Chimera grid system (smaller oversight domain 2)

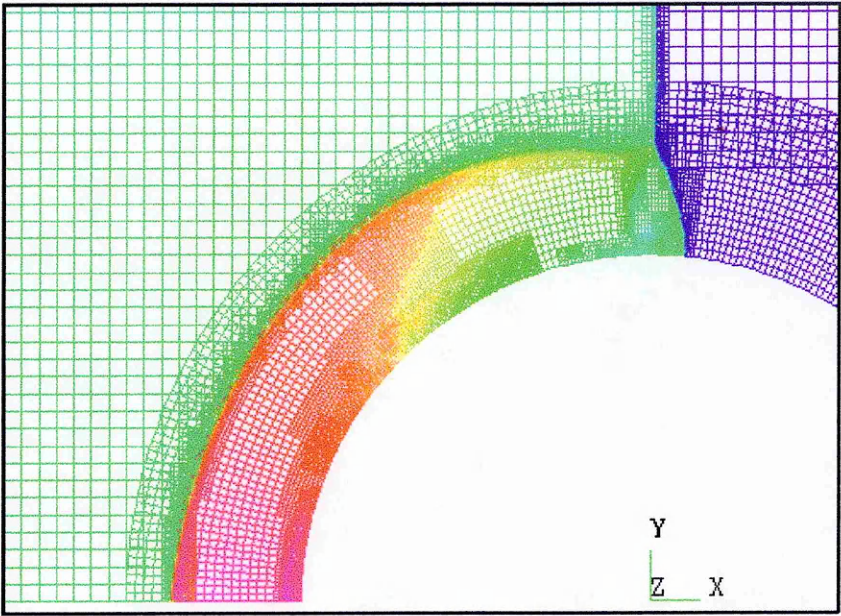


Figure 4-17. Reflecting shock wave crosses hole boundary

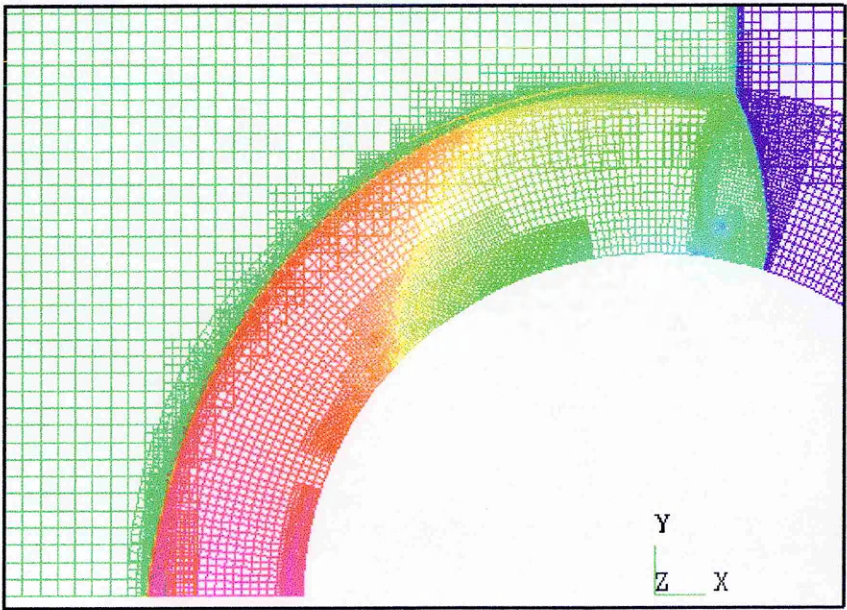


Figure 4-18. Reflecting shock wave crosses overset boundary

Chapter 5

Complex Shock Reflections from Aerofoil at High Angle of Attack

5.1 Introduction

The previous chapter has validated the accuracy of our scheme and carried out the boundary performance investigation on this scheme. In this Chapter, we will apply our scheme to simulate a more realistic complex shock hydrodynamic flow problem where experimental data are available. The flow problem is an unsteady complex shock reflection from an airfoil at high angle of attack and vortex interaction with an aerofoil at zero angle of attack. This study will demonstrate the shock capturing capability, accuracy and the flexibility of grid generation of the AMRC scheme for complicated geometries and multiple bodies.

5.2 Specification of the Simulation Problem

An interesting shock-tube experiment was presented in references^{143, 144}. Moon and Yee simulated parts of this flow problem by second-order explicit upwind and symmetric TVD schemes together with limiters for linear fields and for non-linear fields^{145, 146}. In this Chapter, we shall present the details of the experiment, and the comparison between the simulation results from the AMRC code with the experimental results and the simulation results from references^{145, 146} for the same problem.

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

A schematic sketch of part of the experimental apparatus and the flow field are shown in Figure 5-1. After the diaphragm ruptures, a planar-shock wave of Mach number $M_s=2$ forms and travels down the shock tube. In this experiment, the shock tube has a 5cm×5cm inner cross section and its end is open to the ambient atmosphere. In order to keep the shock wave and its induced flow two-dimensional, two parallel plexi-glass plates are attached to the end of shock tube. An NACA0018 aerofoil (with a chord length of 4.5cm, scaled from the original sketch presented in reference¹⁴⁵) is mounted between the plates at an angle of attack of 30°. The aerofoil is located at a distance of $3\frac{1}{3}$ times the shock tube height away from the shock tube end.

The planar shock starts to diffract as soon as it leaves the shock tube, and forms a curved shock wave, which finally impinges on the aerofoil. The curved shock wave loses its strength in time as it diffracts and the moving shock Mach number decreases to approximately 1.5 at the instant of impingement. Complex shock reflection transition events and shock diffraction occur and a vortex starts to form and grows at the upper nose due to the sudden but mild expansion. Meanwhile, the transition process develops gradually but rather strongly from a Regular to Mach reflection on the compressive lower surface. Eventually, the Mach stem developed on the lower side wraps around the trailing edge of the aerofoil and a vortex is generated due to the sudden strong expansion and sharp trailing edge. This vortex slowly moves forward downstream in a quasi-uniform external flow and, finally, impinges a "target" NACA0012 aerofoil (not shown in Figure 5-1) at near head on conditions, and at essentially zero angle of attack.

The experiment investigated three complex shock hydrodynamic phenomena during the shock wave propagation: including shock diffraction on the 90° convex corner, curved shock wave interaction with a NACA0018 aerofoil at high angle of attack, and the trailing-edge vortex interaction with a NACA0012 aerofoil. A pulsed holographic interferometry method was used to observe these three complex shock hydrodynamic phenomena. With sequential advance of time delays for firing the laser light source, this technique provided a time history of shock propagation, transition of the complex shock diffraction and reflection, and the convection of the vortex in a quasi-uniform external flow. The

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

experimental interferograms taken at different time instants are available from references¹⁴³⁻¹⁴⁵.

5.3 Numerical Simulation by Other Researchers

Figure 5-2 is a schematic of the grid systems Moon and Yee used when they simulated the first two parts of this flow problem. They used one Cartesian grid and one C-type grid.

Mainly due to the grid generation difficulty, their numerical simulation was broken into two parts. (a) Simulation of the shock tube that contains a 90° convex corner (with a computational domain extended to cover the region where the aerofoil is supposed to be located but without the aerofoil. And (b) simulation of a time-dependent curved shock interacting with an aerofoil. The flow field solutions obtained in part (a) are stored and used with bilinear interpolation to obtain the necessary initial and time-dependent boundary conditions for the latter simulation. The time-dependent conditions were obtained assuming the presence of the aerofoil had no effect on the flow at the outer boundaries.

The boundary condition treatment for the second part of the simulation is not physically correct and it must introduce some errors into the numerical solution. In order to reduce the error caused by this treatment, the outer boundary of the C-type grid was chosen to $2\frac{1}{2}$ chord lengths away from the aerofoil.

Since the geometry of the shock tube is symmetric about the centerline, only the upper half was simulated numerically by Moon and Yee. Rectangular Cartesian grids were used for both domains with a mesh having 31×31 points used for the inner region (inside shock tube), and a mesh having 291×245 points used for the outer region. In order to preserve the shock width near the aerofoil, the resolution for the latter 2/3 of the outer region was doubled using a metric-discontinuous grid in the x-direction. A hyperbolic C-type grid consisting of 299 points around the aerofoil and 79 points (stretched to the aerofoil body surface) between the body and outer boundary was used to map the aerofoil.

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

The physical domain of interest in the C-grid was confined to 2.5 times chord lengths away from the aerofoil.

5.4 Current Simulation Details

Using a Chimera grid system, such a geometry configuration can be easily mapped and the boundary conditions can be properly specified. An overset domain can be easily added in or removed if necessary. During our calculation, a baseline domain D_1 (composed of two properly nested meshes D_{1-1} and D_{1-2}) is used to mesh the inner region (inside shock tube) and the outer region respectively. These two meshes are uniform Cartesian grid of 25×25 cells (for D_{1-1}) and 125×125 cells (for D_{1-2}) at level 0. For solely simulating the shock diffraction around the 90° convex corner, this grid system is sufficient. However, in order to compare our simulation results with the experimental results as well as the simulation results obtained by Moon and Yee, for simulating the first two shock phenomena, besides the baseline domain D_1 , one overset domain D_2 is used to map the NACA0018 aerofoil. The existence of the "target" NACA0012 aerofoil is ignored at this stage. The overset domain D_2 composed of two body-fitted meshes of 150×20 cells at level 0. The total cell number is less than 10% of that which Moon and Yee used. Figure 5-3a shows the Chimera grid system and boundary conditions.

The refinement monitor functions chosen for this simulation are density gradient and contact surface. The maximum refinement level $L_{max}(Nd)$ is specified as 2 and the other grid adaptation parameters are shown in Table 5-1. Thus the meshes at level 2 has the same resolution as uniformly spaced cells of 500×500 (for domain D_{1-2}).

Table 5-1. Grid adaptation parameters

l	rI	rJ	F_{tol}	P_{tol}
0	1	1	0.05	0.6
1	2	2	0.05	0.6
2	2	2	N/A	N/A

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

The whole flow field is set as inviscid flow. Since the Euler equations contain no length scale the form of the solution to this problem depends only on the initial shock Mach number and the ratio of the specific heats for the fluid, γ . The normalised initial conditions for the quiescent flow are set as $\bar{\rho}=1$ and $\bar{p}=1$. The simulation was started with the exact shock jump condition, which could be defined using the moving shock relationships given in appendix B. A Mach 2.0 normal shock wave was initially specified just before the open end of the shock tube, see Figure 5-3a. The upstream was set as post-shock condition and downstream was set as pre-shock condition. A CFL number of 0.8 is specified for this simulation. The solver we have used in this calculation is Roe's FDS scheme combined with the SUPERBEE flux limiter.

Actually, using the AMRC scheme, the whole experimental procedure can be numerically duplicated, including the trailing-edge vortex interaction with a NACA0012 aerofoil. This is a multiple body flow problem. Such cases can greatly show the ability and flexibility of the geometry packing of the AMRC scheme. Extending the mesh D_{1-2} of domain D_1 to 175×175 cells, while the cell size is retained the same, and placing the "target" aerofoil 9cm away from the NACA0018 aerofoil trailing edge (approximately scaled from the experimental interferogram presented in reference¹⁴⁴), the full experimental geometry can be simulated. One more overset domain D_3 (composed of two body-fitted meshes of 150×20 cells at level 0) is added to map the NACA0012 aerofoil, see Figure 5-3b. The simulation output at different time instants is then compared with the experimental interferograms. For simulating the whole experiment, including the vortex-aerofoil interaction, we use the Chimera grid system shown in figure 5-3b. All the other parameters are retained the same as those used for simulating the first two shock phenomena.

5.5 Numerical Solution and Comparison

This example is a time-dependent problem. The simulation results presented hereon include resolving the complex shock phenomena at three time stages, i.e. shock diffraction on the 90° convex corner, curved shock interaction with a NACA0018 aerofoil at high

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

angle of attack, and the trailing-edge vortex interaction with a NACA0012. In this section, we will present the simulation results and the comparison with experimental results.

5.5.1 Shock Diffraction on a 90° Convex Corner

According to the experimental interferograms, at the exit of the shock tube, an interesting complex flow pattern was observed. The structure of flow fields behind the diffracting shock behaves in a pseudo-stationary way on a 90° convex corner until the Mach wave hits the plane of symmetry. A slipstream emanating from the convex corner rolls into a vortex-spiral. The reflected Mach wave, the incident planar shock, and its diffracted part form a triple point from which the contact surface emanates and ends at the slipstream encircling the vortex. The second shock appears and forms a triangular configuration with the tail of the Prandtl-Meyer fan and the slipstream. After the reflected Mach wave hits the plane of symmetry, most of the features in the flow field retain their structures for some moments. The second shock grows with time and forms a Mach disk after hitting the plane of symmetry.

Figure 5-4 shows three sequential simulation results (enlarged portion) around the convex corner in the early stages of shock diffraction. The mesh cells around the corner have been refined automatically to the flow solution, see Figures 5-4b, d, and f. The medium grid (level 1) has grown to cover the full extent of the flow activity, while the finest grid (level 2) covers only the diffracted shock and the lip vortex. The efficiency in terms of the number of cells has been demonstrated clearly. At this stage in the calculation a very small part of the computational domain is covered by the fine grid.

Initially, the shock simply travels down the shock tube, but when it reaches the exit it diffracts around the lip, see Figures 5-4a and 5-4b. The pressure gradient at the lip causes the corner slipstream to roll up into a vortex, see Figures 5-4c and 5-4d. This flow pattern remains self-similar in time until the corner reflected Mach waves reach the symmetric plane. From Figure 5-4d, we can clearly see that the reflected Mach wave, the incident planar shock, and its diffracted part form a triple point from which the contact surface

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

emanates and ends at the slipstream encircling the vortex. The vortex and the diffracted shock grow in size, see Figures 5-4d and 5-4f, and eventually the diffracted shock becomes a curved oblique wave. In Figure 5-4f, the second shock appears and forms a triangular configuration with the tail of the Prandtl-Meyer fan and the slipstream.

Figures 5-5 and 5-6 make the comparison of the simulation results from the AMRC scheme with the experimental interferograms and the simulation results obtained by Moon and Yee for the shock diffraction on the 90° convex corner. The simulation results agree quite well with the interferograms at each corresponding time instant. It can be clearly seen that the variety of flow features behind the diffracting shock wave, such as the contact surface, slipstream and the Prandtl-Mayer fan, are well resolved. Their interaction with the vortex is clearly shown in the density contour plots.

From Figures 5-5b and 5-6b, we can also see that the second shock grows with time and eventually hits to the symmetry plane forms a Mach disk.

Since the high resolution of the AMR algorithm, in our simulation results the diffracting shock, second shock and Mach disc are well captured within 2-3 finest cells, i.e. captured within one coarse cell at level 0. Figure 5-7 clearly shows the high resolution from the current calculation by applying the AMR algorithm.

Numerical experiments carried out for the same case by Moon and Yee show that a CFL number greater than 0.5 resulted in instability at the vortex core on their grid system. Again, since our initial coarse Chimera grid system is adapted by the AMR algorithm according to the flow solution, around the vortex core region cells were refined automatically, there is no any problem on stability during our calculation at a CFL number of 0.8.

5.5.2 Oblique Shock Interaction with an Aerofoil

As the flow continues to evolve the moving curved shock subsequently impinges with and progresses over the NACA0018 aerofoil, then transient interaction of the curved incident shock with the aerofoil happens. The experimental results claimed that the moving

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

shock Mach number decreases to approximately 1.5 at the instant of impingement. Figure 5-8 is the simulation result, pressure plot, just before impinging the aerofoil. Figure 5-9 shows the pressure relationship of the quiescent and disturbed fluid state versus x (around the $x/L=0.7$ the aerofoil is located, so the pressure ratio equals zero). By convention a shock Mach number is defined in terms of the shock speed and the speed of sound in the quiescent fluid, i.e. $M_s = \frac{u_s}{a_1}$. Based on the moving shock relationship, see Appendix B equation (B-3), we can calculate the shock Mach number, M_s , at this instant. From Figure 5-9b, we have $p_2/p_1=2.6$.

$$M_s = \sqrt{\frac{\frac{p_2}{p_1}(\gamma + 1) + (\gamma - 1)}{2\gamma}} = \sqrt{\frac{2.5 \times (1.4 + 1) + (1.4 - 1)}{2 \times 1.4}} = 1.512$$

This value is very close to the experimental measurement.

The holographic interferograms obtained in the experiment show very complex unsteady flow fields resulting from the curved moving shock wave striking an aerofoil. The flow field contains not only vortices but also multiple discontinuities including the incident shock, reflected shock, Mach stem, and contact surfaces emanating from shock triple points.

Due to the positive angle of attack of the aerofoil, an interesting feature of shock reflection is observed. There is a transition from a short moment of compression to expansion on the upper surface, and compression on the lower surface. Along the upper surface, the shock reflection retains the regular reflection up to the compressive region and then makes a rapid transition from regular to Mach reflection. A single Mach reflection forms a triple point from which the Mach stem, contact surface, and reflected shock emanate.

Figure 5-10 shows the simulated density plot of the oblique shock interaction with the NACA0018 aerofoil at 30° angle of attack. Most of the flow features mentioned above, and the flow features induced by convex corner are well resolved. Figure 5-11 presents the

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

simulated density plots and makes the comparison with experimental results at the various stages of the interaction and subsequent trailing edge vortex generation. They are close-ups of the interaction around the aerofoil. The first two columns show the current simulation results, density plots and density contours. The third column shows the experimental holographic interferograms and the last column shows the simulated results, density contours, from Moon and Yee. Comparing these Figures we can claim that very good agreement has achieved by our simulations, although very coarse cells (at level 0) are employed.

The top row of Figure 5-11(No.1 to 4) shows that a vortex starts to form and grows very slowly at the upper nose due to the sudden but mild expansion. Meanwhile, the transition process develops gradually but rather strongly from a regular to Mach reflection on the compressive lower surface. The Mach stem and contact surface have already started to appear on the lower surface. The Mach stem stands normal to the aerofoil body. Near the aerofoil body, the thickness of the Mach stem is thinner due to the smaller cell size of the overset domain (about one quarter of the cell size of the baseline domain). It is for this reason (cell size difference) that there is a sudden change of the shock thickness at overset boundaries.

In the middle row of Figure 5-11(No.5 to 8), the Mach stem on the lower surface reached the aerofoil trailing edge. And eventually, the Mach stem developed on the lower surface wraps around the trailing edge of the aerofoil and a vortex is generated due to the sudden strong expansion and sharp trailing edge, see Figures 5-11 (No. 9-12). At this instant, the foot of the lower Mach stem proceeds upstream and collides with the upper diffracted incident moving shock on the upper surface.

On the upper surface, the experimental results show a weak contact surface, which is not clearly captured in both simulations. By increasing the grid resolution around that region, or by decreasing the flagging tolerance F_{tol} , this slip surface would also be expected to be captured. It should be pointed out that the two little bumps on both the upper and lower surfaces at about two thirds of the aerofoil chord length observed in the interferogram resulted from vibration of the experimental setup. Also, because the aerofoil used in the

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

experiment is a hand-made NACA0018 aerofoil, there is a possible manufacturing error resulting in, a slight difference exists between the experimental and computed configurations.

The calculations were performed on a Sun Ultra10 workstation. For the above simulation case, less than two hours real running time were used.

5.5.3 Vortex Interaction with an Aerofoil

As the flow evolves further, the trailing edge vortex detaches from the NACA0018 aerofoil and travels slowly forward to the downstream target NACA0012 aerofoil. The experimental results have revealed the detailed flow structure both before and during the interaction, and provide the flow parameter profiles across the vortex.

Figure 5-12 presents the simulated whole flow fields (density plot) before and during the vortex interaction with the target aerofoil. The complicated flow structures are well resolved. The interaction between the slipstreams, second shock and the vortices induced by 90° convex corners develops and the vortices grow in size. The second shock wave is developing and is about to impinge the NACA0018 aerofoil. The Mach disc moves downstream. The further the Mach disc moves downstream the smaller it becomes and it starts to lag behind the vortex.

The numerical errors caused by the extrapolation boundary conditions at the north and south boundaries of the baseline domain mesh D_{1-2} (curved shock wave was slightly reflected into flow field) are also can be seen, but it has no effect on the vortex generation and interaction with the NACA0012 aerofoil.

Unfortunately, experimental holographic interferograms are not available for the whole flow field. However, some close-up holographic interferograms around the vortex and flow parameter profiles across the vortex were presented in reference¹⁴⁴. Figure 5-13 makes the comparison between the simulated density plot and the experimental result. It is a close-up snapshot with the vortex travelling between the two aerofoils. This Figure shows that the moving curved shock has progressed about 75% of the chord length along the

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

target aerofoil. On the way to the target aerofoil, the vortex is indeed convected in a quasi-uniform flow. This feature is shown in Figure 5-13a and in Figure 5-13b in terms of the very few widely spaced fringes in the external stream surrounding the vortex. The vortex due to the NACA0018 aerofoil trailing edge detaches from the aerofoil travels slowly downstream and eventually impinges the NACA0012 aerofoil.

The vortex induced by the NACA0018 aerofoil nose has experienced a quite different history. It has interacted with the upper surface on the NACA0018 aerofoil and has grown to a larger size. However its downstream progress is clearly impeded in comparison with that of the trailing edge vortex. Figure 5-14 shows the enlarged flow field around the NACA0018 aerofoil. From both density plot and U velocity plot, it can be clearly seen that a couple of smaller vortices appear on the upper surface behind the first large vortex. The U velocity plot shows that the rotations of these vortices are all clockwise. These smaller vortices can be captured due to local refinement of the AMR algorithm. Figure 5-15 shows the local adaptive meshes around the NACA0018.

Figure 5-16 presents profiles of the flow variables across the vortex at the stage just before it reaching the target aerofoil (approximately the same location as the experiment). Figures 5-16a and 5-16c make the comparison between the simulated and the measured radial pressure profiles of the vortex. There is a pronounced partial vacuum at the centre, corresponding to a pressure drop by a factor of about 3 or more. Figures 5-16b and 5-16d show the simulated and measured vortex density profiles. Their shapes are seen to be somewhat similar to that of the pressure profiles. The simulated pressure and density drops are large than those of experimental data. This is because our calculation is inviscid.

Both the simulation results and the experiment results show that the vortex moves more slowly than the surrounding uniform flow does. Figure 5-16e shows the simulated u velocity profiles across the vortex. At the core of the vortex, the u velocity is 179.56 m/s, while the experiment results claimed that the vortex velocity was relatively uniform around 180 m/s. Figures 5-16f and 5-16g show the simulated v velocity profile and that obtained from the measured density and pressure data. The v velocity values show that the vortex rotation is anticlockwise.

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

The large radial pressure variation through the vortex implies that a close encounter between vortex and aerofoil produces serious transient fluctuations in both the lift and drag coefficients. Figure 5-17 shows four successive stages of a head-on interaction with the target aerofoil at zero angle of attack (but at near zero angle of attack in experiment, a slight positive angle of attack can be seen in Figure 5-13b). Rotation in the incident vortex is anticlockwise. Thus as the vortex approaches the aerofoil, there is an upwind component of velocity across the aerofoil nose, i.e. perpendicular to the chord of the aerofoil. As a result, a secondary vortex is produced having clockwise rotation. The interaction between the two vortices is very marked in both simulated and experimental results.

Comparing the simulation results with experimental results at various stages, one can clearly see the high resolution and accuracy achieved by the AMRC scheme.

5.6 Summary

The calculations presented in this chapter have demonstrated the flexibility of grid generation of the AMRC scheme for complicated geometries and multiple bodies. The comparisons with the experimental results have shown the shock capturing capability and the accuracy of the AMRC scheme for simulation complex shock phenomena. The comparisons with the simulated results from literature have greatly shown the grid efficiency and computational efficiency of the AMRC scheme. Using less than 10% cell numbers of that used by other researchers for the same problem, the complex shock features are well resolved.

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

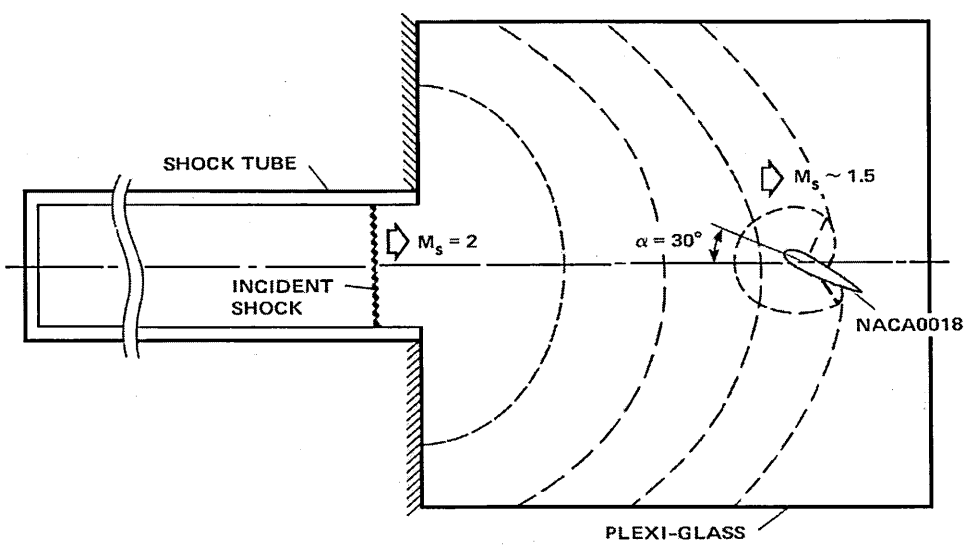


Figure 5-1. Schematic of the experiment [145]

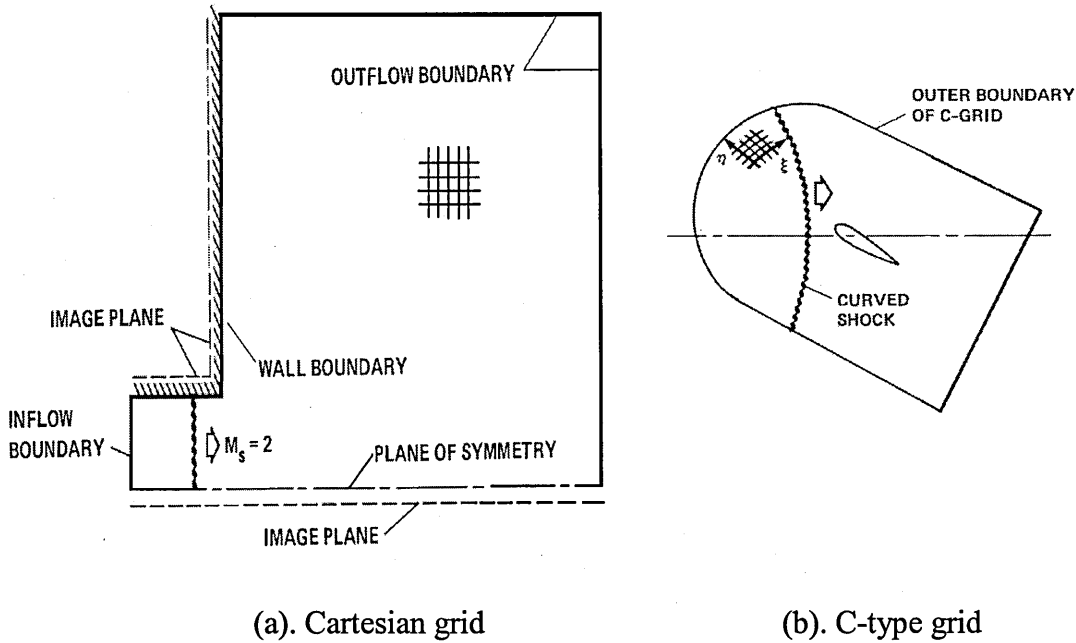
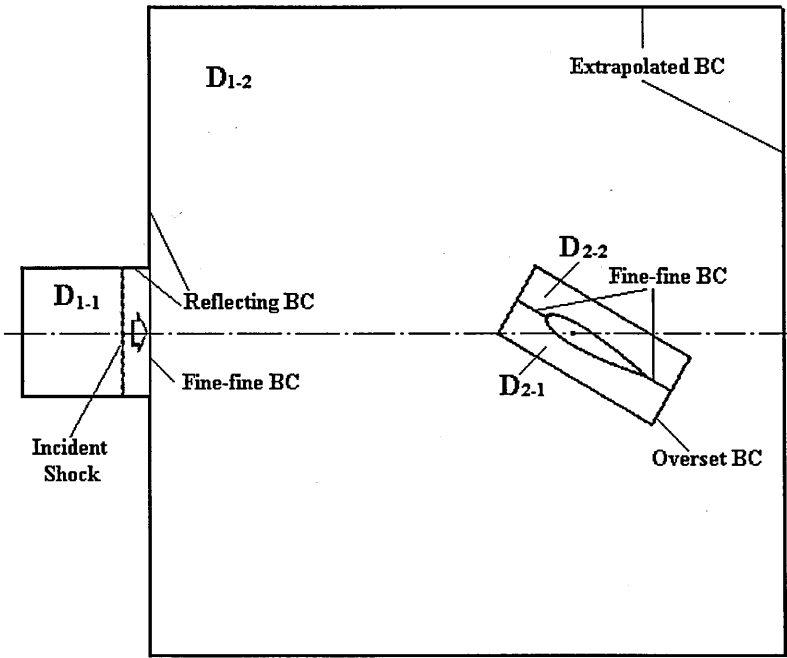
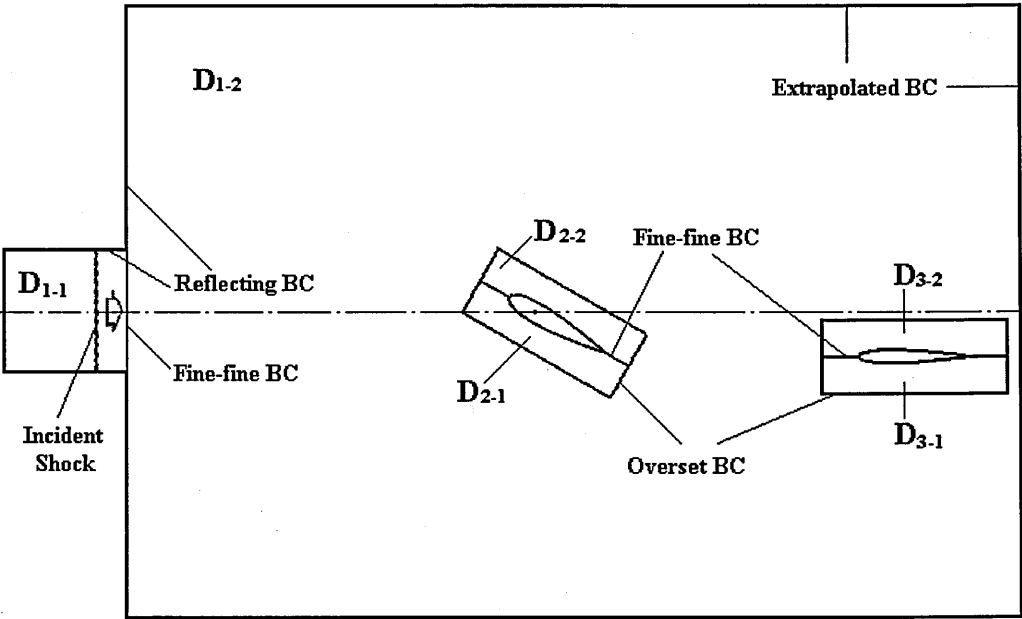


Figure 5-2. Schematic of the computational domain in reference[145]

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack



(a). With one aerofoil



(b). With two aerofoils

Figure 5-3. Chimera grid system and boundary conditions

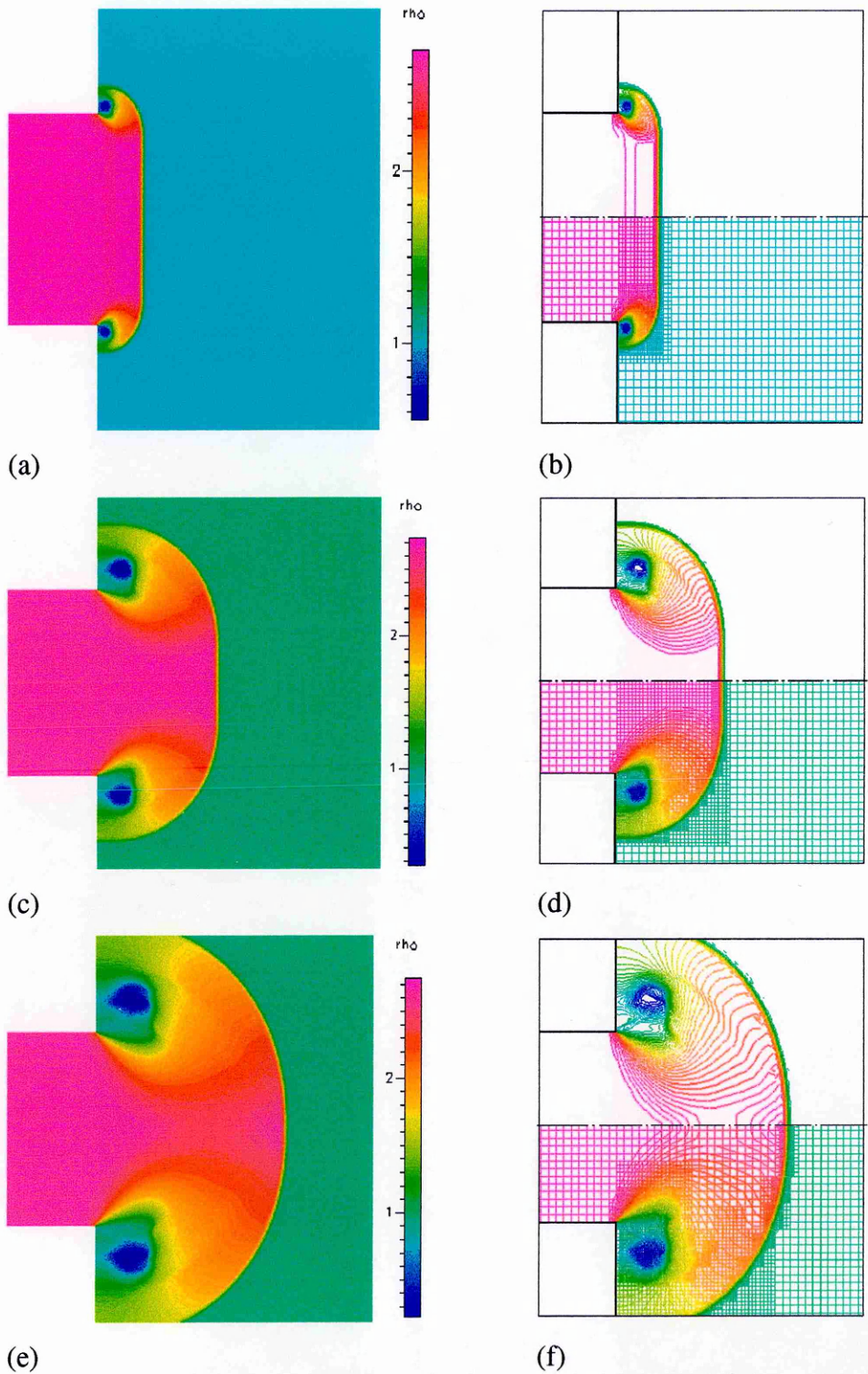
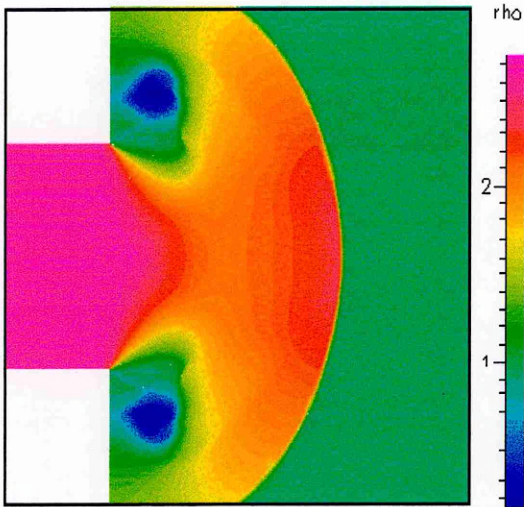
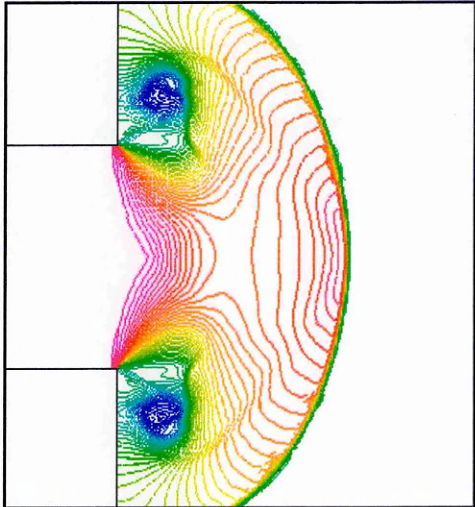


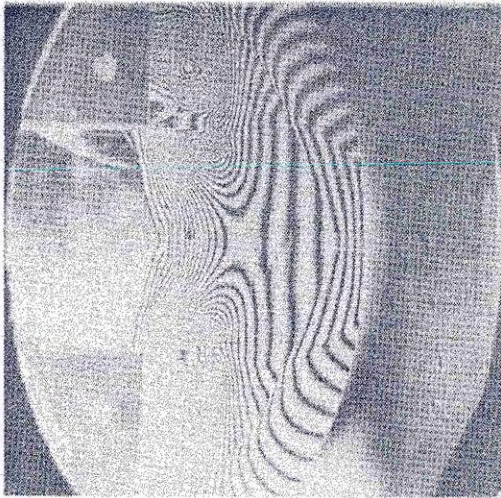
Figure 5-4. Simulation results (density plots and contours) around the convex corner.



