CRANFIELD UNIVERSITY
LIBRARY

CRANFIELD UNIVERSITY

COLLEGE OF AERONAUTICS
DEPARTMENT OF AEROSPACE SCIENCE

PhD THESIS

Academic Year 1994-5

C Y McNEIL

Efficient Upwind Algorithms for Solution of The Euler and Navier-Stokes Equations

Supervisors: P L Roe
N Qin

October 1995

CRANFIELD UNIVERSITY
LIBRARY

ABSTRACT An efficient three-dimensional structured solver for the Euler and Navier-Stokes equations is developed based on a finite volume upwind algorithm using Roe fluxes. Multigrid and optimal smoothing multi-stage time stepping accelerate convergence. The accuracy of the new solver is demonstrated for inviscid flows in the range $0.675 \leq M \leq 25$. A comparative grid convergence study for transonic turbulent flow about a wing is conducted with the present solver and a scalar dissipation central difference industrial design solver. The upwind solver demonstrates faster grid convergence than the central scheme, producing more consistent estimates of lift, drag and boundary layer parameters. In transonic viscous computations, the upwind scheme with convergence acceleration is over 20 times more efficient than without it. The ability of the upwind solver to compute viscous flows of comparable accuracy to scalar dissipation central schemes on grids of one-quarter the density make it a more accurate, cost effective alternative. In addition, an original convergence acceleration method termed *shock acceleration* is proposed. The method is designed to reduce the errors caused by the shock wave singularity $M \rightarrow 1$, based on a localized treatment of discontinuities. Acceleration models are formulated for an inhomogeneous PDE in one variable. Results for the Roe and Engquist-Osher schemes demonstrate an order of magnitude improvement in the rate of convergence. One of the acceleration models is extended to the quasi one-dimensional Euler equations for duct flow. Results for this case demonstrate a marked increase in convergence with negligible loss in accuracy when the acceleration procedure is applied after the shock has settled in its final cell. Typically, the method saves up to 60% in computational expense. Significantly, the performance gain is entirely at the expense of the error modes associated with discrete shock structure. In view of the success achieved, further development of the method is proposed.

ACKNOWLEDGEMENT This work was largely supported by the Sowerby Research Centre, British Aerospace (Operations) Limited, who kindly granted permission to publish the Company data contained in this thesis. I am deeply indebted to Bob Bignell, former Head of the Aerodynamics Research Department at Sowerby for initiating Company support for this work, and to David Hills, CFD Group Leader, for ensuring its continuance. Without their unerring patronage, it would not have materialized nor seen the distance. I would also like to thank fellow staff at Sowerby, BAe Military Aircraft Division, Warton, and at BAe Dynamics who supplied me with several of the grids used to compute solutions presented in this work; access to the flow solvers and results that were used to compare against my own upwind solver; and, software to post-process much of the data generated. In addition, I acknowledge the Ministry of Defence for permitting the inclusion of experimental data for the M165 configuration. I am grateful particularly for the assistance extended to me by the technical monitor of the data, David Lovell of the DRA at Farnborough.

I owe an overwhelming debt of gratitude to my supervisor Professor Phil Roe for introducing me to the upwind methods upon which this study is based and for the privileged insights that he gave. His guidance and example transformed my interest in hyperbolic equations and their numerical solution into fascination. I would also thank Dr. Tito Toro and Dr. Ning Qin. In succession, each assumed the rôle of supervisor following the departure from Cranfield of Phil Roe. Both gave me invaluable support and advice.

Through the course of this study I have benefitted immeasurably from discussions on CFD and related subjects with a host of friends and colleagues at Cranfield, British Aerospace, CFD conferences, and elsewhere. I thank each of those who broadened my knowledge. In particular, I must mention H-C. Lin, Helen Yee, Efstathos Milonidis and the late Russell Morris. I also thank my brother Andrew who was a great source of encouragement to me. Finally, I offer Janice Tudor my heartfelt gratitude for the immense support, understanding and encouragement that she has given me over the years.

Contents

1 Introduction 8
- 1.1 Background and Objectives . . . 8
- 1.2 Outline of Thesis . . . 10

2 An Explicit Upwind Algorithm 13
- 2.1 Conservation Laws . . . 13
- 2.2 Governing Equations . . . 14
  - 2.2.1 Inviscid and viscous flow approximations . . . 14
  - 2.2.2 The Euler and Navier-Stokes equations . . . 15
- 2.3 The Finite Volume Formulation . . . 18
  - 2.3.1 General background to discretization techniques . . . 18
  - 2.3.2 The governing equations in finite volume form . . . 19
  - 2.3.3 Cell volumes . . . 21
  - 2.3.4 Surface normals . . . 23
- 2.4 Spatial Discretization using Roe Fluxes . . . 25
  - 2.4.1 The Riemann problem . . . 25
  - 2.4.2 Discretization of convective terms in 1-D . . . 27
  - 2.4.3 Discretization of convective terms in 3-D . . . 29
  - 2.4.4 Known problems with upwind solvers. . . . 33
  - 2.4.5 Entropy-satisfying sonic expansions . . . 35
  - 2.4.6 Higher-order spatial accuracy . . . 38
  - 2.4.7 Flux limiting in hypersonic flows . . . 43
  - 2.4.8 Limiter stall . . . 43
  - 2.4.9 A second-order numerical flux function . . . 45
- 2.5 Discretization of Diffusive Terms in 3-D . . . 46
  - 2.5.1 Viscous flux discretization . . . 46
  - 2.5.2 The diffusive coefficients . . . 48
  - 2.5.3 Closure for turbulent flow solutions . . . 49
  - 2.5.4 The Baldwin-Lomax turbulence model . . . 50
- 2.6 Time integration using explicit operators . . . 53
  - 2.6.1 The forward-Euler scheme . . . 54
  - 2.6.2 Runge-Kutta schemes . . . 55
  - 2.6.3 Hybrid multi-stage schemes . . . 55
  - 2.6.4 Timestep definition . . . 57
- 2.7 Boundary Treatment . . . 59
  - 2.7.1 Surface boundaries . . . 60
  - 2.7.2 Farfield boundaries . . . 62
  - 2.7.3 Other boundaries . . . 64
- 2.8 Closing Discussion . . . 65

3 Numerical Results for Inviscid Flow **67**
3.1 Inviscid Flow Solutions 67
3.1.1 Transonic channel flow 67
3.1.2 Supersonic flow over a slender forebody 68
3.1.3 Supersonic & hypersonic flow over a sphere 68
3.2 Comparison with Inviscid Solutions from Other Methods 83
3.2.1 Supersonic flow over a blunt-nose body 83
3.2.2 Supersonic flow over a wing-body 84
3.2.3 Transonic flow over a wing-body 91
3.2.4 Vortex flow over a delta wing 92
3.3 Closing Discussion 94

4 Review of Convergence Acceleration Methods **110**
4.1 Background 110
4.2 Geometric and Algebraic Subspace Methods 111
4.2.1 The Multigrid method 113
4.2.2 Krylov methods 120
4.3 Newton-like Methods 124
4.3.1 Sparse finite difference Newton method 126
4.3.2 Sparse quasi-Newton method 126
4.4 Residual Smoothing 127
4.4.1 Explicit residual smoothing 129
4.4.2 Implicit residual smoothing 129
4.4.3 Implicit-explicit residual smoothing 132
4.4.4 Upwind implicit residual smoothing 134
4.5 Optimal Smoothing Multi-stage Time Stepping Schemes 136
4.5.1 The van Leer–Tai–Powell optimization 137
4.5.2 The Lallemand optimization 141
4.5.3 Optimal schemes for residual smoothing 141
4.5.4 Optimal hybrid multi-stage schemes 142
4.6 Time Inclining 142
4.7 Local Preconditioning 146
4.7.1 Choices for time stepping 147
4.7.2 Characteristic time-stepping 148
4.8 Canonical Forms of the Euler Equations 153
4.9 Closing Discussion 154

5 Multigrid Acceleration for Viscous Flow **155**
5.1 Multigrid Strategy 155
5.1.1 The smoothing scheme 156
5.1.2 Solution and residual restriction 156
5.1.3 Coarse grid evolution 157
5.1.4 Correction prolongation 158
5.1.5 Cycling strategy 159
5.2 A Grid Convergence Study of Turbulent Flow Solutions 159
5.2.1 Methodology and test case description 160
5.2.2 Coefficient of pressure 160
5.2.3 Lift and drag coefficients 169
5.2.4 Boundary layer velocity profiles 169

5.2.5 Boundary layer parameters 170
5.2.6 Momentum equation budgets 171
5.3 Computational Issues 172
5.3.1 Convergence histories 172
5.3.2 Coding efficiency 173
5.3.3 Computational expense 173
5.4 Closing Discussion 174

6 The Shock Acceleration Method **186**
6.1 Inhomogeneous Scalar Conservation Laws 186
6.1.1 A model equation with slow convergence 186
6.1.2 The analytic solution 187
6.1.3 The numerical solution –the Roe scheme 190
6.1.4 The numerical solution –the Engquist-Osher scheme . . . 192
6.1.5 Shock Acceleration for Roe's scheme 199
6.1.6 Performance of the accelerated Roe Scheme 209
6.1.7 Shock Acceleration for the Engquist-Osher scheme 214
6.1.8 Performance of the accelerated Engquist-Osher scheme . . 216
6.2 Inhomogeneous Systems of Conservation Laws 217
6.2.1 The inhomogeneous Euler equations for duct flow 219
6.2.2 The analytic solution 219
6.2.3 The numerical solution –the Roe scheme 224
6.2.4 Shock Acceleration for Roe's scheme 228
6.2.5 Performance of the accelerated Roe scheme 233
6.3 Closing Discussion 235

7 Conclusions **244**

References **247**

List of Figures

2.1 Computation of cell volume. Diagram showing (a) an individual cell and (b) its breakdown into five constituent tetrahedra used to compute the volume of the complete cell. 22
2.2 Computation of surface normals. Diagram showing (a) the surface area vectors for a complete cell and (b) an individual cell face with the position vectors defining it. 23
2.3 The Riemann problem. (a) The initial states at time $t = 0$ and (b) the solution at time $t > 0$, showing three centred waves and the intermediate states between them. 26
2.4 Procedure for implementing Roe's first-order upwind scheme. . . 29
2.5 Second-order TVD region. 40
2.6 Common flux limiter functions. 41
2.7 The dual volume used in the computation of the diffusive fluxes. 47
2.8 Schematic diagram of boundary cells. Shading denotes fictitious ghost cells. 60

3.1 GAMM channel transonic flow. Grids and isomach contours. . . . 71
3.2 GAMM channel transonic flow. Convergence histories. 72
3.3 Tangent-ogive foreboby grid, $(89 \times 29 \times 9)$. 73
3.4 Tangent-ogive supersonic flowfield solutions. 74
3.5 Tangent-ogive supersonic flow convergence history. 75
3.6 Sphere flow at $M_\infty = 2.0$. Grid and flowfield solutions. 76
3.7 Sphere flow at $M_\infty = 2.0$. Convergence histories. 77
3.8 Sphere flow at $M_\infty = 8.0$. Grid and flowfield solutions. 78
3.9 Sphere flow at $M_\infty = 8.0$. Convergence histories. 79
3.10 Sphere flow at $M_\infty = 25.0$. Grid and flowfield solutions. 80
3.11 Sphere flow at $M_\infty = 25.0$. Plot of Mach number vs. grid point along rays normal to the surface. 81
3.12 Sphere flow at $M_\infty = 25.0$. Convergence histories. 82
3.13 ONERA blunt-nose body medium grid, $(64 \times 32 \times 32)$. 85
3.14 ONERA blunt-nose body supersonic flow. Isomach contours. . . 86
3.15 ONERA blunt-nose body supersonic flow. Isobar contours. 87
3.16 ONERA blunt-nose body supersonic flow. Isomach contours in the crossflow plane. 88
3.17 ONERA blunt-nose body supersonic flow. Isobar contours in the crossflow plane. 88
3.18 ONERA blunt-nose body supersonic flow. Convergence histories. 89
3.19 M165 wing-body C-H grid, $(144 \times 32 \times 64)$. 96
3.20 Supersonic M165 wing-body sectional C_p profiles at 19% and 30% span. 97

3.21 Supersonic M165 wing-body sectional C_p profiles at 49% and 61% span. 98
3.22 Supersonic M165 wing-body sectional C_p profiles at 74% and 86% span. 99
3.23 Supersonic M165 wing-body convergence histories. 100
3.24 Transonic M165 wing-body sectional C_p profiles at 30% and 74% span for several flow solvers. 101
3.25 Comparison of transonic M165 wing-body convergence histories from several flow solvers. 102
3.26 Comparison of transonic M165 wing-body convergence of force coefficients from several flow solvers. 103
3.27 IEPG W1 delta wing C-H grid, $(144 \times 32 \times 48)$. 104
3.28 IEPG W1 delta wing vortex flow. Upper-surface isobar contours. 105
3.29 IEPG W1 delta wing vortex flow. Isobar contours normal to the trailing edge. Inset indicates viewing plane. 106
3.30 IEPG W1 delta wing vortex flow. Blow-up view of isobar contours normal to the trailing edge. Inset indicates viewing plane. 106
3.31 IEPG W1 delta wing vortex flow. Velocity vectors normal to the trailing edge. Inset indicates viewing plane. 107
3.32 IEPG W1 delta wing vortex flow. Sectional C_p profiles. Insets indicate each chord-wise station. 108
3.33 IEPG W1 delta wing vortex flow. Convergence histories. 109

4.1 Common Multigrid schedules. 117
4.2 An example of Full Multigrid (FMG) using V cycles. 119
4.3 Time inclining. Geometric interpretation of the CFL condition. (a) Normal non-inclined computational plane and (b) Inclined computational plane. 143

5.1 ONERA M6 upper surface C-H grids. 162
5.2 ONERA M6 wing upper-surface isobar contours for fine grid. . . 163
5.3 ONERA M6 wing sectional C_p profiles for fine grid. 164
5.4 ONERA M6 wing upper-surface isobar contours for medium grid. 165
5.5 ONERA M6 wing sectional C_p profiles for medium grid. 166
5.6 ONERA M6 wing upper-surface isobar contours for coarse grid. . 167
5.7 ONERA M6 wing sectional C_p profiles for coarse grid. 168
5.8 Comparison of ONERA M6 wing upper-surface isobar contours. Central solution on fine grid, upwind solution on medium grid. . 175
5.9 Comparative grid convergence of lift coefficient for ONERA M6 wing. 175
5.10 Comparative grid convergence of total drag coefficient for ONERA M6 wing. 176
5.11 Comparative grid convergence of pressure drag coefficient for ONERA M6 wing 176
5.12 Comparative grid convergence of skin-friction coefficient for ONERA M6 wing 177
5.13 Location of boundary layer measurement stations on ONERA M6 upper-surface planform relative to computed isobars. 178