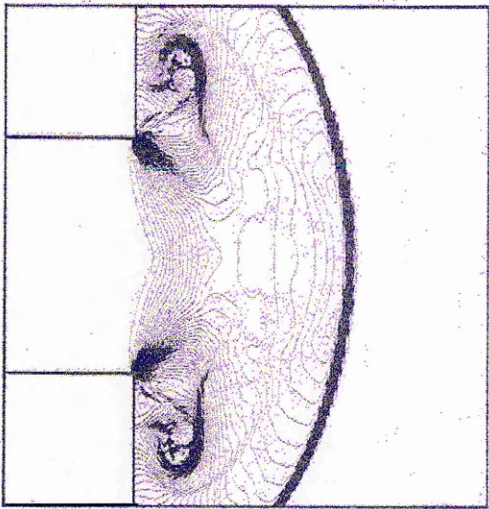
(a). Flooded density flow field plot



(b). Density contours

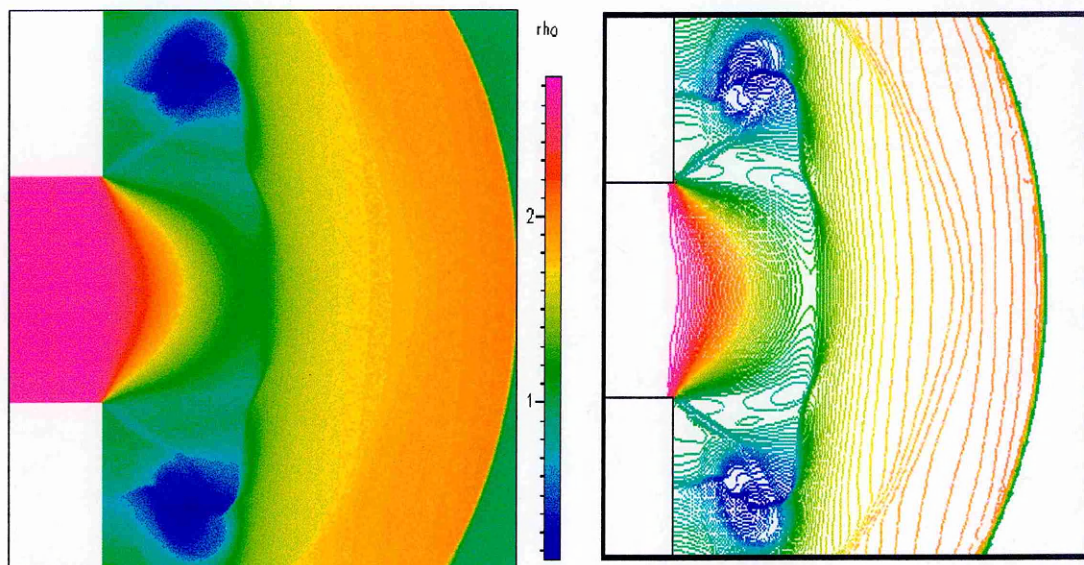


(c). Experimental interferograms [145]



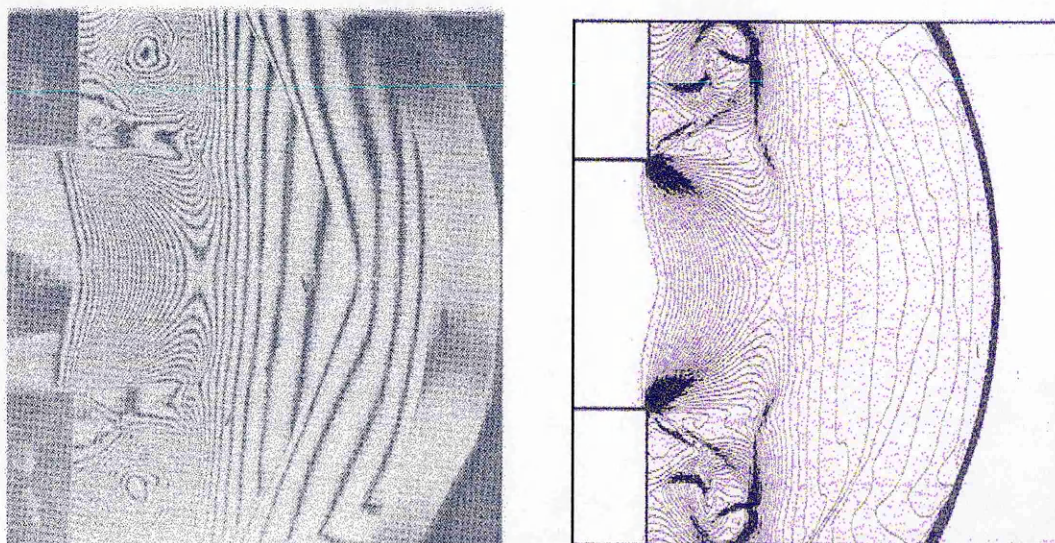
(d) Simulation result presented in reference [145]

Figure 5-5. Comparison of density contours with the experimental Interferograms and simulation result from the literature



(a). Flooded density flow field plot

(b). Density contours



(c). Experimental interferograms [145]

(d) Simulation result presented in reference [145]

Figure 5-6. Comparison of density contours with the experimental Interferograms and simulation result from the literature

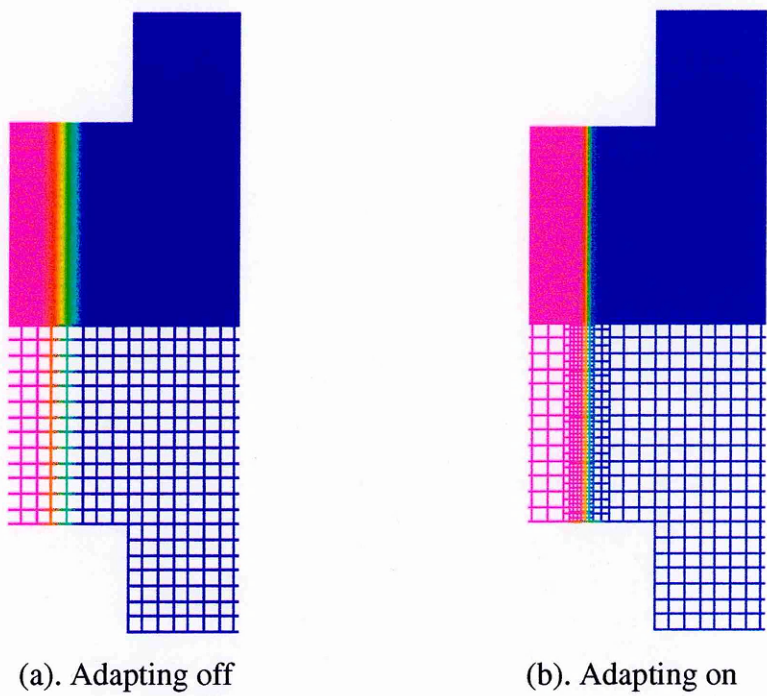


Figure 5-7. Comparison of the resolution

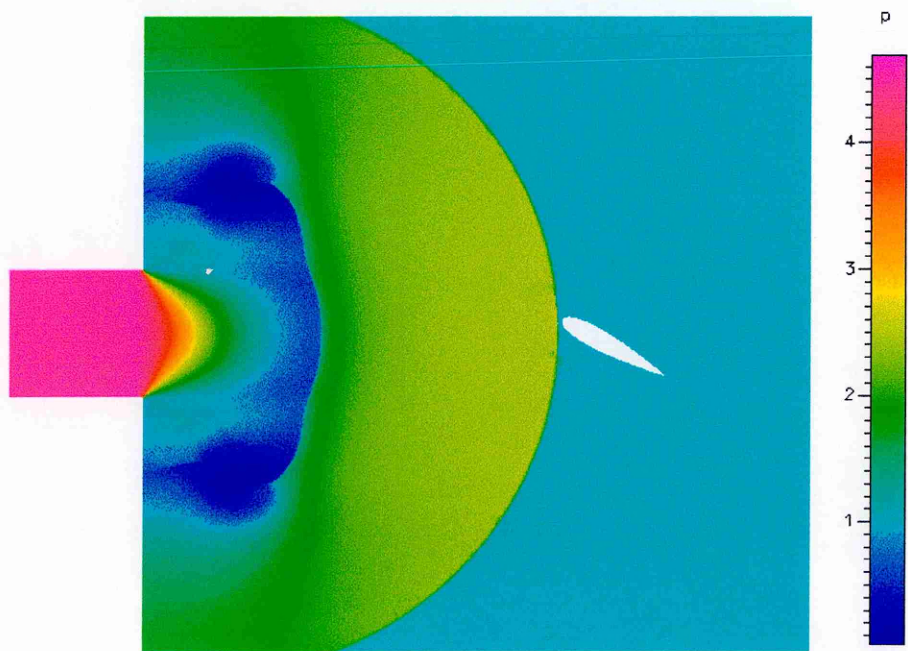


Figure 5-8. Pressure plot just before impinging the aerofoil

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

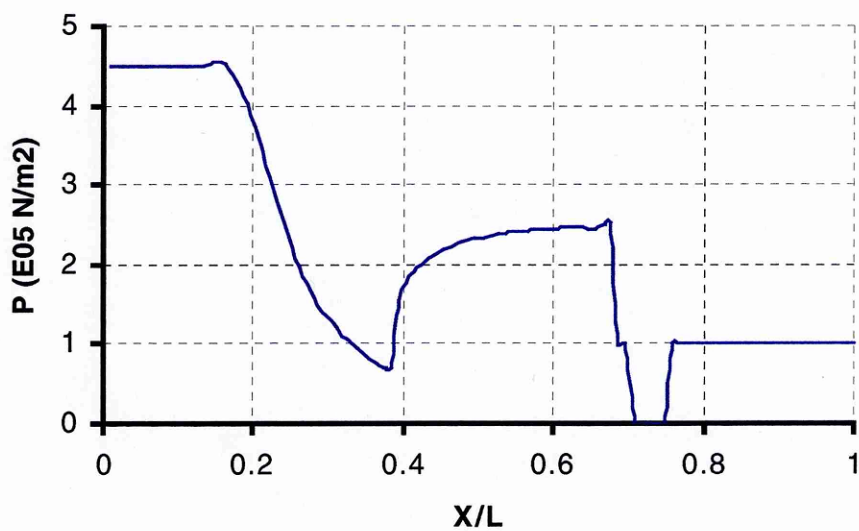


Figure 5-9. Line prove along the symmetric plane

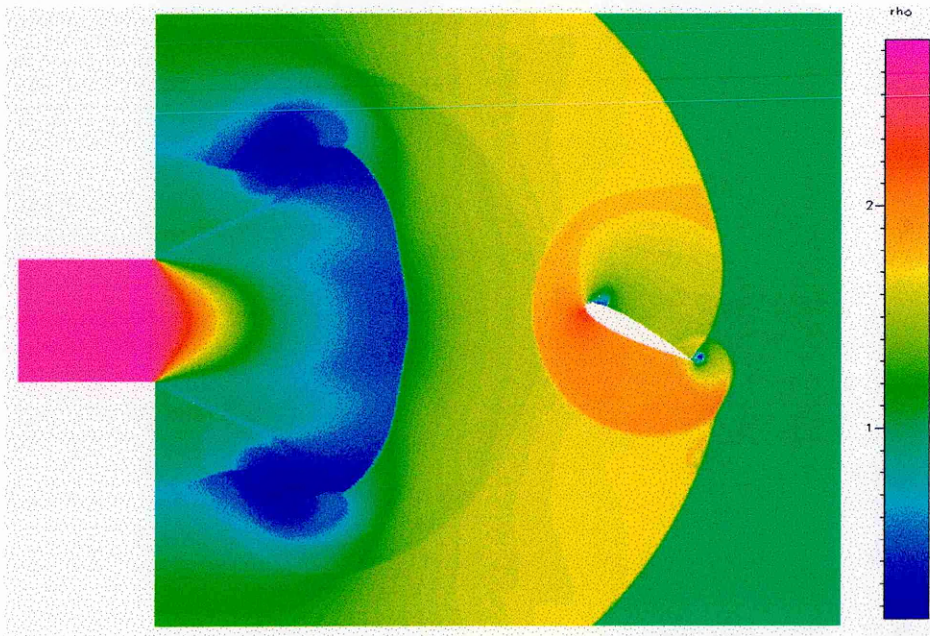


Figure 5-10. Density plot of the oblique shock interaction with the aerofoil at 30° angle of attack

Chapter 5 Complex Shock Reflections from Aerofoil at High Angle of Attack

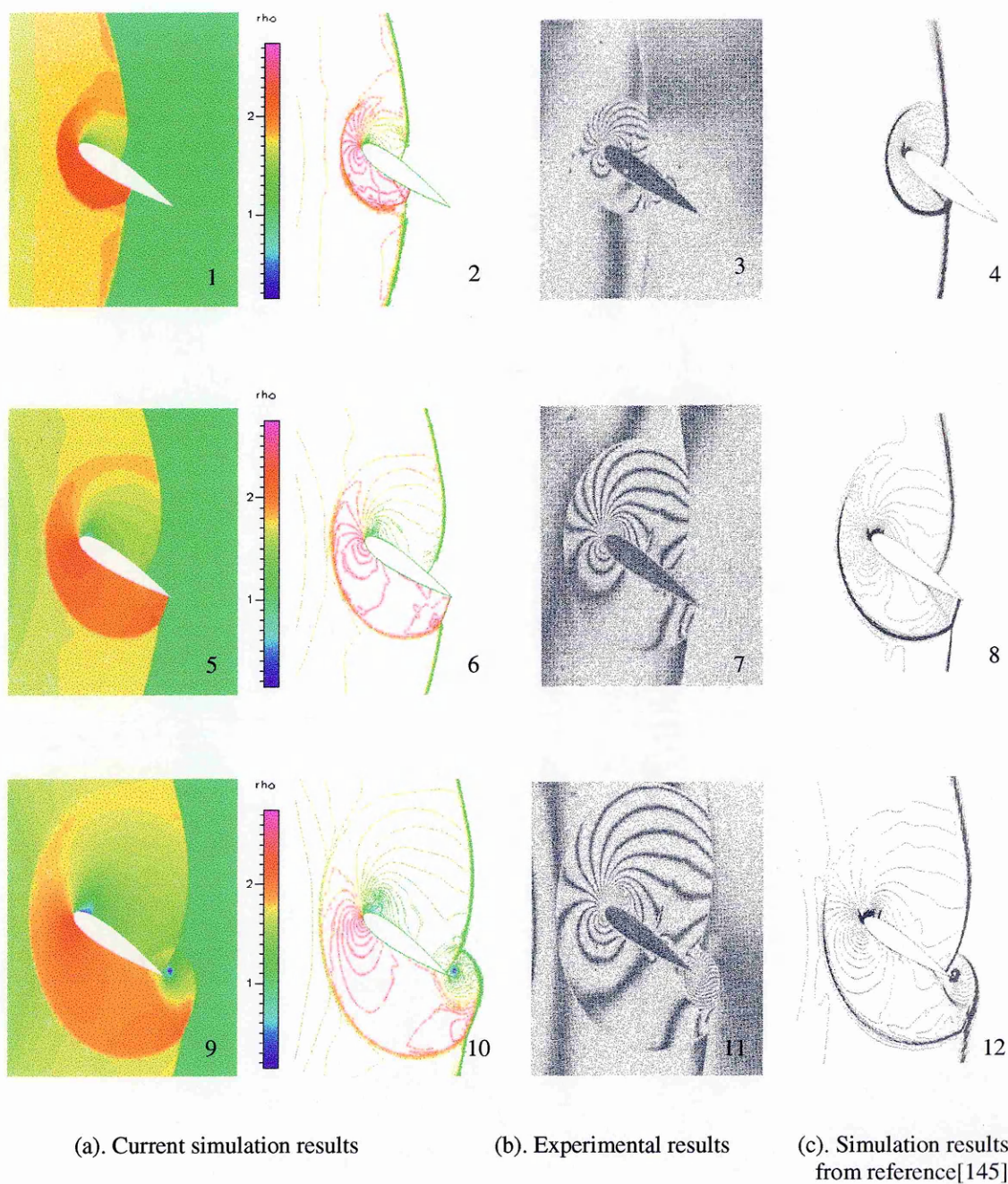
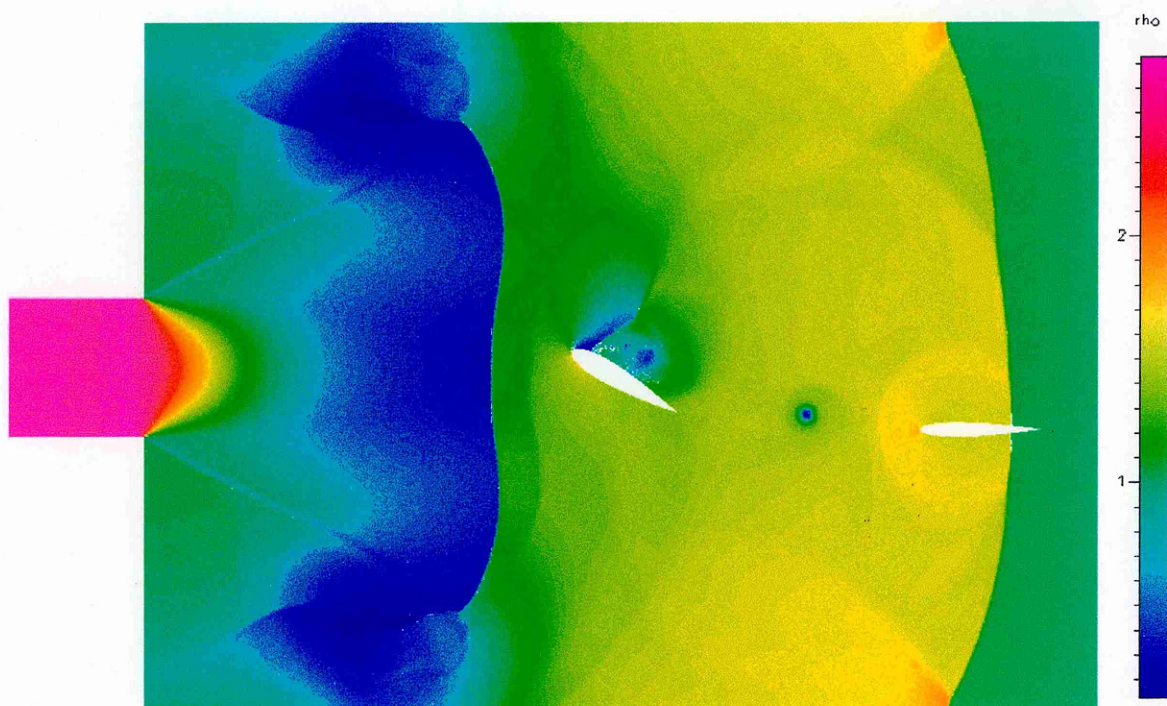
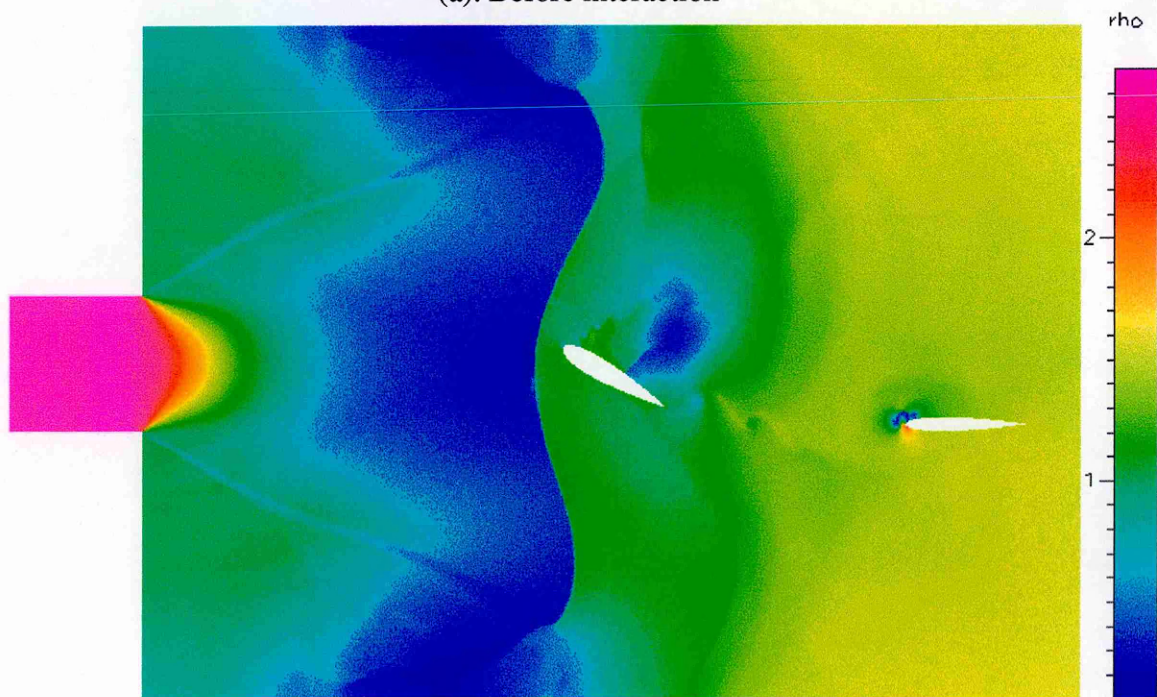


Figure 5-11. Comparison with the experimental Interferograms and simulation result from the literature

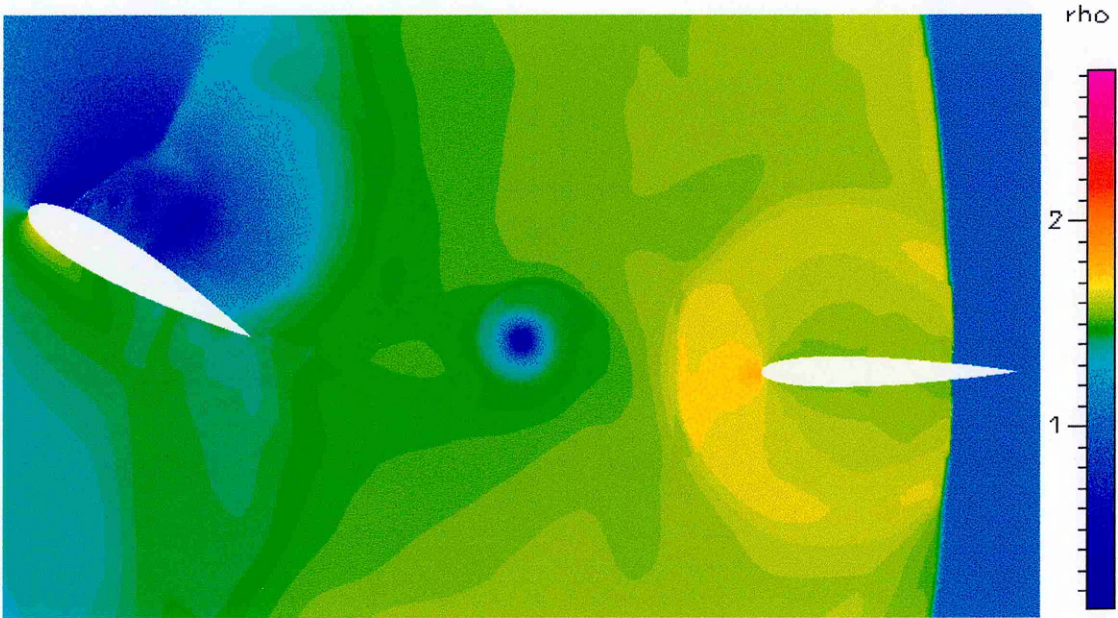


(a). Before interaction

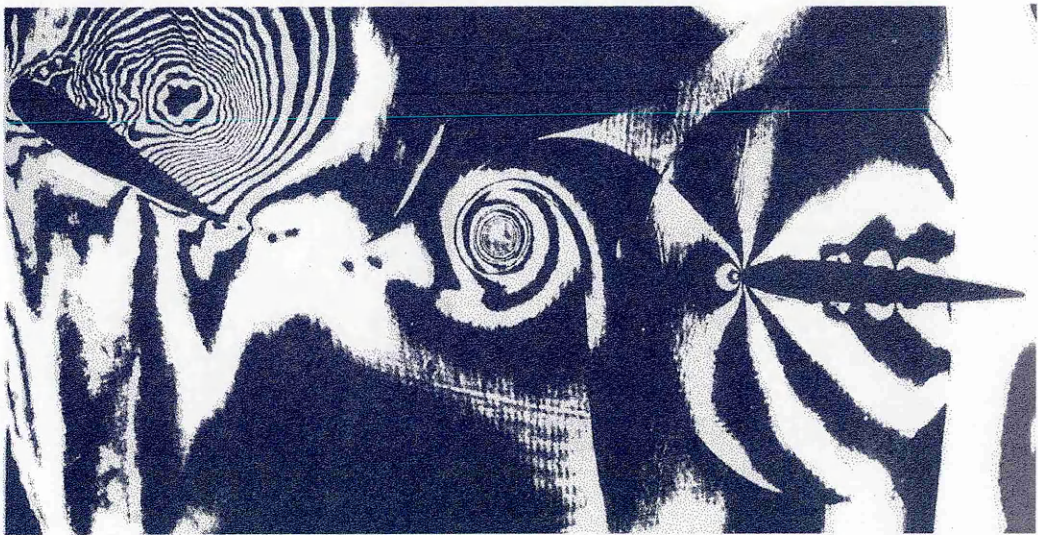


(b). During the interaction

Figure 5-12. Vortex interaction with the target aerofoil

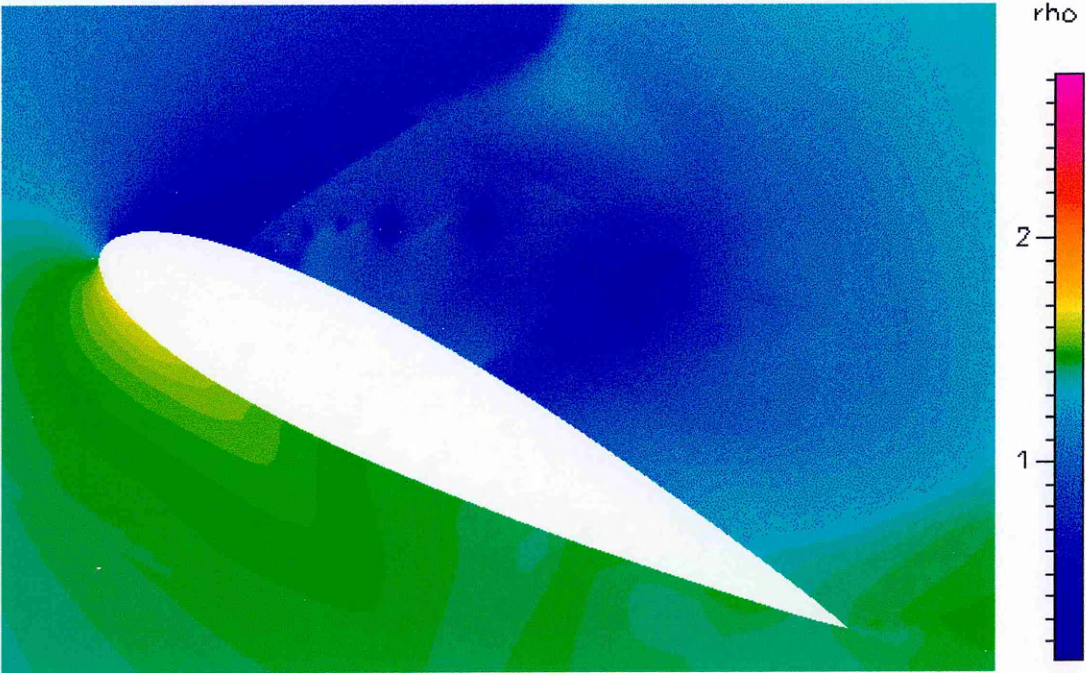


(a). Simulated density plot

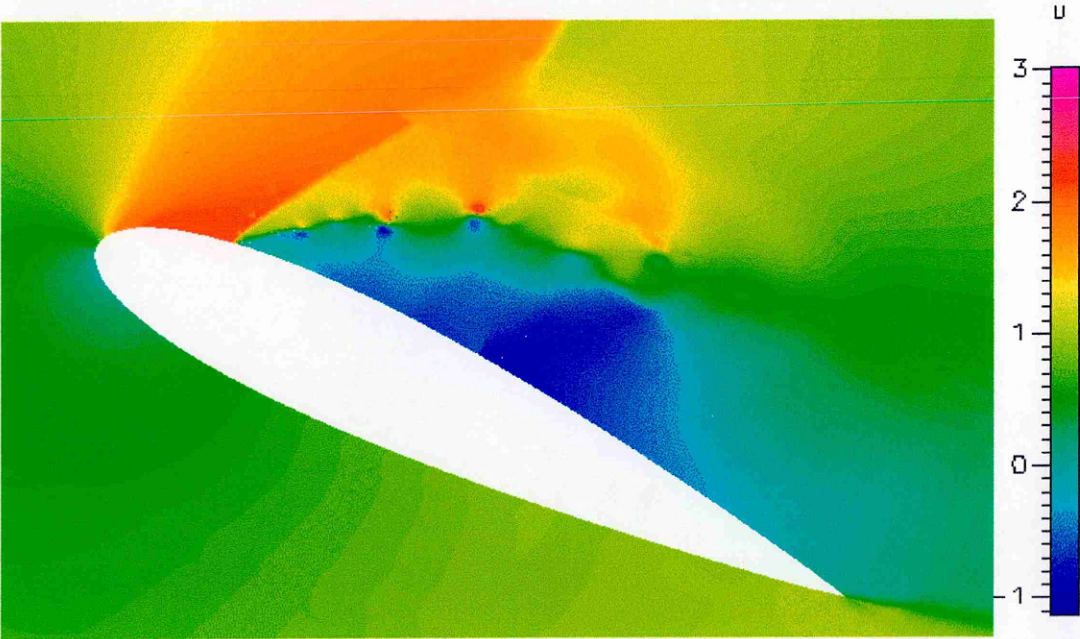


(b). Experimental interferogram

Figure 5-13. Vortex traveling to downstream (enlarged)



(a). Density plot



(b). U velocity plot

Figure 5-14. Enlarged numerical solution around NACA0018 aerofoil

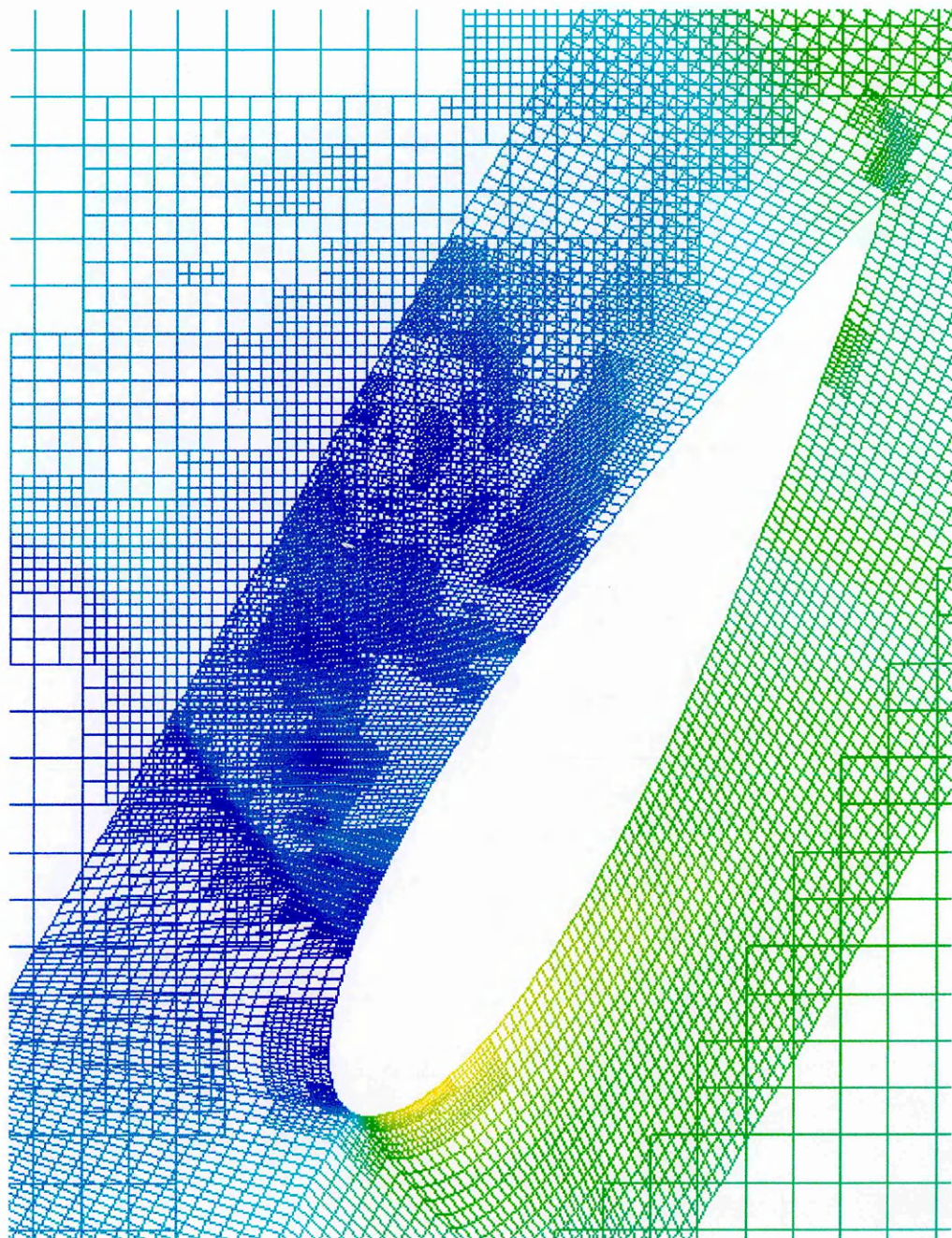
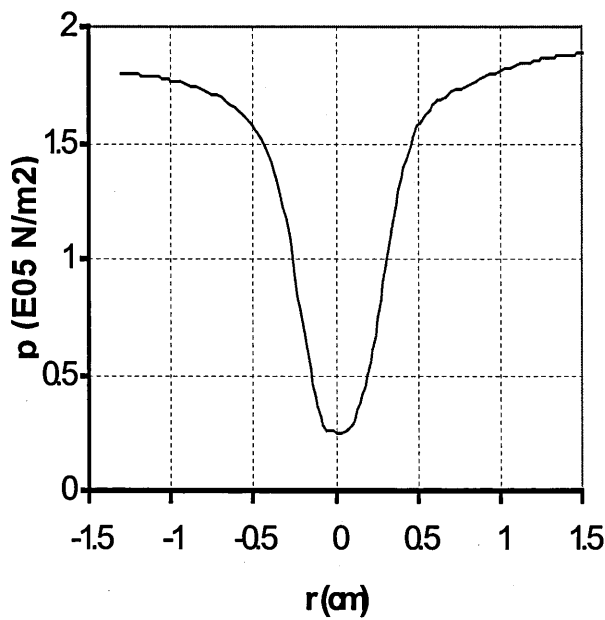
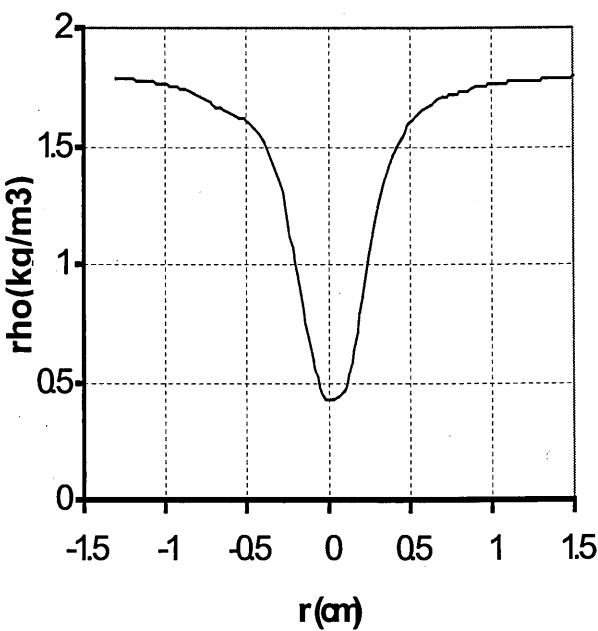


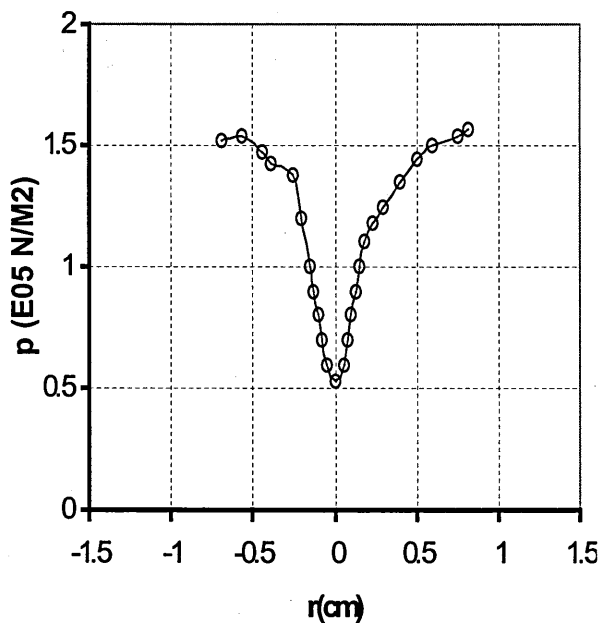
Figure 5-15. Adaptive Meshes (with density) around the NACA0018 aerofoil



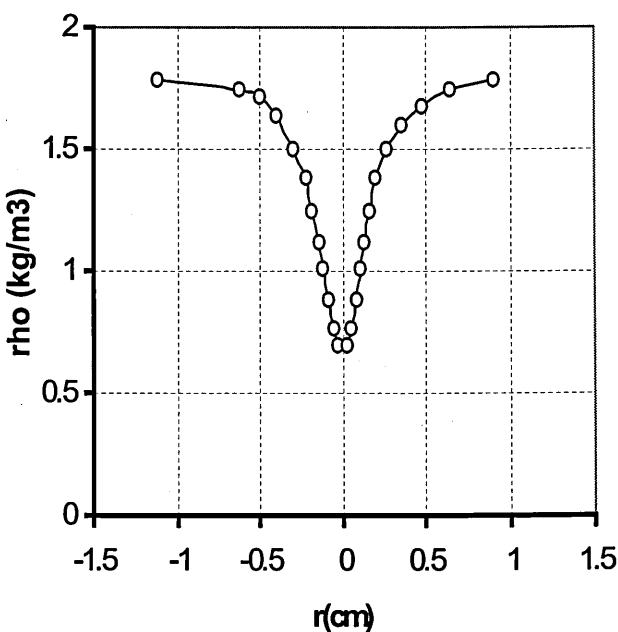
(a). Simulated vortex radial pressure



(b). Simulated vortex radial density

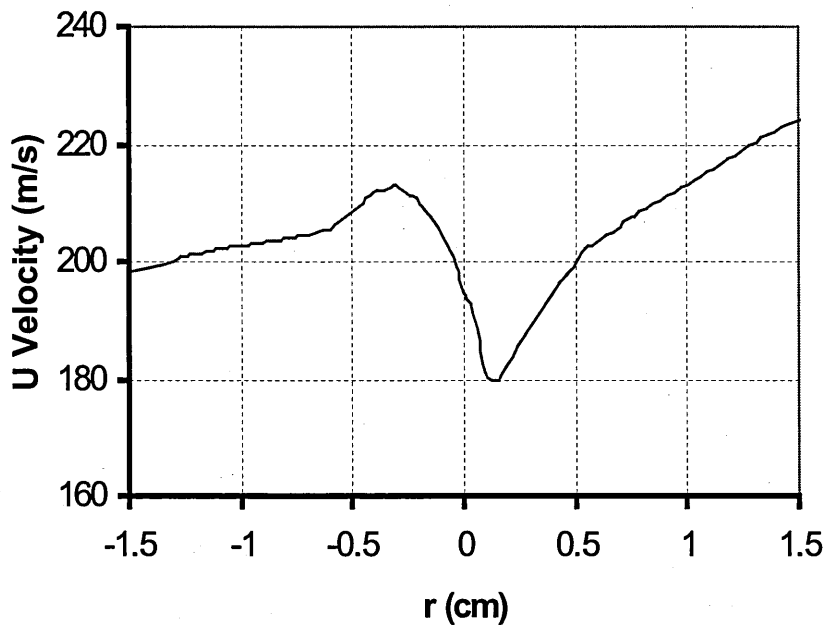


(c). Measured vortex radial pressure

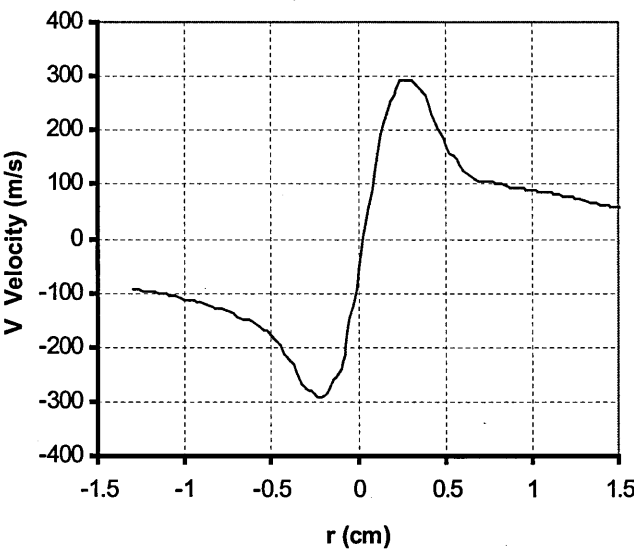


(d). Measured vortex radial density

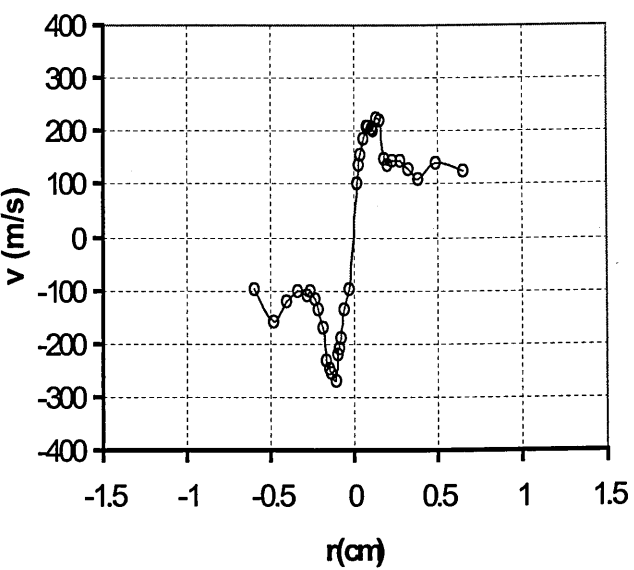
Figure 5-16. Flow variable profiles across vortex



(e). Simulated vortex radial u velocity

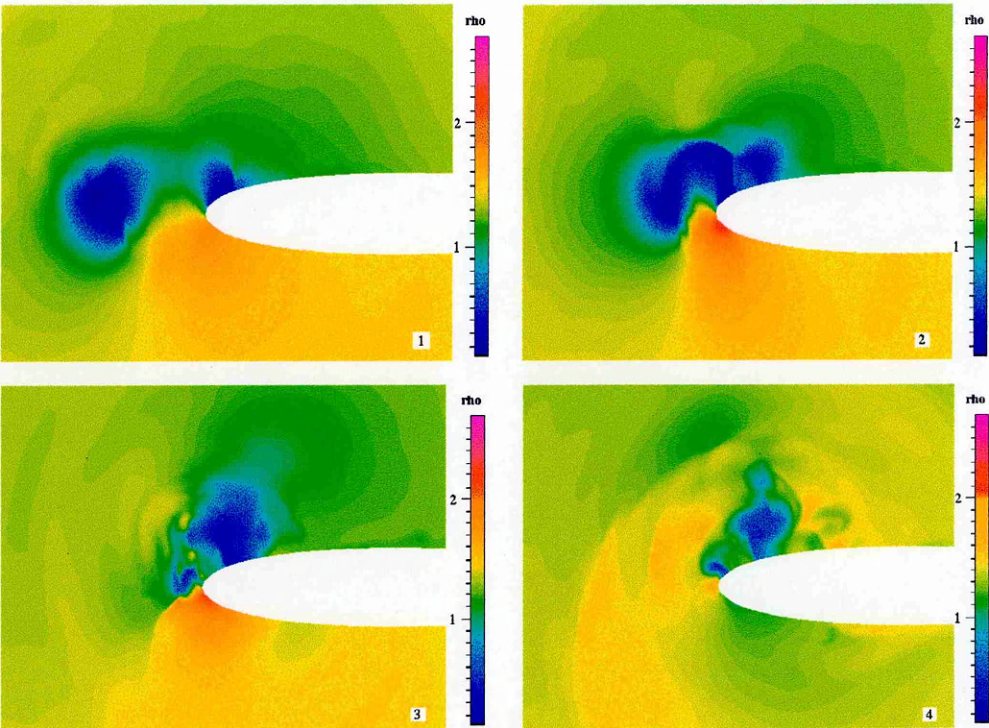


(f). Simulated vortex radial v velocity

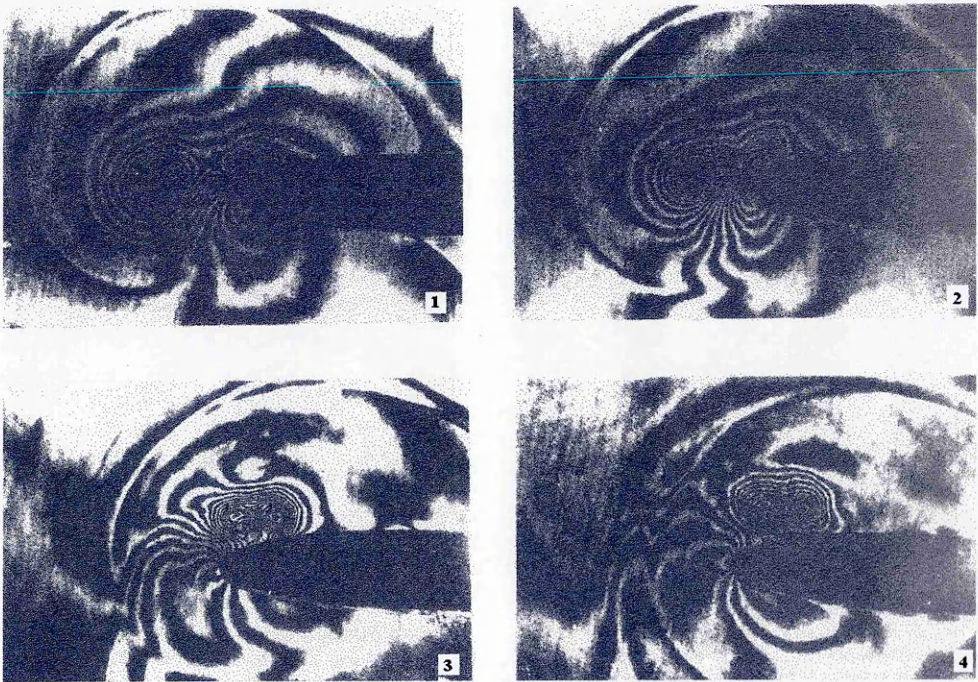


(g). Experimental vortex radial v velocity

Figure 5-16. Flow variable profiles across vortex (continue)



(a). Simulated results



(b). Experimental interferogram

Figure 5-17. Vortex interaction with the target aerofoil nose

Chapter 6

Laminar Viscous Flow Computations

6.1 Introduction

The development of the space shuttle and other high-speed vehicles has drawn attention on the problems of predicting the characteristics of regions of shock wave/boundary layer interaction in supersonic and hypersonic flow. Bow shock/boundary layer interaction, shock/shock interaction and flap and fin interaction are problems of considerable importance and occur under a variety of circumstances. A great deal of research effort, both theoretical and experimental, has been and continues to be directed towards a better understanding of the physical mechanisms involved and the improvement of prediction methods in the field of shock wave boundary layer interactions^{147, 148}.

Numerically predicting the characteristics of such flows is extremely important. Both the Chimera grid scheme and the structured AMR algorithm are very suitable for simulating such viscous flows. In the Chimera grid method, the flow domain can be divided into a number of geometrically simple sub-domains. This method has the advantage of using finer domains where interesting flow features need to be resolved. Moreover, in different domains, different flow physics can be applied if necessary. Since viscous effects are important near the solid boundaries, the Navier-Stokes equations might be solved only on small near body regions. Viscous effects may not be important away from these regions, so the Euler equations might be solved on the major region. This capability of the Chimera grid scheme can greatly reduce computational expense and provide good spatial resolution for thin boundary and shear layers found in viscous flow fields around complex

configurations.

In this study, I modified and corrected the viscous calculation of Quirk's original AMR code. This Chapter is specially designed to evaluate the performance of the proposed AMRC scheme in simulating laminar viscous flow problems. Two different laminar viscous flow problems are addressed. The first one is a shock wave/laminar boundary layer interaction. The second one is shock wave interference heating on a cylindrical leading edge. These studies show the flexibility and accuracy of the AMRC scheme to simulate laminar viscous flows at supersonic and hypersonic speed.

6.2 Shock Wave/Laminar Boundary Layer Interaction

The particular case of an interaction between a two-dimensional laminar boundary layer and a shock wave has been studied extensively for flows with and without heat transfer. The majority of these studies have concerned interactions on flat plates, by oblique shock waves impinging on flat plates or by forward or rearward facing steps on flat plates. A more practical situation of the interaction occurring in a region of streamwise surface curvature was experimentally studied by Dolling¹⁴⁷. In this section, I shall carry out numerical calculations by using the AMRC code viscous simulation, and present the comparison between the simulation results with the experimental results and the simulation results obtained by Dolling¹⁴⁷ for the same problem.

6.2.1 Specification of the Problem

Figure 6-1 shows the flat plate-convex ramp experimental model which Dolling¹⁴⁷ has studied in detail. His model was made in two parts (a forward flat plate section and an aft convex section) together with two small wedges of identical span which could be used to set the ramp at an angle of 7.5° and 10° , see Figure 6-1. To avoid any leakage of air from the separated region through any very small gaps at the hinge line junction, the latter was

smeared with a rubber cement such that when the two halves of the model were brought together a seal was formed.

The forward flat plate section was made of steel and of span 200mm. Its leading edge was ground at a 13° bevel angle. 80mm aft of the leading edge, was the hinge line of the configuration. The convex aft section was of circular arc construction, radius 600 mm, span 200 mm and chord of approximately 90mm.

Actually, the major aim of this experimental study was to examine whether or not the 'equivalence concept'* could also be used for configurations with streamwise curvature downstream of shock impingement. Surface static pressure distributions were measured throughout adiabatic laminar interactions on two different model configurations, namely,

- a). A flat plate-convex ramp model, see Figure 6-2a.
- b). A flat plate-convex afterbody model (with the convex section tangential to the flat plate at the hinge line) with an impinging oblique shock, see Figure 6-2b.

A series of tests (at the same Mach number but different Reynolds numbers) were carried out for the same model at ramp angles of 7.5° and 10° . These angles were chosen for two main reasons. Firstly, to obtain a relatively large separated length but at the same time to keep the overall pressure rise sufficiently low to ensure laminarity of the boundary layer throughout the interaction. Secondly, results could be compared directly with flat plate/flat ramp studies conducted in the same tunnel under similar test conditions (the experimental data are also available from reference¹⁴⁷).

The free stream Mach number was nominally 2.21. The flow total temperature was always approximately equal to the laboratory room temperature (303 °K). The free stream

* Lees and Reeves¹⁴⁹ postulated that the viscous flow over a flat plate-flat ramp configuration is equivalent to the viscous flow over a flat plate interacted with an oblique impinging shock. For the same free stream conditions and the same Reynolds number based on the hinge line position of the ramp model and based on the shock impingement position, the static pressure ratio across the shock system(s) is the same for both cases.

temperature can be calculated as $T_{\infty}=153.28^{\circ}\text{K}$. The wind tunnel test conditions and possible errors in the measurements are shown in Table 6-1 and Table 6-2.

Table 6-1. Wind tunnel test conditions

Parameter	Value
Mach number M_{∞}	2.21
Total Temperature T_o	303 °K
Incidence α	0°
Ramp angles θ	7.5° and 10°

Table 6-2. Error bounds in the measurements

Parameter	Error
Mach number M_{∞}	± 0.03
Temperature T_o	$\pm 5^{\circ}\text{K}$
Incidence α	$\pm 0.25^{\circ}$
Ramp angles θ	$\pm 0.2^{\circ}$
Pressure P_{∞}	$\pm 0.27 \times 10^3 \text{ N/m}^2$
Reynolds number Re_{∞}	$\pm 0.1 \times 10^6 \text{ m}^{-1}$

For the same free stream Mach number and temperature, different total pressure, P_o , naturally results in different free stream Reynolds numbers. A series of experimental measurements were carried out over a range of total pressures from 80 mm Hg ($1.07 \times 10^4 \text{ N/m}^2$) to 180 mm Hg ($2.40 \times 10^4 \text{ N/m}^2$) corresponding to free stream Reynolds numbers of approximately $1.2 \times 10^6 \text{ m}^{-1}$ to $2.8 \times 10^6 \text{ m}^{-1}$ respectively.

6.2.2 Computational Details

Chapter 6 Supersonic Laminar Flow Computations

In the current numerical calculations, 6 free stream Reynolds numbers (three for ramp angles of 7.5° and three for ramp angles of 10°) were considered for comparison with the experimental data. One numerical simulation for a 10° flat plate-flat ramp model is also carried out for comparison with the results for the flat plate-convex ramp model under the same free stream conditions.

According to the known free stream parameters, Mach number M_∞ , temperature T_∞ and unit Reynolds number Re_∞ , the corresponding free stream pressure P_∞ can be calculated. The free stream conditions (and normalised input parameters for the numerical simulation) for the 6 chosen cases are given in Table 6-3 and Table 6-4.

Table 6-3. Free stream conditions for ramp angle of 7.5° case

Case No.	$Re_\infty (m^{-1})$	$P_\infty (N/m^2)$	M_∞	$T_\infty (^\circ K)$	$\bar{p}_\infty$	$\bar{\rho}_\infty$
1	1.15×10^6	0.96×10^4	2.21	153.28	0.146	1
2	1.73×10^6	1.45×10^4	2.21	153.28	0.146	1
3	2.31×10^6	1.94×10^4	2.21	153.28	0.146	1

Table 6-4. Free stream conditions for ramp angle of 10° case

Case No.	$Re_\infty (m^{-1})$	$P_\infty (N/m^2)$	M_∞	$T_\infty (^\circ K)$	$\bar{p}_\infty$	$\bar{\rho}_\infty$
4	1.2×10^6	1.01×10^4	2.21	153.28	0.146	1
5	1.52×10^6	1.27×10^4	2.21	153.28	0.146	1
6	2.16×10^6	1.81×10^4	2.21	153.28	0.146	1

Since viscous effects are only important near the flat plate and convex ramp solid surfaces, the Navier-Stokes equations are solved only in these minor regions. In the simulation, two sub-domains are used to map the physical domain, see Figure 6-3. Inside the near body surface domain D_1 , the flow model is set as viscous flow. The physical property of dynamic viscosity varies with temperature according to Sutherland's Law, refers to equation (4-9). Far away from the solid boundary, the viscous effects are very

small and can be neglected. So inside domain D_2 , the flow model is set as inviscid flow. This arrangement can reduce the computational cost by only carrying out viscous calculation in a minor near body region.

As mentioned in §2.3.4, in normalised simulation, by setting the reference length which is used in the definition of Reynolds number, $L=1\text{m}$, the normalised geometric lengths are physical lengths and the free stream Reynolds number is the unit Reynolds number. Therefore, during generation of the grid system, the dimensions are chosen to be exact physical lengths.

In the Chimera grid system, the domain D_1 is an approximately uniform grid of 180×50 cells. The cell size is between 1×0.22 to 1×0.24 mm at level 0. For a non-adaptive calculation, there are less than 10 coarse cells (at level 0) which fall in the boundary layer. This is certainly not sufficient. However, in this calculation, the mesh cells will be refined by two adaptive levels according to the flow solution. It is expected there will be sufficient fine cells to resolve the boundary layer. As we can see later all the salient features for this shock/boundary layer problem are well resolved.

Domain D_2 is also an approximately uniform grid of 180×40 cells at level 0. It is an inviscid grid. The cell size is about 1×1.3 mm at level 0.

The refinement monitor functions chosen for this simulation are density gradient and shear stress (only within domain D_1). The grid adaptation parameters for this simulation are given in table 6-5. The refinement factors rI , rJ are specified as 2, and the maximum refinement level $L_{\max}(Nd)$ is specified as 2. Thus at level 2, more than 30 cells fall inside the laminar boundary layer.

Table 6-5. Grid adaptation parameters

l	rI	rJ	F_{tol}	P_{tol}
0	1	1	0.05	0.6
1	2	2	0.05	0.6
2	2	2	N/A	N/A

This is a steady state laminar viscous flow problem. The initial conditions are set to free stream conditions and the boundary conditions are shown in Figure 6-3. A CFL number of 0.8 is specified for this simulation. The solver used in this calculation is also the second order Roe's FDS scheme combined with the SUPERBEE flux limiter.

It should be pointed out that the current form of the AMR algorithm was specially designed for simulating transient flows. For a steady flow problem, the flow solution is simply marched to a converged steady solution in a time accurate manner. Consequently, the rate of convergence is very slow. In order to save computational cost, the first 8000 iterations are set as a non-adapting calculation to get a coarse solution, then setting adaption on for both domains and run for another 2000 iterations to get the final high resolution solution. In this way, the present calculation only takes about one day for one case to converge.

6.2.3 Comparison with Experimental Results

Figure 6-4 shows the computational flow structure (density plots). From this Figure, we can clearly see that the leading edge shock, boundary layer, compression wave due to separation and compression wave due to reattachment are all well resolved.

Figure 6-5 presents an enlarged U velocity plot inside the boundary layer at a position upstream $x=75$ mm from the flat plate leading edge. At this point, there are about 22 fine cells (less than 6 coarsest cells) fall in the boundary layer. Since in this region the finest cell size, $\Delta y=0.06$ mm (uniform grid), the resolved boundary layer thickness is,

$$\delta_{99} = 0.06 \times 22 = 1.32 \text{ (mm)}$$

Unfortunately, the boundary layer thickness was not available from the experiment. Although this simulation case is a compressible flow, the Mach number is not very high ($M_\infty=2.21$), the computational boundary layer thickness might be (at least qualitatively) compared with incompressible laminar boundary layer thickness according to the Blasius

solution. For an incompressible laminar flow over a flat plate, the local boundary layer thickness can be calculated,

$$\delta = \frac{5.2x}{\sqrt{\text{Re}_x}} = \frac{5.2 \times 0.075}{\sqrt{0.075 \times 1.15 \times 10^6}} = 1.328 \times 10^{-3} (m) = 1.328 (mm)$$

This at least shows that the simulation results for the boundary layer thickness are reasonable.

Figure 6-6 presents the details of the u velocity vector inside the boundary layer. It can be seen that laminar boundary layer separation occurs in upstream of the hinge line (Figure 6-6a). At the corner, a boundary layer separation bubble appears. The boundary layer reattaches to wall surface at downstream of the corner (Figure 6-6c).

Figure 6-7 shows the resolution of the numerical solution. For the non-adapting computation, after 10000 iterations, the solution is converged, but the resolution is poor, see Figure 6-7a. Figure 6-7b shows the results after another 2000 iterations, but with adaption, i.e. applying the AMR algorithm in both domains D_1 and D_2 . Comparing Figures 6-7a and 6-7b, the high resolution achieved by applying the AMR algorithm is clearly illustrated. Since the AMR algorithm has been applied to both domains, mesh cells in the leading edge region (see Figure 6-8a) and in the interaction region (see Figure 6-8b) are refined due to the high density gradient and high shear stress (only in the near body domain D_1). In particular, far away from the body surface, the compression waves generated in the separation and reattachment regions coalesce to form a shock. This shock is also well resolved by applying the AMR algorithm, see Figures 6-7b and 6-8c.

Figure 6-9 shows the computational pressure distributions for the 7.5° convex ramp model. The vertical axis for this Figure (and for all Figures showing pressure distributions presented later) is the local wall pressure P_w normalised by P_{ow} , the minimum value on the flat plate (for comparison with experimental results). At the leading edge a high peak pressure rise occurs and results in a leading edge shock (referring back to Figure 6-4a). At the ramp corner, $x=80$ mm, a similar phenomenon of an oblique incident shock interaction

with laminar boundary layer appears. Actually, to examine the similarity of the pressure distributions for two different flows, as shown in Figure 6-2, was the aim of the experiment.

The purpose here is to examine the accuracy of the AMRC scheme for simulating laminar viscous flow, to show the flexibility of domain arrangement and assess the computational efficiency of this scheme. In this study, I simulate the flat plate-convex ramp model and compare the results with the experimental results. Figure 6-10 shows the comparisons of computed pressure distributions and those of measured (and those predicted for inviscid flow theory presented in reference¹⁴⁷) on the 7.5° convex ramp models under different free stream conditions.

For the 7.5° ramp model at the two lower Reynolds numbers (Figure 6-10a and Figure 6-10b), the computational results agree well with the experimental results throughout the interaction. The upstream influence and the 'plateau' pressures are well predicted. Between the hinge line and the start of the interaction, the computational pressures are slightly lower than those measured. Downstream of the corner, the agreement in the reattachment and pressure peak region is excellent. At the higher Reynolds number (Figure 6-10c), the comparison with the experimental results is less satisfactory than for the lower Reynolds numbers. The computed pressures are slightly lower than those measured throughout the interaction, but have the identical trends with the experimental results.

Similar comparisons for the 10° convex ramp model are shown in Figure 6-11. Generally speaking, adequate agreements have been achieved. For the 10° convex ramp model at the higher Reynolds numbers, the pressure distribution ahead of the corner appears a trend to form a plateau region.

The slight difference between the computational results and the experimental results is likely due to the fact that a slight difference exists between the experimental and computed configurations. Since the convex aft section is of circular arc with a large radius of 600 mm, it not easy to measure the ramp angle. As shown in Table 6-2, an error band ($\pm 0.2^\circ$) exists in the ramp angle. Also there is a large error band (± 20 mm) exists in the

convex radius. As we can see later, the convex configuration has a significant influence on the pressure distributions.

For both ramp angles the same basic trends of the pressure distribution with Reynolds number are observed from both simulation and experiment. Figure 6-12 shows the influence of free stream Reynolds number on pressure distribution on the 10° convex ramp model. The vertical axis is the same as before, but the x-axis is normalised by the flat plate length ($x=80$ mm, from leading edge to hinge line). For the 10° convex model, the rise to peak pressure shows a strong trend with Reynolds number. The overall rise in pressure, $[(P_{pk}/P_{ow})-1]$, increases with Reynolds number. The 'free interaction' length upstream of the hinge line is about $x/L < 0.6$. The computational results clearly show that at higher Reynolds numbers, both the upstream influence region and downstream reattachment region move closer to the corner.

6.2.4 Comparison with Flat Ramp Model

Although the convex ramp radius used in this study is quite large (600 mm), it has significant influence on the important features of the interaction. In comparison with the flat compression ramp of the same ramp angle of 10° and hinge line position under the identical free stream and wall conditions, see Figure 6-13, the convex curvature downstream of the hinge line considerably weakens the interaction. The start point of the interaction upstream influence region moves from $x \approx 40$ mm (see Figure 6-13a) to $x \approx 50$ mm (see Figure 6-13b). That is, the upstream influence length (from the hinge line to the start point of the interaction) is diminished by about 25%. The downstream reattachment point also moves close to the hinge line (interaction point) largely.

Besides the length scales, pressure levels throughout the interaction are decreased as well, notably the peak pressure level attained downstream of reattachment. The pressure level at the corner is less and with the smaller separated length. Between the separation and the corner, both computational and measured pressures continue to rise steeply. There is not a distinct 'plateau pressure' region, as in the flat ramp case. Downstream of the corner, the

peak pressure is reached in a shorter streamwise distance and is significantly reduced. For this case, the overall rise in pressure is about 55~60% of that which occurs on the flat ramp under the same free stream conditions.

Since the reattachment pressure gradients are also reduced, for the convex ramp model the transition from laminar flow to turbulent flow would not occur until a higher Reynolds number compared with the flat ramp model.

Figures 6-14 and 6-15 presents the numerical solutions for both 10° flat ramp and convex ramp models. We can see that for the convex ramp model, the separation occurs closer to the corner and the down stream pressure level is considerably lower than for the flat ramp.

6.3 Shock Wave Interference on a Cylindrical Leading Edge

The understanding of the physics of complex flow phenomena in hypersonic flow is important. One example of these phenomena is shock-shock interaction. Shock wave impingement is a critical problem in the structural design of the thermal protection system and the load carrying structure of high-speed vehicles. Extremely high pressure and complex flows can occur in highly localised regions where the interference pattern impinges on the surface. Numerically solving this kind of flows requires good local resolution. The AMRC scheme is particularly suitable for the simulation of such a flow problem. In this section, I will present a computational example of shock wave interference on a cylindrical leading edge at a hypersonic speed where experimental data are available.

The physics of such flow phenomena, a weak shock generated by means of a wedge interacts with the bow shock in front of a cylinder, were experimentally investigated in a two-dimensional model flow by Edney¹⁵⁰. Based on these experiments, six different types of shock-shock interactions were classified, which depended on the position of the interaction location and the strength of the impinging shock wave, see Figure 6-16. Typically the Type III, Type IV and Type V patterns have higher surface pressure and heating associated with them than others. These patterns are more common and the flow

structures are more complicated. In this study, I shall simulate Type III and Type IV interference patterns for which the experimental data are available from reference¹⁵¹.

6.3.1 Simulation Problem

In the literature¹⁵¹, Wieting and Holden carried out a series of experimental studies of shock wave interference on a cylindrical leading edge at different hypersonic speeds (Mach numbers of 6.3, 6.5 and 8.0). The test model consisted of a 3-inch diameter cylinder and a shock generator wedge articulated to angles of 10, 12.5 and 15 degrees. Model position (the exact position dimensions are not mentioned) was varied to obtain Type III, Type IV and Type V interference patterns, as defined by Edney¹⁵⁰. The majority of the results are for the type IV interference pattern as it results in the most severe pressures and heat transfer rates. Among these experimental studies, for the test case at Mach number of 8 and wedge angle of 12.5 degrees, more detailed experimental results were presented. This experimental case was conducted in Calspan 48-inch hypersonic Shock Tunnel (48" HST). The test time was 15 milliseconds. In the simulation, this case was chosen to demonstrate the ability of AMRC scheme for simulating such kind of flow problem. The wind tunnel test conditions are given in Table 6-6.

Table 6-6. Wind tunnel test conditions

Parameter	Value
Mach Number M_∞	8.03
Reynolds Number Re	$1.55 \times 10^6 \text{ ft}^{-1} = 5.09 \times 10^6 \text{ m}^{-1}$
Total temperature T_0	$2800^\circ\text{R} = 1555.56^\circ\text{K}$

6.3.2 Computational Details

This study intends to simulate similar flow fields including the modelling of the incident shock wave generated by the wedge. Figure 6-17 shows the simulation problem

and the Chimera grid system and boundary conditions. The domain D_1 is composed of two meshes. Mesh D_{1-1} , a body fitted grid of 100×40 cells, is used to map the oblique shock generator wedge. Mesh D_{1-2} , a uniform Cartesian grid of 80×100 cells, is used to form a baseline grid which overlaps with domain D_2 . The bottom mesh edge of D_{1-2} is connected to the wedge end and is specified to extrapolated boundary condition. Thus the expansion wave effect resulted from the wedge end can be diminished. The near body domain D_2 , a very thin polar grid of 180×20 cells, is used to map the cylinder. In the numerical simulation, by changing the position of the domain D_2 Type III and Type IV interference patterns can be easily obtained.

For the same reasons as mentioned in §6.2.2, during generating the grid system, the dimensions are chosen as exact physical lengths, and in this simulation only inside the near body surface domain D_2 the flow model is specified as viscous flow and a viscous grid system is applied. In the circumfluent direction, 181 rays with a uniform angular spacing of one degree are used. In the radial direction, 20 grid cells are distributed, and is clustered towards the cylinder wall. The cell size near to the cylinder surface is 0.665×0.1 mm. Near to the overset boundary the cell size is 0.776×0.4 mm. This is certainly small enough to capture the boundary layer and salient flow features inside this domain. It is not necessary to apply the AMR algorithm in this small domain.

In this way, it is only inside domain D_1 necessary to apply the AMR algorithm to capture the bow wave, the oblique incident shock and the impingement point. This is another one of the flexibility of the AMRC scheme, i.e. in different domains the adaptive procedure can be adopted differently if necessary. Therefore, in this simulation the logical variables, *Euler(Nd)* and *Adapting(Nd)*, are specified as Table 6-7.

Table 6-7. Flow physics and adaptive calculation setting

	<i>Euler(Nd)</i>	<i>Adapting(Nd)</i>
Domain D_1	.T.	.T.
Domain D_2	.F.	.F.

Chapter 6 Supersonic Laminar Flow Computations

The refinement monitor functions chosen for this simulation are density gradient. Since this is a hypersonic flow problem, very strong discontinuities exist in the flow field. Therefore, the refinement factors, rI and rJ , are set to 8. The maximum refinement level $Lmax(Nd)$ is specified as 1 and the other grid adaptation parameters are shown in Table 6-8. Thus the meshes at level 1 has the same resolution as uniformly spaced meshes of 640×800 (for domain D_{1-2}).

Table 6-8. Grid adaptation parameters for domain D_1

l	rI	rJ	F_{tol}	P_{tol}
0	1	1	0.04	0.6
1	8	8	N/A	N/A

This is a steady flow problem. The simulation was started with an initial condition (set to be the same as the free stream condition). According to the known wind tunnel test conditions (see Table 6-6), the free stream conditions can be calculated out for the numerical simulation, see Table 6-9.

It should be pointed out that since the cylinder wall temperature was not mentioned in the reference¹⁵¹ and because the test time was only 15 milliseconds, the wall boundary condition is specified as no-slip adiabatic one.

Table 6-9. Free stream conditions

$Re_\infty (m^{-1})$	$P_\infty (N/m^2)$	M_∞	$T_\infty (^\circ K)$	$\bar{p}_\infty$	$\bar{\rho}_\infty$
5.09×10^6	0.752×10^3	8.03	112	0.0112	1

The experimental pressure distributions were presented in terms of P_w/P_{ow} vs angular position, θ , measured in degrees from the horizontal centreline of the cylinder. The P_{ow} indicates the pressure value at the undisturbed stagnation point in front of the cylinder. In order to compare with the experimental data and schlieren photograph, one polar grid of 90×60 cells is used, see Figure 6-18, to simulate the undisturbed flow field. The refinement parameters are the same as those shown in Table 6-8.

Since Roe's FDS scheme suffers from the so-called "carbuncle phenomenon"^{140, 141} near the symmetry line in this problem, especially for the undisturbed bow wave case, the WAF solver¹⁸ was chosen to carry out all simulations. The CFL number is set to 0.2 for the calculation stability.

6.3.3 Comparison with Experimental Results

Figure 6-19 presents results for the undisturbed flow field. The computed bow shock wave shape agrees well with the experimental schlieren photograph. Wieting and Holden¹⁵¹ state that the bow shock displacement was 0.58 inches away from the cylinder at the stagnation point. From the computational flow field, it can be measured out that the bow shock displacement is 0.6 inches.

Figure 6-20 presents the computational flow structure, Mach number plot and pressure plot of a Type IV interference pattern. The intersection point of the incident shock and bow shock is just below the cylinder centreline, i.e. the incident shock intersects closer to the normal shock portion of the bow shock. From the Mach number plot, we can see that just behind the impingement point of the incident shock with the bow shock a high Mach number region is resolved. A supersonic jet embeds within the subsonic regions between the bow shock wave and the surface, and is separated by shear layers from the subsonic flow. Comparing with interference patterns shown in Figure 6-16, this flow structure can be classified as a Type IV interference pattern.

The jet flow becomes subsonic through a strong normal shock just in front of the cylinder. A jet bow shock is produced when the jet impinges on the cylinder surface. The jet flow is split in two directions (upward and downward) and is accelerated to supersonic speed.

The Weak compression of the flow in the jet and then strong compression through the jet bow shock results in high-localised pressure and heating at the wall stagnation point. The maximum pressure occurs when the jet impinges perpendicular to the cylinder wall surface, see Figure 6-20b.

Figure 6-21 makes a comparison of the computational density plot with experimental schlieren photograph. Good agreement has been achieved. For this case, the experiment stated that the bow shock wave displacement due to the intersection was increased from 0.58 inches to 1.13 inches. The computational displacement is increased from 0.6 inches to 1.2 inches. Some of differences may be because that the exact position of the cylinder relative to the wedge was not mentioned in the literature¹⁵¹, in this computation, the Type IV interference pattern is from an approximately scaled position from the schlieren photograph.

Figure 6-22 makes a comparison of pressure distributions on the cylinder surface obtained from the simulation results and those presented in the literature¹⁵¹. For comparison, the pressure value is normalised by undisturbed stagnation pressure and is plotted as a function of the angular position (θ) expressed in degrees from the horizontal centreline of the cylinder. The peak pressure position agrees very well with the experimental result, $\theta = -19.1$ degrees. The pressure distribution trends have adequate agreement. However, the computational pressure peak is not as sharp as the experimental one. The measured peak pressure amplification P_w/P_o is 8.02 and the computational P_w/P_o is 6.74. In the region of $\theta < -25$ degrees the computational pressure is higher than the experimental data. This indicates that in the near body domain D_2 , a polar grid of 180×20 cells, the uniform angular spacing of one degree is not fine enough, and the AMR algorithm was not applied in this domain, see Table 6-7. Another reason is because that the specification of the bottom boundary condition (Extrapolation BC, see Figure 6-17) of mesh D_{1-2} does not properly represent the physical flow field.

Figure 6-23 presents the computational Type III interference pattern. The incident shock intersects the bow shock (the flow behind shock subsonic). Behind the intersection point, the flow in the region above the shear layer is subsonic (see Figure 6-23a point A) and the flow between the shear layer and the transmitted shock is supersonic (see Figure 6-23a point B). However, the pressure is identical (see Figure 6-23b point C). These two regions are separated by a shear layer, which attaches to the wall boundary layer causing a

shear-layer/boundary layer interaction (see Figure 6-23c point D). The supersonic flow is deflected downward through an oblique shock (see Figure 6-23c point E).

Comparing with the Type IV interference, the jet flow turns away from the wall. The peak pressure rise on the cylinder surface decreases largely. The typical peak pressure levels vanish with the development of a type IV interaction to a Type III interaction where the supersonic jet runs along the surface.