5.14 Comparison of velocity profiles computed by central and upwind solvers for ONERA M6 wing. 179
5.15 Comparison of x-momentum budget plots obtained from central and upwind solutions of ONERA M6 wing. 180
5.16 Comparison of convergence histories for ONERA M6 wing from multigrid and single-grid upwind solutions. 181
5.17 Comparison of force coefficient convergence for ONERA M6 wing from multigrid and single-grid upwind solutions. 182
5.18 Comparison of convergence histories for ONERA M6 wing from central solution and two upwind solutions. 183
5.19 Comparison of force coefficient convergence for ONERA M6 wing from central solution and two upwind solutions. 184
5.20 Extracts from Cray Perftraces obtained for central and upwind solvers. 185

6.1 Analytic solutions of the equation $[u + 1/u]_x = 0$. 188
6.2 Analytic solutions of the equation $[u + 1/u]_x = kx/u$. 189
6.3 Flow of information at a shock wave for the model equation using using two different upwind methods. 191
6.4 Generic types of flow. 194
6.5 Two-point shock transfer models. 206
6.6 Comparison of convergence histories of Roe's scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.42$. . 212
6.7 Comparison of convergence histories of Roe's scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$. . 212
6.8 Comparison of convergence histories of Roe's scheme using different shock acceleration procedures for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$. 213
6.9 Comparison of convergence histories of the Engquist-Osher scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.42$. 218
6.10 Comparison of convergence histories of the Engquist-Osher scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$. 218
6.11 Graph of the function defined by equation (6.124). 221
6.12 Various graphs of Mach number against distance along a nozzle. (a) Shockless steady flows and (b) steady flows with a standing shock. 222
6.13 Flow of information at a shock wave for the Euler equations using Roe's method. 230
6.14 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced on shock formation. Grid size=40. 238
6.15 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 1000. Grid size=40. 239
6.16 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 1500. Grid size=40. 240

6.17 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 2000. Grid size=40. 241
6.18 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 5500. Grid size=100. 242
6.19 Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.65$. Accelerated solution commenced at iteration 3000. Grid size=100. 243

1

Introduction

1.1 Background and Objectives

Computational Fluid Dynamics (CFD) is a new method of analysing fluid flow about bodies in motion through fluid media. It is founded on the numerical solution of approximations to the governing field equations. Over the last twenty-five years, its use in aerodynamic prediction has grown at an amazingly rapid pace. Today, CFD methods are routinely used in a variety of fields. These include aircraft design, ship, car and train design, aeroacoustics, geophysics and oil recovery, oceanography, meteorology, astrophysics, the modelling of pollutant dispersal and even in bio-medical research. However, CFD has seen enormous growth in the aerospace sciences. To a large extent this has been facilitated by the increasing availability of powerful computers that have made the solution of large-scale field problems of interest to industrial designers both feasible and affordable.

In the field of CFD algorithm development for aerospace applications, a landmark was reached in 1971 with a paper by Murman and Cole [105]. In that work, a method for solving the Transonic Small-Disturbance Potential (TSP) equation was presented and for the first time, flow in the highly nonlinear transonic regime was predicted. This breakthrough was achieved using a first-order accurate upwind scheme. As supercomputer technology leapt ahead, numerical methods began to emerge that could solve higher level approximations of the governing equations, culminating in solution of the Euler and the Navier-Stokes equations. At the same time, the type and complexity of problems that became tractable grew. In the aerospace industry, there are powerful motivations for growth in technology. The economics of aircraft operation are such that even a small improvement in aerodynamic or propulsive efficiency can yield a substantial savings in operational cost. Furthermore the increasingly competitive nature of the aircraft industry which is driven by potentially massive profit provides manufacturers with a compelling incentive to design more efficient aircraft in the most cost effective manner. For these reasons CFD has become a key aerospace technology that is now used alongside the wind tunnel as a complementary method in the analysis and design of aircraft. In addition, CFD is also used to obtain simulations of flows that cannot be obtained from ground-based facilities. Indeed, the design of the space shuttle has benefitted from CFD simulations. Yet despite the leaps and bounds that have been made in computational methods for simulating fluid flow, aircraft designers still face daunting requirements that cannot adequately be met by existing CFD solvers. There are four basic requirements which all successful CFD algorithms must meet.

The first requirement is accuracy. As CFD methods are used more and more in the preliminary design phase to set bounds on the design space and to assist in

screening candidate configurations for wind tunnel testing, meeting this requirement becomes essential. Regrettably, the accuracy of existing CFD algorithms can often be wanting. Even in the prediction of relatively benign flows such as steady transonic attached flow past wings, there are still major limitations in the accuracy of absolute drag prediction [124]. This is a sobering thought for any who see CFD as a panacea to all aerodynamic problems.

The second requirement for a successful CFD solver is the ability to model flows past complex geometries. Frequently flow simulations about realistic configurations cannot be attempted. This is due to the inability of existing methods to adequately cope with certain types of geometric complexity. Solutions of flow past swept wings are routine but solutions for the same geometries with control-surfaces deployed part-span, are only beginning to be seen for relatively simplistic configurations[98]. Unstructured methods have a clear advantage over structured methods in meeting the requirement for dealing with complex geometries, but even they have some way to go.

The third requirement is efficiency. This encompasses CPU usage as well as memory usage. Both need to be budgeted for since they determine the final turn-around of a given computation. Traditionally, the efficiency of CFD methods was equated directly with CPU/memory usage. However, today efficiency is frequently equated to turn-around which is more representative of the true cost than computer usage alone. For example, a structured grid algorithm may require less CPU time than an unstructured one. However, as the time taken in man-hours to generate an unstructured grid for a realistic configuration is usually less than that needed for an equivalent structured grid, overall efficiency in terms of turnaround will favour the former. The prevalence of fixed-cost design projects necessitates pre-determined budgets. Consequently the efficiency of design tools such as CFD ultimately influence product cost.

The last major requirement is often taken for granted since it is a parameter which cannot be assessed as readily as the preceding requirements. Robustness. It necessitates that a CFD method be applicable to a wide range of configurations and flow conditions. It is a factor which becomes increasingly important in a time-critical design environment. Robustness also promotes confidence in the reliability of CFD methods. Ease of use or user-friendliness are qualities which go a long way to enhance robustness. Without reliability and ease of use, the application of CFD flow codes is severely restricted.

Most current CFD algorithms meet all four of these requirements to varying degrees. The present study is particularly motivated by three of them, namely, accuracy, efficiency and robustness. The outstanding requirement not specifically considered here is the ability to model complex geometries. This is not to say that the methods considered here are not amenable to far more complex geometries than those considered. Rather, that in a study of this nature, the subject matter must be realistically bounded. To meet the demands of accuracy and robustness, the algorithms developed in this study are based on the current generation of upwind schemes which provide accurate solutions to a wide range of flow conditions. These methods, often called *High Resolution* schemes, have several beneficial properties manifest in their construction. They are at least second-order accurate in smooth parts of a solution; they resolve discontinuities sharply without generating spurious numerical oscillations; and, being naturally dissipative they require no free parameters to adjust the numerical dissipation.

The unifying concept behind all such algorithms is their reliance on physically based models which can account for the wave propagation phenomena of hyperbolic systems. In addition they make use of higher-order schemes in smooth and smoothly-varying regions of flow which is supplemented with additional dissipation in regions where steep gradients are present. It is this property of adapting the dissipation selectively to individual flow features based on characteristic principles that distinguishes the modern upwind schemes from the classical shock-capturing schemes. This is not to say that the present upwind methods represent the pinacle of algorithm development. However, they do represent the height of the current generation of structured CFD methods. Their biggest draw back is expense, arising from the relatively sophisticated matrix dissipation models upon which they are founded. This weakness is tackled by the remaining thrust of the present study –the development of methods to accelerate the convergence of upwind flow solvers such that they may compete more favorably with earlier less accurate, less robust but cheaper CFD methods.

In pace with the evolution of CFD methods, convergence acceleration has received increasing interest from numerical analysts, scientists and engineers. The majority of new CFD codes are developed with a view to improved algorithmic efficiency through some form convergence acceleration. The motivation comes from the need to reduce the expense of large-scale computation and to increase the utility of numerical methods so that even larger problems may be tackled in the future. Although the advent of modern digital computers has given a massive impetus to the tractability of large-scale computations, it alone is insufficient to abate the massive costs of the projected CFD simulations of the future. Consequently, major effort is devoted to the development of convergence acceleration methods. In the present study, the interest in convergence acceleration relates to time marching techniques which are routinely used to solve steady-state problems arising from a wide range of CFD aerospace applications. In particular, interest is confined to the convergence acceleration of *explicit* time marching methods for upwind spatial discretizations, appropriate to the solution of initial value and initial boundary value problems posed by the Euler and Navier-Stokes equations. The way in which the present study attempts to develop efficient upwind algorithms for solution of these equations and the contribution that it makes to this field is now outlined.

1.2 Outline of Thesis

This study has two major parts. The first consists of utilizing existing ingredients to develop an efficient three-dimensional CFD solver for inviscid and viscous flows while the second part formulates a novel new approach to accelerate the convergence of upwind solvers. To achieve the latter, the study moves back a step to a more fundamental level to develop a new convergence acceleration technique for a one-dimensional hyperbolic equation in a single variable. Some of the techniques developed for the model scalar equation are then extended to the one-dimensional Euler equations. The various areas covered are now outlined by chapter.

The second chapter is devoted to describing the basic upwind solver, a structured grid, cell-centred, finite volume formulation for solution of the three-

dimensional Euler and Navier-Stokes equations. Some necessary background is explained to place the present algorithm in the context of conservation laws, the particular equations of interest to the study, and numerical discretization techniques. A detailed description of the finite volume formulation is given. The upwind discretization is then presented and details are given of how the algorithm is extended to achieve higher-order accuracy by flux limiting. The known vices and virtues of the chosen spatial discretization are discussed. The numerical modelling of the physically diffusive terms is described and an algebraic turbulence model that is commonly used for closure is detailed. Where appropriate, implementation details are given. Attention is also devoted to the time integration operators. The chapter concludes with a brief description of the boundary condition procedures. Though most of the ingredients of the present upwind solver are well known, the choice and manner of its formulation represents the contribution of this part of the work.

The third chapter presents several numerical solutions to inviscid flow test cases which were used to validate the new solver. These cover a wide range of conditions, including transonic, supersonic and hypersonic inviscid flows. Where possible, the results from the present method are analyzed and compared to results from experimental data and other numerical methods. These results demonstrate the high accuracy that may be achieved from this type of upwind solver, often under extreme flow conditions. The contribution of this chapter lies in validation of the new solver and demonstration of its accuracy and robustness across a significant range of Mach number.

The fourth chapter reviews the principal convergence acceleration methods in use in CFD today for explicit time marching methods. The underlying concepts that unify and distinguish the various approaches to convergence acceleration are discussed at some length. This is done to place in the wider context the particular convergence acceleration methods developed later in the study.

The fifth chapter begins with a description of the convergence acceleration methods developed for the upwind solver of chapter 2. These comprise a multigrid algorithm used in conjunction with optimal multi-stage time stepping schemes. The combined components of the present accelerated upwind Navier-Stokes solver have not previously been synergised for a 3-D upwind solver following the present formulation. A comparative grid convergence study for high Reynolds number turbulent flow solutions over a wing is undertaken in which numerical results from the accelerated algorithm are compared to ones obtained from an established industrial design code based on a central difference algorithm using scalar dissipation. The accuracy, reliability and efficiency of the two flow solvers which are representative of two major classes of CFD algorithm are comprehensively analyzed and assessed. The findings of the comparative study corroborate and extend those of earlier studies.

The sixth chapter proposes an original method of accelerating nonlinear inhomogeneous hyperbolic equations in one variable, based on a localized treatment of the discrete shock structure. We have called the method *shock acceleration*, as it tends to accelerate the motion of the shock wave to its terminal position. This is achieved by damping the error modes associated with the movement of discontinuities. Initially a model hyperbolic equation is examined and its analytic solution described. Then two standard upwind methods, the Roe scheme and the Engquist-Osher scheme, are applied to the inhomogeneous problem. With

the baseline methods in place, shock acceleration models are formulated. To prevent departure solutions arising in instances when the acceleration procedure is applied before a solution has attained its basin of attraction and, when a shock is moving from cell to cell, admissibility criteria have been developed to screen the accelerated solution updates. However, some of the models do permit acceleration to be maintained while a shock is in the process of transferring from one cell to the next. Numerical results are presented for the accelerated schemes which demonstrate an order of magnitude improvement in the number of iterations required to achieve convergence, with no adverse effect on accuracy and low computational overhead.

Moving on to systems of equations, an extension of Roe's method to the quasi one-dimensional Euler equations for duct flow is detailed, based on source-term splitting. One of the shock acceleration methods developed in the scalar study is extended to these equations for which numerical results are obtained. With the particular acceleration model used for the Euler equations, it is not possible to accelerate convergence prior to the shock wave attaining its terminal cell. However, the feasibility of the new method is demonstrated for cases in which the acceleration procedure is applied after the shock wave has settled in its terminal cell. The saving that the method achieves in computational efficiency is quantified. Since convergence acceleration is achieved at the expense of error modes associated with the $M \rightarrow 1$ singularity of discrete shock structure, the method is unique. The formulation of shock acceleration methods and the demonstration of their feasibility represent the major contribution of the present work. The final chapter concludes the study by summarizing the main findings and briefly discussing future work.

2

An Explicit Upwind Algorithm

In this chapter, the background of the present numerical algorithm is discussed and details are given of where it rests within the broader field. The chapter begins by considering the important physical principles upon which conservative schemes are based, before moving on to discuss the governing equations in terms of the implicit approximations contained in the Euler and Reynolds Averaged Navier-Stokes equations. Then the mathematical formulation is presented. Numerical discretization techniques are considered and the necessary theory needed to understand some of the major developments in modern shock capturing schemes is discussed. In particular, upwind methods based on characteristic or wave decomposition techniques are discussed, and issues such as entropy satisfaction; the property of monotonicity in the presence of discontinuous solutions (TVD theory); the extension of first-order schemes to higher-order; and the formulation of explicit temporal operators are covered. Where it is instructive, implementation details are mentioned and other necessary information such as closure for turbulent flow solutions and boundary treatment is given.

2.1 Conservation Laws

Conservation laws are of considerable importance because of their frequent occurrence in physical problems where they arise naturally from integral formulations of quantities that are conserved. In the abstract sense, they arise from the modelling of physical processes which involves the following three steps[94]:

1. The appropriate physical balance laws are derived for each of the m physical quantities $q_{1,\ldots,}q_m$ in state space where $q(x_j,t) = [q_1,\ldots,q_m]^T$ for $x_j \in R^N$ $(N = 1, 2, or\, 3)$ and $t \geq 0$. The state space arises because physical quantities such as the density or total energy should always be positive;
2. The physical balance laws contain flux functions and they may also contain source terms. The flux functions are idealized through prescribing nonlinear functions $F_j(q)$ that provide a mapping in state space, while any source terms are represented by smooth functions $S_j(q,x,t)$ of the state variables. In this idealization, detailed microscopic effects such as diffusion and dissipation are ignored; and,
3. A generalized version of the principle of virtual work is applied to the resulting system.