6.4 Summary

The results presented in this Chapter demonstrate the application of the AMRC scheme for simulating steady state supersonic and hypersonic laminar viscous flows. The calculations have demonstrated the flexibility of the domain arrangement (near body domain) and computational efficiency. The comparisons with the experimental results have shown the accuracy of the AMRC scheme for simulating laminar flows.

Note that the computed flow fields are smooth, even for the hypersonic flow case. This indicates that the flow field information is being passed correctly between the domains. Experience gained from these model problems indicates that the AMRC scheme can be used with confidence to simulate supersonic or hypersonic laminar viscous flow problems.

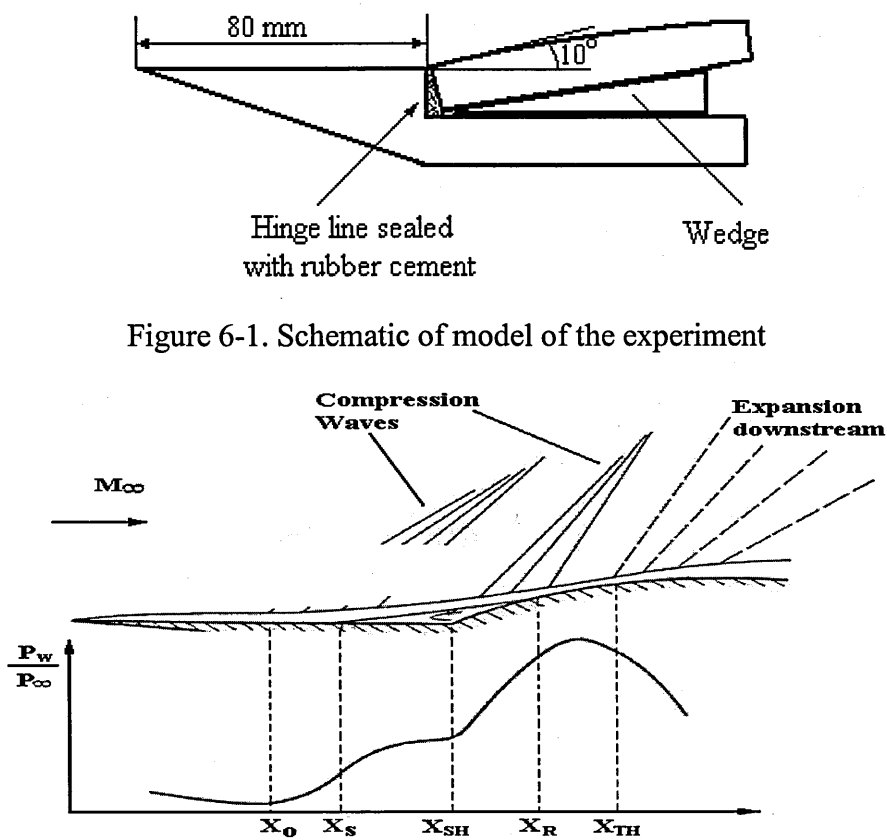
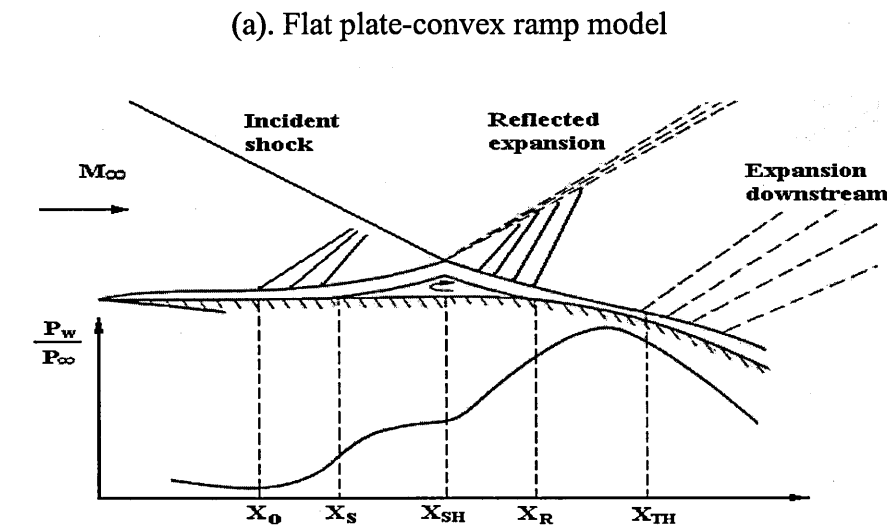


Figure 6-1. Schematic of model of the experiment



(b). Flat plate-convex afterbody model

Figure 6-2. Schematic of shock wave/laminar boundary layer interaction

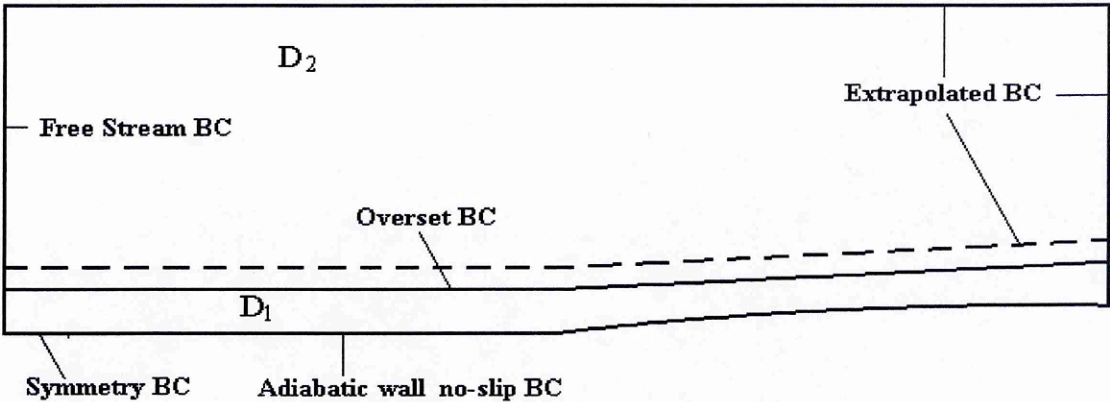
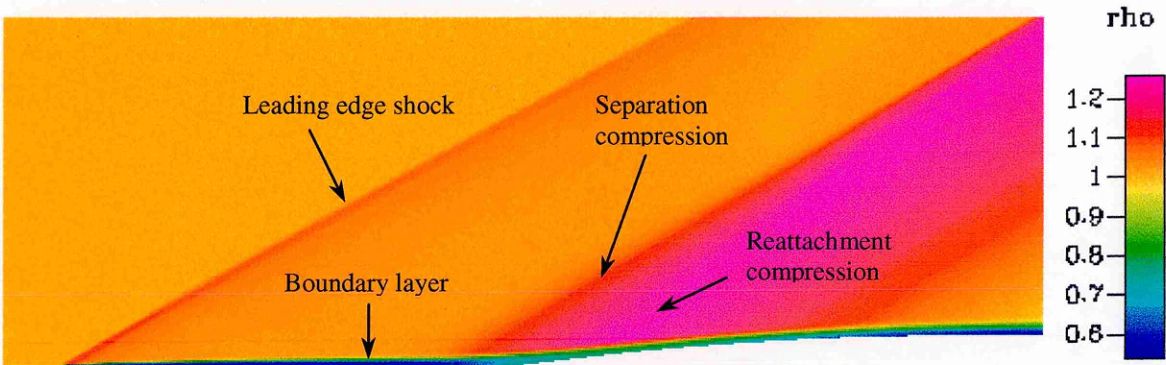
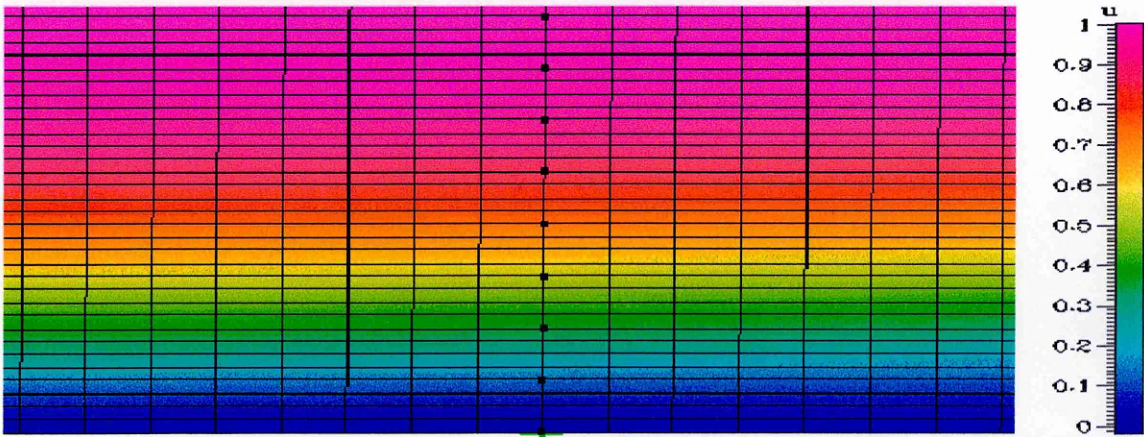


Figure 6-3. Chimera grid system and boundary conditions



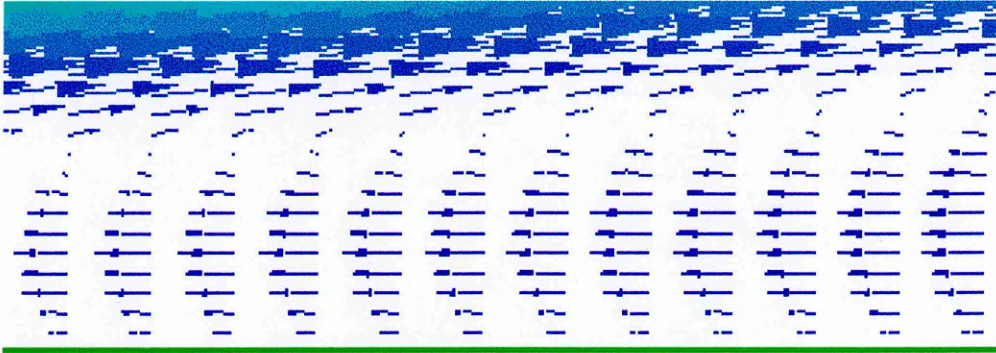
$M_\infty=2.21, Re=1.15 \times 10^6 \text{ m}^{-1}, \alpha=7.5^\circ$

Figure 6-4. Computational flow structure (density plot)

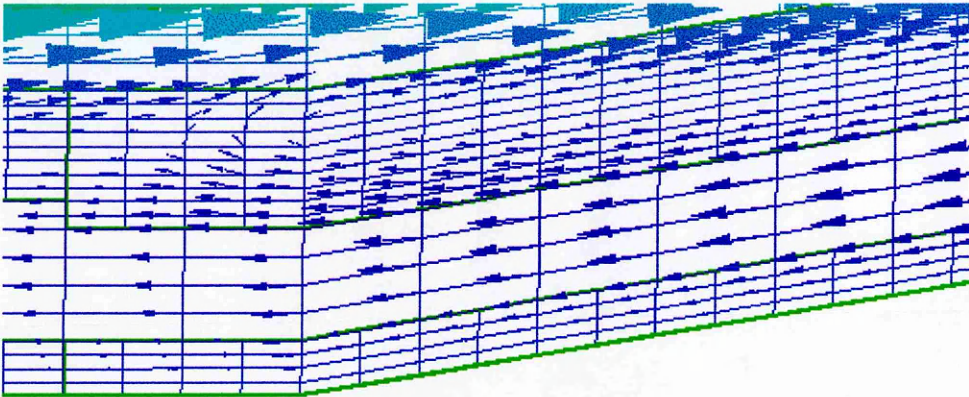


$M_\infty=2.21, Re=1.15 \times 10^6 \text{ m}^{-1},$

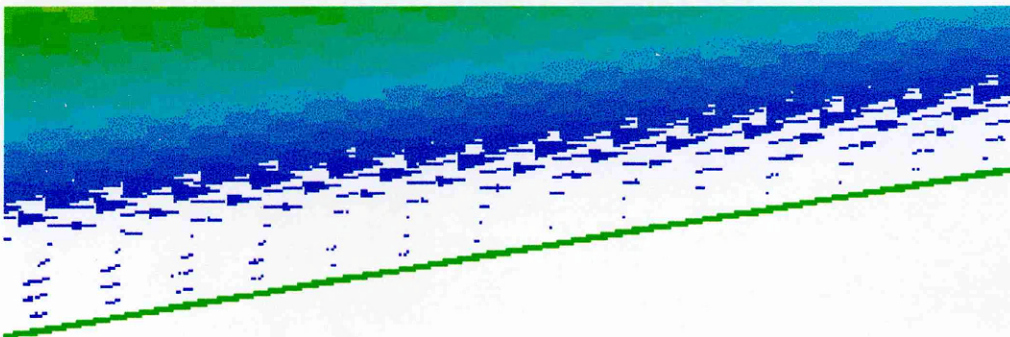
Figure 6-5. Enlarged U velocity plot at $x=75\text{mm}$



(a). In upstream separation region

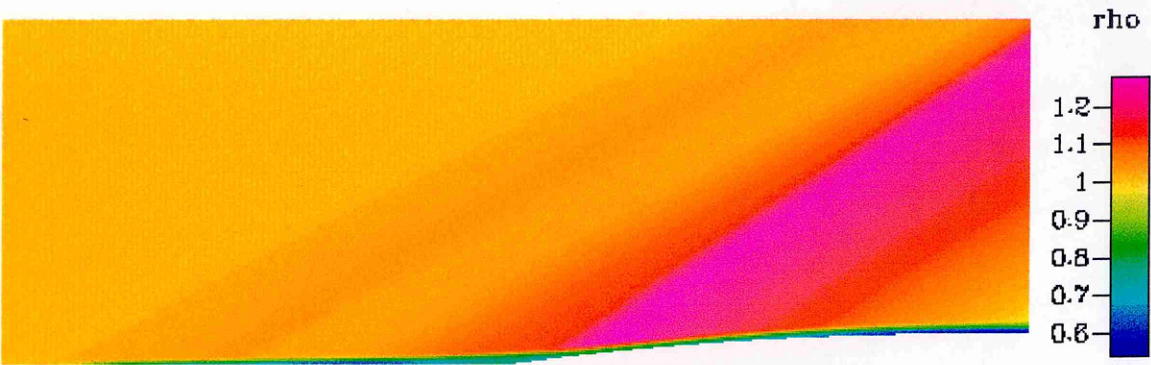


(b). At the corner

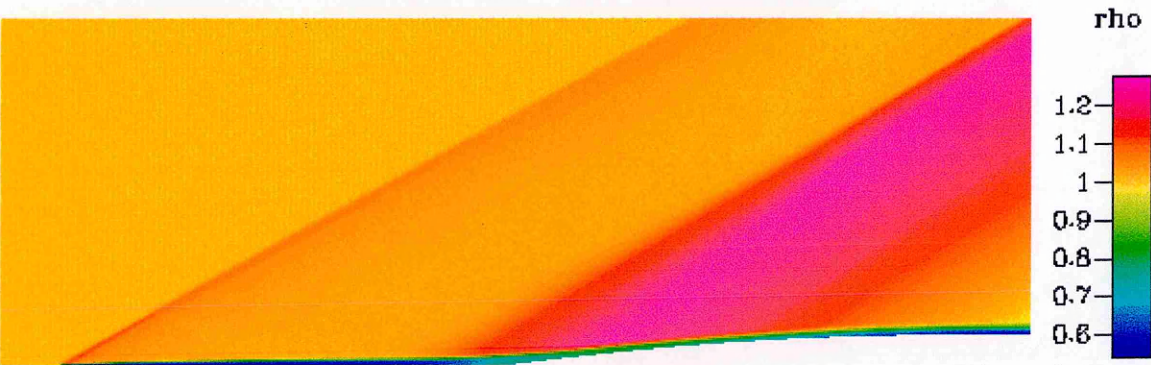


(c). In reattachment region

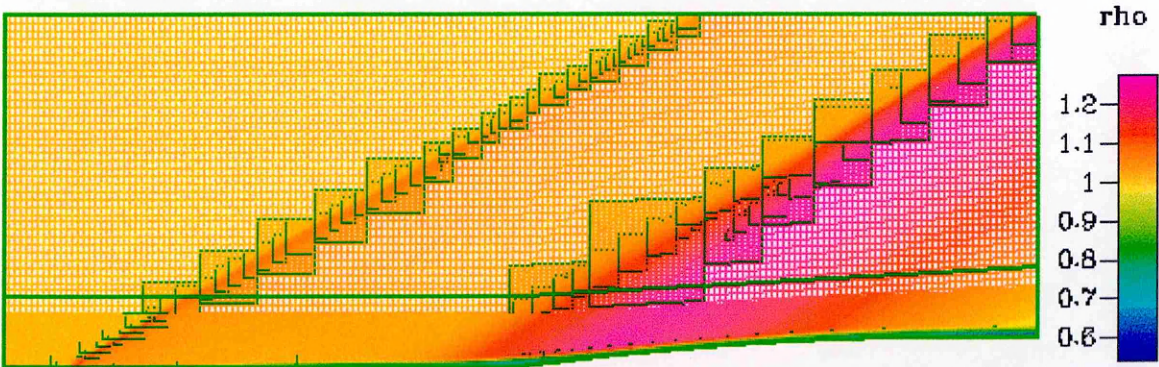
Figure 6-6. Velocity vector inside boundary layer



(a). Numerical solution on coarse grid



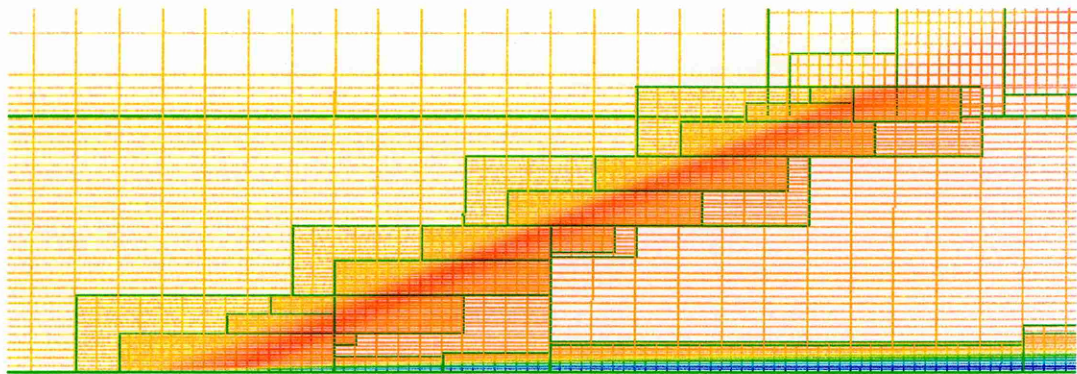
(b). Numerical solution on refined grid



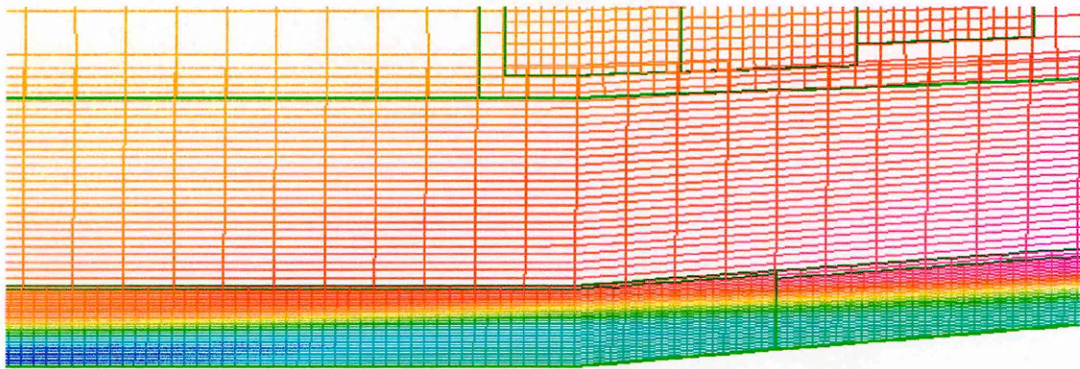
(c). Refined Chimera grid system

$M_{\infty}=2.21, Re=1.73\times10^6\text{ m}^{-1}, \text{convex model}, \alpha=7.5^{\circ}$

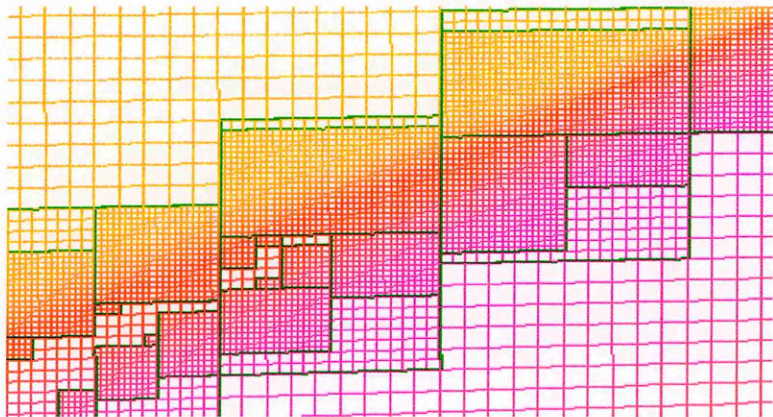
Figure 6-7. Comparison of numerical solution resolution (density plots)



(a). In leading edge region

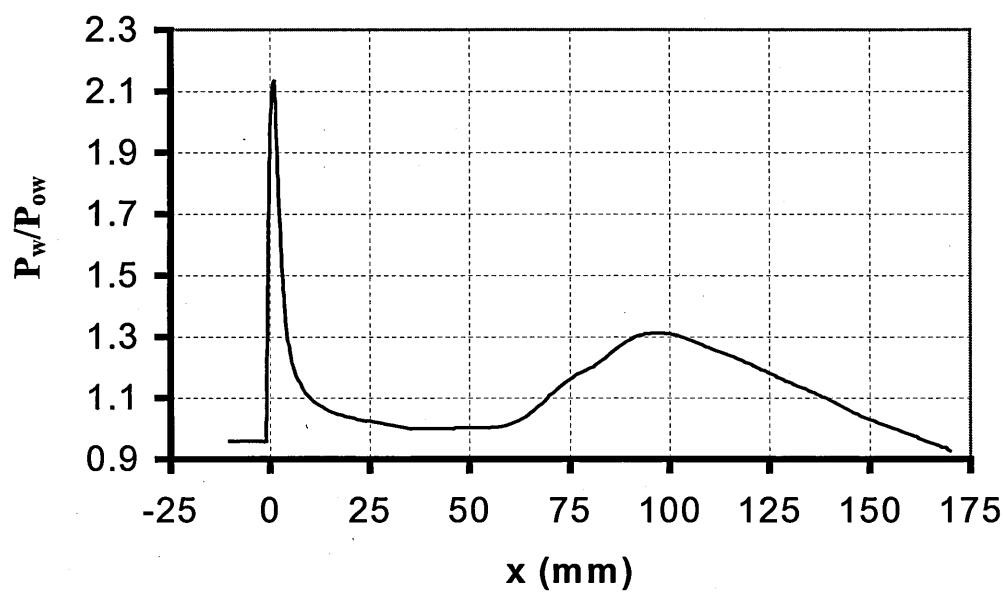


(b). At the corner



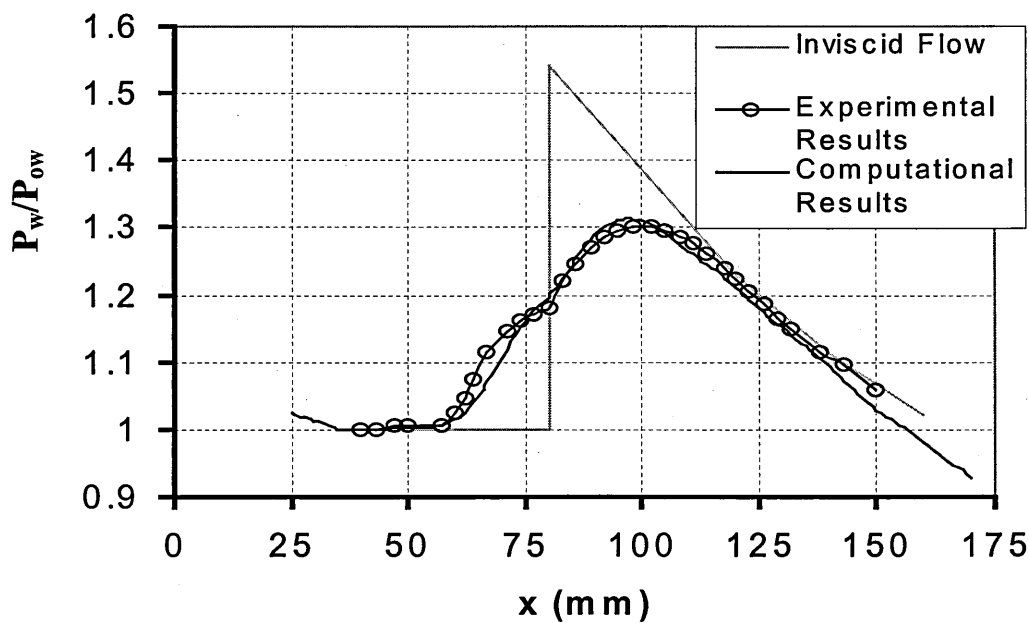
(c). In the compression wave coalescing region

Figure 6-8. Enlarged grid system



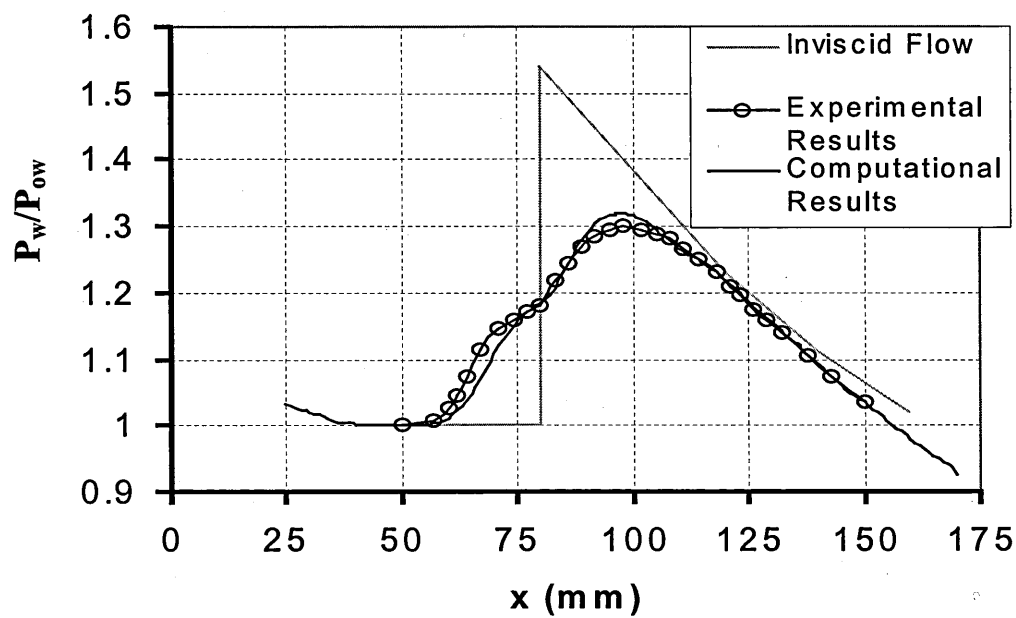
$M_\infty=2.21$, $Re=1.15\times10^6\text{ m}^{-1}$, convex model, $\alpha=7.5^\circ$

Figure 6-9. Computational pressure distributions

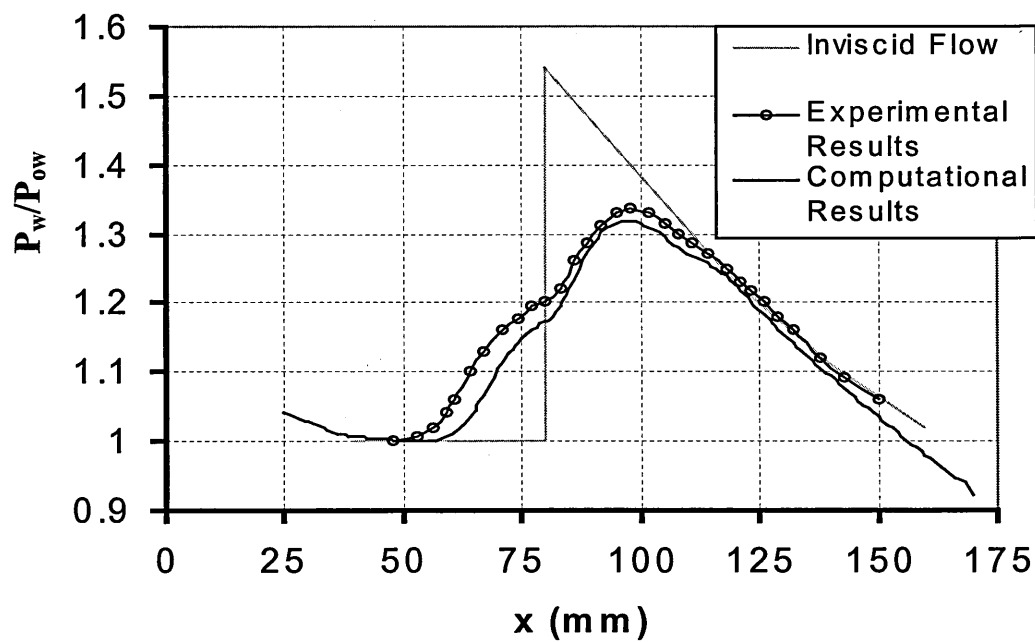


(a). $M_\infty=2.21$, $Re=1.15\times10^6\text{ m}^{-1}$, convex model, $\alpha=7.5^\circ$

Figure 6-10. Comparison of pressure distributions with Experimental results

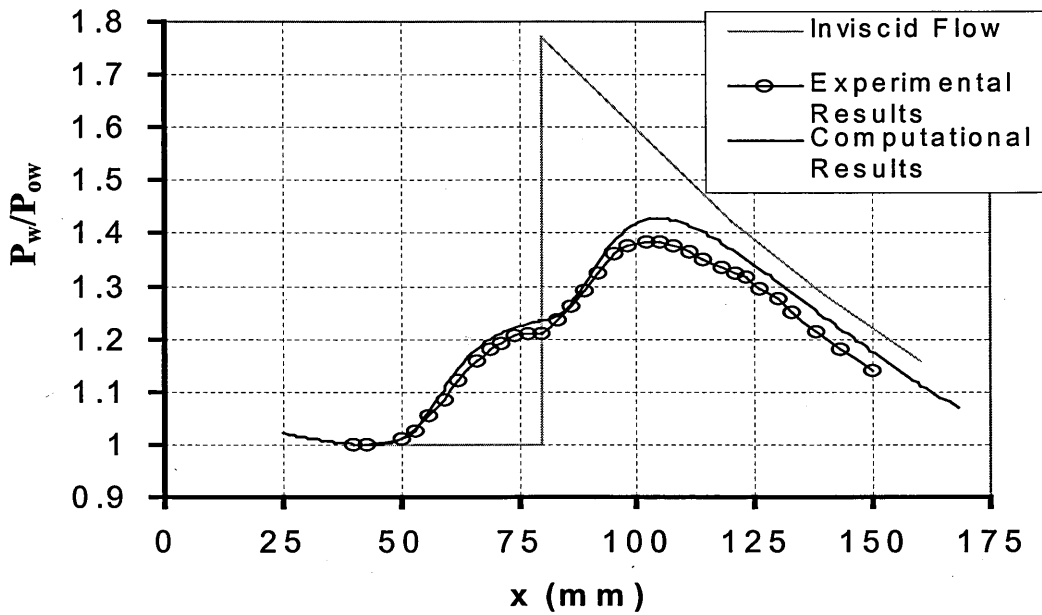


(b). $M_\infty=2.21$, $Re=1.73 \times 10^6 \text{ m}^{-1}$, convex model, $\alpha=7.5^\circ$

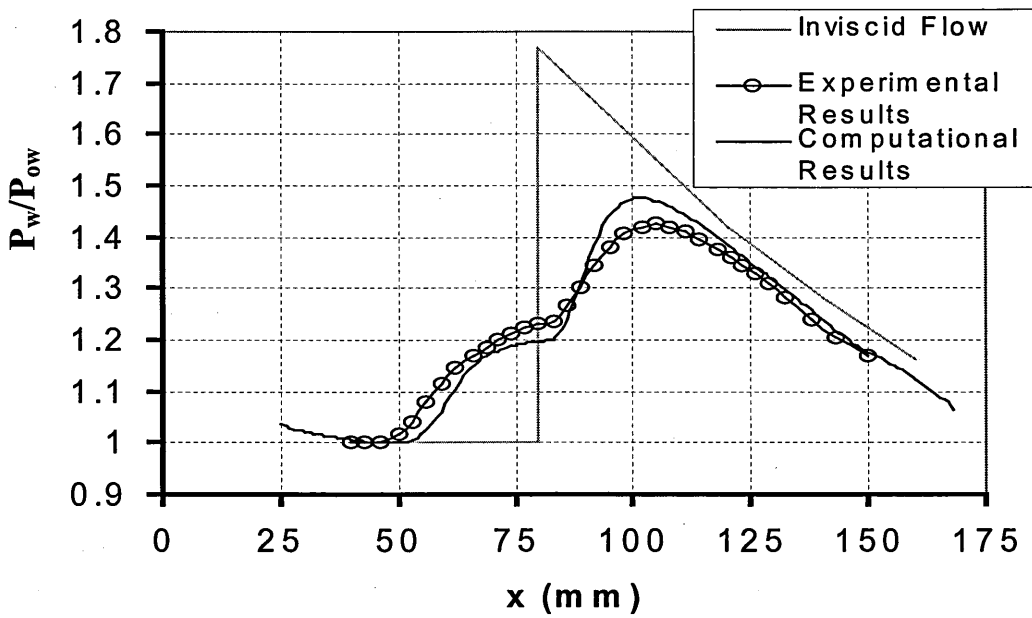


(c). $M_\infty=2.21$, $Re=2.31 \times 10^6 \text{ m}^{-1}$, convex model, $\alpha=7.5^\circ$

Figure 6-10. Comparison of pressure distributions with Experimental results (continue)

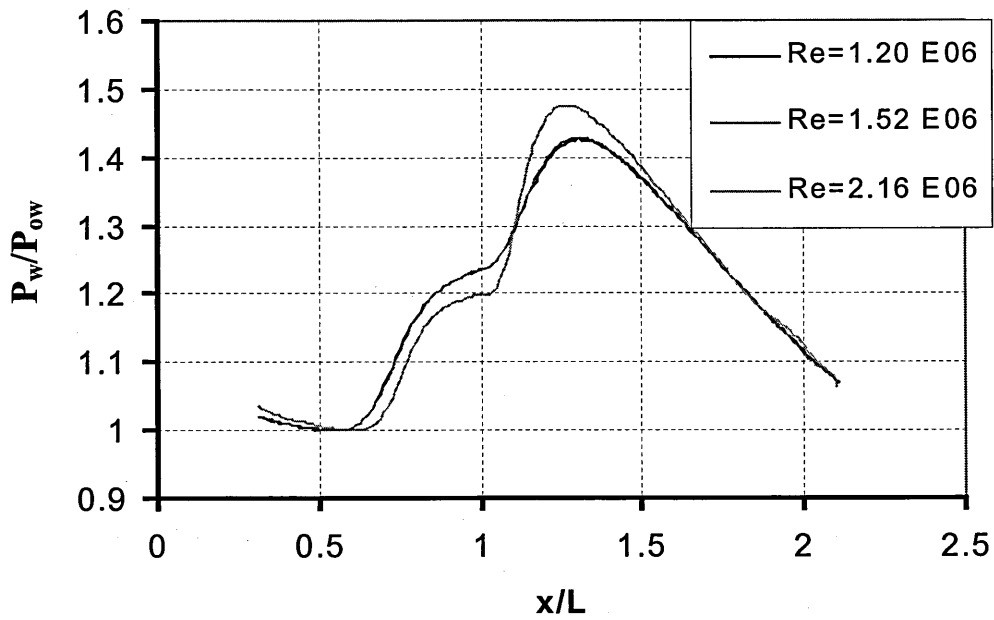


(a). $M_\infty=2.21$, $Re=1.52 \times 10^6 \text{ m}^{-1}$, convex model, $\alpha=10^\circ$

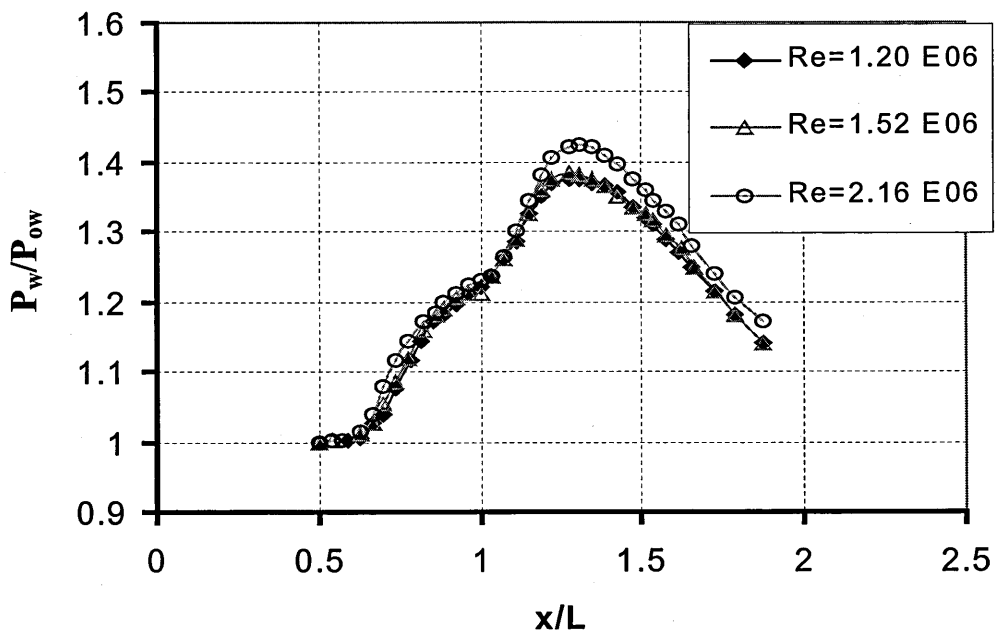


(b). $M_\infty=2.21$, $Re=2.16 \times 10^6 \text{ m}^{-1}$, convex model, $\alpha=10^\circ$