The formal result of applying the above three steps is that the m physical quantities q define a weak solution to an $m \times m$ system of conservation laws

that permit the integrand to be discontinuous. In problems of gas dynamics, the existence of weak solutions ensures the admission of physical phenomenon such as shock waves.

2.2 Governing Equations

2.2.1 Inviscid and viscous flow approximations

The Euler equations describe inviscid, adiabatic continuum flows. They are a sub-set of the Navier-Stokes equations, representing non-viscous and non-heat conducting flows and they provide an approximation of the governing equations of gas dynamics for high Reynolds Number ($1/Re \rightarrow 0$), convection dominated flows in the absence of separation. They represent a first-order system of Partial Differential Equations (PDEs) which are hyperbolic in nature in the case of time-dependent (unsteady) flows. They may be written in conservative, non-conservative or characteristic form. In the present study, the conservation form is used. This is done by expressing the vector of dependent variables (the state vector) in terms of density, the components of momentum and the total energy. In the theoretical context, the form chosen is not of issue since one formulation may be obtained from either of the other two. However, from a numerical point of view, these formulations are not equivalent and the issue here arises from the need for the numerical scheme to be consistent not only with the differential equations, but also with the jump relations over a discontinuity. Since the Euler equations admit discontinuous solutions, the conservative form of the equations is preferred over the non-conservative at least in instances in which shock waves are present in the solution. Furthermore, the integral form of the conservation laws applied to an infinitesimal control volume around a discontinuity has been shown to reduce to the Rankine-Hugoniot relations [88].

The Euler equations may also admit solutions that are mathematically correct but physically implausible. For instance, expansion shocks in which the jump in entropy is negative are valid Euler solutions representing reversible transformations in the absence of heat transfer. To ensure that such solutions –which contravene the Second law of Thermodynamics– are not admitted, some form of constraint or condition on the solution is necessary. To this end, Lax [83] has shown that through the addition of a dissipative mechanism to the inviscid flow model and in the limit of vanishing viscosity, the solution is guaranteed to attain the solution that represents correctly the physical fluid behavior. This is termed an entropy condition.

Turning to the viscous flow approximations, the equation set considered here is valid for uniform, homogeneous fluids without mass diffusion and without chemical reactions. Furthermore, body forces, defined as those acting at a distance from the volumetric mass of the fluid element under consideration (e.g. gravitational, electro-magnetic etc.), are assumed to be negligible. In addition, the fluid is considered to be a continuum –an assumption that is generally valid for most gases and liquids such as those found in fluid dynamics. Finally, the study is restricted to Newtonian fluids, that is continuum flows in which frictional stresses are proportional to the fluids time rate of strain or velocity gradients. For this class of fluids, it is possible to formulate a general deformation law which relates

the stress tensor to the pressure and velocity components. As discussed above, inviscid flow approximations are convection dominated and hence they contain first-order nonlinear terms, being hyperbolic in character for Mach numbers, $\dot{M} > 1$. For processes driven purely by diffusion, the governing PDEs admit unsteady solutions that are of parabolic nature and steady-state solutions that are of elliptic nature. Consequently, the Navier-Stokes equations are of mixed type with their nature dependant upon the given flow conditions. In high Reynolds Number flows, in which viscosity is negligible, they reduce to the Euler equations, whereas for low Reynolds Number viscous dominated flows in which the convective terms are negligible, they reduce to the Stokes equations that describe isotropic diffusion processes.

The Navier-Stokes equations are valid for laminar and turbulent flows. In reality, a flow will remain laminar up to a critical value of Reynolds number above which it can transition to turbulent flow. Laminar flow is well ordered in nature whereas turbulent flow is characterized by the appearance of fluctuations in velocity, pressure, density and temperature around mean values. As these fluctuations are of a statistical nature, they cannot be described in a deterministic way. However, they may be computed numerically in Direct Numerical Simulations (DNS), or by using the Large Eddy Simulation (LES) approach in which only small scale turbulent fluctuations are modelled while the larger scale ones are computed directly. Though there is little doubt that DNS and LES will become increasingly important in the future, at present they are still far from being applicable to practical industrial applications due to their massive computational requirements. In contrast to that approach, the present study is based on a lower level approximation, whereby the full equation set is averaged out in time, over the turbulent fluctuations. This leads to the Reynolds-averaged Navier-Stokes equations which additionally require closure by way of semi-empirical information. This time-averaging removes the influence of the turbulent fluctuations though it does not destroy the time-dependence of other phenomena which have time scales distinct from those of the turbulence.

Shear stress and flow separation are two major ramifications of viscous flow. Shear stress is the cause of skin friction drag while flow separation can be a significant source of pressure drag. As drag is of great importance to the aerodynamic prediction of fluid flow, the significance of the Navier-Stokes equations becomes more apparent. (Other approximations to the Navier-Stokes equations, such as the Thin Layer equations and the Parabolised Navier-Stokes equations are not considered in the present study).

The inviscid and viscous flow approximations relevant to the current work have been discussed and the intrinsic limitations of these approximations have been identified. The potential difficulties with which a numerical method should be expected to cope have also been considered. With this background in mind, the mathematical formulation is now presented.

2.2.2 The Euler and Navier-Stokes equations

The Navier-Stokes equations are derived by applying Newton's Second law either to an infinitesimally small moving fluid element or to a small control volume, under the action of both external pressure forces and stresses. Additionally, the net flux of heat into the element must be taken into account in deriving the

energy equation. By the same approach, the Euler equations may be obtained, this time by considering only external pressure forces. (The derivation of both equation sets is standard though laborious and given in several texts of fluid dynamic such as that by Anderson[2]. As such it is not repeated here). In modern parlance, the conservation laws for mass, momentum and energy make up the Euler and Navier-Stokes systems of equations. The unsteady three-dimensional Navier-Stokes equations in the physical coordinate system x, y, z and time t may be expressed in vectorized conservation form as

$$\mathbf{Q}_t + (\mathbf{F}^\iota - \mathbf{F}^\upsilon)_x + (\mathbf{G}^\iota - \mathbf{G}^\upsilon)_y + (\mathbf{H}^\iota - \mathbf{H}^\upsilon)_z = \mathbf{0} \tag{2.1}$$

The subscripts represent spatial and temporal derivatives while the superscripts ι and υ indicate the inviscid and viscous flux vectors respectively. The Euler equations are readily obtained by setting the viscous flux vectors $\mathbf{F}^\upsilon_x, \mathbf{G}^\upsilon_y$ and $\mathbf{H}^\upsilon_z$ to zero. The vector of dependent variables is given by

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

in which ρ is the fluid density and u, v and w are the Cartesian velocity components in the respective x, y and z directions. The inviscid and viscous flux vectors in each direction are given by

$$\mathbf{F}^\iota = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{bmatrix}, \qquad \mathbf{F}^\upsilon = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \tag{2.3}$$

$$\mathbf{G}^\iota = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{bmatrix}, \qquad \mathbf{G}^\upsilon = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \tag{2.4}$$

$$\mathbf{H}^\iota = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{bmatrix}, \qquad \mathbf{H}^\upsilon = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix} \tag{2.5}$$

In the above equations p is the pressure, e the total energy per unit volume, τ_{ij} the stresses and q_i the heat fluxes. The system is closed by the equation of state for an ideal gas which is given by

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

in which γ is defined as the ratio of the specific heat at constant pressure to that at constant volume, i.e. $\gamma \equiv c_p/c_v$. For a perfect gas it is taken to have a value of 1.4. The Temperature T is obtained from standard thermodynamic relations as

$$T = \frac{\gamma p}{c_p(\gamma - 1)\rho} \tag{2.7}$$

The stress tensor τ_{ij} comprising the normal stresses ($i = j$) and shear stresses ($i \neq j$) and the heat flux vector q_i are given by

$$\begin{aligned}
\tau_{xx} &= 2\mu u_x - \lambda^v \Delta \\
\tau_{yy} &= 2\mu v_y - \lambda^v \Delta \\
\tau_{zz} &= 2\mu w_z - \lambda^v \Delta \\
\tau_{xy} &= \tau_{yx} = \mu(u_y + v_x) \\
\tau_{xz} &= \tau_{zx} = \mu(u_z + w_x) \\
\tau_{yz} &= \tau_{zy} = \mu(v_z + w_y)
\end{aligned} \tag{2.8}$$

$$q_x = -\kappa \frac{\partial T}{\partial x}, \quad q_y = -\kappa \frac{\partial T}{\partial y}, \quad q_z = -\kappa \frac{\partial T}{\partial z} \tag{2.9}$$

where

$$\Delta = u_x + v_y + w_z$$

and κ is the thermal conductivity coefficient, μ is the dynamic viscosity and λ^v a second viscosity coefficient. In equation (2.8), the fluid is assumed to be in local equilibrium. The diffusive coefficients are examined in greater detail in section 2.5.2.

Use of the governing equations in approximating turbulent flow solutions requires some further explanation. The properties of turbulent flow would be known if the exact Navier-Stokes equations could be solved with the correct boundary conditions. Unfortunately, the large range of time and spatial scales present in turbulent flows, combined with the limitations of present day computer power, precludes this possibility for nearly all flows of industrial interest. The governing equations may, however, be simplified by an averaging procedure to produce the Reynolds averaged Navier-Stokes equations. This involves decomposing the instantaneous flow variables of the governing equations (2.1) into mean and fluctuating components and averaging the equations over a sufficiently small time interval. The resulting equations contain the so called Reynolds stresses. This simplification introduces the issue of closure, needed to reduce the number of unknowns to the number of equations. The Reynolds-averaged equations are formally equivalent to the laminar form of equation (2.1). However for turbulent flow, the relationships between the mean flow quantities and the viscous stresses and turbulent heat diffusion are unknown. As such it becomes necessary to model them. Following the Boussinesq approximation, the Reynolds stresses can be related to a turbulent viscosity. The classical closure of the governing equations is obtained by replacing the dynamic viscosity by an *effective viscosity* which is defined as the sum of the molecular (laminar) dynamic viscosity and an eddy (turbulent) viscosity, i.e.

$$\mu_{eff} = \mu_l + \mu_t \tag{2.10}$$

With the effective viscosity defined in this way, the averaging procedure leads to a system of equations which is formally identical to (2.1), once the association between μ_{eff} and the viscosity term μ appearing in the equations (2.8) is made. Since the Reynolds averaged equations constitute the viscous approximation of interest to the present study, henceforth the term μ will be used interchangeably to mean μ_{eff}. The question of closure is considered further in section 2.5.3.

Finally, it should be noted that the system of governing equations may be derived in an alternative yet equivalent integral form which constitute the basic equations of the *Finite Volume Method*, described in the next section.

2.3 The Finite Volume Formulation

2.3.1 General background to discretization techniques

In the numerical solution of field problems, a computational domain is set up to discretize the physical space of the continuum by defining a mesh or grid from which the solution to the continuum problem may be approximated at a finite number of points. At each point, the governing equations are represented in either the differential or integral form of the conservation laws. Moving from the level of the continuum to its discrete counterpart, the order of accuracy of the approximation becomes an issue. In theory, the derivative of a continuous and sufficiently smooth function may be evaluated by Taylor series expansions to any order of accuracy, given infinite support. Most numerical methods rely on this property of continuous functions. Indeed, provided that the value of the function and a sufficiently high number of its derivatives at a given point are known, it is possible to reconstruct the function in other parts of the domain. Furthermore, for a numerical method to be consistent with its governing PDE, in the limit of vanishing mesh size, the error of the discretization must tend to zero at a rate that is proportional to the order of the discretization. Once the computational mesh has been defined, the next step is to discretize the spatial terms in the governing equations. For time-dependent problems, this spatial discretization transforms the governing system of PDEs into a system of ordinary differential equations. The final step in the solution process is the time discretization which produces a system of algebraic equations that may be solved readily on a digital computer.

For problems of typical interest to aeronautical applications, the discretization of differential operators may be classified into three principal categories, namely Finite Difference Methods, Finite Element Methods and Finite Volume Methods. Most attribute the first definitive work of importance on the finite difference method to Richardson [121], who near the turn of the century introduced point iterative schemes for numerically solving the Laplace equation and the biharmonic equation. He actually carried out hand calculations for the stress distribution in a masonry dam. In addition, he defined the distinction between problems which must be solved by a relaxation scheme and those which are referred to as marching problems. Finite difference methods are based on Taylor series expansions, relying on the approximation of differential operators by a finite difference representation of derivatives. Finite difference formulae of arbitrary order of accuracy may be constructed provided there is sufficiently large support. However, the bandwidth of the resulting algebraic system which must be inverted to obtain the solution, is proportional to the support of the difference stencil. The computational cost of inverting large systems of equations of wide bandwidth therefore places a practical constraint on the use of formulations of very high-order. In addition, finite difference methods must use structured grids with a high degree of mesh regularity. Excessive mesh stretching or skewing and abrupt changes in mesh spacing reduce their solution accuracy.

As it is known today, the finite element method was first presented in 1956 by Turner, Clough, Martin and Topp. Their paper [159] which describes the application of triangular finite elements in plain stress problems, is considered

a seminal contribution to the development of the method. After its successful development in a wide variety of problems in structural mechanics, the method was applied to the solution of continuous field problems including ones in fluid dynamics. With finite element methods, the physical domain is subdivided into *elements* which form the mesh. The elements may have any shape and unlike finite difference methods, the grid may be unstructured. The solution of the discrete problem belongs to a functional space, having a pre-determined character. For example, the order of the function at nodal points may be linear, quadratic or higher. The integral form of PDEs is a natural framework for the finite element method and indeed the one in which it is usually met. In recent years, the finite element method has developed much mathematical rigour and sophistication, particularly in the field of structural mechanics. However, within the field of fluid mechanics and in particular for hyperbolic problems, the development of the finite element method is less advanced.

The finite volume method was first published independently by McDonald [100] and MacCormack and Paullay [93] in the early seventies for the solution of the two-dimensional, time-dependent Euler equations. It was subsequently extended by others to three-dimensional flows. The technique uses the integral formulation of the conservation laws which are discretized directly in the physical space. It is viewed by some as a finite difference method applied to the differential form of the conservation laws written in arbitrary coordinates while others view it as a variant of a weak formulation of the conservation laws written in integral form. The finite volume method may be formulated to use structured, unstructured or even hybrid mesh. In addition, by the direct discretization of the integral form of the conservation laws, mass, momentum and energy remain conserved at the discrete level. This is a property of fundamental significance to numerical schemes in that it guarantees that when a numerical solution converges, it will do so to a solution of the governing PDEs, with the correct satisfaction of the Rankine-Hugoniot relations in the presence of discontinuities. A mathematical statement of this is given in the fundamental theorem of Lax and Wendroff [84]. A rigorous analysis of finite difference and finite volume methods is given in the review article by Vinokur [177].

2.3.2 The governing equations in finite volume form

Since the differential form of a conservation law is only valid where state quantities are differentiable, the integral form of the governing equations provides a more natural setting in which to consider hyperbolic systems which admit weak solutions. With this in mind, the governing differential equations (2.1) may be written in the compact form

$$\mathbf{Q}_t + \nabla \bar{F}(\mathbf{Q}) = 0 \tag{2.11}$$

in which the flux density tensor $\bar{F}$ is introduced as

$$\bar{F} = (\mathbf{F}^i - \mathbf{F}^v)\mathbf{i}_x + (\mathbf{G}^i - \mathbf{G}^v)\mathbf{i}_y + (\mathbf{H}^i - \mathbf{H}^v)\mathbf{i}_z$$

Integrating equation (2.11) over an arbitrary control volume $\mathcal{V}$ with boundary $\mathcal{S}$ and outer normal $\mathbf{n}$ and applying the Gauss divergence theorem produces

$$\frac{\partial}{\partial t}\iiint_{\mathcal{V}} \mathbf{Q}\, d\mathcal{V} + \iint_{S} \bar{F}\cdot\mathbf{n}\, dS = 0 \tag{2.12}$$

which is the basic equation for the finite volume method. It can be interpreted as a statement that the rate of change of the conserved variables in a volume $\mathcal{V}$ is balanced by the net flux $\bar{F}$ passing through its boundary S. This formulation corresponds to a flux summation algorithm in physical space. It is equivalent to the strong conservation finite-difference form of solution in logical or computational space if appropriate associations are made between the metric terms of the finite difference formulation and the physical dimensions (volumes and areas), of the integral equations. This equivalence becomes apparent when the system of governing equations (2.1) is transformed from physical (x, y, z) space to logical space (ξ, η, ζ) using a time-invariant grid, aligned with the physical geometry. Then they become

$$\hat{\mathbf{Q}}_t + (\hat{\mathbf{F}}^\iota - \hat{\mathbf{F}}^\upsilon)_\xi + (\hat{\mathbf{G}}^\iota - \hat{\mathbf{G}}^\upsilon)_\eta + (\hat{\mathbf{H}}^\iota - \hat{\mathbf{H}}^\upsilon)_\zeta = 0 \tag{2.13}$$

where

$$\begin{aligned}
\hat{\mathbf{Q}} &= \mathbf{Q}/J \\
\hat{\mathbf{F}}^\iota &= (\xi_x\mathbf{F}^\iota + \xi_y\mathbf{G}^\iota + \xi_z\mathbf{H}^\iota)/J \\
\hat{\mathbf{G}}^\iota &= (\eta_x\mathbf{F}^\iota + \eta_y\mathbf{G}^\iota + \eta_z\mathbf{H}^\iota)/J \\
\hat{\mathbf{H}}^\iota &= (\zeta_x\mathbf{F}^\iota + \zeta_y\mathbf{G}^\iota + \zeta_z\mathbf{H}^\iota)/J \\
\hat{\mathbf{F}}^\upsilon &= (\xi_x\mathbf{F}^\upsilon + \xi_y\mathbf{G}^\upsilon + \xi_z\mathbf{H}^\upsilon)/J \\
\hat{\mathbf{G}}^\upsilon &= (\eta_x\mathbf{F}^\upsilon + \eta_y\mathbf{G}^\upsilon + \eta_z\mathbf{H}^\upsilon)/J \\
\hat{\mathbf{H}}^\upsilon &= (\zeta_x\mathbf{F}^\upsilon + \zeta_y\mathbf{G}^\upsilon + \zeta_z\mathbf{H}^\upsilon)/J
\end{aligned} \tag{2.14}$$

with J being the Jacobian of the transformation given by

$$J = \left|\frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)}\right| \tag{2.15}$$

The inviscid and viscous flux vectors may now be re-expressed in the computationally efficient form as

$$\hat{\mathbf{F}}^\iota = J^{-1}\begin{bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ U(e+p) \end{bmatrix}, \qquad \mathbf{F}^\upsilon = J^{-1}\begin{bmatrix} 0 \\ \xi_x\tau_{xx} + \xi_y\tau_{xy} + \xi_z\tau_{xz} \\ \xi_x\tau_{yx} + \xi_y\tau_{yy} + \xi_z\tau_{yz} \\ \xi_x\tau_{zx} + \xi_y\tau_{zy} + \xi_z\tau_{zz} \\ u\Pi_x^\xi + v\Pi_y^\xi + w\Pi_z^\xi \end{bmatrix} \tag{2.16}$$

$$\hat{\mathbf{G}}^\iota = J^{-1}\begin{bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ V(e+p) \end{bmatrix}, \qquad \mathbf{G}^\upsilon = J^{-1}\begin{bmatrix} 0 \\ \eta_x\tau_{xx} + \eta_y\tau_{xy} + \eta_z\tau_{xz} \\ \eta_x\tau_{yx} + \eta_y\tau_{yy} + \eta_z\tau_{yz} \\ \eta_x\tau_{zx} + \eta_y\tau_{zy} + \eta_z\tau_{zz} \\ u\Pi_x^\eta + v\Pi_y^\eta + w\Pi_z^\eta \end{bmatrix} \tag{2.17}$$

$$\hat{\mathbf{H}}^\iota = J^{-1}\begin{bmatrix} \rho W \\ \rho W u + \zeta_x p \\ \rho W v + \zeta_y p \\ \rho W w + \zeta_z p \\ W(e+p) \end{bmatrix}, \qquad \mathbf{H}^\upsilon = J^{-1}\begin{bmatrix} 0 \\ \zeta_x\tau_{xx} + \zeta_y\tau_{xy} + \zeta_z\tau_{xz} \\ \zeta_x\tau_{yx} + \zeta_y\tau_{yy} + \zeta_z\tau_{yz} \\ \zeta_x\tau_{zx} + \zeta_y\tau_{zy} + \zeta_z\tau_{zz} \\ u\Pi_x^\zeta + v\Pi_y^\zeta + w\Pi_z^\zeta \end{bmatrix} \tag{2.18}$$

where

$$\begin{aligned} \Pi_x^\phi &= \phi_x\tau_{xx} + \phi_y\tau_{xy} + \phi_z\tau_{xz} - q_x \\ \Pi_y^\phi &= \phi_x\tau_{yx} + \phi_y\tau_{yy} + \phi_z\tau_{yz} - q_y \\ \Pi_z^\phi &= \phi_x\tau_{zx} + \phi_y\tau_{zy} + \phi_z\tau_{zz} - q_z \end{aligned} \tag{2.19}$$

and U, V, W are the velocities normal to each cell face (the contravariant velocities) in each of the respective directions ξ, η, ζ. They are given by

$$\begin{aligned} U &= k_x u + k_y v + k_z w \\ V &= l_x u + l_y v + l_z w \\ W &= m_x u + m_y v + m_z w \end{aligned} \tag{2.20}$$

For a finite difference representation, the metric terms $(\xi, \eta, \zeta)_{x,y,z}$ represent the actual mapping from physical to computational space. However, for a finite volume formulation they are interpreted as the cell surface normals while the inverse of the Jacobian transformation J^{-1}, is associated directly with the volume of individual cells in the domain. With the metric terms so interpreted, the formulation corresponds to a physical space discretization in a body conforming coordinate system. Thus the finite volume method can be expressed equivalently using either equation (2.12) or by equation (2.13) once the necessary associations have been made.

As noted already, to obtain the inviscid equations the viscous flux vectors in the governing equations are set to zero. Thus the Euler equations are a subset of equation (2.13) given by

$$\hat{\mathbf{Q}}_t + \hat{\mathbf{F}}'_\xi + \hat{\mathbf{G}}'_\eta + \hat{\mathbf{H}}'_\zeta = 0 \tag{2.21}$$

In this framework, the cell volumes, surface normals and direction cosines are the metric parameters that are required for the computation. Their derivation is considered in the following sections.

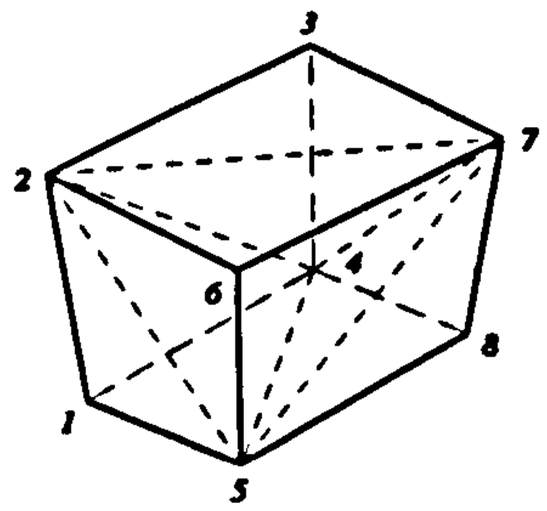
2.3.3 Cell volumes

With structured grids, the volume of any general three–dimensional hexahedral cell may be determined as the sum of constituent tetrahedra. Several of the ways in which this can be done are discussed by Vinokur [177]. Here the volume is computed from five tetrahedra. Figure 2.1 shows an arbitrary finite volume cell together with the individual tetrahedra that make up its volume. The volume of the complete cell $\mathcal{V}_{i,j,k}$ is therefore given by

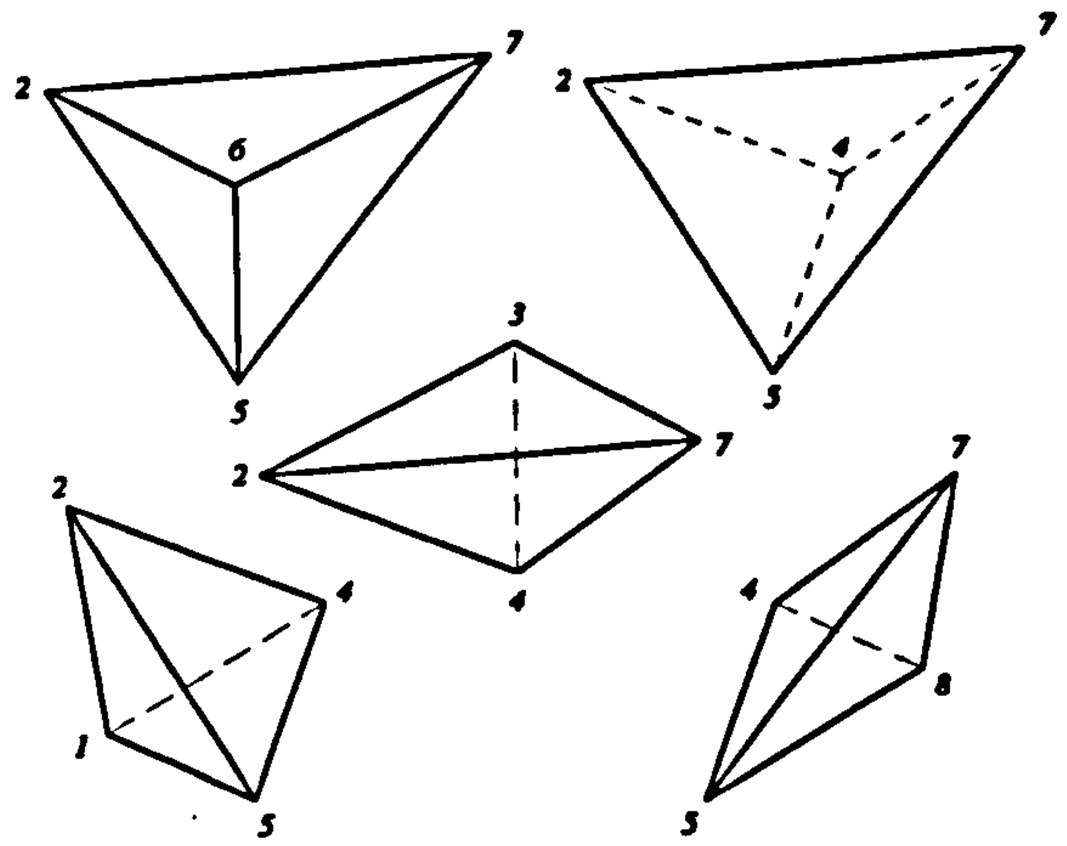
$$\mathcal{V}_{i,j,k} = \mathcal{V}_{4527} + \mathcal{V}_{4521} + \mathcal{V}_{4578} + \mathcal{V}_{4273} + \mathcal{V}_{5276} \tag{2.22}$$

in which the terms on the right hand side represent the volume of the constituent tetrahedra. Each is defined by subscript in terms of the numbered edges that bound it according to the notation of the figure. Now, consider an arbitrary tetrahedron t with a base (a, b, c) and vertex d. It may be defined in terms of the vectors $\mathbf{a} = ac, \mathbf{b} = ab, \mathbf{c} = ad$, and the angle θ made between $\mathbf{c}$ and the vertical. Its volume is given by

$$\mathcal{V}_t = \frac{1}{3}\,(\text{vertical height}) \times (\text{base area } abc)$$



(a)



(b)

FIGURE 2.1. Computation of cell volume. Diagram showing (a) an individual cell and (b) its breakdown into five constituent tetrahedra used to compute the volume of the complete cell.

$$\begin{aligned} &= \frac{1}{3}\left(|\mathbf{c}|\cos\theta\right) \times \frac{1}{2}\left(\mathbf{a}\times\mathbf{b}\right) \\ &= \mathbf{c}.(\mathbf{a}\times\mathbf{b})/6 \end{aligned} \tag{2.23}$$