Figure 6-11. Comparison of pressure distributions with Experimental results



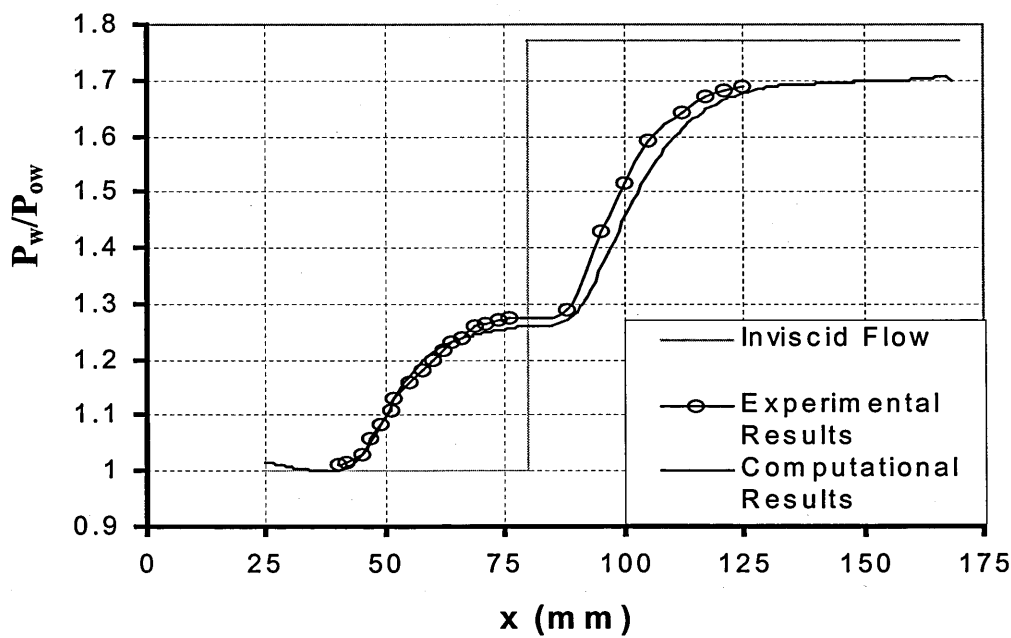
(a). Computational results



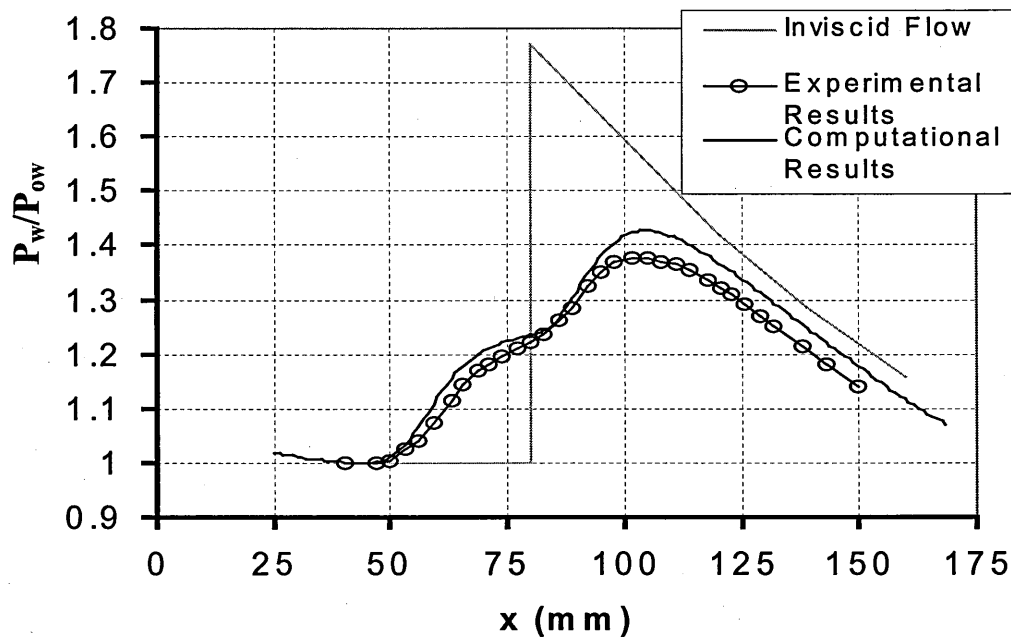
(b). Experimental results

$M_\infty=2.21$, Convex ramp model, $\alpha=10^\circ$

Figure 6-12. Effects of free stream Reynolds number on pressure distributions



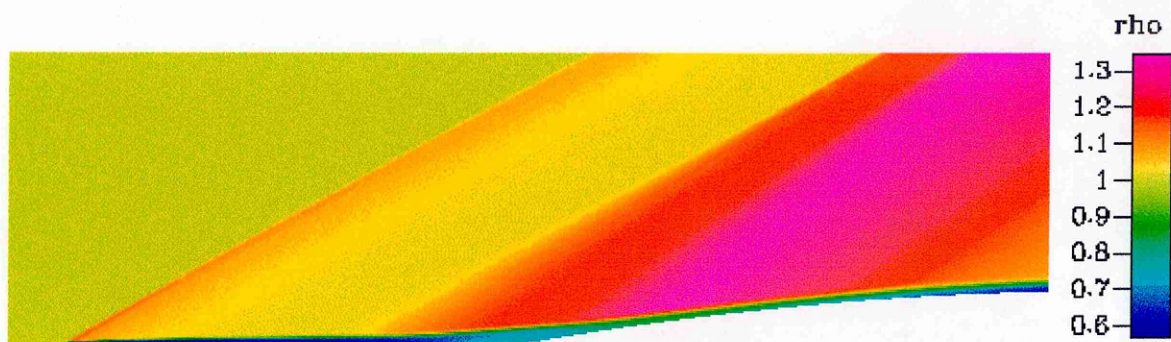
(a). Flat ramp model



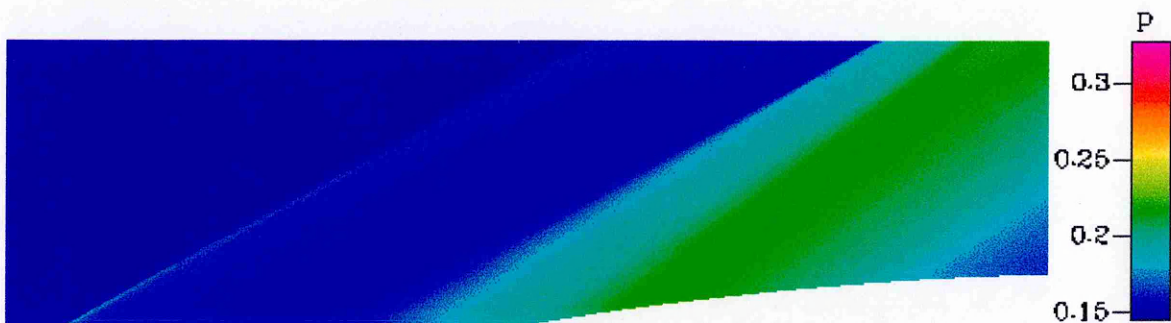
(b). Convex ramp model

$M_\infty=2.21, Re=1.20 \times 10^6 \text{ m}^{-1}, \alpha=10^\circ$

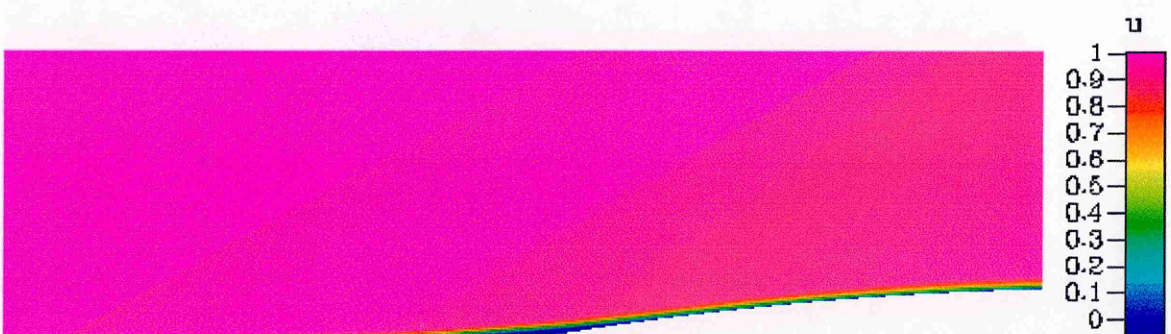
Figure 6-13. Comparison of pressure distributions with flat ramp model



(a). Density plot



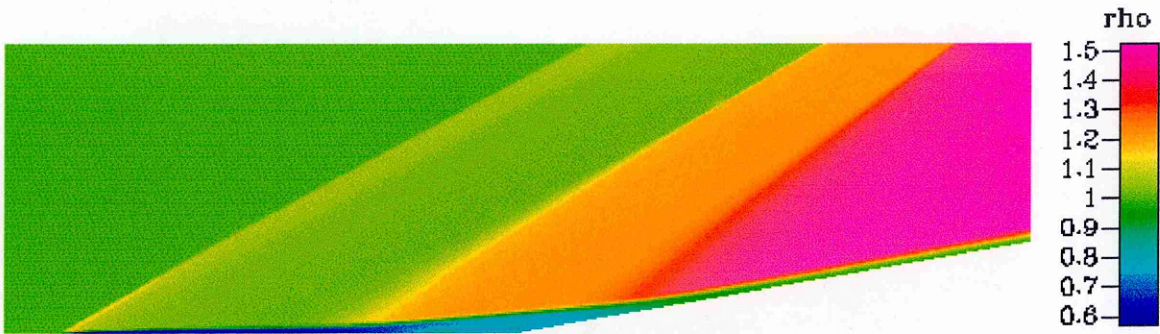
(b). Pressure plot



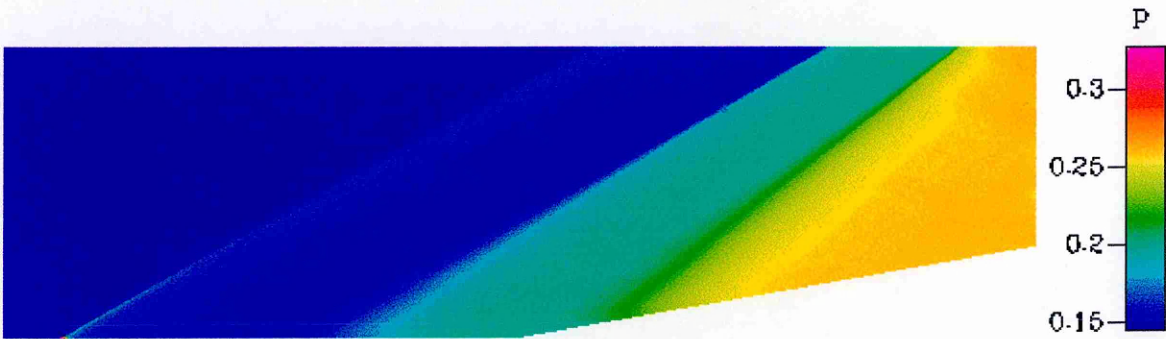
(c). U velocity plot

$M_\infty=2.21, Re=1.20\times10^6\text{ m}^{-1}, \alpha=10^\circ$

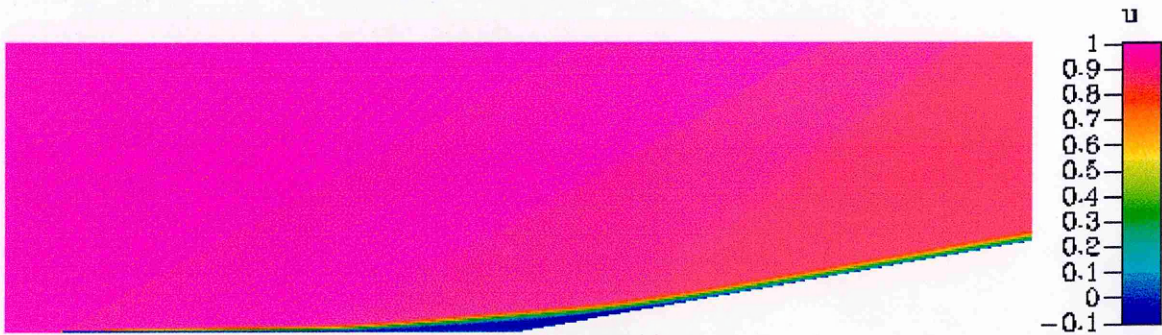
Figure 6-14. Simulation results of convex ramp model, (cf. Figure 6-15)



(a). Density plot



(b). Pressure plot



(c). U velocity plot

$M_\infty=2.21, Re=1.20\times10^6\text{ m}^{-1}, \alpha=10^\circ$

Figure 6-15. Simulation results of flat ramp model, (cf. Figure 6-14)

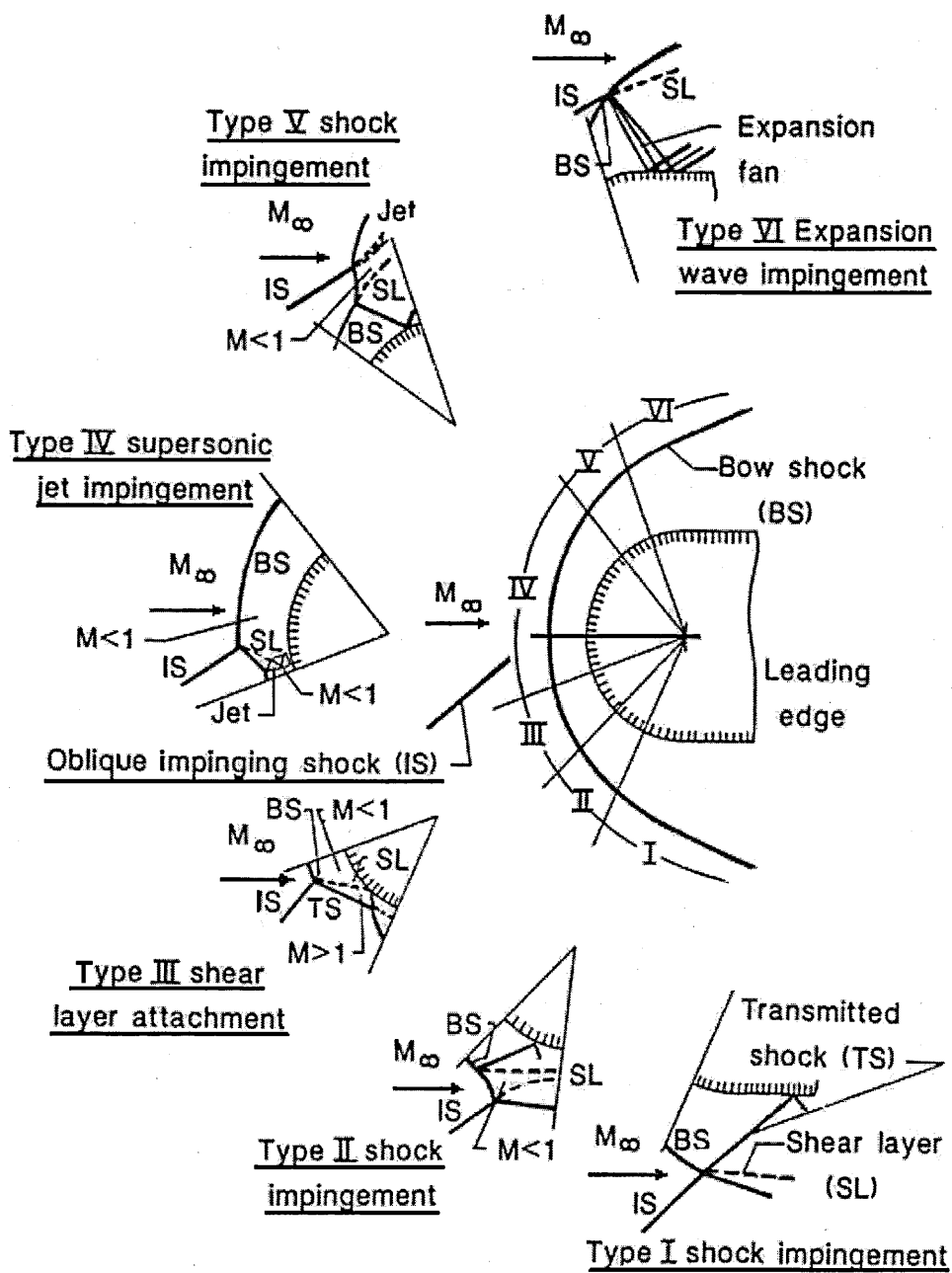


Figure 6-16. Six types of shock wave interference patterns [151]

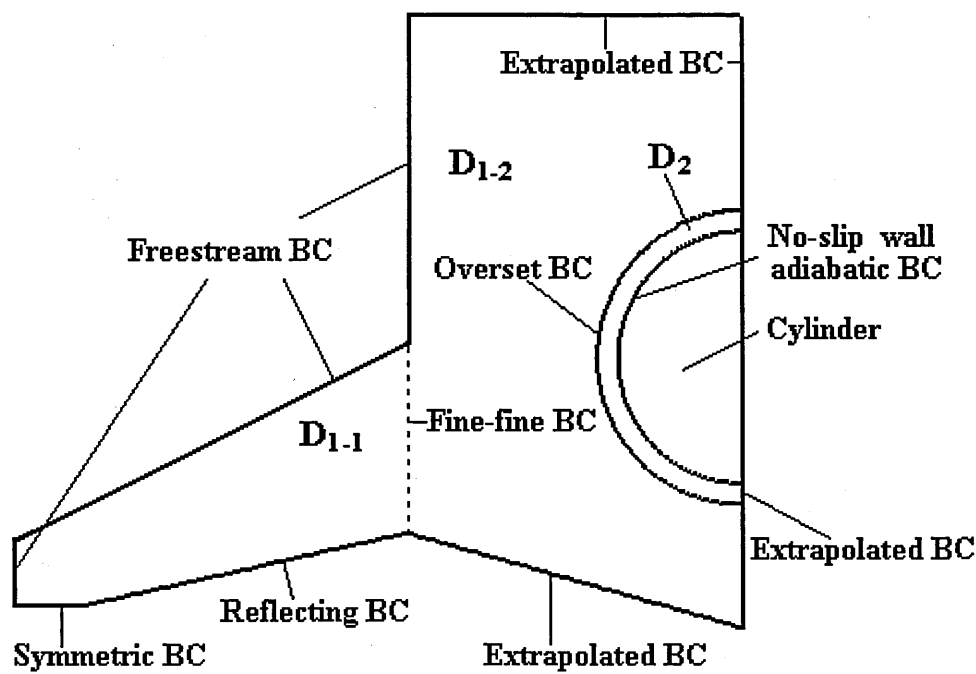


Figure 6-17. Chimera grid system and boundary conditions

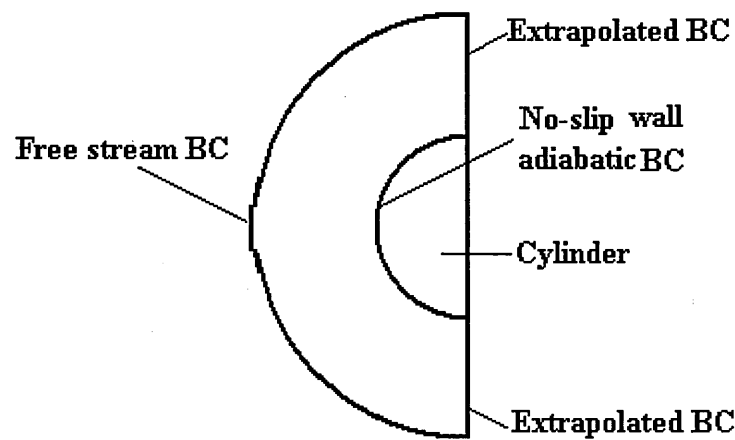
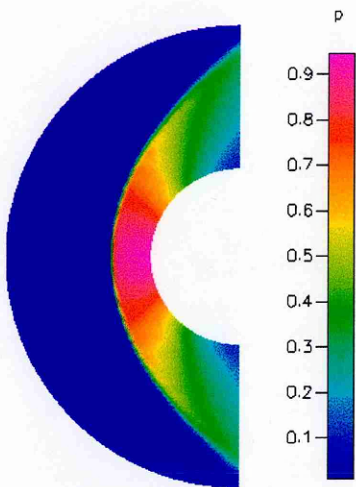
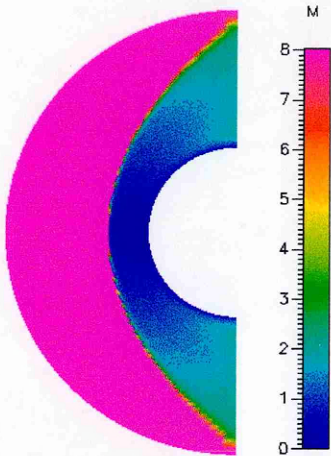


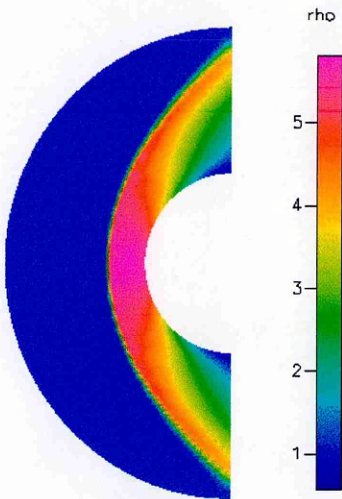
Figure 6-18. Grid system for simulating undisturbed bow wave



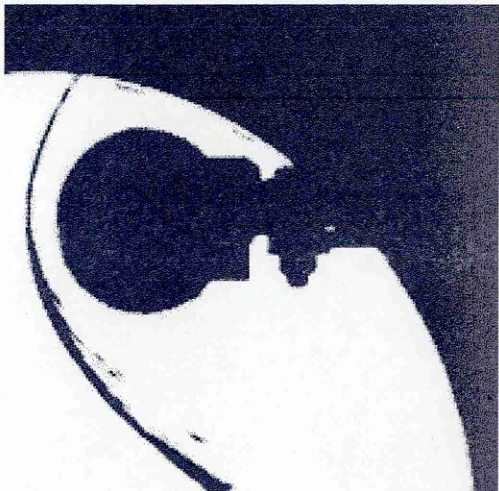
(a). Computational pressure plot.



(b). Computational Mach number plot

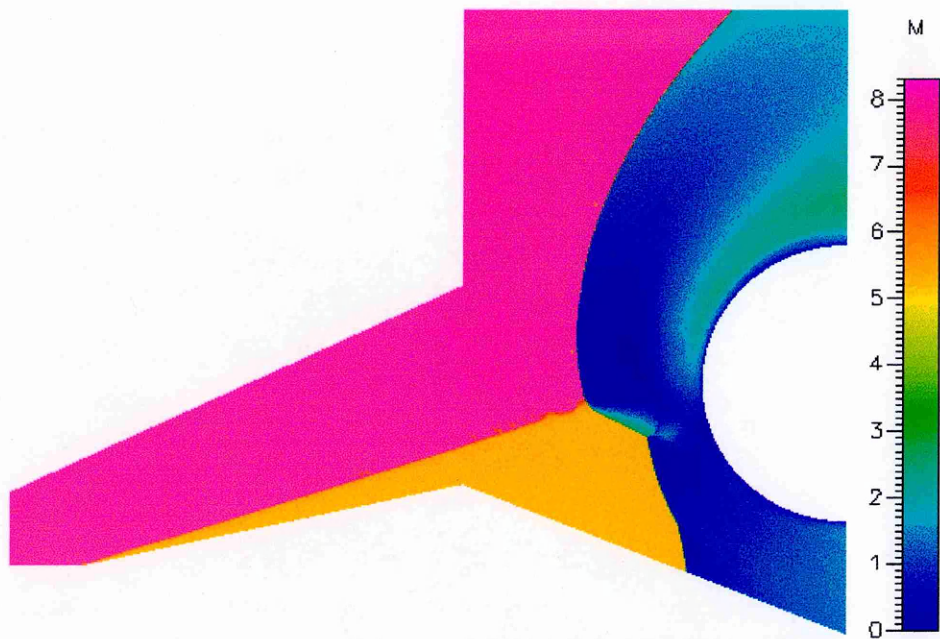


(c). Computational density plot

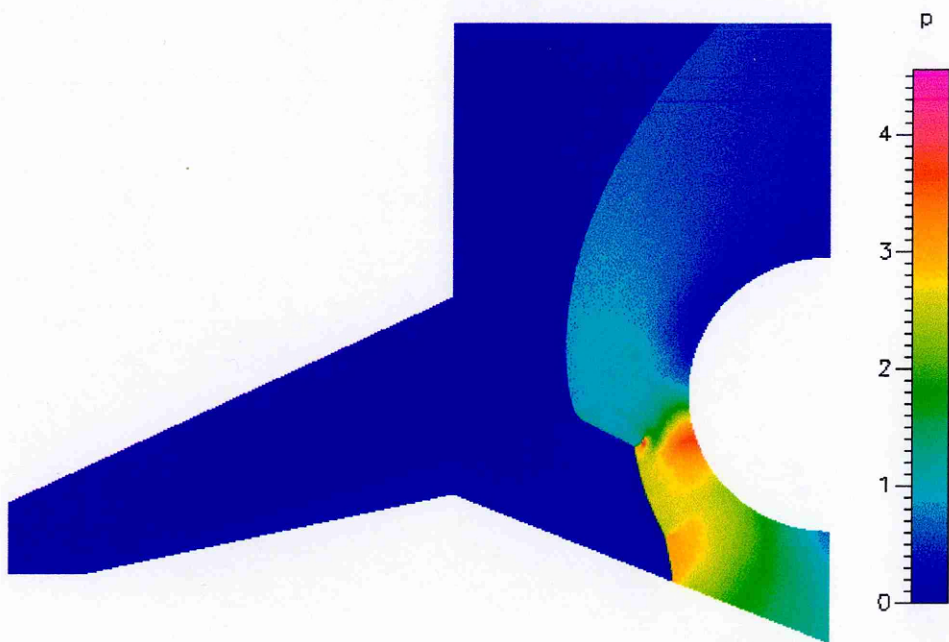


(d). Experimental schlieren photograph

Figure 6-19. Comparison of the undisturbed flow

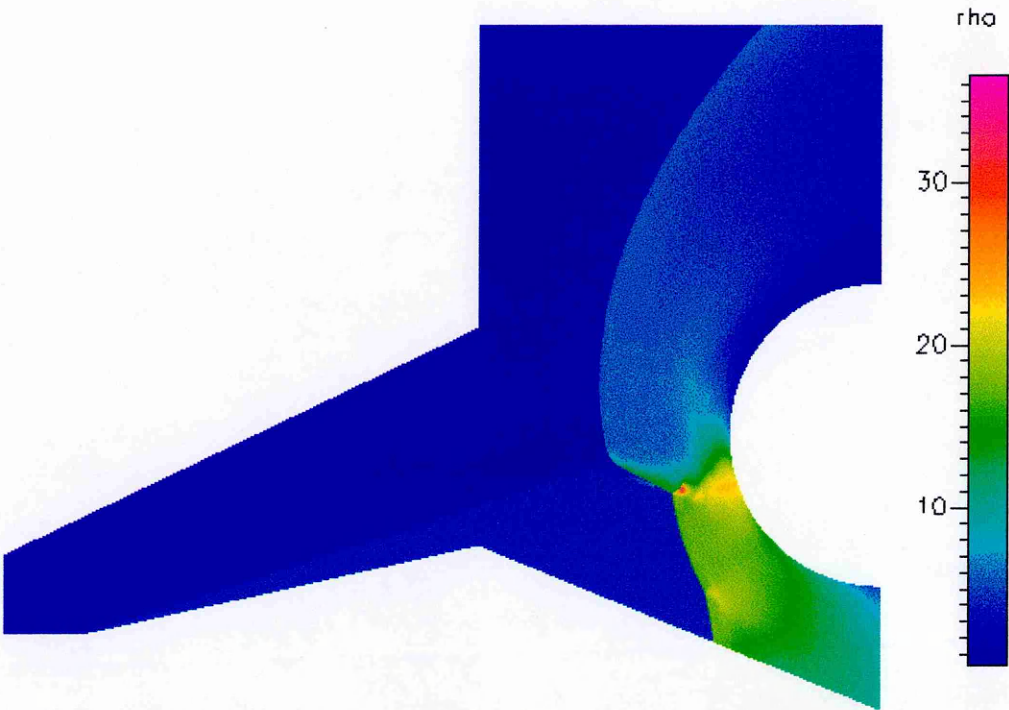


(a). Mach plot

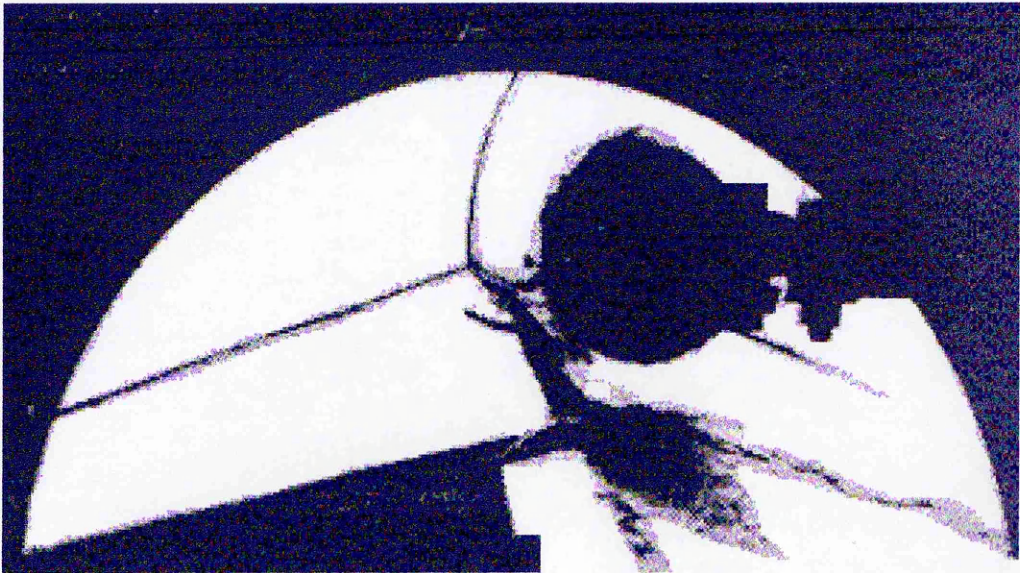


(b). Pressure plot

Figure 6-20. Computational Type IV Interference pattern

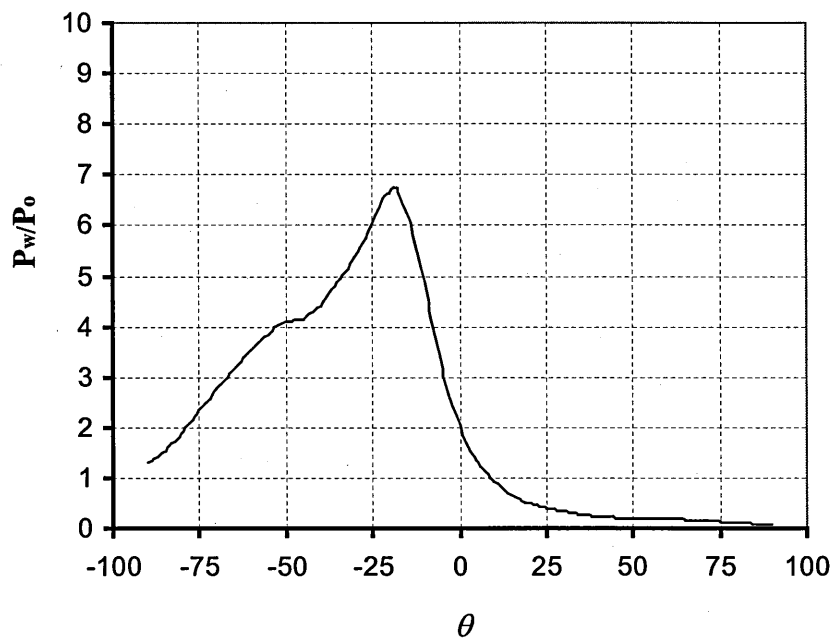


(a). Computational density plot

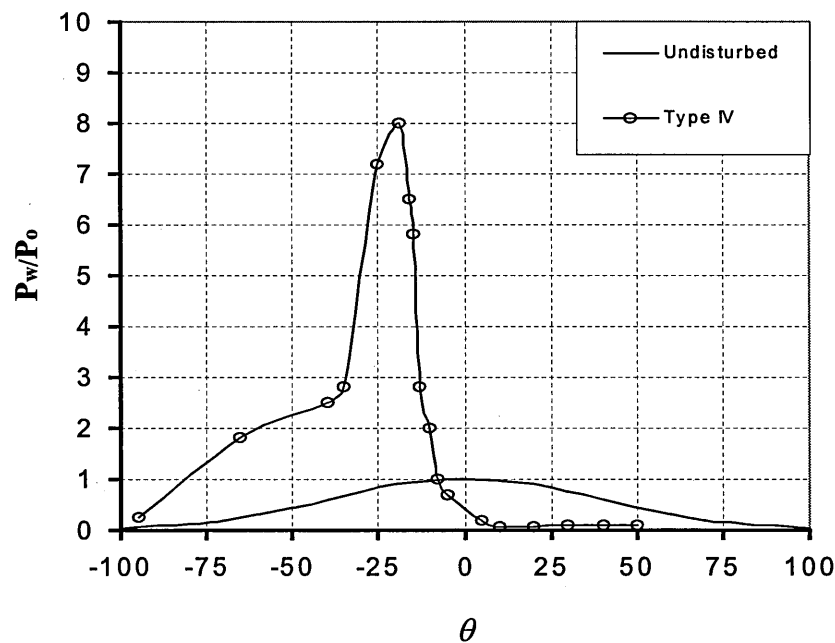


(b). Experimental schlieren photograph

Figure 6-21. Comparison of the Type IV Interference pattern

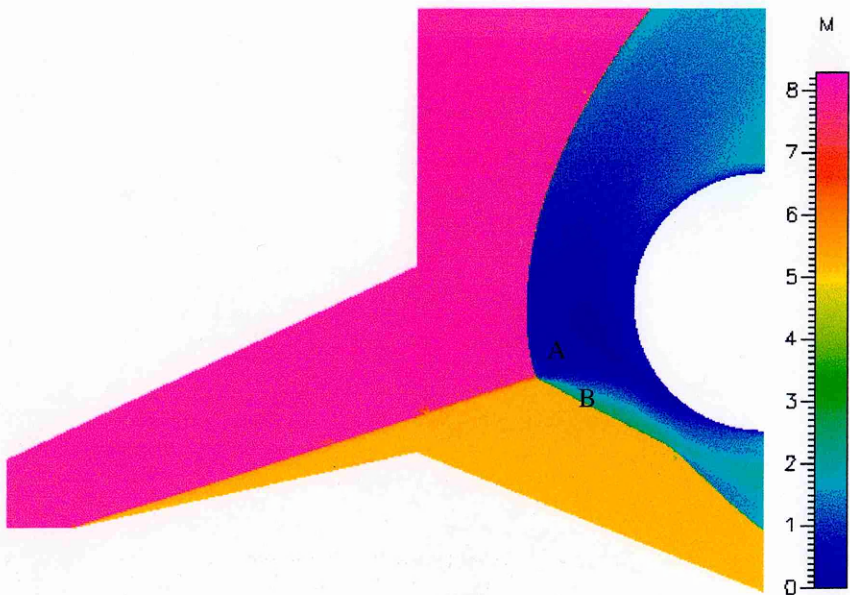


(a). Computational pressure distributions

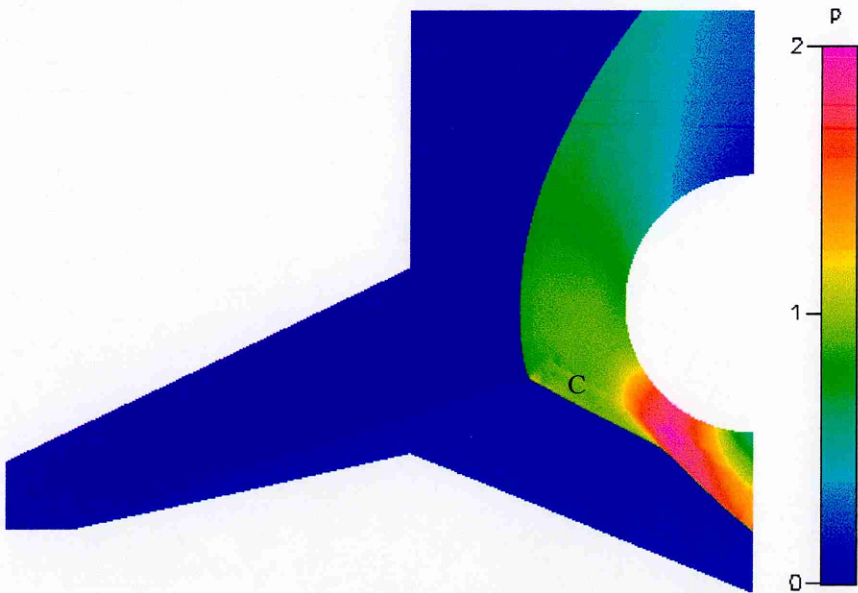


(b). Measured pressure distributions

Figure 6-22. Comparison of the pressure distributions of Type IV pattern

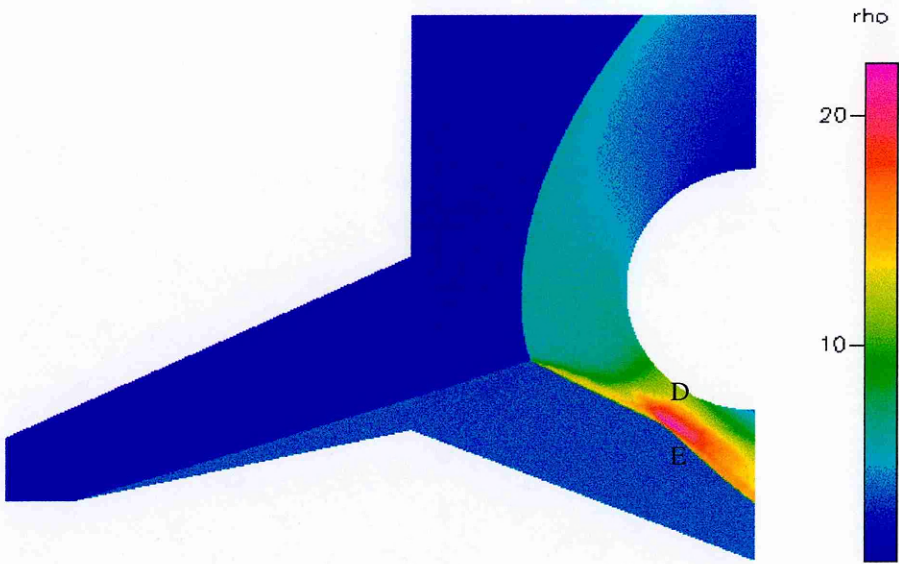


(a). Mach plot



(b). Pressure plot

Figure 6-23. Computational Type III interference pattern



(c). Density plot

Figure 6-23. Computational Type III interference pattern (continue)

Chapter 7

Moving Body Flow Computations

7.1 Introduction

The ability to accurately simulate unsteady flow fields about multiple and/or moving body flow is becoming increasingly important in the analysis of modern aircraft and launch vehicles. The ability to simulate this kind of flow problem is the remarkable feature of the Chimera grid method. The presence of moving bodies continually changes the relative orientation of the overset domains during the flow simulation. In such cases, it is difficult for the user to provide apriori all the input necessary for hole-cutting and stencil identification. Therefore it is required that all the tasks associated with a Chimera scheme such as hole-cutting, stencil identification, and interpolation coefficients can be performed in a fully automated mode. As mentioned in Chapter 3, the AMRC scheme is a fully automated solution adaptive Chimera grid scheme. In this Chapter, multiple and moving body flows are simulated to show the flexibility of the AMRC scheme to simulate this kind of flow at high speed.