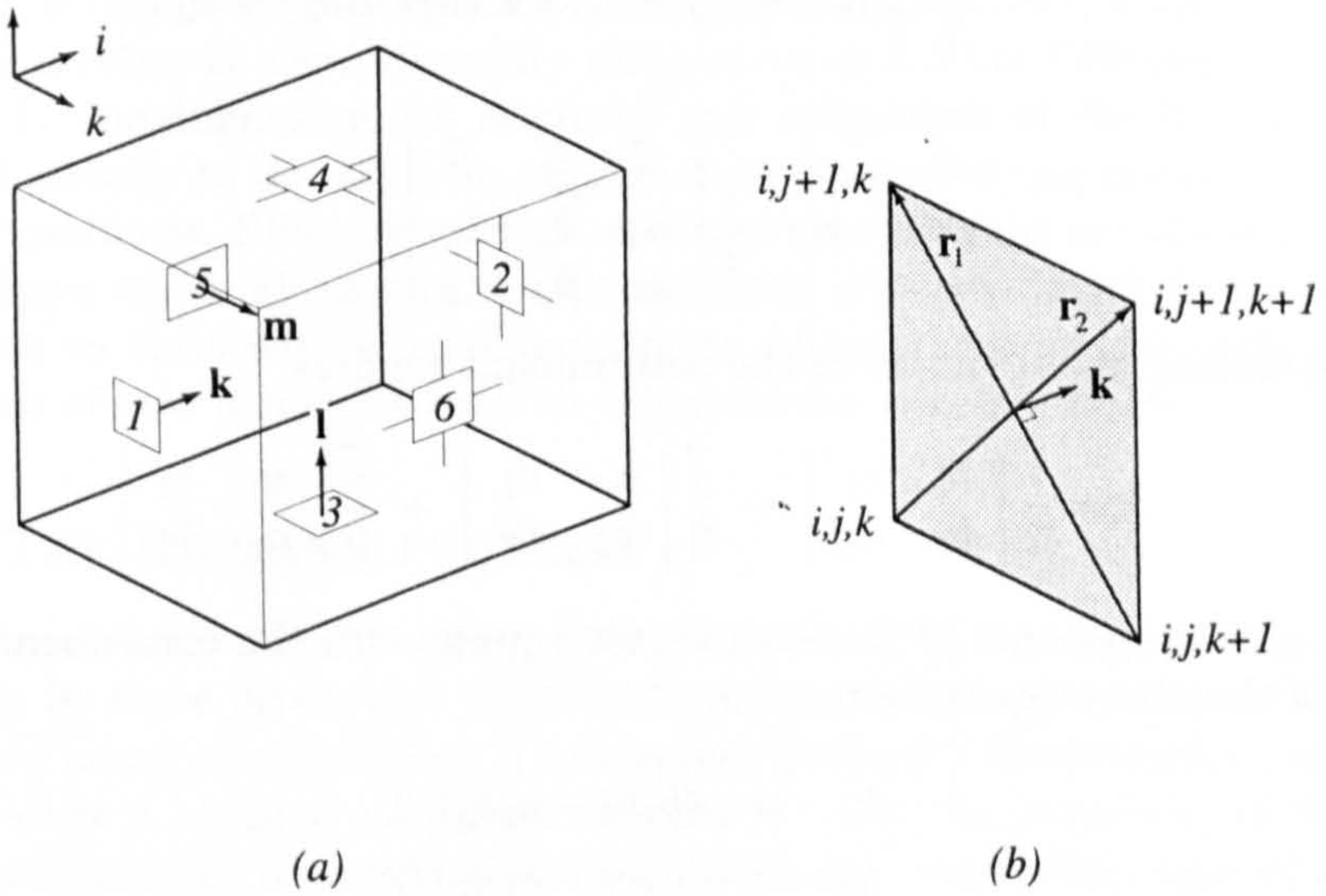
From equation (2.23), the tetrahedron volume is evaluated algebraically as

$$\begin{aligned} \mathcal{V}_t = \; & |z_a\left[x_b(y_c-y_d)-x_c(y_b-y_d)+x_d(y_b-y_c)\right]- \\ & z_b\left[x_a(y_c-y_d)-x_c(y_a-y_d)+x_d(y_a-y_c)\right]+ \\ & z_c\left[x_a(y_b-y_d)-x_b(y_a-y_d)+x_d(y_a-y_b)\right]- \\ & z_d\left[x_a(y_b-y_c)-x_b(y_a-y_c)+x_c(y_a-y_b)\right]|/6 \end{aligned} \tag{2.24}$$

Once the volume of each constituent tetrahedron has been computed in this fashion, the cell volume is computed directly from equation (2.22).

2.3.4 Surface normals

Consider the finite volume cell in logical space that is illustrated in Figure 2.2(a).



(a) *(b)*

FIGURE 2.2. Computation of surface normals. Diagram showing (a) the surface area vectors for a complete cell and (b) an individual cell face with the position vectors defining it.

Each cell face is numbered from 1 to 6. By definition, **k** is the vector representing the area perpendicular to cell face 1 directed inside of the cell. Similarly **l** and **m** are the respective area vectors of cell faces 3 and 5. It is only cell faces 1, 3 and 5, hereafter termed the *low faces*, that are of concern as the remaining *high faces* may be computed as the low faces of adjoining cells. Resolving each area vector into the Cartesian directions renders nine components $k_{x,y,z}, l_{x,y,z}, m_{x,y,z}$, such that

$$\begin{aligned} |\mathbf{k}| &= (k_x^2 + k_y^2 + k_z^2)^{\frac{1}{2}} \\ |\mathbf{l}| &= (l_x^2 + l_y^2 + l_z^2)^{\frac{1}{2}} \\ |\mathbf{m}| &= (m_x^2 + m_y^2 + m_z^2)^{\frac{1}{2}} \end{aligned} \tag{2.25}$$

In part (b) of Figure 2.2 an individual cell face is shown with its area vector **k**, and the two diagonal position vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ which define it. The coordinates of the face edges are annotated in terms of the logical counters. (In computation, the counters reference the physical coordinates of each grid node in the respective x, y, z directions). The diagonals are defined as

$$\mathbf{r}_1 = a_1\vec{i} + b_1\vec{j} + c_1\vec{k}$$

$$\mathbf{r}_2 = a_2\vec{i} + b_2\vec{j} + c_2\vec{k}$$

where $\vec{i}, \vec{j}, \vec{k}$ are unit vectors in the x, y, z directions. It follows that

$$\begin{aligned} \mathbf{k} &= \frac{1}{2}(\mathbf{r}_1 \times \mathbf{r}_2) \\ &= \frac{1}{2}\left[(a_1\vec{i} + b_1\vec{j} + c_1\vec{k}) \times (a_2\vec{i} + b_2\vec{j} + c_2\vec{k})\right] \end{aligned} \tag{2.26}$$

Hence

$$\mathbf{k} = \frac{1}{2}\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} \tag{2.27}$$

and expanding the cofactors of the determinant renders

$$\mathbf{k} = \frac{\vec{i}}{2}\begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} - \frac{\vec{j}}{2}\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} + \frac{\vec{k}}{2}\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \tag{2.28}$$

The x, y, z components of the area vector equate with the component determinants of the above equation to give

$$\begin{aligned} k_x &= (b_1c_2 - c_1b_2)/2 \\ k_y &= -(a_1c_2 - c_1a_2)/2 \\ k_z &= (a_1b_2 - b_1a_2)/2 \end{aligned} \tag{2.29}$$

As the lengths $a_1, b_1, c_1, a_2, b_2, c_2$, are defined in terms of the logical coordinates of the particular cell face under consideration, those of cell face 1 are

$$\begin{aligned} a_1 &= x_{i,j+1,k} - x_{i,j,k+1} \\ b_1 &= y_{i,j+1,k} - y_{,j,k+1} \\ c_1 &= z_{i,j+1,k} - z_{i,j,k+1} \\ a_2 &= x_{i,j+1,k+1} - x_{i,j,k} \\ b_2 &= y_{i,j+1,k+1} - y_{i,j,k} \\ c_2 &= z_{i,j+1,k+1} - z_{i,j,k} \end{aligned}$$

Hence the area vector components of face 1 are evaluated from equation (2.29) in terms of the above coefficients. The area vector components for the remaining low faces $l_{x,y,z}, m_{x,y,z}$ are of the same form as $k_{x,y,z}$ and are calculated in the same fashion using their own logical coordinates.

The three direction cosines of each cell face are defined simply in terms of the individual area vector components normalized by the magnitude of the respective resultant vector as

$$\begin{aligned} (\hat{k}_x, \hat{k}_y, \hat{k}_z) &= (k_x, k_y, k_z)/|\mathbf{k}| \\ (\hat{l}_x, \hat{l}_y, \hat{l}_z) &= (l_x, l_y, l_z)/|\mathbf{l}| \\ (\hat{m}_x, \hat{m}_y, \hat{m}_z) &= (m_x, m_y, m_z)/|\mathbf{m}| \end{aligned} \tag{2.30}$$

2.4 Spatial Discretization using Roe Fluxes

Riemann problems and their solution procedures were first introduced into computational fluid dynamics by Godunov through finite difference methods developed for solving one and two-dimensional unsteady inviscid flows [53, 67]. The algorithm discussed in this section is a cell-centred scheme known as Roe's method. It is one of several schemes classed as a Godunov-type method. (Note that Roe's scheme is also commonly referred to as a Flux Difference Splitting Method). In one-dimension the accuracy and robustness of the Roe scheme is attributed directly to its ability to approximate the underlying physics well, over a range of problems. Effectively this is achieved through solution of the Riemann problem upon which all Godunov type methods are formulated. It is therefore appropriate to start with a brief description of the Riemann problem and an explanation of how it may be used to calculate the interface fluxes.

2.4.1 The Riemann problem

For one-dimensional non-stationary flows, the state $\mathbf{Q}$ of a perfect gas is specified completely by three dependent variables together with the properties of the particular fluid under consideration. A convenient choice for the dependent variables is the pressure p, density ρ and flow velocity u while the properties of the fluid may be conveniently represented by two constants, namely the ratio of specific heats γ and the gas constant R. Note that other choices of variables and fluid properties are equally valid. From these variables and constants all other state properties may be obtained, where necessary using supplementary relationships such as the equation of state and the definition of the speed of sound. Hence, the state $\mathbf{Q}$ is completely defined by (p,ρ,u,γ,R). In this context consider the Riemann problem in one-dimension, a classical initial value problem that may be posed for any hyperbolic system of conservation laws

$$\mathbf{Q}_t + \mathbf{F}_x = 0 \tag{2.31}$$

In this notation, the subscripts t and x represent temporal and spatial partial derivatives, respectively. The problem is defined at time $t = 0$ by considering initial data consisting of piecewise constant left and right states L and R. These are shown in Figure 2.3(a). For times $t > 0$, the states interact to form a pattern of three centred waves (characteristic lines), along which the signals known as Riemann invariants propagate. These waves split the domain into four regions of constant state Q_L, Q_L^*, Q_R^*, Q_R as indicated in part (b) of the figure. The inner wave is a contact discontinuity separating states at different temperature while the outer waves are acoustic. Each of the acoustic waves may be may be either a shock wave or a rarefaction fan (expansion wave). For the Euler equations no closed form analytic solution exists to the Riemann problem. However, from the observation that pressure p and velocity u are constant across a contact discontinuity, i.e.

$$\begin{aligned} p_L^* &= p_R^* \\ u_L^* &= u_R^* \end{aligned}$$

it is possible to derive a set of equations that may be solved iteratively for p^*. Hence, the values of u^*, ρ_L^* and ρ_R^* may be determined. This basic strategy is

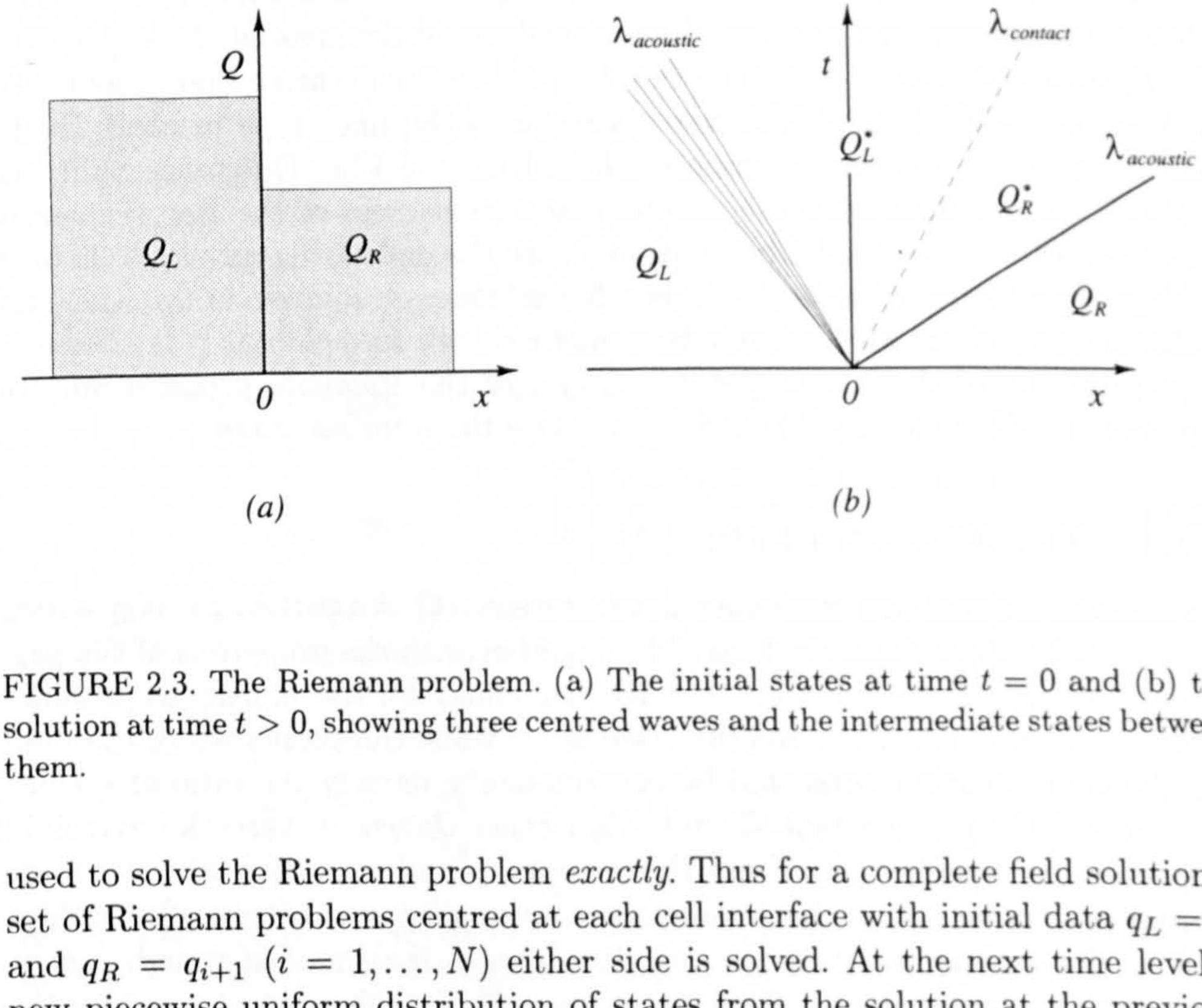


FIGURE 2.3. The Riemann problem. (a) The initial states at time $t = 0$ and (b) the solution at time $t > 0$, showing three centred waves and the intermediate states between them.

used to solve the Riemann problem *exactly*. Thus for a complete field solution a set of Riemann problems centred at each cell interface with initial data $q_L = q_i$ and $q_R = q_{i+1}$ $(i = 1, \ldots, N)$ either side is solved. At the next time level a new piecewise uniform distribution of states from the solution at the previous time level is used and the process repeated successively until a steady state solution is attained or some desired point in time is reached. As exact solution of the Riemann problem is an expensive procedure, effort has been devoted to developing efficient techniques for its solution. (A comprehensive discussion of one-dimensional exact Riemann solvers is given by Gottlieb et al. [56]). However, instead of solving the Riemann problem exactly, an alternative approach is possible.

Since the Riemann problem arising in the above solution method relates only to an approximation of the data, it may be reasonable to accept approximate solutions provided these solutions still retain the essential nonlinear behaviour of the hyperbolic system. This, in fact, is an argument advanced by Roe [129] and several others to justify numerical algorithms that are based on solution of an *Approximate* Riemann problem. These algorithms are therefore termed Approximate Riemann Solvers. Their advantage over exact Riemann solvers lies in that approximate solutions are substantially cheaper to compute than exact solutions. As a consequence, several Approximate Riemann algorithms have been developed, notably by Osher and Solomon [107] and several others [109, 35, 154].

2.4.2 Discretization of convective terms in 1-D

The basic idea of Flux Difference Splitting methods is to construct the interface fluxes by solving the underlying Riemann problems so that each cell interface flux may be constructed accurately via summation of the component flux of each wave that is present locally. In Roe's method [127], the exact solution to an *Approximate* Riemann problem is found. To achieve this, equation (2.31) is approximated as

$$\mathbf{Q}_t + \tilde{\mathbf{A}}(\mathbf{Q}_L, \mathbf{Q}_R)\mathbf{Q}_x = 0 \tag{2.32}$$

in which $\tilde{\mathbf{A}}(\mathbf{Q}_L, \mathbf{Q}_R)$ is a constant matrix which locally approximates the actual flux Jacobian $\mathbf{A} \equiv \partial\mathbf{F}/\partial\mathbf{Q}$, and is referred as the Roe matrix. Three properties are manifest in its construction:

(i) $\tilde{\mathbf{A}}(\mathbf{Q}_L, \mathbf{Q}_R) \rightarrow \mathbf{A}(\mathbf{Q})$ smoothly as $\mathbf{Q}_L, \mathbf{Q}_R \rightarrow \mathbf{Q}$

(ii) $\tilde{\mathbf{A}}(\mathbf{Q}_L, \mathbf{Q}_R)(\mathbf{Q}_R - \mathbf{Q}_L) = \mathbf{F}_L - \mathbf{F}_R$

(iii) $\tilde{\mathbf{A}}(\mathbf{Q}_L, \mathbf{Q}_R)$ is diagonalizable with real eigenvectors

The first property guarantees that the method behaves reasonably on smooth solutions while the second property ensures both consistency and conservation. Another effect of the second property is that in the special case where $\mathbf{Q}_L$ and $\mathbf{Q}_R$ are connected by a single shock or contact, the Approximate Riemann solution agrees with the exact Riemann solution. This follows from satisfaction of the Rankine-Hugoniot condition. Finally, the third property is required in order that equation (2.32) is hyperbolic and solvable.

Roe expresses the difference in flux between left and right states as

$$\begin{aligned} \Delta\mathbf{F} &= \mathbf{F}_R - \mathbf{F}_L \\ &= \sum_{k=1}^{n} \alpha_k \lambda_k \mathbf{r}_k \end{aligned} \tag{2.33}$$

where each term $\alpha_k\lambda_k\mathbf{r}_k$ represents the difference in flux across an individual wave, given n waves present in the solution. Physically, λ_k represents the speed of the kth wave while α_k represents its strength or amplitude. Mathematically, λ_k and r_k are the respective eigenvalues and (right) eigenvectors of the Flux Jacobian $\tilde{\mathbf{A}}$ while α_k is the projection of the jump in $\mathbf{Q}$ between left and right states onto the (left) eigenvectors of $\tilde{\mathbf{A}}$.

With this (characteristic based) decomposition, the interface flux can be determined either by adding the wave flux associated with *negative* wave speeds to the flux of the left state, $\mathbf{F}_L$, or, by subtracting the *positive* wave flux from the right state, $\mathbf{F}_R$. Alternatively, taking an average of these two approaches gives the first-order numerical flux function

$$\mathbf{F}_{i+\frac{1}{2}}(\mathbf{Q}_L, \mathbf{Q}_R) = \frac{1}{2}\left[(\mathbf{F}_L + \mathbf{F}_R) - \sum_{k=1}^{n} \alpha_k \left|\lambda_k\right| \mathbf{r}_k\right] \tag{2.34}$$

where the summation terms are calculated from cell interface values. Note that the numerical dissipation arises out of the last term on the right-hand side of the equation. This flux function affords wiggle-free capture of discontinuities. To

determine the eigenvalues and eigenvectors of the approximate Jacobian $\tilde{\mathbf{A}}$, it is diagonalized as

$$\tilde{\mathbf{A}} = \mathbf{R}\Lambda\mathbf{R}^{-1} \tag{2.35}$$

in which the columns of the matrix $\mathbf{R}$ and rows of the matrix $\mathbf{R}^{-1}$ are the respective right and left eigenvectors. Accordingly, equation (2.33) may be recast as

$$\begin{aligned} \Delta\mathbf{F} &= \tilde{\mathbf{A}}\Delta\mathbf{Q} \\ &= \mathbf{R}\Lambda\mathbf{R}^{-1}\Delta\mathbf{Q} \end{aligned} \tag{2.36}$$

in which Λ is a diagonal matrix containing the eigenvalues and $\mathbf{R}^{-1}\Delta\mathbf{Q}$ is the change (jump) in characteristic variables across a cell interface. The latter is equivalent to wave strength. Thus, the interface flux in equation (2.34) may be expressed equivalently as

$$\mathbf{F}_{i+\frac{1}{2}}(\mathbf{Q}_L, \mathbf{Q}_R) = \frac{1}{2}\left[(\mathbf{F}_L + \mathbf{F}_R) - \mathbf{R}|\Lambda|\mathbf{R}^{-1}\Delta\mathbf{Q}\right] \tag{2.37}$$

In what follows, the enthalpy H and the speed of sound a, must be determined from the states L and R via the standard relations

$$H = (e + p)/\rho \tag{2.38}$$

and

$$a^2 = (\gamma - 1)\left[H - u^2/2\right] \tag{2.39}$$

The second property of the Roe matrix necessitates the use of a special averaging procedure to determine flow parameters at cell interfaces. The resulting interface values, commonly called Roe averages and denoted by a tilde, are obtained directly from Property (ii). The Roe averages for the velocity and enthalpy are given by

$$\tilde{\mathcal{R}} = (\mathcal{R}_R + \omega\mathcal{R}_L)/(1 + \omega) \tag{2.40}$$

in which $\tilde{\mathcal{R}}$ represents Roe averaged velocity $\tilde{u}$ and enthalpy $\tilde{H}$, while the Roe-averaged density is obtained as

$$\tilde{\rho} = \rho_R\omega \tag{2.41}$$

in which

$$\omega = (\rho_L/\rho_R)^{\frac{1}{2}} \tag{2.42}$$

These averages are required for construction of the numerical interface fluxes defined by equations (2.34) and (2.37).

If equation (2.31) is taken to represent the unsteady one-dimensional Euler equations, the respective state and flux vectors of the equation are

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad \mathbf{F}(\mathbf{Q}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e + p) \end{bmatrix} \tag{2.43}$$

in which density is denoted as ρ, velocity as u, pressure as p and, energy as e. A semi-discrete finite volume representation of the system is given by

$$\mathbf{Q}_t + \frac{1}{\Delta x}(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}) = 0 \tag{2.44}$$

The interface flux $\mathbf{F}_{i+\frac{1}{2}}$, is now determined using the results that follow. The wave speeds (λ_k), are determined from

$$\begin{aligned} \Lambda &= \text{diag } [\lambda_1, \lambda_2, \lambda_3] \\ &= \text{diag } [(\tilde{u}-\tilde{a}), \tilde{u}, (\tilde{u}+\tilde{a})] \end{aligned} \tag{2.45}$$

while the matrix $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 1 \\ \tilde{u}-\tilde{a} & \tilde{u} & \tilde{u}+\tilde{a} \\ \tilde{H}-\tilde{u}\tilde{a} & \tilde{u}^2/2 & \tilde{H}+\tilde{u}\tilde{a} \end{bmatrix} \tag{2.46}$$

The (right) eigenvectors are the column entries of this matrix while the characteristic variables $\mathbf{R}^{-1}\Delta\mathbf{Q}$, or wave strengths α_k are

$$\mathbf{R}^{-1}\Delta\mathbf{Q} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{1}{2\tilde{a}^2} \begin{bmatrix} \Delta p - \tilde{\rho}\tilde{a}\Delta u \\ 2(\tilde{a}^2\Delta\rho - \Delta p) \\ \Delta p + \tilde{\rho}\tilde{a}\Delta u \end{bmatrix} \tag{2.47}$$

with quantities of the form $\Delta(\bullet)$ evaluated as

$$\Delta(\bullet) \equiv (\bullet)_R - (\bullet)_L \tag{2.48}$$

To conclude this description of the one-dimensional upwind discretization of the convective terms, the solution procedure for Roe's first-order scheme using a forward-Euler time discretization is given in Figure 2.4.

1. Compute Roe-averaged interface quantities $\tilde{\rho}, \tilde{u}, \tilde{H}, \tilde{a}$
2. Compute eigenvalues, eigenvectors and wave strengths $\lambda_k, \mathbf{r}_k, \alpha_k$
3. Construct interface flux:
   $\mathbf{F}_{i+\frac{1}{2}}(\mathbf{Q}_L, \mathbf{Q}_R) = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\sum_{k=1}^{n} \alpha_k |\lambda_k| \mathbf{r}_k$
4. Update State vector:
   $\mathbf{Q}^{n+1} = \mathbf{Q}^n - \frac{\Delta t}{\Delta x}(\mathbf{F}^n_{i+\frac{1}{2}} - \mathbf{F}^n_{i-\frac{1}{2}})$
5. Update pressure via eqn of state:
   $p^{n+1} = (\gamma - 1)[e - \frac{1}{2}\rho u^2]$
6. Repeat 1-5 until solution converged
7. Stop

FIGURE 2.4. Procedure for implementing Roe's first-order upwind scheme.

2.4.3 Discretization of convective terms in 3-D

Fundamentally, there are two approaches to the discretization of convective terms for problems in two- or three-dimensions. The oldest and most common approach is to extend the one-dimensional theory to the multi-dimensional problem on a dimension-by-dimension basis. This approach constitutes state-of-the-art practice in industry for computing inviscid and viscous flows. However, from the

theoretical point of view, it is not wholly satisfactory. In the case of upwind discretizations based on Godunov-type methods, for example, the one-dimensional Riemann problem is a weak model for the complex wave interactions in a two- or three-dimensional flow with waves propagating and interacting in an infinite number of space directions. Also, by solving the one-dimensional Riemann problem in mesh dependent directions, viz. along grid lines at cell interfaces, non-physical grid dependency is introduced. With most non-upwind discretizations, the problems of the dimension-by-dimension approach are even worse. This is because in order to avoid odd-even decoupling (an instability that can arise in smooth flow), and oscillations at shock waves, scalar dissipation operators are included in the discretization. The problem of oscillations about shock waves arises through the inability of these discretizations to account for the relevant characteristic directions. Thus in the presence of discontinuities, the robustness of central schemes can become questionable since the dissipation must be tuned for individual flow conditions. This becomes problematic in certain cases, particularly ones involving very strong shock waves.

In recognition of the weaknesses of the dimensionally-split schemes, a new approach has been developed known as *intrinsically multi-dimensional methods*, and variously referred to as Intrinsic schemes, Truly Multi-dimensional methods and simply Multi-dimensional Upwind Methods. This approach aims to cure the deficiencies of the traditional methods by introducing true multi-dimensional physics based on compact stencils. By definition this implies reducing grid dependency. The need for this new approach was recognized by Raithby [120] as early as 1976 who devised a *skew-upstream differencing scheme* for the scalar convection equation which improved on the standard dimensionally-split upwind schemes. However, multi-dimensional upwind methods for realistic systems of equations only began to emerge nearly a decade later out of the pioneering work of Roe [130, 133] and Deconinck [31]. At the present time, two-dimensional upwind methods are beginning to exhibit some maturity and some excellent results have been presented by Paillère, Deconinck and Roe [108]. For a valuable insight into 2-D multi-dimensional upwind methods applied to the Euler equations, the interested reader is referred to van Leer [167]. However, this approach has yet to mature for use in 3-D and is not dealt with here. The remainder of this section details the dimensionally-split formulation of the present study.

To evaluate the convective terms of the Euler or Navier-Stokes equations in three-dimensions, the spatial discretization is done independently of the time discretization following the method of lines. (This approach was popularized in CFD largely through the work of Jameson [74, 76, 72] and the success of his widely available series of finite volume FLO codes). Initially each coordinate direction is treated separately based on one-dimensional theory.

As discussed in section 2.3.2, the governing equations are cast in the finite volume framework by making the association

$$\hat{\mathbf{Q}}_{i,j,k} = \mathcal{V}_{i,j,k}\mathbf{Q}_{i,j,k} \tag{2.49}$$

and so obviating the need to evaluate the transformation Jacobian $J_{i,j,k}$ since $J^{-1}_{i,j,k} \equiv \mathcal{V}_{i,j,k}$, where $\mathcal{V}_{i,j,k}$ is the cell volume. Using this relationship and dropping subscripts for clarity, the Navier-Stokes equations (2.13) become

$$\mathcal{V}\mathbf{Q}_t + (\hat{\mathbf{F}}^\iota - \hat{\mathbf{F}}^\upsilon)_\xi + (\hat{\mathbf{G}}^\iota - \hat{\mathbf{G}}^\upsilon)_\eta + (\hat{\mathbf{H}}^\iota - \hat{\mathbf{H}}^\upsilon)_\zeta = 0 \tag{2.50}$$

They may also be expressed in the abstract form

$$\mathbf{Q}_t = \mathcal{L}_s \tag{2.51}$$

in which the following relation holds:

$$\mathcal{L}_s = \mathcal{L}_\xi + \mathcal{L}_\eta + \mathcal{L}_\zeta \tag{2.52}$$

Here the one-dimensional operators $\mathcal{L}_\xi, \mathcal{L}_\eta, \mathcal{L}_\zeta$ are associated with the evaluation of both convective and diffusive derivatives in each of the respective coordinate directions such that

$$\begin{aligned} \mathcal{L}_\xi &= \mathcal{L}_\xi^\iota + \mathcal{L}_\xi^\upsilon \\ \mathcal{L}_\eta &= \mathcal{L}_\eta^\iota + \mathcal{L}_\eta^\upsilon \\ \mathcal{L}_\zeta &= \mathcal{L}_\zeta^\iota + \mathcal{L}_\zeta^\upsilon \end{aligned} \tag{2.53}$$

where the superscripts ι and υ correspond to the respective convective and diffusive operators. To effect an update, all spatial operators are applied simultaneously. Now a semi-discrete finite volume representation of equation (2.50) leads to

$$\begin{aligned} \mathbf{Q}_t = - \Big[& (\hat{\mathbf{F}}^\iota - \hat{\mathbf{F}}^\upsilon)_{i+\frac{1}{2},j,k} - (\hat{\mathbf{F}}^\iota - \hat{\mathbf{F}}^\upsilon)_{i-\frac{1}{2},j,k} \\ & (\hat{\mathbf{G}}^\iota - \hat{\mathbf{G}}^\upsilon)_{i,j+\frac{1}{2},k} - (\hat{\mathbf{G}}^\iota - \hat{\mathbf{G}}^\upsilon)_{i,j-\frac{1}{2},k} + \\ & (\hat{\mathbf{H}}^\iota - \hat{\mathbf{H}}^\upsilon)_{i,j,k+\frac{1}{2}} - (\hat{\mathbf{H}}^\iota - \hat{\mathbf{H}}^\upsilon)_{i,j,k-\frac{1}{2}} \Big] / \mathcal{V} \end{aligned} \tag{2.54}$$