7.2 Multiple Moving Body Flow Computations

The Chimera grid method has been proven to be an efficient and versatile approach for movement in that grids for different parts of the configuration can be moved relative to one another to simulate unsteady body dynamics⁸⁸⁻¹⁰⁴. In this section, I shall simulate a multiple moving body supersonic flow. Since it is not easy to obtain suitable experimental

data for comparison, I shall consider two identical double-wedge bodies and simulate a pre-described body motion. Although the whole flow field and moving body trajectory can not be compared with experiment or other computation, it is still possible to check some of the simulation results with theoretical analyses.

7.2.1 Specification of the Problem

Figure 7-1 shows the geometry definition of a double wedge aerofoil. The chord length is 2 inches with a half angle of 10° and the maximum thickness is 0.353 inches. Nose and tail angles are equal to 20°. In this simulation, two identical double-wedge bodies were chosen as the geometric configuration.

Figure 7-2 presents the Chimera grid system and boundary conditions. The #2 double wedge aerofoil is placed downstream of the #1 aerofoil and is constrained to move forward towards and below aerofoil #1. The baseline domain D_1 is a uniform Cartesian mesh of 80×80 cells, each of 2.0mm×2.0mm in dimension. The two overset domains covering the aerofoils, D_2 and D_3 , are identical, formed by two body fitted meshes, each of 40×10 cells.

The flow model is specified as inviscid flow. It is supposed that the total temperature to be ambient temperature, $T_0=288\text{ }^{\circ}\text{K}$, the total pressure to be $1.0\times10^5\text{ (N/m}^2\text{)}$ and the free stream Mach number equals 2. The other flow variables can be calculated, as shown in Table 7-1. Some of the simulation results can be compared with analytical results. For example, the oblique shock wave angle should be 39.3 degrees for this case, and the expansion at the ridge point should be 32.6 degrees.

Table 7-1. Free stream conditions

M_{∞}	$P_{\infty}\text{ (N/m}^2\text{)}$	$T_{\infty}\text{ (}^{\circ}\text{K)}$	$\bar{p}_{\infty}$	$\bar{\rho}_{\infty}$
2.0	1.278×10^4	160	0.179	1

In the simulation, the refinement factors rI and rJ are set as 4. The maximum refinement level $Lmax(Nd)$ is specified as 1, i.e. the coarse grid system will be only refined

by one fine grid level. The grid adaptation parameters are shown in Table 7-2. The CFL number is specified as 0.8. For this simulation, the second order Roe's FDS scheme combined with the SUPERBEE limiter function was chosen as flow solver.

Table 7-2. Grid adaptation parameters

l	rI	rJ	F_{tol}	P_{tol}
0	1	1	0.05	0.6
1	4	4	N/A	N/A

Initial conditions for this moving body flow are an initial converged steady solution. The initial position of the moving body (#2 aerofoil) is placed downstream of the #1 aerofoil (65mm afterward and 40mm underneath). Therefore, the steady solution around the #1 aerofoil can be compared with results from theoretical analyses. After obtaining a converged steady solution, the #2 aerofoil moves forward to pass underneath flow field of the #1 aerofoil at a constant normalised speed of $U/u_\infty = -0.5$.

7.2.2 Steady State Solution

To obtain the initial converged steady state solution, the simulation is started with the free stream condition, i.e. the whole flow field initial conditions are specified as free stream Mach number, normalised free stream density and pressure and free stream physical temperature, see Table 7-1.

For the sake of reducing computational expense, at the beginning, all three domains are set to carry out a non-adapting calculation. After some iterations, a converged flow solution on the coarse grid system is achieved, then adaption is turned on to obtain high resolution solution to resolve the complex flow features.

Figure 7-3 presents the steady state numerical solutions. Complex shock-shock interaction flow structures are well captured. The trailing edge oblique shock of the #1 aerofoil is deflected three times. It first interacts with the oblique shock due to the #2

aerofoil leading edge, then with the expansion wave due to the ridge and finally with the oblique shock due to trailing edge. From Figure 7-3a, Mach plot, we can see that at point "A", a small step appears on the oblique shock. This is mainly due to the sudden change of cell sizes in different domains.

Figure 7-4 presents the pressure coefficient distributions on #2 aerofoil resulting from the numerical solution (coarse level). On the upper fore-body surface the pressure coefficients are effected by the interaction of the #1 aerofoil expansion wave.

Figure 7-5 makes the comparison of the steady state numerical solution with the theoretical results of streamlines, shock wave and expansions around the #1 aerofoil. For this aerofoil, the flow field is not disturbed by the down stream #2 aerofoil. The oblique shock wave angle $\theta=39.5$ degrees. The angle of the expansion fan is about 30 degrees. These two angles can be compared with theory results. From the oblique shock chart¹⁵⁴ with $M_1=2.0$ and $\delta=10^\circ$ we have $\theta=39.3^\circ$. From Prandtl-Meyer table¹⁵⁴ with $M_1=2.0$ and $\delta=10^\circ$ we can get $M_2=1.64$, $v_2=16^\circ$ and $\mu_2=37.6^\circ$. For this case, the angle the flow turns through in the expansion fan at the ridge point is 20° (hence $v_3=16+20=36^\circ$), so we have $M_3=2.37$ and $\mu_3=25^\circ$. It can be calculated that the angle of the expansion fan $\phi=37.6-(25-20)=32.6^\circ$.

Figure 7-6 presents the computational pressure coefficient distribution on #1 aerofoil surface. The computational C_p values on fore-body and aft-body are $C_{p2}=0.271$ and $C_{p3}= -0.164$ respectively. Table 7-3 presents the comparison of the computational results (pressure coefficients and Mach numbers) in regions 2 and 3 (see Figure 7-5b) and those from the shock expansion theory and Ackeret's theory¹⁵⁴ for supersonic flow.

Table 7-3. Comparison of pressure coefficients and Mach numbers

	C_{p2}	C_{p3}	M_2	M_3
Computational results	0.271	-0.164	1.62	2.39
Shock expansion theory	0.258	-0.159	1.64	2.37
Ackeret's theory	0.201	-0.201	2.0	2.0

The computational results agree well with the results from the shock expansion theory, but differ with the results from Ackeret's theory. This is because that for this case, both the free stream Mach number, $M_\infty=2.0$, and surface slope angle, $\delta=10^\circ$, are all a bit too large to apply the Ackeret's theory. This theory is suitable for Mach number in the region of $1.2 < M < 2.0$, and the slope angle, δ , must everywhere be small, i.e. $\sin(\delta) \approx \delta$.

7.2.3 Prescribed Motion

After obtaining a converged steady solution for this case, the domain D_3 , which is around the #2 aerofoil, is set to move with the aerofoil together at a constant normalised velocity $U/u_\infty = -0.5$. This is a prescribed translation movement model. The aerofoil could also be released to rotate around a fix rotation axis (x_r, y_r) at a specified rotational angular velocity. In this case, it will be a prescribed translation movement model.

When using the Chimera grid scheme to simulate a moving body flow, although there is no need to regenerate grids, the communication among the domains must be recomputed for each movement or time step. During the simulation, the hole-cutting and recipient-donor stencil searching processes are required to be implemented at every time step. A problem may arise. Since the hole-cutting, recipient-donor stencil searching process is expensive, as the time step decreases the computational cost of solving a moving body problem increases. However, if the time step is large, the flow field at every time step may be not sufficiently accurate. In order to solve this problem, in the AMRC scheme, a sub-iteration procedure is employed. That is, at every time level T_1 , dividing the time step $\Delta\tau$ by a sub-iteration number, which can be specified by user, one can obtain a smaller time step. Then the sub-iteration calculations are carried out until reaching next time level $T_2=T_1+\Delta\tau$. At the new time level, the hole-cutting and recipient-donor stencil searching and the inter-grid communication processes are carried out.

Figures 7-7 and 7-8 present the computational flow fields, Mach number plots and density plots, at different times. The complex flow structures are well resolved. As the #2 aerofoil crosses through the expansion wave from the #1 aerofoil ridge, a reflection shock

induced on the #2 aerofoil upper forebody surface occurs, see Figures 7-7a and 7-7b. While the #2 aerofoil crossing through the oblique shock wave due to #1 aerofoil leading edge, a shock stem appears and grows in size see Figure 7-7c. Near to the #2 aerofoil leading edge point, a normal shock is formed (note the legend) and is going to vanish at this stage. In Figure 7-7d, a Mach disc, two triple points and two slip lines are clearly captured. As the movement evolves further, two oblique shock waves induced by two leading edges intersects with each other, see Figures 7-7e and 7-7f.

7.2.4 Impacting Moving Bodies

The intention of this sub-section is to show the ability of the AMRC scheme to simulate impacting moving bodies. We again chose the two identical double-wedge aerofoils used in previous sub-section as the geometric configuration and simulate a pre-described body motion problem. The #2 aerofoil is placed downstream of the #1 aerofoil, 65mm afterward but just 10mm underneath the centreline of #1 aerofoil. Figure 7-9 shows the initial aerofoil positions, the Chimera grid system and boundary conditions. In this simulation case, domain D_2 oversets the baseline domain D_1 and overlaps with domain D_3 . As the #2 aerofoil moving forward (at a constant normalised speed of $U/u_\infty = -0.5$) it will move very close to the #1 aerofoil lower surface. The #1 aerofoil will introduce a Chimera hole into domain D_3 . As a result, for this kind of moving body flow case the flow variable communication between the multiple domains is more complex. Extra care must be taken to carry out Chimera hole cutting and to locate the recipient-donor stencil.

The simulation strategies and the grid adaptation parameters are the same as those used in the previous sub-section.

Figure 7-10 presents the initial steady state numerical solution, pressure and Mach plots. Figure 7-11 presents the computational flow fields, density plots, at different time instants. At this stage, the lower oblique shock wave induced by #1 aerofoil trailing edge impinges #2 aerofoil forebody upper surface and the reflected shock wave interacts with the upper oblique shock wave induced by #1 aerofoil trailing edge, see Figure 7-11b. As the #2

aerofoil movement evolves, the lower expansion wave induced by #1 aerofoil ridge will impinge on the #2 aerofoil, see Figure 7-11c. The #2 aerofoil will move very close towards to #1 aerofoil, see Figure 7-11d. At this stage, the expansion wave is reflected by a few times between two aerofoil surfaces inside this small gap.

Figure 7-12 shows the Chimera holes introduced by intersection of the domains (Figure 7-12a) and by the solid body (Figure 7-12b). This case illustrates the flexibility of the domain arrangement of the AMRC code due to the graph data structure and orthogonal list storage.

7.3 Freely Oscillating Aerofoil Flow Simulations

In this section, a supersonic flow over a freely oscillating aerofoil is simulated by using the AMRC scheme and by using the CFD-FASTRAN[®] flow solver. Both steady and unsteady supersonic flow simulation results are presented, analysed and compared with experimental results¹⁵⁵ and the results obtained by using the Adaptive Mesh Refinement scheme (adaptive body recovery) within DRAMR^{131, 132}.

7.3.1 Specification of the Problem

The geometric configuration of this simulation is a 3-inch chord NACA65-009 aerofoil model. It is a symmetric aerofoil with a round leading edge and a sharp pointed trailing edge¹⁵⁵. For this aerofoil, steady state experimental data, i.e. surface pressure coefficient distributions, normal-force, and pitching-moment coefficients are available from reference¹⁵⁵ at various angles of attack (0° – 10°) in a supersonic flow at Mach 1.62.

The wind tunnel test conditions for the experimental investigations presented in reference¹⁵⁵ are shown in Table 7-4. It is assumed that a laminar boundary layer develops from the leading edge and transition does not occur on the aerofoil at this Reynolds

number. A laminar trailing edge separation is expected to occur at incidence above about 4°. In the steady simulation, we only chose incidences of 0, 2, 4 and 10 degrees.

Table 7-4. Wind tunnel test conditions

M_∞	1.62	
Re (Based on chord length)	1.07E+06	
α	0,1,2,3,4,6,8,10	degrees

7.3.2 Computational Details

Figure 7-13 shows the details of the Chimera grid system at 0° incidence and the boundary conditions. The whole grid system is composed of two domains. The baseline domain D_1 is a uniform Cartesian grid of 100×100 cells, each of 3.81mm×3.81mm in dimension, the same as the baseline mesh used in reference¹³¹ (in reference¹⁵⁶, The baseline domain D_1 is a uniform Cartesian grid of 100×100 cells, each of 2.25mm×2.25mm in dimension). The overset domain D_2 is formed by two identical body fitted meshes around the aerofoil. They are stretched grids of 100×40 cells. The grid is clustered towards the wall to form a viscous mesh. Near the aerofoil surface, the smallest cell size is 0.1mm×0.387mm in dimension. For the various incidences, the overset domain meshes are rotated around a fixed axis along the chord. There is no need to regenerate the grid system with body motion. For unsteady flow, the overset domain D_2 moves together with the aerofoil.

In the simulation, the free stream conditions are set to be the same as the wind tunnel test conditions¹⁵⁵, see Table 7-3. Steady state calculations are only carried out for incidences of 0 2, 4 and 10 degrees. Note that it is supposed that the free stream air was heated to ambient temperature, $T_\infty=288$ °K, for comparison (the same as Roper¹³¹ did for simulating the same problem using DRAMR code). Therefore, the other flow variables can be calculated, as shown in Table 7-5.

Table 7-5. Free stream conditions

M_∞	$Re_\infty \text{ (m}^{-1}\text{)}$	$P_\infty \text{ (N/m}^2\text{)}$	$T_\infty \text{ (}^\circ\text{K)}$	$\bar{p}_\infty$	$\bar{\rho}_\infty$
1.62	1.404×10^7	3.77×10^4	288	0.272	1

In the simulation, inside the overset domain D_2 , the flow model is set as laminar flow. The physical property of dynamic viscosity varies with temperature according to Sutherland’s Law. Far away from the solid boundary, the viscous effects are very small and can be neglected. So inside the baseline domain D_1 , the flow model is set as inviscid flow.

For the baseline domain, the refinement factors rI and rJ are set as 4. Since within the overset domain we have already applied a finer viscous grid, the refinement factors rI and rJ are set as 2. The maximum refinement level $Lmax(Nd)$ is specified as 1. The grid adaptation parameters are shown in Table 7-6.

Table 7-6. Grid adaptation parameters

l	rI		rJ		F_{tol}	P_{tol}
	D_1	D_2	D_1	D_2		
0	1	1	1	1	0.05	0.6
1	4	2	4	2	N/A	N/A

7.3.3 Steady State Simulation

Steady state simulations were performed for the NACA65-009 aerofoil at various angles of attack in a supersonic flow at Mach 1.62. Figures 7-14 and 7-15 compare the computational Mach plots (from the AMRC and a commercial CFD solver, CFD-FASTRAN) at incidences of 0°, 2°, 4° and 10° respectively. From these plots, we can see that the bow wave, the oblique shock waves at the sharp pointed trailing edge and the laminar boundary layer are all well captured. The solution resolution obtained from the AMRC is higher than that from CFD-FASTRAN. Since in AMRC, the solution is marched to a converged steady solution in a time accurate manner and high gradient region to be refined, when the incidence is larger than 4 degrees the vortexes due to laminar boundary

separation and sharp pointed trailing edge are also well resolved by the AMRC scheme, but they are not captured by the CFD-FASTRAN, see Figure 7-14c, 7-14d and 7-15c, 7-15d. In fact, the initial coarse grid used in the calculation by the AMRC is much coarser than used in the calculation by the CFD-FASTRAN.

Note that the computed Mach plots from the AMRC are smooth, even through the boundaries of the overset domain. This indicates that the flow field information is being passed correctly between overset boundaries in different domains.

Figure 7-16 compares the pressure coefficients for the present numerical results, the experimental data presented in the literature¹⁵⁵, the numerical results from the CFD-FASTRAN¹⁵⁶ and those from DRAMR¹³¹ at various angles of attack. The black lines with squares and triangles indicate the test data. The black lines with circles and stars indicate the results from DRAMR (only for incidences of 2 and 4 degrees). Different coloured lines indicate numerical results on the upper and lower surface of the aerofoil from the AMRC and the CFD-FASTRAN respectively.

From Figure 7-16a, at 0° incidence, we can see that the test results on the upper and on the lower surface near the leading edge are different. It indicates that the nose section of the aerofoil appears to be at a slight positive angle. At the trailing edge, there is a similar effect. These are suspected to be due to test errors caused by model accuracy or flow imperfection in the wind tunnel.

From Figures 7-16b and 7-16c, we can see that the numerical results agree very well with the experimental results. Numerical results from the AMRC and DRAMR are closer to the experimental data, especially near the leading edge and trailing edge. This is mainly because both AMRC and DRAMR apply the AMR algorithm. However the current version of CFD-FASTRAN has no grid adaptation ability. In Figure 7-16d, the computational pressure coefficients from CFD-FASTRAN on the lower surface at 40% of the chord length show a small peak. This is caused by the reflected wave at the point of oblique shock wave crosses through the Chimera overset boundary¹⁵⁶. The numerical solution from the AMRC is smooth, see Figure 7-14. This indicates that the oblique shock wave crosses through the Chimera overset boundary properly.

It obviously can be concluded that the grid adaptation or cell clustering according to the flow solution is very important in modelling such a supersonic flow problem.

Figure 7-17 presents an integrated aerodynamic characteristic, the pitching moment coefficient about half chord length, resulting from the surface pressures. The numerical results are slightly smaller than the test data, but have similar slope. At 0° incidence the pitching moment coefficients both from the AMRC and CFD-FASTRAN equal zero but the test data $C_m \approx 0.004$. In reference¹⁵¹, Rainey pointed out that the maximum probable error of the experimental data for incidence α was less than 0.05° . The differences of the simulation results and experimental data for C_m appear to be mainly due to an effective incidence on the model. The curves are displaced slightly for which no correction has been applied.

7.3.4 Unsteady State Simulation

For the unsteady flow simulation, the axis of rotation of the aerofoil is fixed at 20% chord length from the leading edge. The initial condition is a steady state converged solution at an incidence of 4° . The aerofoil is then allowed to move under the action of aerodynamic forces and moments. In the simulation, gravity is neglected. The aerofoil section was supposed to be made of solid aluminium ($\rho = 2700 \text{ kg/m}^3$). The mass for a unit (1m) length of aerofoil section is 967g and the moment of inertia about 20% chord from the leading edge is $5.82 \times 10^{-4} \text{ kgm}^2$. These conditions are the same as those defined in reference^{131, 132}. Due to the fact that the aerofoil is constrained in the stream, it can move only in one-degree of freedom (pitch), i.e. oscillating around the axis of rotation.

In this simulation, the aerofoil movement is relatively slow. As we can see later (Figure 7-18) the wavelength of the oscillation is $\approx 96 \text{ Hz}$ and the damping is $\approx 22\%$ per cycle. A dynamically moving body with associated flow interaction will approximate a sequence of static solutions if the body's reduced angular frequency ($\omega c/2u$) is of order less than one¹³¹ (for inviscid flows). For this case (which is dominated by the inviscid flow), $\omega c/2u \approx 0.04$ and hence the solution is likely to be quasi-steady. The aerofoil is supposed to

interact with flow field in a quasi-steady state, i.e. the solution at any instant will correspond to the static solution with body in the same static orientation.

The moments acting on the aerofoil about the rotation axis varied linearly with α for small α , i.e. less than 10° . At 0° incidence the moment should equal zero¹⁵⁵. Therefore, the body motion (oscillation) may also be predicted analytically by undertaking a one-degree of freedom trajectory simulation using forces and moments obtained from the static experimental data. The analytic frequency ≈ 105 Hz¹⁵⁶. The grey line in Figure 7-18 indicates the analytical motion.

In Figure 7-18, the black line shows the unsteady simulation result from the AMRC. The frequency is ≈ 96 Hz and the damping is $\approx 22\%$ per cycle. The pink line indicates the simulation results from DRAMR¹³¹. Here the frequency is ≈ 98 Hz and the damping is $\approx 20\%$ per cycle. The result from the CFD-FASTRAN is indicated by the green line. The frequency of the oscillation is ≈ 107 Hz. The wave amplitude is damped by about 25% per cycle.

Forces and moments should be computed by integrating the pressure and viscous stresses on the surface grids as they are calculated in the CFD-FASTRAN. In the current calculation, for the results from the AMRC code, only pressure is considered.

Since the surface grids for overset grids are allowed to overlap arbitrarily, the regions in the overlapped zones might be counted more than once if integration is performed over all the surface grids. In order for the AMRC scheme to be widely accepted for use in practical applications a procedure should be developed to accurately extract force and moment information from the numerical solution. Currently, body surfaces from which the aerodynamic forces and moments are extracted need to be specified manually.

7.4 Summary

The calculations presented in this chapter have demonstrated the flexibility of grid generation of the AMRC scheme for simulating multiple and moving body flows. The simulations show adequate agreement with the experimental results and demonstrate the

flexibility of the Chimera grid method in modelling supersonic moving body flows. Experience gained from these model problems indicates that the Chimera grid technique incorporated with the AMR algorithm could be used for simulating multiple and moving body high speed flow problems. It does not overly complicate the flow solver or cause any other adverse effects, but can greatly increase the flexibility of grid generation and solution resolution.

It should be pointed out that the Quirk's AMR code was designed for simulating transient shock hydrodynamic flows. After integrating with the Chimera grid method, the AMRC scheme is particularly suitable for simulating transient shock hydrodynamic flows around multiple bodies or complex geometry configurations. However, for a steady flow problem, the flow solution is simply marched to a converged steady solution in a time accurate manner. Simulating a steady state flow problem in this manner with $CFL < 1$ means that the rate of convergence is very slow. This is especially true for viscous flows. The implication is that the AMRC scheme is not particularly suitable to simulate a quasi-steady moving body flow problem. Currently, it is possible to restrict the viscous calculation in small near body regions by using the Chimera grid method. In author's opinion, it is still necessary to develop a suitable integration scheme or acceleration method for simulating steady state flow problems.

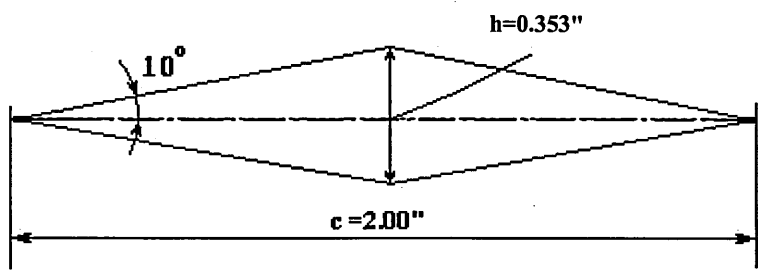


Figure 7-1. Double wedge aerofoil geometry dimensions

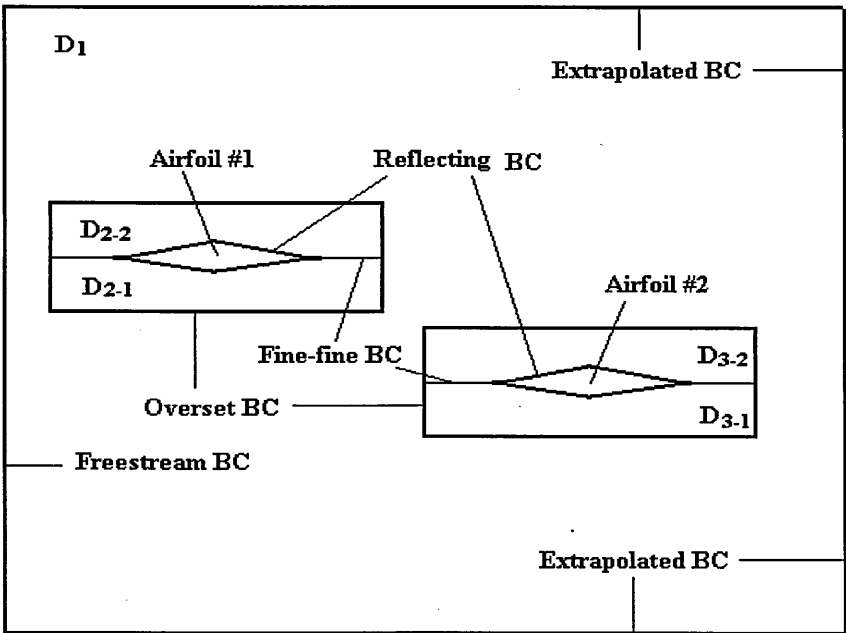
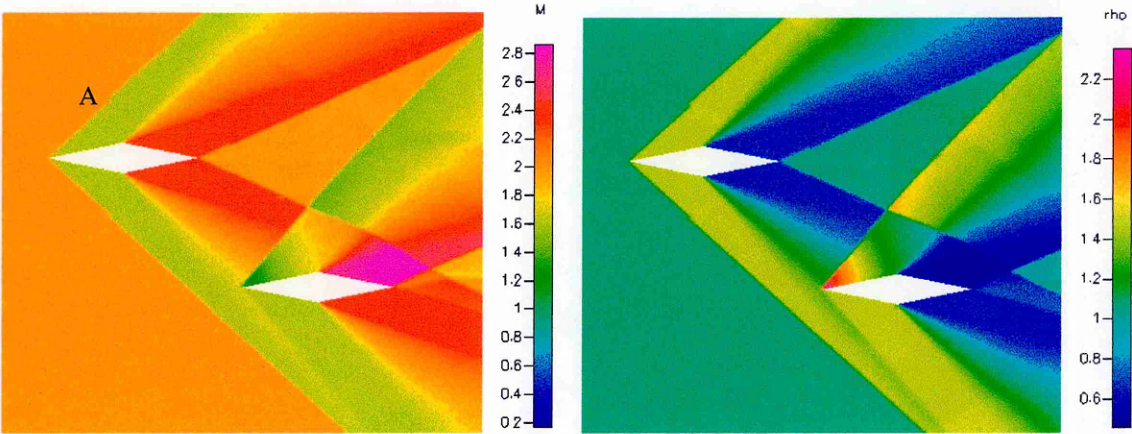
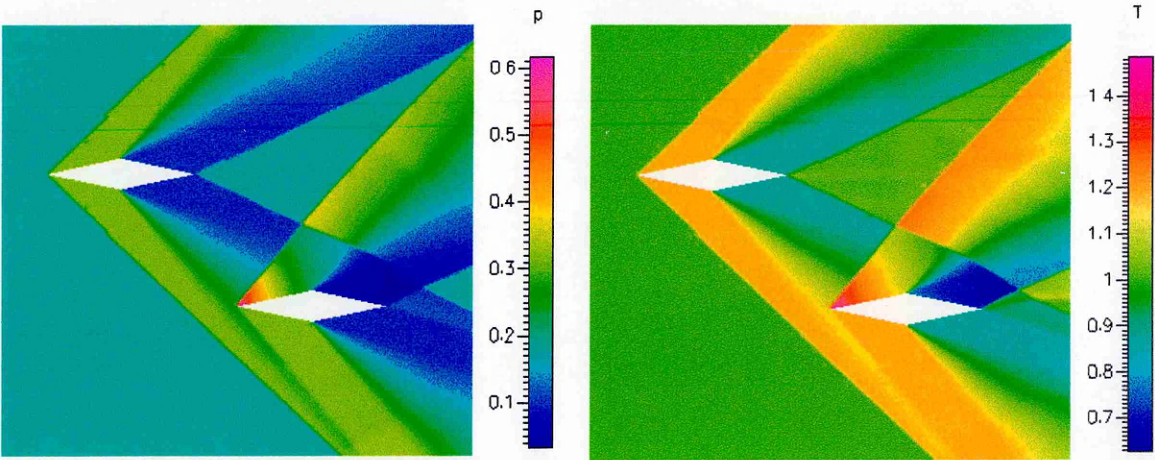