As the discretization of the viscous terms is treated in section 2.5, it will be left for the time being so as to concentrate on the convective discretization. Dropping them from the above equation leaves the semi-discrete finite volume representation of the Euler equations which is given by

$$\begin{aligned} \mathbf{Q}_t = - \Big[& (\hat{\mathbf{F}}^\iota_{i+\frac{1}{2},j,k} - \hat{\mathbf{F}}^\iota_{i-\frac{1}{2},j,k}) + (\hat{\mathbf{G}}^\iota_{i,j+\frac{1}{2},k} - \hat{\mathbf{G}}^\iota_{i,j-\frac{1}{2},k}) + \\ & (\hat{\mathbf{H}}^\iota_{i,j,k+\frac{1}{2}} - \hat{\mathbf{H}}^\iota_{i,j,k-\frac{1}{2}}) \Big] / \mathcal{V} \end{aligned} \tag{2.55}$$

Now the spatial terms may be evaluated by the one-dimensional approximations

$$\begin{aligned} L_\xi &\equiv -(\hat{\mathbf{F}}^\iota_{i+\frac{1}{2},j,k} - \hat{\mathbf{F}}^\iota_{i-\frac{1}{2},j,k})/\mathcal{V} \\ L_\eta &\equiv -(\hat{\mathbf{G}}^\iota_{i,j+\frac{1}{2},k} - \hat{\mathbf{G}}^\iota_{i,j-\frac{1}{2},k})/\mathcal{V} \\ L_\zeta &\equiv -(\hat{\mathbf{H}}^\iota_{i,j,k+\frac{1}{2}} - \hat{\mathbf{H}}^\iota_{i,j,k-\frac{1}{2}})/\mathcal{V} \end{aligned} \tag{2.56}$$

The terms L_ξ, L_η, L_ζ are the numerical counterparts to the convective operators appearing in equations (2.53).

In the finite volume context, the normals to the cell faces serve as the local coordinate at which the one-dimensional Riemann solver is applied. The interface fluxes given in equation (2.55) may be expressed in terms of the one-dimensional Roe flux formula, equation (2.37), which may be generalized and written in the (ξ, η, ζ) coordinate system as

$$\hat{\mathbf{E}}_{m+\frac{1}{2}}(\mathbf{Q}_m, \mathbf{Q}_{m+1}) = \frac{1}{2}\left[(\hat{\mathbf{E}}_m + \hat{\mathbf{E}}_{m+1}) - \mathbf{R}|\Lambda|\mathbf{R}^{-1}\Delta\mathbf{Q}\right] \tag{2.57}$$

in which $\hat{\mathbf{E}}$ denotes each of the directional fluxes $\hat{\mathbf{F}}, \hat{\mathbf{G}}, \hat{\mathbf{H}}$ in turn, while the index m denotes the incrementing cell counter i, j, or k associated with the flux in each respective direction. This expression is used for evaluating the interface fluxes for a scheme which is first-order accurate in space. Second-order schemes are dealt with later in section 2.4.6.

Aside from the increase in complexity resulting from the body-fitted coordinate system, the flux at each interface is found analogously to the one-dimensional case described in the preceding section. The algebraic formulae for the components of the three-dimensional approximate flux Jacobian $\tilde{\mathbf{A}}$, necessary to evaluate the interface fluxes, are now presented. As in the one-dimensional case, the cell interface values for density $\tilde{\rho}$ are given by equations (2.41) and (2.42), while those of the three Cartesian velocity components $\tilde{u}, \tilde{v}, \tilde{w}$, and enthalpy $\tilde{H}$, are given by equations (2.40) and (2.42). The speed of sound at the interface is computed from

$$\tilde{a} = \left[(\gamma - 1)\left\{\tilde{H} - \tilde{q}^2/2\right\}\right]^{\frac{1}{2}} \tag{2.58}$$

where

$$\tilde{q} = \sqrt{\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2} \tag{2.59}$$

Further generalized notation necessary to simplify the presentation of the formulae to follow is now introduced. First the surface normals (area vectors) $(k, l, m)_{x,y,z}$ and unit normals (direction cosines) $(\hat{k}, \hat{l}, \hat{m})_{x,y,z}$ will be expressed such that an arbitrary surface normal n may be taken to represent k, l or m, while $\hat{n}$ represents any of the unit normals $\hat{k}, \hat{l}$ or $\hat{m}$. Similarly $\bar{U}$ is taken to represent any of the contravariant velocities U, V or W. Thus

$$|\mathbf{n}| = (n_x^2 + n_y^2 + n_z^2)^{\frac{1}{2}} \tag{2.60}$$

$$\hat{n}_{x,y,z} = n_{x,y,z}/|\mathbf{n}| \tag{2.61}$$

$$\bar{U} = n_x\tilde{u} + n_y\tilde{v} + n_z\tilde{w} \tag{2.62}$$

and the normalized contravariant velocities are generalized as

$$\hat{U} = \hat{n}_x\tilde{u} + \hat{n}_y\tilde{v} + \hat{n}_z\tilde{w} \tag{2.63}$$

The choice of eigenvectors for the linearized Roe matrix (viz. the approximate flux Jacobian) is not unique. Those adopted here are proposed by Manna [95]. With this choice, the matrix $\mathbf{R}$ containing the right eigenvectors (by column) is given by

$$\begin{bmatrix} 1 & 0 & 0 & \tilde{\rho}/2\tilde{a} & \tilde{\rho}/2\tilde{a} \\ \tilde{u} & 0 & \tilde{\rho}(\hat{n}_y^2 + \hat{n}_z^2) & \tilde{\rho}(\tilde{u} + \hat{n}_x\tilde{a})/2\tilde{a} & \tilde{\rho}(\tilde{u} - \hat{n}_x\tilde{a})/2\tilde{a} \\ \tilde{v} & -\tilde{\rho}\hat{n}_z & -\tilde{\rho}\hat{n}_x\hat{n}_y & \tilde{\rho}(\tilde{v} + \hat{n}_y\tilde{a})/2\tilde{a} & \tilde{\rho}(\tilde{v} - \hat{n}_y\tilde{a})/2\tilde{a} \\ \tilde{w} & \tilde{\rho}\hat{n}_y & -\tilde{\rho}\hat{n}_x\hat{n}_z & \tilde{\rho}(\tilde{w} + \hat{n}_z\tilde{a})/2\tilde{a} & \tilde{\rho}(\tilde{w} - \hat{n}_z\tilde{a})/2\tilde{a} \\ \tilde{q}^2/2 & \tilde{\rho}(\hat{n}_y\tilde{w} - \hat{n}_z\tilde{v}) & \tilde{\rho}(\tilde{u} - \hat{n}_x\hat{U}) & \tilde{\rho}(\tilde{H} + \hat{U}\tilde{a})/2\tilde{a} & \tilde{\rho}(\tilde{H} - \hat{U}\tilde{a})/2\tilde{a} \end{bmatrix} \tag{2.64}$$

while the corresponding characteristic variables $\mathbf{R}^{-1}\Delta\mathbf{Q}$ are

$$\mathbf{R}^{-1}\Delta\mathbf{Q} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} \Delta\rho - \Delta p/\tilde{a}^2 \\ \Delta\hat{U}\cdot\vec{s} \\ \Delta\hat{U}\cdot\vec{t} \\ \Delta p/\rho\tilde{a} + \Delta\hat{U}\cdot\vec{n} \\ \Delta p/\rho\tilde{a} - \Delta\hat{U}\cdot\vec{n} \end{bmatrix} \tag{2.65}$$

in which

$$\Delta\hat{U} = [\hat{n}_x\Delta u,\ \hat{n}_y\Delta v,\ \hat{n}_z\Delta w]^T \tag{2.66}$$

with the quantities $\Delta(\bullet)$ evaluated as a first-order difference, cf. equation (2.48). The newly introduced geometric vector quantities $\vec{n}, \vec{s}$ and $\vec{t}$ also require explanation. The vector $\vec{n}$ is defined as

$$\vec{n} \equiv [\hat{n}_x,\ \hat{n}_y,\ \hat{n}_z]^T \tag{2.67}$$

while the vectors $\vec{s}$ and $\vec{t}$ form an ortho-normal cyclic base with $\vec{n}$, such that $\vec{n} \wedge \vec{s} = \vec{t}$. The components of $\vec{s}$ and $\vec{t}$ are

$$\vec{s} = \left[0,\ \frac{-\hat{n}_z}{\hat{n}_y^2+\hat{n}_z^2},\ \frac{\hat{n}_y}{\hat{n}_y^2+\hat{n}_z^2}\right]^T \tag{2.68}$$

$$\vec{t} = \left[1,\ \frac{-\hat{n}_x\hat{n}_y}{\hat{n}_y^2+\hat{n}_z^2},\ \frac{-\hat{n}_x\hat{n}_z}{\hat{n}_y^2+\hat{n}_z^2}\right]^T \tag{2.69}$$

In coding the expressions for $\vec{s}$ and $\vec{t}$ allowance must be made for the denominator in the second and third components becoming zero, in order to avoid possible floating-point exceptions occuring in computation. The present decomposition of the numerical flux Jacobian models five waves in the solution of the Riemann problem. Their eigenvalues (wave speeds) are given by

$$\begin{aligned}\Lambda &= \text{diag}\ [\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5] \\ &= \text{diag}\ [\bar{U},\ \bar{U},\ \bar{U},\ (\bar{U}+|\mathbf{n}|\tilde{a}),\ (\bar{U}-|\mathbf{n}|\tilde{a})]\end{aligned} \tag{2.70}$$

With the present choice of eigenvectors these waves may be interpreted respectively as an entropy wave, two shear (vorticity) waves and, two acoustic waves. Each is oriented with reference to the local cell face. This completes the description of the chosen spatial discretization for the convective terms of the first-order algorithm. Lastly it should be noted that the inviscid fluxes represent the most computationally intensive part of the present algorithm. Thus, time taken to optimize their coding is well spent.

The issues of entropy-satisfaction and extending the algorithm to second-order accuracy is described below after several known problems with upwind schemes have been discussed.

2.4.4 Known problems with upwind solvers.

As a consequence of the non-uniqueness of the eigenvectors, difficulties can arise and some sets of eigenvectors are a better choice than others. Problems that do arise are often caused by scaling which produces poor numerical conditioning. Yee [188] has observed that in flows of moderate Mach number, different choices of eigenvectors have little effect. However, in hypersonic flows, the magnitude of all variables at the jump of a discontinuity are not the same. For example, jumps in pressure are generally much larger than those of density or total energy. In such circumstances, the choice of eigenvectors can effect the stability and the rate of convergence of the algorithm. In another instance, Forth [45] has reported an apparent deficiency with the eigenvectors given in [20] when used to compute supersonic flow about a cone at zero incidence. The solution to this test case is

expected to exhibit radial symmetry and did so for a first-order calculation but not for a second-order one. It was discovered that the loss of symmetry was due to the spatial orientation of the three eigenvectors corresponding to the linear degenerate waves. These particular eigenvectors are aligned in the directions of the coordinate system and consequently the corresponding wave strengths varied radially around the cone. Combined with the nonlinearity of the flux limiting, this produced a solution exhibiting a non-physical radial variation. Forth cured the problem by formulating a new set of eigenvectors which were chosen to lie in more natural, mutually orthogonal directions.

The eigenvectors that produced the spurious solution [45], distribute the jumps in the characteristic variables corresponding to the linear degenerate waves symmetrically in each coordinate direction. This implies a wave model consisting of three shear waves and two acoustic waves in contrast to the model originally suggested by Roe [127] and indeed, the model implied by equations (2.64) and (2.65). It should also be noted that different choices of eigenvectors may not be dimensionally consistent. At the present time, the implications of different sets of eigenvectors and their implied wave models is an issue that has not been adequately addressed in the literature for grid-aligned upwind solvers. However, van Leer [167] has pointed out that when the actual waves in a solution are far from alignment with the grid, such as the case of a grid-oblique stationary wave, grid-aligned upwind schemes can misrepresent them by an incorrect combination of waves. This is an inherent weakness in the wave models of all grid-aligned upwind solvers. In contrast, with truly multi-dimensional methods, wave models/residual distribution schemes are one of the most active areas of ongoing research. It may be postured that the less rigorous physics of grid-aligned upwind solvers is more forgiving than those of truly multi-dimensional methods. Other failings with some upwind discretizations have been catalogued by Quirk [117]. They include the admission of non-entropy-satisfying solutions such as expansion shocks, the carbuncle phenomenon, post-shock oscillations in slowly moving shock waves, negative internal energies, kinked Mach stems and odd-even decoupling. Here each of these failings will be discussed briefly. However, it should first be noted that not all upwind schemes suffer from these weaknesses. For example, the naturally entropy-satisfying scheme of Osher [107] does not produce expansion shocks nor the carbuncle phenomenon. In addition, Osher's scheme exhibits better behaviour in the slowly moving shock problem than the upwind scheme of Roe [90].

Expansion shocks are mathematically plausible but physically inadmissable solutions that can arise in the solution of upwind methods that are not entropy-satisfying. They are by far the most common failing of Riemann solvers. Their cure is discussed along with one approach to curing the carbuncle phenomenon (a blunt-nose instability that arises for Mach numbers $M > 4$), in the section that follows. Another approach to curing the carbuncle phenomenon is given by Wada and Liou [180], based on switching to a more dissipative solver only at shock transition points.

The post-shock oscillations that arise when Godunov type methods are applied to problems involving slowly moving shock waves were first reported by Roberts [123]. These spurious oscillations arise as low-frequency numerical noise, being symptomatic of the discrete shock structure produced by any scheme with flux functions that give exact shock resolution such as Godunov's or Roe's scheme. Lin

[89] has considered the slowly moving shock problem in some detail and demonstrated an approach which substantially rectifies anomalous solutions, based on selectively increasing the dissipation of the scheme and switching-off the limiter in shock cells that contain sonic points.