Figure 7-2. The Chimera grid system and boundary conditions



(a) Mach plot

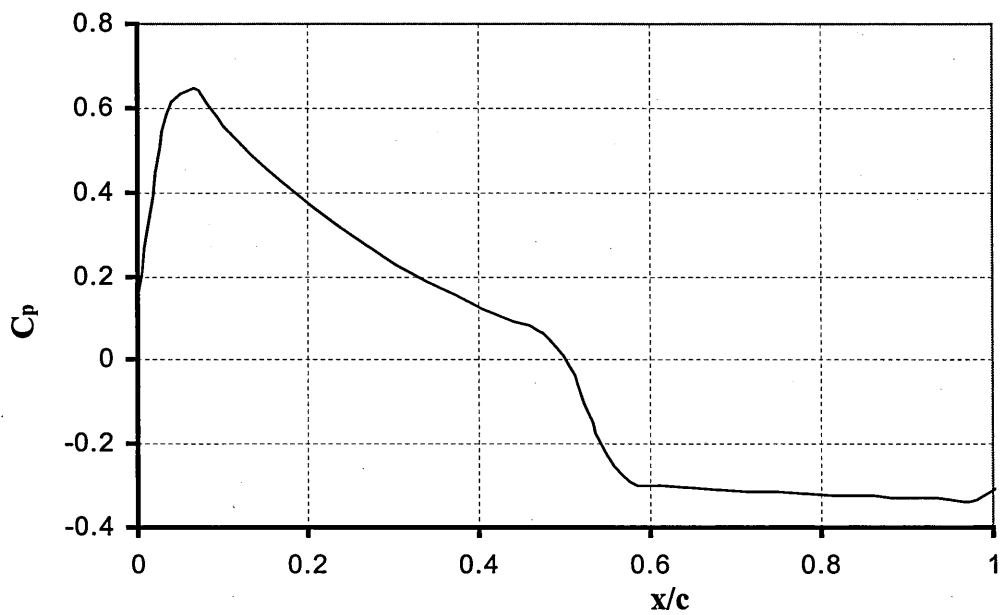
(b). Density plot



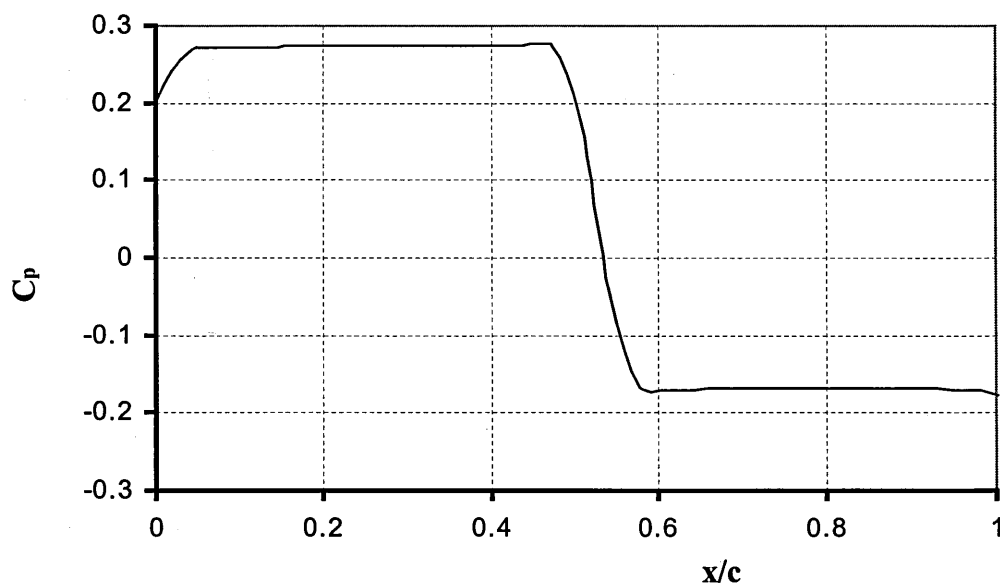
(c). Pressure plot

(d). Temperature plot

Figure 7-3. Steady state simulation results

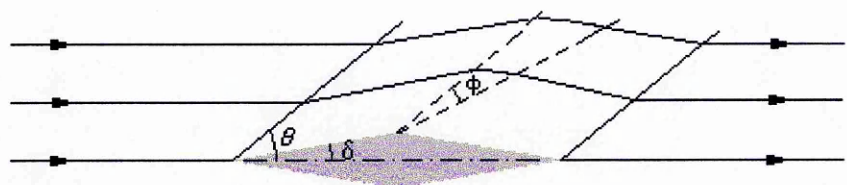


(a). Upper surface

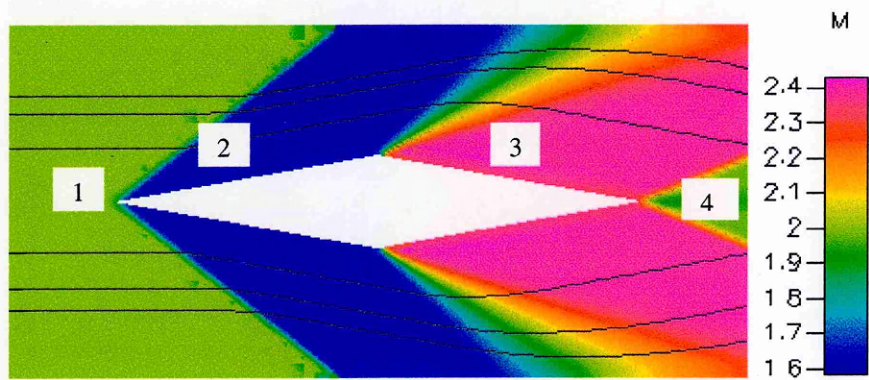


(b). Lower surface

Figure 7-4 Pressure coefficient distributions on #2 aerofoil



(a). Theoretical results



(b). Simulation results

Figure 7-5. Comparison of streamlines, shock wave and expansions

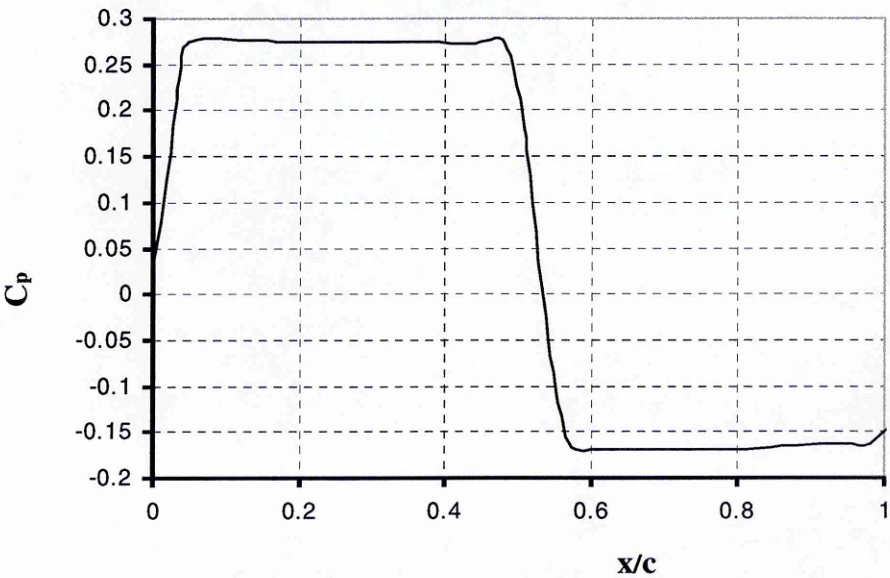


Figure 7-6 Pressure coefficient distribution on #1 aerofoil

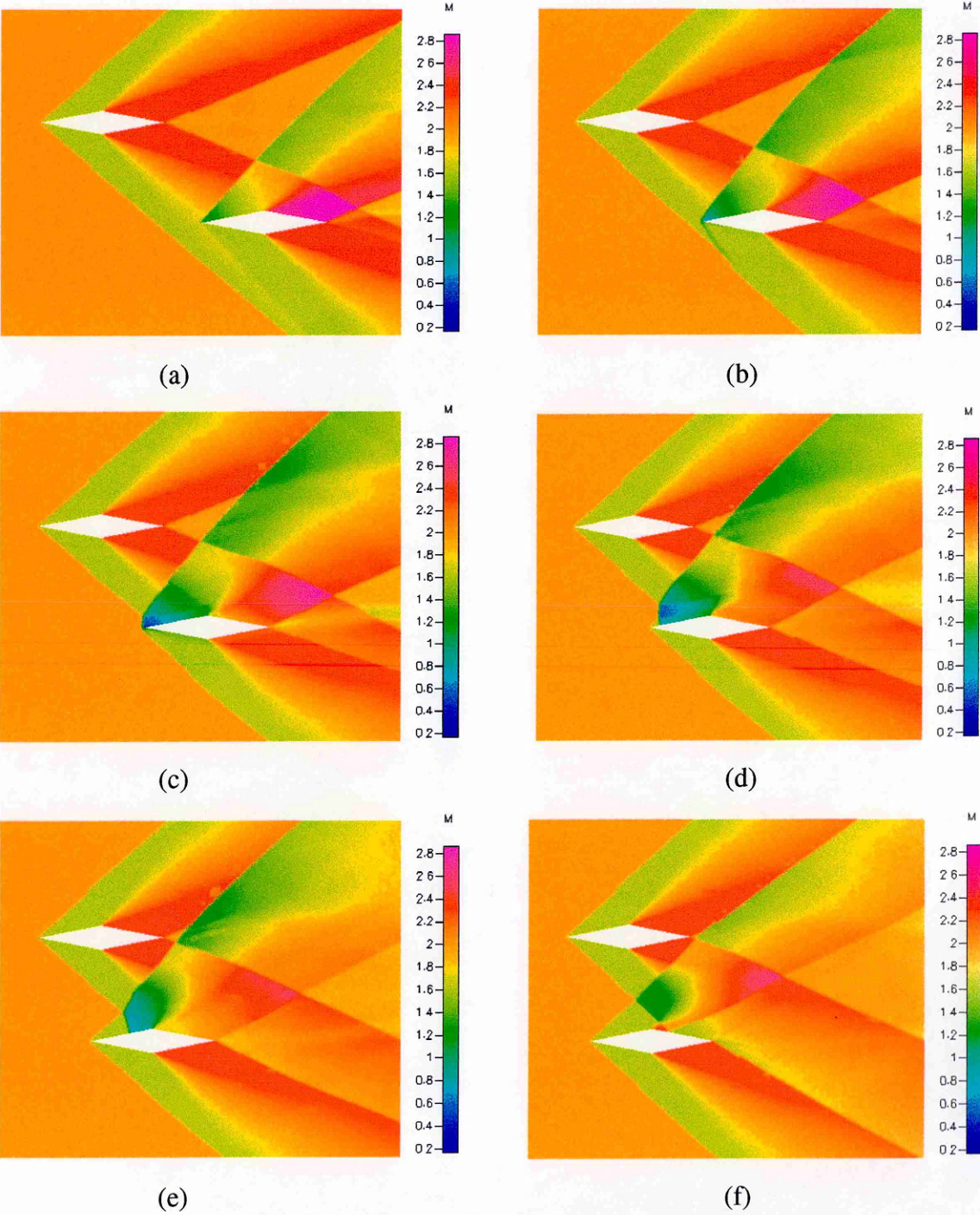


Figure 7-7 Moving body numerical solutions, Mach plots

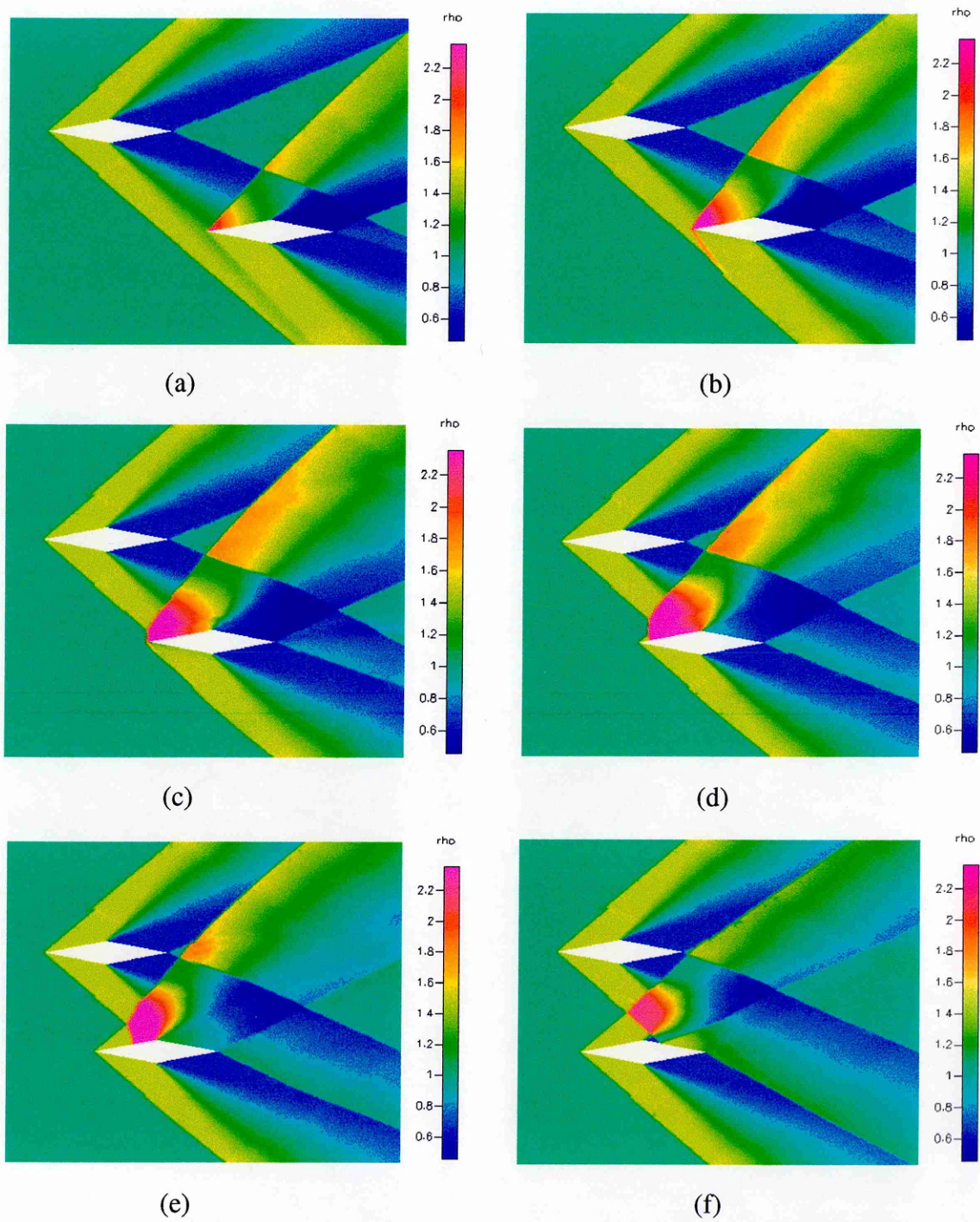


Figure 7-8 Moving body numerical solutions, density plots

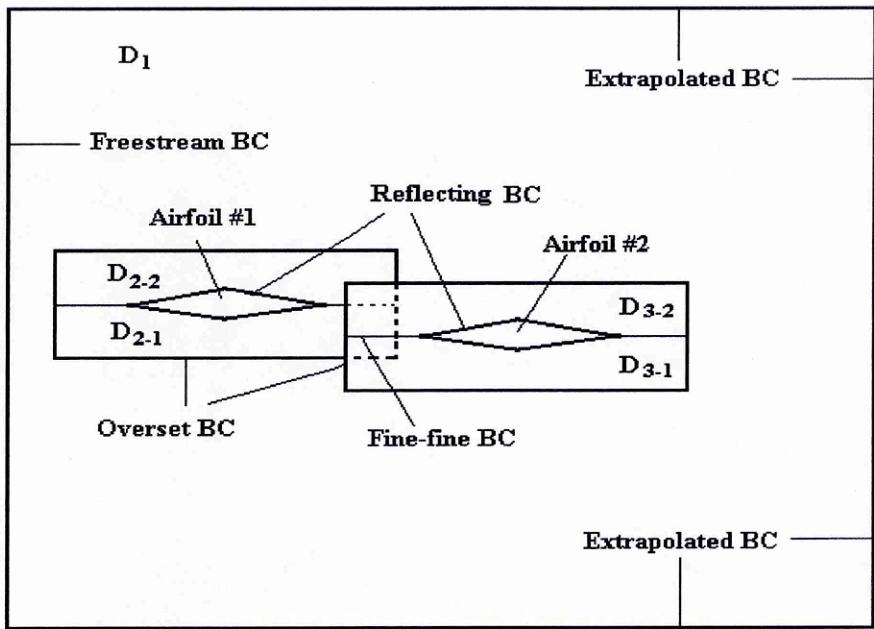
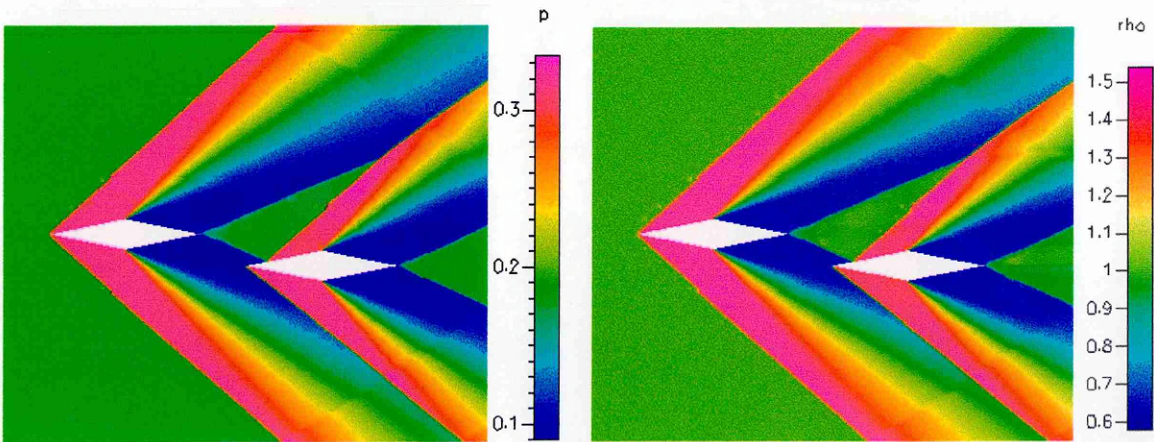


Figure 7-9. The Chimera grid system and boundary conditions



(a). Pressure plot (b). Density plot

Figure 7-10. Steady state simulation results (impact flow)

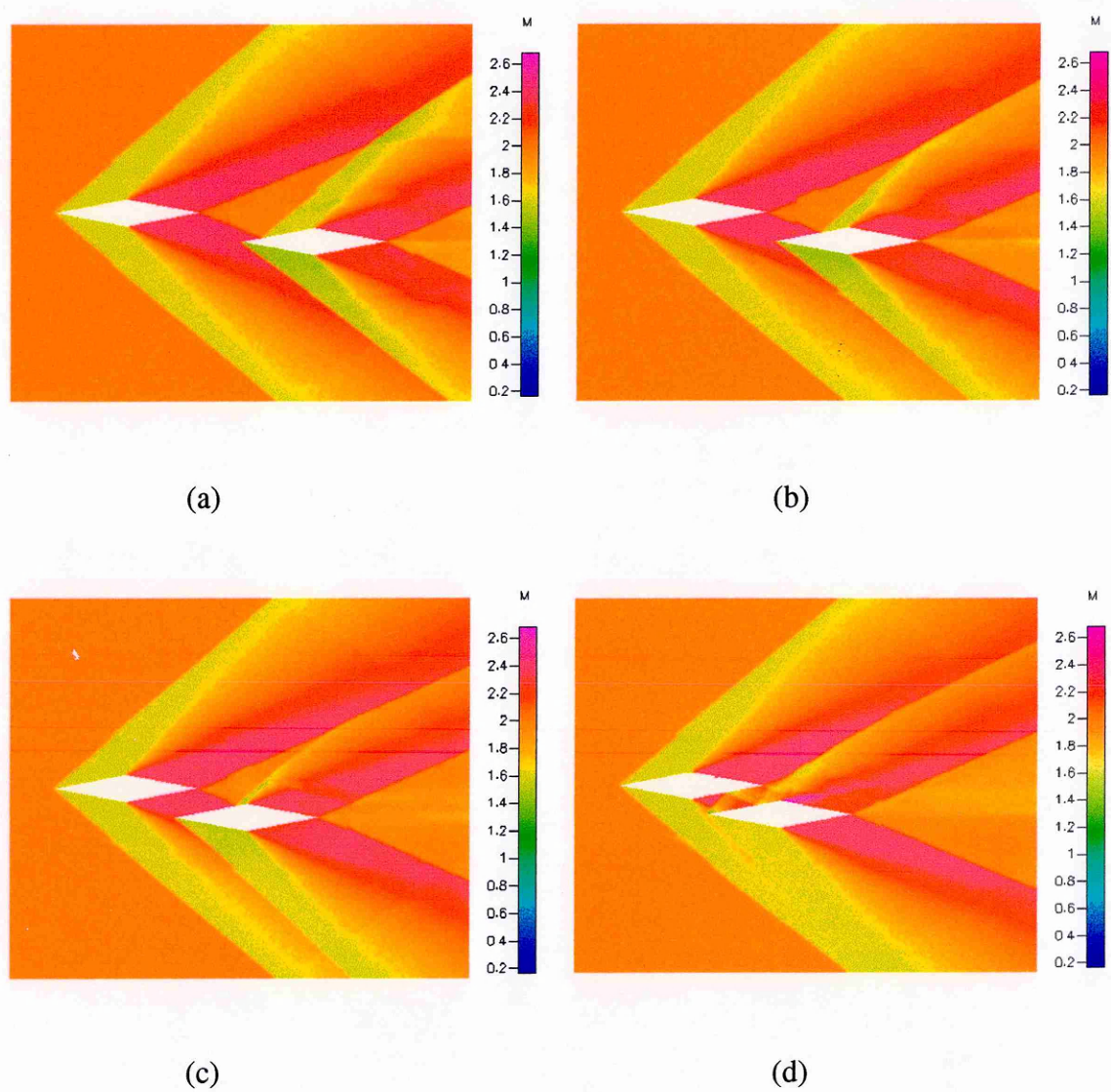


Figure 7-11. Impact moving body flow numerical solutions, Mach plots

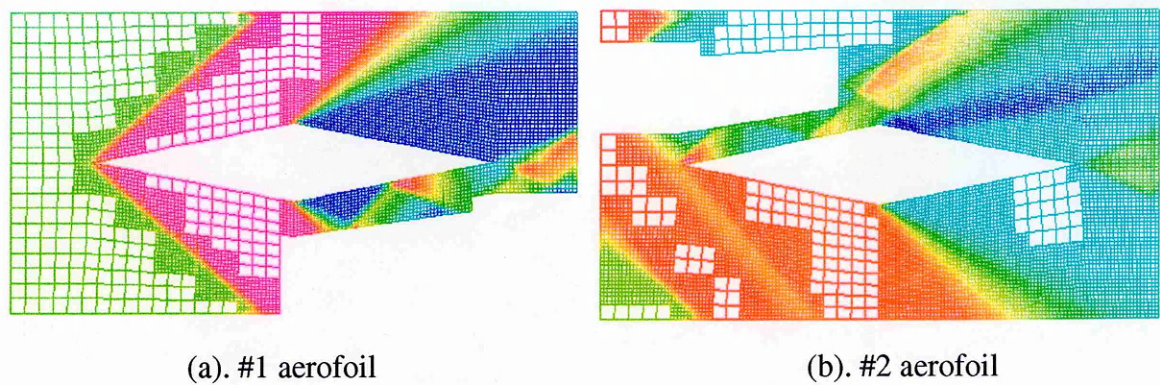


Figure 7-12 Chimera holes and adaptive meshes

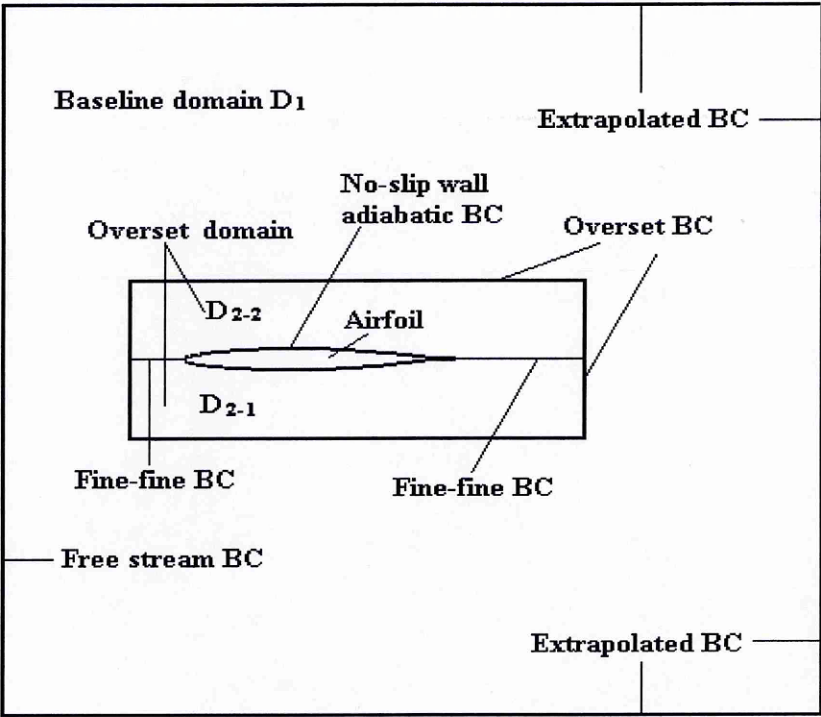
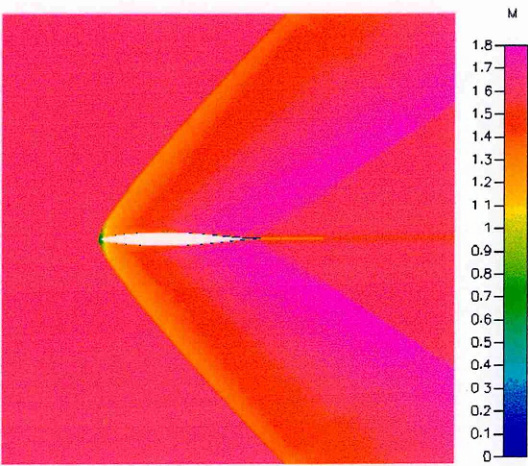
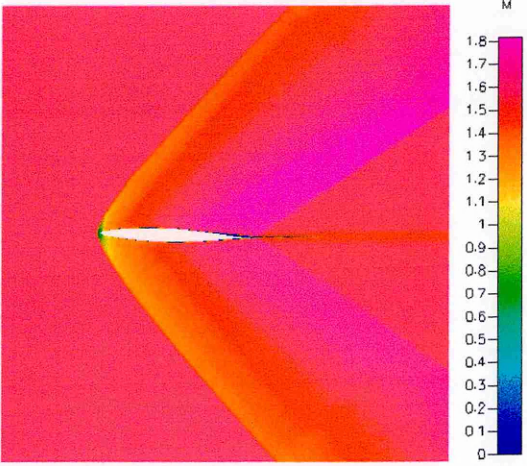


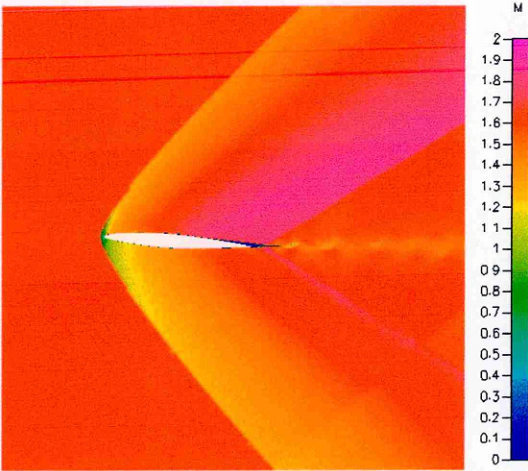
Figure 7-13 Chimera grid system and boundary conditions for NACA65-009 aerofoil



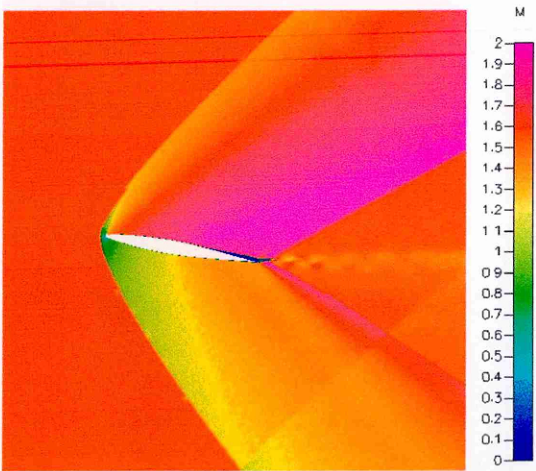
(a). $\alpha=0^\circ$



(b). $\alpha=2^\circ$



(c). $\alpha=4^\circ$



(d). $\alpha=10^\circ$

Figure 7-14. Computational Mach plots from the AMRC code

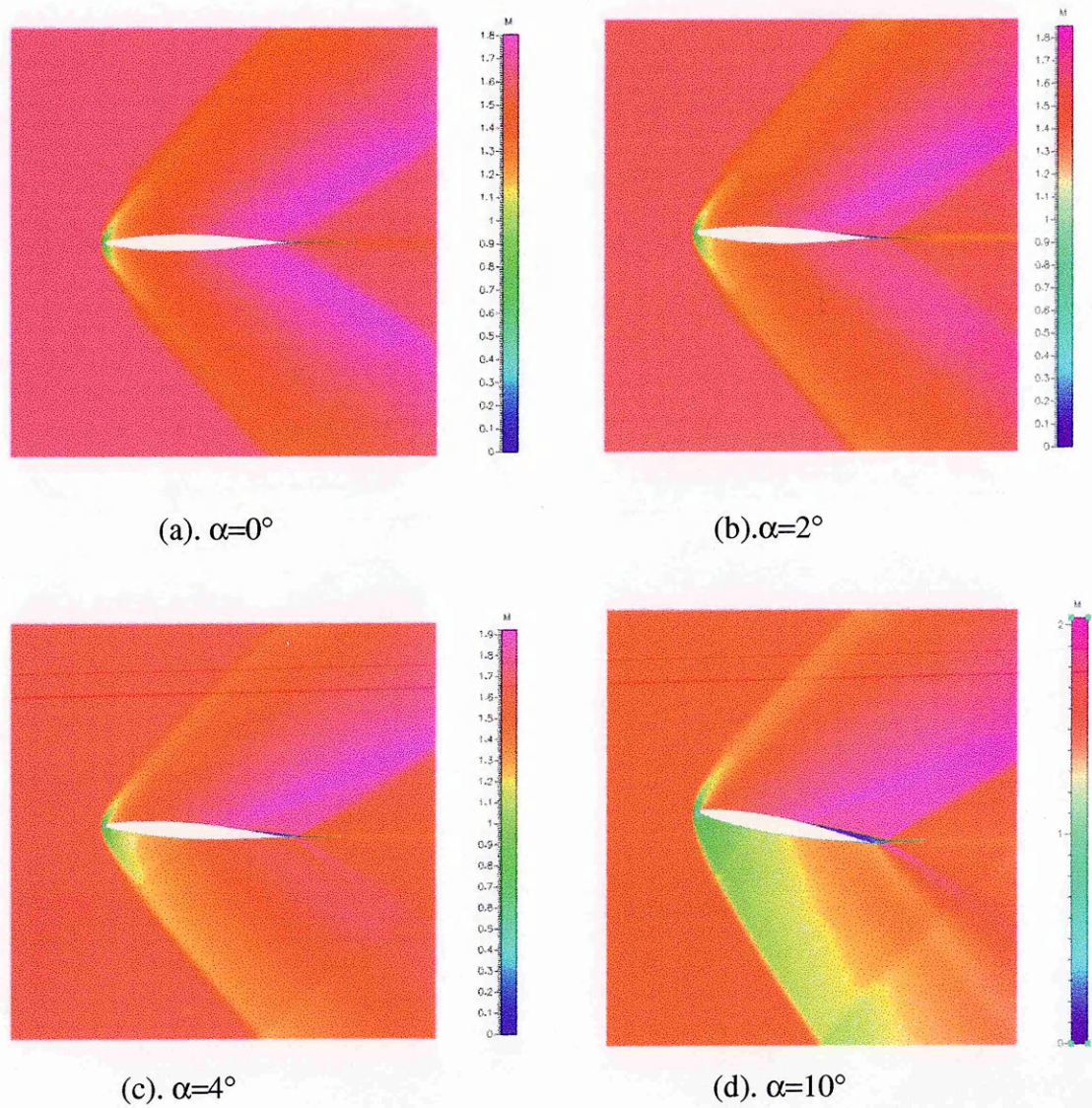
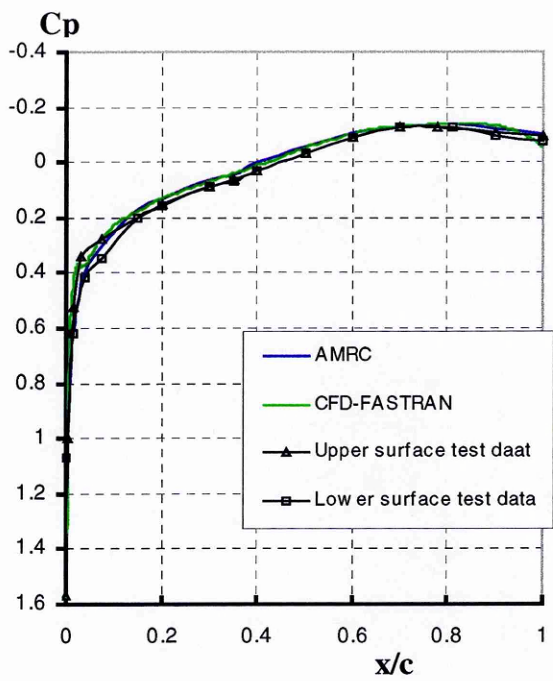
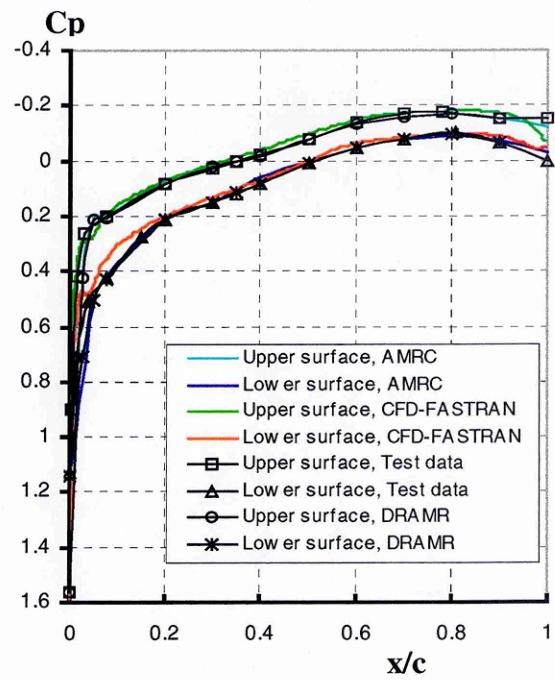


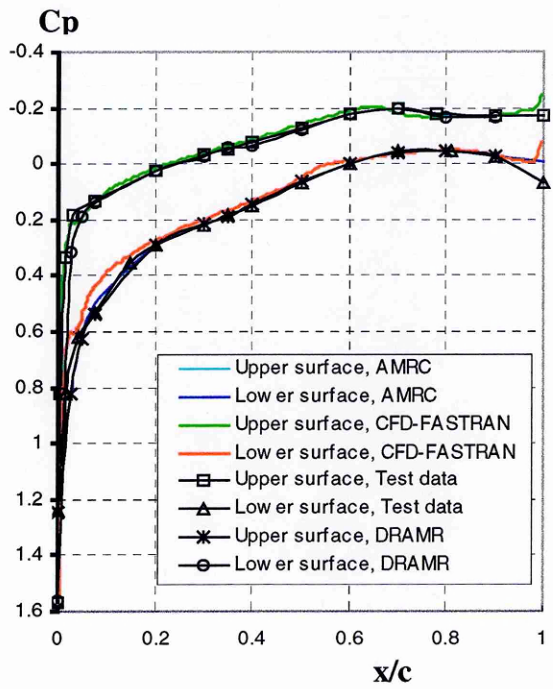
Figure 7-15. Computational Mach plots from the CFD-FASTRAN



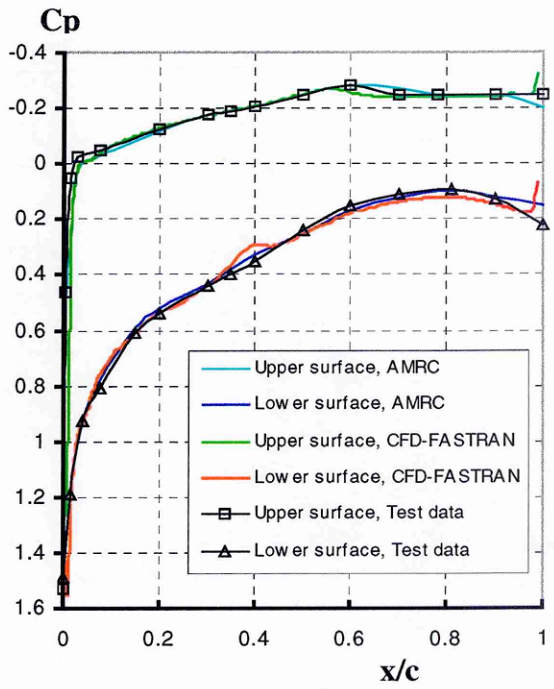
(a). $\alpha=0^\circ$



(b). $\alpha=2^\circ$



(c). $\alpha=4^\circ$



(d). $\alpha=10^\circ$

Figure 7-16. Comparison of the pressure coefficients

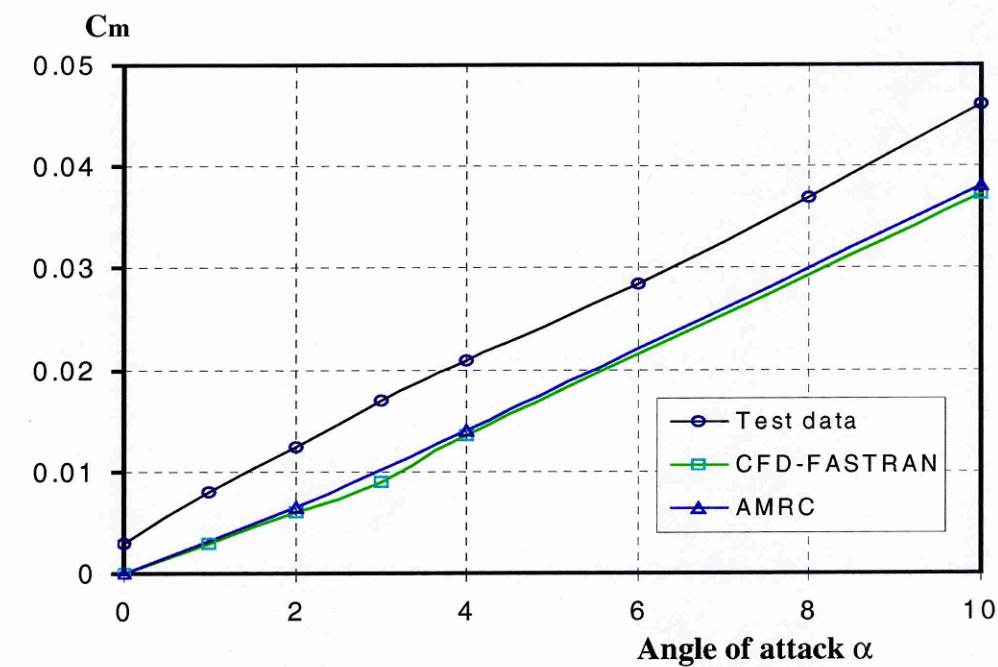


Figure 7-17. Comparison of pitching moment coefficients

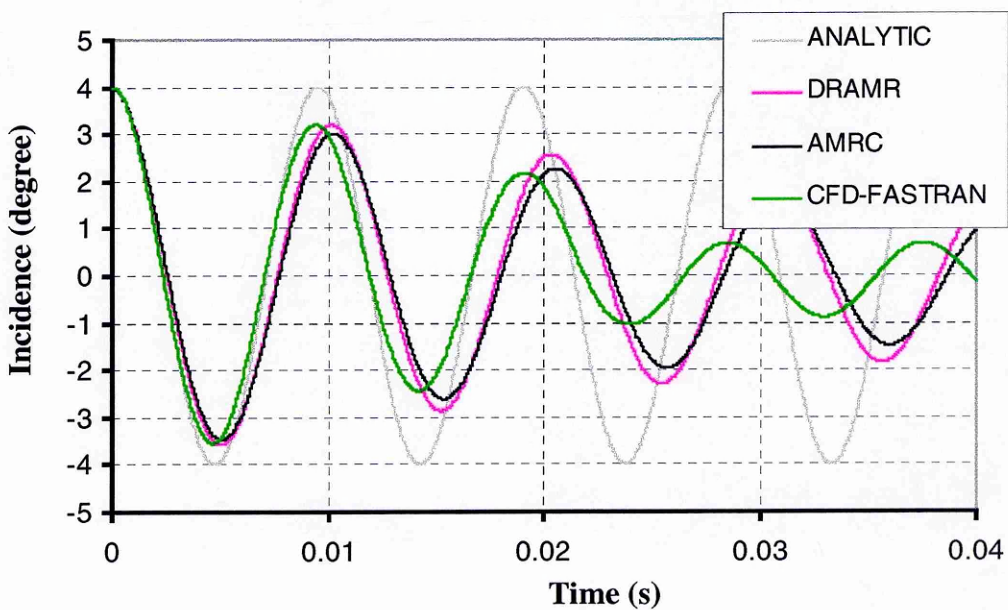


Figure 7-18. Aerofoil oscillation incidence vs time

Chapter 8

Closure

The research objective defined at the start of this thesis was to develop a solution adaptive Chimera overset grid scheme for modelling complex geometry problems and moving multiple body problems in an adaptive environment at high speed. This objective has been achieved. The Chimera overset grid method has been successfully integrated with the AMR algorithm. The procedure is robust, effective and efficient. The current version of the AMRC code works well. The results presented in this thesis have shown the ability of the Chimera grid method to map complicated configurations and multiple moving bodies, and demonstrates the high resolution resulting from the AMR algorithm. Discussion of the results and comparison of the various simulations is included in the body of the text.

In this chapter, we complete our discussion by relating conclusions and suggesting further research work.

8.1 Conclusions

This research work has successfully integrated the Chimera overset grid method with the AMR algorithm. The innovative aspects of this study are as below.

- 1). During this study a new solution adaptive scheme was developed which applied the AMR algorithm directly into a Chimera grid system.
- 2). Both Graph data structures and orthogonal list storage are employed in this new scheme to describing the Chimera sub-domains relationship and improve the region and point searching efficiency. These methods make the AMRC scheme more flexible and more efficient than existing Chimera grid schemes and AMR algorithms.

- 3). This study extended the application of the AMR algorithm. The main limitation of Quirk's AMR code, that is the poor geometric packing ability, has been replaced by a versatile Chimera grid scheme.
- 4). This study improved the accuracy of the Chimera inter-grid communication when simulating supersonic or hypersonic flows with strong discontinuities due to applying the AMR algorithm.
- 5). The combination of the versatile geometric packing ability of the Chimera grid method and the efficient computation ability of the AMR algorithm makes the AMRC scheme capable of simulating multiple and moving body flow problem with strong discontinuities using moderate computational resources.

The highlights of this scheme are as below:

- 1) Most of the tasks associated with the Chimera grid scheme, such as hole cutting, recipient-donor stencil identification, and inter-grid communication, are performed in a fully automated mode. The domain de-composition and Chimera hole cutting curve or buffer cell is defined by the user when generating sub-domains and constructing the overall Chimera grid system.
- 2) Each initial coarse Chimera overset sub-domain is refined in high-gradient regions by applying the AMR algorithm according to the flow solution. The solution features, such as shock waves, contact surfaces, boundary layers, and shear layers, can be easily resolved.
- 3) Due to the fact that the AMR algorithm employs local refinement strategies, about a further 5~10 percent (depending on the problem) of computational resources are required than those are needed by a non-adaptive calculation method. Furthermore, a couple of methods have been used to improve the efficiency of the Chimera grid hole cutting and point searching.
- 4) Integrating the Chimera overset grid method with the AMR algorithm makes the AMRC a versatile and high-resolution scheme for modelling shock wave phenomena and steady and unsteady supersonic and hypersonic flows.

This scheme has been used to solve time-dependent and steady state shock hydrodynamic problems and to simulate moving multiple body flows at supersonic and hypersonic speed. Based on the simulations, applications, comparisons and analyses, the following conclusions can be drawn:

- 1) The AMR algorithm has been successfully employed in the Chimera grid system. This greatly enhances the solution resolution inside high gradient regions and extends the application of both the AMR scheme and the Chimera grid method.
- 2) The simulation results presented in this thesis show the flexibility and capability of the Chimera grid method in modelling multiple body problems, and the reasonable accuracy.
- 3) The technique is very powerful and spans a wide range of application areas. Some of the examples presented within this thesis have demonstrated that it is possible to simulate complex geometry flows and multiple moving body flows at high speed.
- 4) To simulate a supersonic flow problem with shock waves, a conservative Chimera grid interface algorithm is likely to be important. However the grid adaptation according to the flow solution is more important, since in a supersonic flow bow wave, laminar boundary layer, and laminar boundary layer separation occur often. Most importantly, grid refinement is a more efficient way for the Chimera common boundary treatment.
- 5) Since Quirk's AMR code was designed for simulating transient shock hydrodynamic flows, the rate of convergence is very slow for simulating steady state viscous flow problems. It is possible to restrict the viscous calculation in small near body regions by using the Chimera grid method.

8.2 Suggestions for Future Research Work

The results obtained have been very encouraging and have given a basis for future study and further applications. There are a number of potential areas in which this work

could be either improved or extended. The followings are suggestions for future research work

1. Further extensions into three dimensions

Although our work to date is in two space dimensions, it is recognised that axis-symmetric and three-dimensional flows are a natural progression. The real world of fluid dynamics is mainly a three-dimensional world. Both multiple component grids and mesh refinement will become more essential in three-dimensional calculations.

The Chimera overset grid method has been widely used in simulating three-dimensional flows. There are great possibilities for extending this scheme in three dimensions. In fact, during the development of our Chimera overset scheme, we have considered the extension to three dimensions. Most of the procedures associated with the Chimera overset grid scheme in the AMRC code are already, or can easily be after some minor modifications, extended to three dimensions. Currently, the array for z direction coordinates is set as $z(1,1)$, we can simply adjust it to be the same size as the x and y arrays for a three-dimensional problem. The grid data input and output procedures are also easy to extend to three dimensions. Two major modifications are as below,

- The Chimera hole cell flagging procedure needs to be totally replaced. Currently we employ the 'Non-Exterior Winding Role' to identify whether a cell is within or outside of a closed curve. This is easy to code and well suits a 2D problem. Unfortunately, it is not applicable to 3D problems. The algorithm for locating hole points presented in reference¹ is generalised to three dimensions. It is recommended that this algorithm be employed.
- The inter-grid communication procedure needs to be modified. For a 3D problem, using tri-linear interpolation is a popular way to carry out the transfer of flow information.

So far, the majority of the research work on the structured AMR algorithm presented in the literature is mainly concerned with two-dimensional flows. However, there is no inherent reason why the AMR algorithm cannot be extended to three dimensions. In fact, a

number of 3D applications of the AMR algorithm have been reported in the literature¹¹⁵⁻¹¹⁶. James Quirk addressed in his thesis¹⁸ that the relatively poor geometric packing capability of the AMR code is a limitation. After integrating the Chimera overset grid scheme with the AMR algorithm, this is no longer a problem. It is the opinion of the author that the primary issue in extending the AMRC code to three dimensions relates to the original AMR code itself, since the AMR algorithm requires a recursive process which is not explicitly supported in Fortran-77. In the AMR code, a number of procedures have been implemented recursively; therefore the code is not easy to support, modify or improve. As this is the case, rewriting the AMR code in C would probably consume less effort than trying to modify it to handle three dimensions.

Nevertheless, it is feasible and necessary to extend all the algorithms to cope with three dimensions; it would probably be possible to implement a general piece of code with the number of dimensions as an input.

2. Parallelisation

There are also opportunities to extend this research work using parallel computing technology.

Firstly, for a large simulation, if the sizes of different sub-domains are of the same scale, the multi-domain feature of the Chimera overset grid method can be adopted to take advantage of multi-processor parallel computers. In the AMRC code, every domain is treated as a separate entity. Different overset domains are solved separately no matter whether it is an adaptive calculation that is being used or not. Moreover, most procedures associated with the Chimera overset method, such as the flagging of hole cells, the identification of recipient-donor stencils and interpolation, also can be extracted and employ parallel computing techniques. A few references exist on research work of this type⁴³⁻⁴⁸.

Secondly, although the AMR algorithm is quite involved, its block-structured nature is also inherently parallel. In fact, in reference¹²⁹ a technique describing how Quirk's AMR

algorithm may be made parallel is presented. References^{125, 130} also contain further parallel adaptive mesh refinement algorithm information.

Despite the flexibility and the efficiency of the AMRC scheme, it is clear that the long-term usefulness of the AMRC scheme, especially after having been extended to deal with three dimensions, is undoubtedly dependent on its parallelization.

3. Turbulence model

Currently the potential of this scheme is still greatly limited due to the lack of a turbulence model. The most important feature of the Chimera overset grid method is its ability to model complex configuration and moving body flows. However, to simulate such a practical problem often requires the accurate prediction of transitional or turbulent boundary layers and turbulent wake flows. This is especially true when analysing supersonic or hypersonic moving body flows, such as the flow fields around modern aerospace vehicles, store separation and free flight projectiles. It is obviously worth adding a turbulence model into the AMRC scheme. This will significantly extend its ability to simulate practical multiple moving body flows or geometrically complicated flows.

4. Re-writing the scheme in C or Fortran 90

There are a number of reasons for rewriting this scheme in C or Fortran 90.

- Firstly, the main procedures associated with the AMR algorithm need to be implemented recursively at different grid levels and time steps; unfortunately, recursive processes are not explicitly supported in Fortran 77. For this reason, Quirk's AMR code is quite complex and difficult to enhance. Both C and Fortran 90 allow for recursive processes, thus rewriting the code in one of these languages would be beneficial for both easier coding and for improving the scheme.
- Secondly, both the Chimera grid scheme and the AMR algorithm require the precise organisation and management of data in different domains or upon different levels.

the computational grid structure used by the AMR algorithm is dynamic during a calculation the grid adapts to the evolving flow solution. Coding this scheme in C or Fortran 90 would allow for memory usage to be easily controlled through dynamic memory allocation (when adding or removing grid points).

- Finally, it is especially true for extending into 3D that rewriting this scheme in C would simplify both the coding work and organising the data structures.

In order to provide easy access to its solving power, a user-friendly interface should be included.

The author believes that the research work presented in this thesis is basic to multiple and/or moving body CFD research in an adaptive environment. The above suggestions for future work are not completely exhaustive; as CFD methodology and computational technology develop year by year, it will be possible to find many new and interesting projects in this area. The author is grateful to have had the opportunity to work in this field which he found to be very interesting. He has, therefore, done his utmost to present the work in this thesis to the best of his abilities.

References

- [1]. Steger J L, Dougherty F C and Benek J A. "A Chimera grid scheme". In: ASME mini-symposium on advances in grid generation, June, 1983, Houston Texas, USA, 59-69
- [2]. Benek J A, Steger J L and Dougherty F C. "A flexible grid embedding technique with application to the Euler equations". AIAA 83-1944, 1983
- [3]. Benek J A, Buning P G and Steger J L, "A 3-D Chimera grid embedding technique". AIAA 85-1523, 1985.
- [4]. Dougherty F C, "Development of a Chimera grid scheme with applications to unsteady problems". PhD Thesis, Stanford University, 1985.
- [5]. Benek J A, Steger J L, Dougherty F C and Buning P G, "Chimera: a grid-embedding technique". AEDC-TR-85-64, April 1986.
- [6]. Benek J A, Tonegan T L and Suhs N E, "Extended Chimera grid embedding scheme with application to viscous flows". AIAA 87-1126, 1987.
- [7]. Berger M J, "Adaptive mesh refinement for hyperbolic partial differential equations", PhD Thesis, Stanford University, 1982.
- [8]. Berger M J and Oliger J E, "Adaptive mesh refinement for hyperbolic partial differential equations", Journal of Computational Physics, 53: 484-512, 1984.
- [9]. Berger M J and Jameson A, "Automatic adaptive grid refinement for the Euler equations", AIAA Journal, 23(4): 561-568, April 1985.
- [10]. Berger M J and Collela P, "Local adaptive mesh refinement for shock hydrodynamics", Journal of Computational Physics, 82(1): 64-84, 1989.
- [11]. Berger M J and Jameson A, "Automatic adaptive grid refinement for the Euler equations", AIAA Journal, 23(4):561-568, April 1985.
- [12]. Berger M J, "Data Structures for adaptive grid generation", SIAM Journal of Sci. Statist. Comput., 7(3) 904-916, July 1986.

References

- [13]. Berger M J, Saltzman and Welcome M, "A three dimensional adaptive mesh refinement for hyperbolic conservation laws", SIAM Journal of Sci. Statist. Comput., 15(1): 127-138, 1994.
- [14]. Bell J, Berger M et al, "Three dimensional Adaptive Mesh Refinement for hyperbolic conservation laws", SIAM Journal of Scientific Computing, 15(1):127-138, January 1994.
- [15]. Berger M J and Leveque R L, "Adaptive mesh refinement using wave-propagation algorithm for hyperbolic systems", SIAM Journal of Numer. Anal., 35(6): 2298-2316, 1998.
- [16]. Berger M J and Bokhari S H, "A partitioning strategy for nonuniform problems on multiprocessors", IEEE Transactions on Computers, 36(5): 570-580, May 1987.
- [17]. Berger M J and Saltzman J S, "AMR on the CM-2", Applied Numerical Mathematics, 14, 1994.
- [18]. Quirk J J, "An adaptive grid algorithm for computational shock hydrodynamics". PhD Thesis, Cranfield University, 1991.
- [19]. Kao K H, Liou M S and Chow C Y, "Grid adaptation using composite overlapping meshes". AIAA 93-3389-CP, 1993.
- [20]. Chawla K and Banks D W, "Tracking flow features using overset grids ". AIAA 93-0197, 1993.
- [21]. Meakin R L, "An efficient means of adaptive refinement within systems of overset grids". AIAA 95-1722-CP, 1995.
- [22]. Meakin R L, "On adaptive refinement and overset structured grids". AIAA 97-1858, 1997.
- [23]. Eberhardt S and Daganoff D, "Overset grids in compressible flow". AIAA 85-1542, 1985.
- [24]. Wang Y. "A three-dimensional inviscid flow solver in Chimera flow simulation". AIAA 93-0190, 1993.
- [25]. Meakin R L, "A new method for establishing inter-grid communication among system of overset grids". AIAA 91-1586, 1991.

References

- [26]. Chiu I T and Meakin R, "On automating domain connectivity for overset grids". AIAA 95-0854, 1995.
- [27]. Holst T L, "A Chimera donor cell search algorithm suitable for solving the full potential equation". AIAA 97-2259, 1997.
- [28]. Wang Z J, Parthasarathy V and Hariharan N, "A fully automated Chimera methodology for multiple moving body problems". AIAA 98-0217, 1998.
- [29]. Wang Z J, "A conservative overlapped (Chimera) grid algorithm for multiple moving body flows". AIAA 96-0823, 1996.
- [30]. Wagner C, Davis D, et al, "Computation of free surface flows using a hybrid multiblock/Chimera approach". AIAA 98-0228, 1998.
- [31]. Rock S G, Habchi S D et al, "A computational methodology for the rapid simulation of jettisoned aircraft canopy trajectories". AIAA 97-0169, 1997.
- [32]. Chan W M and Buning P G, "Zipper grids for force and moment computation on overset grids", AIAA 95-1681-CP, 1995.
- [33]. Lee S, Park M, et al, "A new automated Chimera method for the prediction of store trajectory". AIAA 99-3131, 1999.
- [34]. Brown D L, Henshaw W D and Quinlan D J, "Overture: Object-oriented tools for overset grid applications". AIAA 99-3130, 1999.
- [35]. Chan W M and Meakin R L, "Advances towards automatic surface domain decomposition and grid generation for overset grids", AIAA 97-1979, 1997.
- [36]. Chan W M and Gomez III R J, "Advances in automatic overset grid generation around surface discontinuities", AIAA 99-3303, 1999.
- [37]. Lutton M J and Visbal M R, "Time-accurate validation of the Chimera method for unsteady vertical flows". AIAA 96-2077, 1996.
- [38]. Holst T L, "Fast transonic aerodynamic analysis suitable for the preliminary design environment". AIAA 99-3133, 1999.
- [39]. Rock S G and Habchi S D, "Validation of an automated Chimera methodology for aircraft escape system analysis". AIAA 98-0767, 1998.

- [40]. Wang Z J and Przekwas A J, "Unsteady flow computation using moving grid with mesh enrichment". AIAA 94-0285, 1994.
- [41]. Wang Z J, Mitchell C R et al, "An enriched hybrid grid approach for unsteady multi-body flow computation". AIAA 94-1916, 1994.
- [42]. Ingram C L, McRae D S. and Benson R A, "Time-accurate simulation of a self-excited oscillatory supersonic external flow with a multi-block solution-adaptive mesh algorithm". AIAA 93-3387-CP, 1993.
- [43]. Meakin R L and Wissink A M, "Unsteady aerodynamic simulation of static and moving bodies using scalable computers", AIAA 99-3302. 1999.
- [44]. Weeratunga S and Chawla K, "Overset grid applications on distributed memory MIMD computers". AIAA 95-0573, 1995.
- [45]. Weeratunga S K, Barszcz E and Chawla K. "Moving body overset grid applications on distributed memory MIMD computers". AIAA 95-1751-CP, 1995.
- [46]. Barszcz E, Weeratunga S K and Meakin R L, "Dynamic overset grid communication on distributed memory parallel processors", AIAA 93-3311-CP, 1993.
- [47]. Smith M H and Pallis J M, "MEDUSA-an overset grid flow solver for network-based parallel computer systems", AIAA93-3312-CP, 1993.
- [48]. Stoll P, Gerlinger P and Bruggemann D, "Domain decomposition for an implicit LU-SGS scheme using overlapping grids", AIAA 97-1896, 1997.
- [49]. Wang Z J and Yang H Q, "A unified conservative zonal interface treatment for arbitrarily patched and overlapped grids". AIAA 94-0320, 1994.
- [50]. Wang Z J, "A fully conservative interface algorithm for overlapped grids". Journal of Computational Physics, Volume 122, 96-109, 1995.
- [51]. Moon Y J and Liou M S, "Conservative treatment of boundary interfaces for overlaid multi-level grid adaptations". AIAA 89-1980-CP, 1989.
- [52]. Wang Z J, Yang H Q and Przekwas A J, "Implicit conservative interfacing for 3-D overlapped Chimera grids". AIAA 95-1683, 1995.
- [53]. Wang Z J, "A fully conservative structured/unstructured Chimera grid scheme". AIAA 95-0671, 1995.

References

- [54]. Wang Z J, Buning P et al, "Critical evaluation of conservative and non-conservative interface treatment for Chimera grids". AIAA 95-0077. 1995.
- [55]. Wang Z J, Hariharan N and Chen R, "Recent developments on the conservation property of Chimera". AIAA 98-0216, 1998.
- [56]. Tamura Y, Fujii K, "Conservation law for moving and transformed grids". AIAA 93-3365-CP, 1993.
- [57]. Hubbard B and Chen H C, "A Chimera scheme for incompressible viscous flows with application to submarine hydrodynamics". AIAA 94-2210, 1994.
- [58]. Paraschivoiu M, Cai X C et al, "Multi-domain multi-model formulation for compressible flows: Conservative interface coupling and parallel implicit solvers for 3D unstructured meshes". AIAA 99-0784, 1999.
- [59]. Rogers S E and Pulliam T H, "Accuracy enhancements for overset grids using a defect correction approach". AIAA 94-0523, 1994.
- [60]. Rai M M, "A conservative treatment of zonal boundaries for Euler equation calculations". Journal of Computational Physics, Vol. 62, pp. 472-503, 1986.
- [61]. Rai M M, "An Implicit, conservative, zonal boundary scheme for Euler equation calculations". NACA CR 3865, 1985.
- [62]. Berger M J, "On conservation at grid interfaces", SIAM Journal of Numerical Analysis, 24(5): 967-984, October 1987.
- [63]. Meakin R L, "On the spatial and temporal accuracy of overset grid methods for moving body problems". AIAA 94-1925, 1994.
- [64]. Holst T L, "Numerical solution of the full potential equation using a Chimera grid approach", NASA TM 110360, July, 1995.
- [65]. Holst T L, "Full potential equation solutions using a Chimera grid approach", AIAA 96-2423-CP, 1996.
- [66]. Kiris C, Rogers S E, Kwak D and Chang I D, "Computation of incompressible viscous flow through artificial heart devices with boundaries", Proceedings of an AMS-IMS-SIAM joint research conference, edited by Cheer A Y and Van Dam C P, American Mathematical society, providence, R. I., 1991, pp.237-247.

References

- [67]. Sakata T and Jameson A, " Multi-body flow field calculations with overlapping mesh method", AIAA 89-2179-CP. 1989.
- [68]. Nirschl H Dwyer H A and Denk V, " A Chimera grid scheme for the calculation of particle flows". AIAA 94-0519, 1994.
- [69]. Chen K H and Bulzan D L, "Numerical study of cooling system design in a high pressure combustion rig with Chimera grids", AIAA 96-3126, 1996.
- [70]. Ostrander M J, Cedar R D, "Analysis of a high bypass ratio engine installation using the Chimera domain decomposition technique". AIAA 93-1808, 1993.
- [71]. Kuan J H, Lin C K and Shih L L, "Analysis of three-dimensional turbomachinery flows on Chimera grid scheme using a time accurate implicit Euler solver". AIAA 93-0334, 1993.
- [72]. Stephens M A, Rimlinger M J et al, "Chimera grids in the simulation of turbine-blade-coolant passages". AIAA 93-2559, 1993.
- [73]. Riner W C, Jolly B A and Brock J, "Verification of transonic Euler solutions of an F-16 aircraft with stores using a Chimera grid scheme". AIAA 95-2288, 1995.
- [74]. Hariharan N, Wang Z J and Buning P G, "Application of conservative Chimera methodology in finite difference settings". AIAA 97-0627, 1997.
- [75]. Parks S J, Buning P G et al, "Collar grids for intersecting geometric components within the Chimera overlapped grid scheme". AIAA 91-1587, 1991.
- [76]. Ostrander M J and Cedar R D, "Analysis of a high bypass ratio engine installation using the Chimera domain decomposition technique". AIAA 96-1808, 1996.
- [77]. Yungster S and Chen K H, "Application of Chimera grid scheme to combustor flowfields at all speeds". AIAA 96-0731, 1996.
- [78]. Rogers S E, Wiltberger N L and Kwak D, "Efficient simulation of incompressible viscous flow over single and multi-element airfoils", AIAA 92-0405, 1992.
- [79]. Rogers S E, "Progress in high-lift aerodynamic calculations", AIAA 93-0194, 1993.
- [80]. Rogers S E, Menter F et al, "A comparison of turbulence models in computing multi-element airfoil flows", AIAA 94-0291, 1994

References

- [81]. Renze K J, Buning P G et al, "A comparative study of turbulence models for overset grids". AIAA 92-0437, 1992.
- [82]. Rock S G and Tramel R W, "A three-dimensional thermo-chemical non-equilibrium Chimera flow solver for moving grids, part I: steady state". AIAA 95-0151, 1995.
- [83]. Duque E P and Srinivasan G R "Numerical simulation of a hovering rotor using embedded grids", In: the AHS 48th annual forum and technology display, 429-445 Washington, D.C., June3-5, 1992.
- [84]. Thoms R D, Jordan J K, "Investigation of multiple-body trajectory prediction using time-accurate computational fluid dynamics". AIAA 95-1870, 1995.
- [85]. Tuncer I H, "A 2-D unsteady Navier-Stokes solution method with overlapping /overset moving grids". AIAA 96-0822, 1996.
- [86]. Blake D C and Buter T A, "Overset grid methods applied to a finite-volume time-domain Maxwell equation solver". AIAA 96-2338, 1996.
- [87]. Boniface J C, Guillen P et al, "Development of a Chimera unsteady method for the numerical simulation of rotorcraft flowfields". AIAA 98-0421, 1998.
- [88]. Meakin R L, "Moving body overset grid methods for complete aircraft tiltrotor simulations", AIAA-93-3350-CP, 1993.
- [89]. Sahu J Nietubicz C J, "Application of Chimera technique to projectiles in relative motion", AIAA 93-3632-CP, 1993.
- [90]. Duque E P, Strawn R C, Ahmad J and Biswas R, "An overset grid Navier-Stokes Kirchhoff-Surface method for rotorcraft aeroacoustic predictions". AIAA 96-0152, 1996.
- [91]. Pearce D G, Stanley S A et al, "Development of a large scale Chimera grid system for the space shuttle launch vehicle". AIAA 93-0533, 1993.
- [92]. Dominik D, Rajagopal K, et al, "Navier-Stokes solution for the space shuttle vehicle using a full-scale high fidelity grid model". AIAA 93-0419, 1993.
- [93]. Dominik D, Wisneski J et al "Grid generation of a high fidelity complex multi-body space shuttle mated vehicle". AIAA 93-0432, 1993.

References

- [94]. Noack R W and Belk D M, "Improved interpolation for viscous overset grids". AIAA 97-0199, 1997.
- [95]. Prewitt N C, Belk D M and Maple R C, "Multiple body trajectory calculations using the Beggar code". AIAA 96-3384-CP, 1996.
- [96]. Hall L H and Parthasarathy V, "Validation of an automated Chimera/6-DOF methodology for multiple moving body problems". AIAA 98-0753, 1998.
- [97]. Meakin R L and Suhs N E, "Unsteady aerodynamic simulation of multiple bodies in relative motion". AIAA 89-1996, 1989.
- [98]. Yen G W and Baysal O, "Dynamic-overlapped-grid simulation of aerodynamically determined relative motion". AIAA 93-3018, 1993.
- [99]. Lijewski L E and Suhs N E, "Time-accurate computational fluid dynamics approach to transonic store separation trajectory prediction". Journal of Aircraft, Volume 31, No. 4, pp. 886-891.
- [100]. Lijewski L and Suhs N, "Chimera-eagle store separation". AIAA 92-4569, 1992.
- [101]. Lijewski L E, "Comparison of transonic store separation trajectory predictions using the PEGASUS/DXEAGLE and BEGGAR codes". AIAA 97-2202, 1997.
- [102]. Cvrlje T, "Unsteady separation of a two-stage hypersonic vehicle". AIAA 99-3412, 1999.
- [103]. Ferry E, Sahu J and Heavey K, "Navier-Stokes computations of sabot discard using a Chimera scheme". AIAA 98-0752, 1998.
- [104]. Edge H L, Sahu J and Heavey K R, "Computational fluid dynamics modeling of submunition separation from missile". AIAA 99-3129, 1999.
- [105]. Li S and Petzold L, "Moving mesh methods with upwinding schemes for time-dependent PDEs", Journal of Computational Physics, 130: 368-377, 1997.
- [106]. Gropp W D, "A test of moving mesh refinement for 2-D scalar hyperbolic problems", SIAM Journal of Sci. Statist. Comput., 1(2): 191-197, June 1980.
- [107]. Gropp W D, "Local uniform mesh refinement with moving grids", SIAM Journal of Sci. Statist. Comput., 8(3): 292-304, May 1987.

References

- [108]. Bayyuk S A, Powell K G and Van Leer B, "A simulation technique for 2-D unsteady inviscid flows around arbitrarily moving and deforming bodies arbitrary geometry". AIAA 93-3391-CP, 1993.
- [109]. Aftosmis M J and Baron J R, "Adaptive grid embedding in nonequilibrium hypersonic flow", AIAA 89-1652, 1989.
- [110]. Davis R L and Dannenhoffer J F, "Adaptive grid embedding Navier-Stokes technique for cascade flows", AIAA 89-0204, 1989.
- [111]. Dannenhoffer J F, "A comparison of adaptive-grid redistribution and embedding for steady transonic flows", AIAA 90-1565, 1990.
- [112]. Davis R L and Dannenhoffer J F, "3-D adaptive grid-embedding Euler technique", AIAA 93-0330, 1993.
- [113]. Davis R L, Choi D and Dannenhoffer J F, "Application of two-equation turbulence models with adaptive grid-embedding procedures", AIAA 94-2716, 1994.
- [114]. Warren G P, "Adaptive grid embedding for the two-dimensional Euler equations", AIAA 90-3049, 1990.
- [115]. Ochs S S and Rajagopalan R G, "An adaptively-refined quadtree grid method for incompressible flows", AIAA 98-0613, 1998.
- [116]. Charlton E F and Powell K G, "An octree solution to conservation-laws over arbitrary regions (OSCAR)", AIAA 97-0198.
- [117]. Moas E and Pageau S S, "Coupling of adaptive and non-adaptive mesh refinement regions using interface elements". AIAA 99-1284, 1999.
- [118]. Dadone A and Palma P D, "An adaptive multigrid upwind solver for compressible viscous flows", AIAA 95-2181, 1995.
- [119]. Lezeau P A, "An adaptive quasi-newton coupled multigrid solver for the solution of steady multiphase flows", PhD Thesis, Cranfield University, 1997
- [120]. Skamarock W, Olinger J E and Street R L, "Adaptive grid refinement for numerical weather prediction", Journal of Computational physics, 80(1): 27-60, 1989.

References

- [121]. Steinhilber E, Modiano D and Colella P, "Computations of unsteady viscous compressible flows using adaptive mesh refinement in curvilinear body-fitted grid systems", AIAA 94-2330, 1994.
- [122]. Jones B J, "The status of shamrock: a 2-dimensional, adaptive mesh refinement (AMR), Eulerian production code", In: Proceedings of new models and numerical codes for shock wave processes in condensed media, 402-412, Oxford, UK, 15-19th, Sep., 1997.
- [123]. Neeman H J, "Autonomous hierarchical adaptive mesh refinement for multiscale simulations", PhD Thesis, University of Illinois at Urbana-Champaign, 1996.
- [124]. Quirk J J, "A parallel adaptive mesh refinement algorithm". NASA/CR 191530, August 1993.
- [125]. Zeeuw D D and Powell K G, "An adaptively refined Cartesian mesh solver for the Euler equations". Journal of Computational Physics, Vol. 104(1): 56-68, 1993.
- [126]. Aftosmis M J Melton J E and Berger M J, "Adaptation and surface modeling for Cartesian mesh method", AIAA-95-1725-CP, 1995.
- [127]. Pember R B, Bell J B et al, "An adaptive Cartesian grid method for unsteady compressible flow in irregular regions", Journal of computational physics, 120(2): 278-304, September 1995.
- [128]. Quirk, J J "A Cartesian grid approach with hierarchical refinement for compressible flows", NASA/CR-194938, 1994.
- [129]. Coirier W J and Powell K G, "A Cartesian cell based approach for adaptively refined solutions of the Euler and Navier-Stokes equations", AIAA 95-0566, 1995.
- [130]. Pember R B, Bell J B and Colella P et al, "An adaptive Cartesian grid method for unsteady compressible flow in irregular regions", Journal of Computational Physics, Vol. 120(2): 278-304, 1995.
- [131]. Roper J J, "An investigation of computational techniques for the prediction of supersonic dynamic flows", PhD Thesis, Cranfield University, 2000.

References

- [132]. Edwards J A, and Roper J J, "Adaptive Moving Body Flow Computations". AIAA 99-0393, 37th AIAA Aerospace Sciences Meeting and Exhibit, January 11-14, 1999, Reno, NV.
- [133]. Roper J J and Edwards J A, "Computation of a dynamically deployed flap employing adaptive body recovery", ICAS 2000 Congress.
- [134]. Thompson J F Soni B K and Weatherill N P(eds)(1998), "Handbook of grid generation", CRC Press LLC
- [135]. "CFD-VIEW User's Manual", Version 5.0.98, CFD Research Corporation, 1998.
- [136]. "FIELDVIEW User's Manual", Intelligent Light, Version 5.5, 1996.
- [137]. Ren T, "Integration of the Chimera grid method with the Adaptive Mesh Refinement Algorithm", Cranfield University, College of Aeronautics report No. NFP 0002, March, 2000.
- [138]. Bryson A E and Gross R W F, "Diffraction of strong shocks, by cones, cylinders, and spheres", Journal of Fluid Mech., Vol.10 (1961), pp1-16.
- [139]. Bendor G, Takayama K and Kawauchi T, " The transition from regular to Mach reflexon and from Mach to regular reflexion in truly non-stationary flows", J. Fluid Mech. Vol.100, pp147-160, 1980.
- [140]. Wada Y and Liou M S, "A flux splitting scheme with high-resolution and robustness for discontinuities", AIAA 94-0083, 1994.
- [141]. Qin N and Ludlow D K, "A cure for anomalies of Osher and AUSM+ schemes for hypersonic viscous flows around swept cylinders". In: proceedings of the 22nd International symposium on shock wakes, 18-23, July, 1999, Imperial college, London, UK, pp635-640.
- [142]. Young V Y C and Yee H C, "Numerical simulation of shock wave diffraction by TVD schemes", AIAA 87-0112. 1987.
- [143]. Mandella M, "Experimental and analytical studies of compressible vortices", PhD thesis, Dept. of applied physics, Stanford University, 1987.
- [144]. Mandella M and Bershader D, "Quantitative study of the compressible vortex: Generation, structure and interaction with airfoils", AIAA 87-0328. 1987.

References

- [145]. Yee H C, "A class of high-resolution explicit and implicit shock-capturing methods", NASA TM-101088, Feb, 1989.
- [146]. Young J M and Yee H C, "Numerical simulation by TVD schemes of complex shock reflections from airfoils at high angle of attack", AIAA 87-0350. 1987.
- [147]. Dolling D S, "Two dimensional shock wave/laminar boundary layer interactions on curved walls", PhD Thesis, University of London, Faculty of Engineering, 1977
- [148]. Hankey W L and Holden M S, "Two-dimensional shock-wave boundary layer interactions in high speed flows", Reports AGARD-AG-203, 1975.
- [149]. Lees L and Reeves B L, "Supersonic separated and reattaching laminar flows. Part I—general theory and application to adiabatic boundary layer/shock wave interactions". AIAA J., Vol. 2, No.11, November 1964, pp. 1907-20.
- [150]. Edney B, "Anomalous heat transfer and pressure distributions on blunt bodies at hypersonic speeds in the presence of an impinging shock", FFA Report 115, Stockholm, Sweden. 1968.
- [151]. Wieting A R and Holden M S, "Experimental study of shock wave interference heating on a cylindrical leading edge at Mach 6 and 8", AIAA 87-1511, 1987.
- [152]. Holden M S, Sweet S et al, "A review of the aerothermal characteristics of laminar, transitional and turbulent shock/shock interaction regions in hypersonic flows", AIAA 98-0899. 1998.
- [153]. Glass C E, "Numerical simulation of low-density shock-wave interactions", NASA/TM-1999-209358, 1999.
- [154]. Stollery J L, "Supersonic aircraft: Msc. Course in aerospace dynamics", lecture notes, Cranfield University college of aeronautics, October 1998.
- [155]. Rainey R W, "Pressure measurement at supersonic speeds on a section of a rectangular wing having an NACA 65-009 Profile". NACA RM L9L16, March 1950.
- [156]. Edwards J A, Ren T & Roper J J, " Progress in the simulation of supersonic flow over a freely oscillating aerofoil", AIAA 2000-3923, 2000.

Appendix A

Bilinear Interpolation

Using the bilinear interpolation method, each cell containing a point at which a function value is to be interpolated must first be transformed to a square, see Figure A-1.

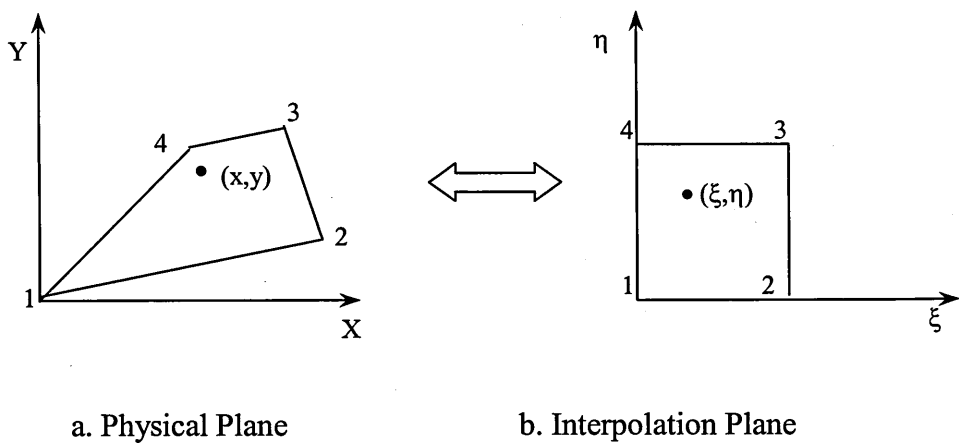


Figure A-1 Isoparametric mapping used for bilinear interpolation

For simplicity, set the square as a unit square, the coordinates (ξ_i, η_i) of each vertex are set as 1 or 0. The relationship of coordinates in the physical domain (x, y) and in the unit square (ξ, η) can be simply defined as:

$$\begin{cases} x = a_1 + a_2\xi + a_3\eta + a_4\xi\eta \\ y = b_1 + b_2\xi + b_3\eta + b_4\xi\eta \end{cases} \tag{A-1}$$

Appendix A Bilinear Interpolation

The coefficients a_i, b_i are easily defined according to the coordinates (x_i, y_i) and (ξ_i, η_i) of each vertex. For example, since

$$\begin{cases} x_1 = a_1 \\ x_2 = a_2 + a_2 \\ x_3 = a_1 + a_2 + a_3 + a_4 \\ x_4 = a_1 + a_3 \end{cases} \quad (\text{A-2})$$

Therefore,

$$\begin{cases} a = x_1 \\ a_2 = x_2 - x_1 \\ a_3 = x_4 - x_1 \\ a_4 = x_3 + x_1 - x_2 - x_4 \end{cases} \quad (\text{A-3})$$

Hence, in equation (A-1), the co-ordinate (x, y) and the coefficient a_i and b_i are known. Using the Newton method, we can get the co-ordinate (ξ, η) in the unit square corresponding to (x, y) .

Rearranging equation (A-1), gives,

$$\begin{cases} F_1 = (a_1 - x) + a_2\xi + a_3\eta + a_4\xi\eta = 0 \\ F_2 = (b_1 - y) + b_2\xi + b_3\eta + b_4\xi\eta = 0 \end{cases} \quad (\text{A-4})$$

According to the Newton method, we have,

$$\xi_{n+1} = \xi_n - \frac{F_i(\xi_n, \eta_n)}{F'_i(\xi_n, \eta_n)}; \quad \eta_{n+1} = \eta_n - \frac{F_{2\eta}(\xi_n, \eta_n)}{F'_{2\eta}(\xi_n, \eta_n)} \quad (\text{A-5})$$

The Jacobian of the transformation can be defined as below,

$$M_{ij} = \begin{bmatrix} \frac{\partial F_1}{\partial \xi} & \frac{\partial F_1}{\partial \eta} \\ \frac{\partial F_2}{\partial \xi} & \frac{\partial F_2}{\partial \eta} \end{bmatrix} = \begin{bmatrix} a_2 + a_4\eta & a_3 + a_4\xi \\ b_2 + b_4\eta & b_3 + b_4\xi \end{bmatrix} \quad (\text{A-6})$$

After figuring out the interpolation coefficients ξ and η for each interpolation cells, the flow solution for the recipient cell can be achieved by the following bilinear interpolation equation.

$$W_R = \sum_{i=1}^4 (-1)^{\xi_i + \eta_i} (\xi + \xi_i - 1)(\eta + \eta_i - 1)W_{Di} \quad (\text{A-7})$$

Appendix B

Moving Shock Relationships

Consider a plane shock moving through a quiescent fluid with speed u_s , and let the two fluid states be referenced by the subscripts $_1$ and $_2$, see figure B-1. By convention a shock Mach number, M_s , is defined in terms of the shock speed and the speed of sound in the quiescent fluid,

$$M_s = \frac{u_s}{a_1} \tag{B-1}$$

For a perfect gas,

$$a = \sqrt{\frac{\gamma p}{\rho}} \tag{B-2}$$

the following relationships which connect the quiescent and disturbed fluid states,

$$\frac{p_2}{p_1} = \frac{2\gamma M_s^2 - (\gamma - 1)}{(\gamma + 1)} \tag{B-3}$$

$$\frac{\rho_2}{\rho_1} = \frac{\left[\left(\frac{\gamma + 1}{\gamma - 1}\right)\right] \frac{p_2}{p_1} + 1}{\left[\left(\frac{\gamma + 1}{\gamma - 1}\right)\right] + \frac{p_2}{p_1}} \tag{B-4}$$

$$\frac{u_2}{a_1} = M_s \left[1 - \frac{(\gamma - 1)M_s^2 + 2}{(\gamma - 1)M_s^2} \right] \tag{B-5}$$

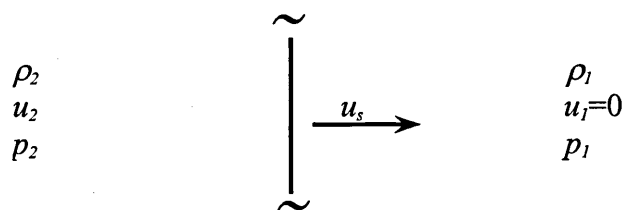


Figure B-1 Disturbed and quiescent fluid states