Negative internal energies may occur in highly energetic flows in which the dominant energy mode is kinetic rather than thermal. The negative energy leads to negative pressure causing the scheme to fail. Einfeldt *et al.* [36] have considered such cases in some detail. They call methods that do not suffer from this defect *positively conservative.* Godunov's scheme [53] as well as Einfeldt's HLLE scheme [35] are both positively conservative for the Euler equations while Godunov methods based on linearized Riemann solvers are not. However, Einfeldt [35] has demonstrated a way of rectifying non-positive Riemann solvers by simple modification.

Solutions computed by Roe's scheme containing kinked Mach stems were first reported by Quirk [116]. They can arise when the reflection of a plane shock wave from a ramp lies inside the double Mach reflection range. This oddity is caused through the principal Mach stem being aligned obliquely to the grid. In such instances Quirk conjectures that the dissipation in the flux components arising from the contact and shear waves is insufficient to counteract acoustic perturbations.

Quirk [117] was also the first to notice the tendency in an operator-split Roe scheme for odd-even decoupling to occur along the length of planar shock waves which are aligned with the mesh. This phenomenon is only observed in very fine grid solutions with strong shocks which undergo systematic perturbation.

In addition to the above problems, Vinokur [177] has demonstrated that the use of Roe averages can produce an interface value for the square of the speed of sound $\tilde{a}^2$, that is greater than the weighted average of its neighbouring states a_L^2 and a_R^2. It follows that for either of the acoustic waves, it is possible that the Roe-averaged eigenvalue could lie outside the range determined by its neighbouring states. In particular, if the normal velocity is close to sonic in both states, the corresponding eigenvalues could both be of one sign, while the Roe-averaged eigenvalue could have the opposite sign. Vinokur gives three numerical examples that illustrate that this finding is far from hypothetical. So far the implications of this on algorithms that rely on the use of Roe averages, does not appear to have been examined. Clearly the matter warrants further investigation.

Having discussed this apparent catalogue of horrors arising from the use of upwind discretizations, it must be stressed that several of the algorithm failings that have been identified are unique to particular classes of problem. Most of the list apply only to problems involving strong shock waves. With the exception of non-entropy-satisfying solutions, they would not be encountered in transonic flow problems. Furthermore, fixes are available for most of these weaknesses. Finally, it should be stressed that in terms of grid-aligned flow solvers, today upwind discretizations are generally considered to be superior to other available space discretizations, both in terms of accuracy and robustness.

2.4.5 Entropy-satisfying sonic expansions

A disadvantage of Roe's linearization is that the solution resulting from the formulation described so far may not converge to the correct entropy-satisfying

solution in regions of flow containing sonic expansions. Entropy violation arises as a consequence of vanishing wave speed associated with the occurrence of the singularity in the governing equations that occurs at $M = 1$ when flow is expanding. In these regions of flow any of the following phenomena may occur: an expansion shock (i.e. an arbitrarily large discontinuity where otherwise a smooth transition through the sonic point would be expected), a jump (glitch) of magnitude $\mathcal{O}(\Delta x)$, a two-cell plateau (dog leg), or something in between. In addition to being non-unique, solutions containing these defects are often slow to converge. Non-entropy-satisfying solutions may arise when

$$\lambda_{m-1} \leq 0 \leq \lambda_m \tag{2.71}$$

Here, subscripts denote the indices of the pair of cells straddling the sonic region of the expansion in the given coordinate direction. To cure this deficiency so that entropy-satisfying solutions are always obtained, it is necessary to modify the scheme locally in any potentially troublesome sonic region.

Various methods have been devised for this purpose. Broadly speaking they may be divided into two classes, namely empirically based methods generally referred to as *entropy fixes* and the theoretically better-posed *sonic flux formulae*. The former are routinely used for steady-state solutions and involve modifying conditions only at the sonic interface. They are not time accurate and require the specification of empirical coefficients which must be tuned for particular flow regimes. However, they are simple and relatively inexpensive to code. Furthermore, they usually render good solutions particularly for inviscid steady flow problems. On the other hand, sonic flux formulae are time accurate being modelled on the proper decay of the stationary wave. They are non-parametrized and involve modifying the sonic interface flux as well as both neighbouring fluxes. They offer greater accuracy than entropy fixes but require more sophisticated coding and are more computationally intensive. In addition, they offer the prospect of enhanced convergence, being based on the correct rate of decay of sonic gradients. The work of van Leer, Lee and Powell [168] offers valuable insight into the formulation of entropy fixes and presents the first extension of this type of method to inhomogeneous equations. Entropy-satisfaction is approached from a more fundamental standpoint by Roe [132], through the development of accurate sonic flux formulae in one-dimension. However, earlier work by Goodman and Leveque [55] deserves note as it originally suggested the principal ingredient to solving the puzzle by showing that sonic gradients always decay in convex problems. For present purposes, only entropy fixes are considered further. The details of two such methods are now described.

The first method considered here was originally proposed by Harten [64] as a smooth curve fit to the vanishing wave speed in an expansion fan, approximated geometrically by an absolute-value function. Like all entropy fixes, it is based on one-dimensional analysis. The uniqueness of Harten's fit was not appreciated until a rigorous analysis and derivation was presented by van Leer, Lee and Powell [168]. Since the offending waves in a sonic expansion are the acoustic ones, normally the acoustic wave speeds are modified to prevent the wave speed becoming too small. This approach is the basis for all entropy fixes. It has the effect of locally increasing the dissipative flux associated with the nonlinear fields. The Harten/van Leer entropy fix redefines the absolute characteristic speeds $|\lambda_k|$

for the acoustic waves as

$$|\lambda_k^*| = \begin{cases} |\lambda_k|, & |\lambda_k| \geq \frac{1}{2}\delta\lambda_k \\ \frac{\lambda_k^2}{\delta\lambda_k} + \frac{1}{4}\delta\lambda_k, & |\lambda_k| < \frac{1}{2}\delta\lambda_k \end{cases}, \qquad \forall k \in \{\text{acoustic waves}\} \tag{2.72}$$

where

$$\begin{aligned} \Delta\lambda_k &= \lambda_{k,R} - \lambda_{k,L} \\ \delta\lambda_k &= \max(4|\Delta\lambda_k|, 0) \end{aligned} \tag{2.73}$$

For the linear-degenerate waves no modification is required and hence

$$|\lambda_k^*| = |\lambda_k|, \qquad \forall k \in \{\text{linear-degenerate waves}\} \tag{2.74}$$

This particular fix is robust for most flows up to moderate supersonic speeds and may be applied to two- and three-dimensional problems on a dimension by dimension basis without the need for tuning. It also appears to work well for viscous flows in this Mach number range and was used in all the viscous computations presented in section 5.2. A description of its extension to equations involving source terms is given later in section 6.2.3.

For high-supersonic and hypersonic flows over blunt bodies, Roe's approximate Riemann solver can produce an instability, named by researchers at NASA *the carbuncle phenomenon*. The blunt-nose instability was first reported in the literature by Peerey and Imlay [110] who demonstrated a means of curing it by an appropriately formulated entropy fix. Their approach involves using a pressure gradient to tune the magnitude of the dissipation. The second entropy fix used in the present study is simpler than that of Peerey and Imlay, and is based on the Harten fix, modified according to Lin [89], along similar lines to those originally suggested by Yee [188]. To implement this entropy correction, the absolute characteristic speeds $|\lambda_k|$ are redefined for all waves (acoustic and linear degenerate) as

$$|\lambda_k^*| = \begin{cases} |\lambda_k|, & |\lambda_k| \geq \delta_k \\ \frac{1}{2}\left(\frac{\lambda_k^2+\delta_k^2}{2\delta_k} + \lambda_k\right), & |\lambda_k| < \delta_k \end{cases}, \qquad \forall k \tag{2.75}$$

where the entropy parameter is given by

$$\delta_k = \delta_k^*(|\bar{U}| + |\mathbf{n}|\tilde{a}) \tag{2.76}$$

Here $\bar{U}$ is the contravariant velocity in the specified coordinate direction and $\tilde{a}$ is the speed of sound, the flow variables being Roe averaged. The parameter(s) δ_k^* is an empirical constant in the range

$$0 < \delta_k^* < 1 \tag{2.77}$$

which is tuned according to the freestream Mach number. See [89, 188] for recommended values. Theoretically, it should of course be necessary to apply the modification only to the acoustic waves as linearly degenerate waves cannot violate entropy. However, in practice Yee [188] has found that additions to the linear degenerate waves are necessary to exclude the blunt-nose instability. Furthermore, the equations are better conditioned when the technique is applied to all wave fields and convergence improves. For Mach numbers up to $M_\infty \simeq 4$,

δ_k^* is set at a constant value for all waves. However, for higher Mach numbers ($M_\infty > 4$), Lin [89] suggests that the value of the entropy parameter δ_k^* should be set according to the wave type (viz. whether it is acoustic or linear-degenerate). In fact, for very high Mach number flows the value of this parameter for the linear degenerate waves needs to be set quite high to avoid instability arising.

Unfortunately, the addition of dissipation to the linear-degenerate waves via an entropy fix usually degrades the accuracy of viscous flow computations by artificially thickening the boundary layer. To overcome this difficulty, Lin [89, 90] has proposed an improvement to the entropy fix of the Peerey and Imlay [110] which appears to work well. Müller [104] also offers an improved formulation for hypersonic viscous flows in which the entropy correction is constructed as a function of cell aspect ratio. In this way it accounts for the high aspect ratio cells which are common to viscous grids. Finally, Dubois and Mehlman [33] have introduced a non-parametrized entropy fix based on 3-point Hermite interpolation of the sonic flux, which has apparently been used successfully in the computation of inviscid hypersonic real-gas flows.

In the present study, the Harten/van Leer entropy fix given by equations (2.72–2.74) was used for the 3-D transonic inviscid computations and all the viscous computations. For the computations of Mach 4 and above, the formulation given by equation (2.75–2.76) was adopted. For viscous high Mach number computations, the formulation of Lin [90] or that of Müller [104] is recommended. The approach of Dubois [33] also warrants closer scrutiny.

2.4.6 Higher-order spatial accuracy

Godunov [53] was the first to realize that no numerical scheme of second- or higher-order applied to the linear advection equation can preserve monotonicity if its differencing stencil is of constant coefficient. For some time this was widely interpreted as meaning that monotonic second-order solutions to nonlinear PDEs such as those of fluid dynamics could not be produced from any shock capturing algorithm. However, van Leer [162] demonstrated that monotonic solutions are possible provided a nonlinear algorithm is adopted, even in the case of linear problems. As a result of his work and that of others, notably Roe [125, 128], several nonlinear algorithms emerged to afford monotonic second-order solutions to simplified problems. Many of these algorithms have subsequently been extended to sophisticated numerical schemes for the Euler and Navier-Stokes equations.

From an organizational point of view, modern higher-order shock capturing schemes may be divided into *geometric* or *algebraic* schemes, these terms being coined by Goodman and Leveque [55]. In the geometric approach, attempts are made to reconstruct the dependent variables within each control volume, subject to certain monotonic constraints. These constraints allow values of the variables to be constructed at both sides of the interface using a polynomial interpolation of the average values at cell centres. The higher accuracy is thus obtained through a process of data reconstruction such that the original piecewise constant data is replaced by higher-order (piecewise linear, for example) data. Hence these methods are also called *slope limiter methods*. Straightforward application of a numerical method to the reconstructed data usually results in oscillations of the flow variables in the region of discontinuities. To ensure monotonicity, the magnitude of the higher-order terms are bounded or *limited* in regions of large

gradient by the flux limiter function which by nature is nonlinear. The interface flux is then computed. As the data is reconstructed to obtain higher-order accuracy before the flux is obtained, these methods are also referred to generically as *pre-processing methods.* One of the best known schemes in this class is the original MUSCL scheme (standing for Monotonic Upstream-centred Scheme for Conservation Laws) of van Leer, with which geometric (pre-processing) methods have become synonymous [163]. In contrast, most of the algebraic schemes are constructed in a hybrid form using both a low-order flux and a high-order one. These fluxes are blended together according to the local flow conditions again using a flux limiter function. The resulting scheme combines the two fluxes into a single flux so that in smooth regions, the high-order flux is adopted while in non-smooth regions, the low-order flux is used. This is facilitated by expressing the numerical flux as a low-order flux plus a high-order flux correction or *anti-diffusive flux.* The magnitude of the corrective flux is then limited according to the solution data. This is achieved via a nonlinear function $B(r)$ known as a *flux limiter* or *limiter function.* Thus in the vicinity of a discontinuity where greater numerical dissipation is required to maintain monotonicity, the limiter function $B(r)$ will assume its lower-bound of zero producing a final flux of first-order to be used in the solution update. In smooth regions the limiter function will assume its upper bound of say unity, ensuring that the final flux has a sufficiently small amount of dissipation to produce second-order accuracy. (In practice, allowing a wider upper-range for the upper-bound of $B(r)$ often improves flowfield resolution). Since the higher-order reconstruction of the algebraic approach is carried out after the component fluxes have been computed, the approach is also referred to as a *post-processing method.* In the present study, it is the algebraic (post-processing) approach which is adopted.

The success of each approach lies in properly computing $B(r)$ so that the solution from the numerical flux function will satisfy certain desired properties. One such property is called Total Variation Diminishing (TVD) which was introduced by Harten [64]. Defining the total variation of a solution u, at time $(n+1)\Delta t$ to be

$$\mathrm{TV}\left(u^{n+1}\right) = \sum_{k} \left| u_{k+1}^{n+1} - u_{k}^{n+1} \right| \tag{2.78}$$

where the subscripts denote the spatial index and superscripts denote the time level, a TVD scheme is one in which

$$\mathrm{TV}\left(u^{n+1}\right) \leq \mathrm{TV}\left(u^{n}\right) \tag{2.79}$$

Apart from affording a uniform bound on the variation of a solution which is needed in proving convergence for nonlinear schemes, an immediately useful property of any TVD scheme is that it is monotonicity preserving and hence oscillation-free. Harten also derived a set of sufficiency conditions which are important in the construction of second-order TVD schemes. Drawing on TVD theory, Sweby [147] derived some algebraic conditions for limiter functions that guarantee monotonic second-order accurate solutions for homogeneous equations. To achieve this, the limiter is taken to be a function of consecutive flow disturbances or gradients. For systems of equations the limiting may be applied independently to each field that is to be limited. i.e.

$$B \equiv B\left(r_k\right) \tag{2.80}$$

where the subscript k denotes the given field. Typically the argument is defined to be the ratio of the gradient at the local cell-interface to its upstream neighbour and is computed as

$$r_k = \frac{\Delta u(k)_{i+1/2-\sigma_k}}{\Delta u(k)_{i+1/2}} \tag{2.81}$$

where $\Delta u(k)$ is a representative flow gradient of the kth field. The quantities σ_k, needed to determine the counter of the upstream interface for each field, are determined as

$$\sigma_k = \text{sgn}\ (\lambda_k)$$

in which λ_k is the eigenvalue of the kth field and sgn $(\lambda_k) = \pm 1$ depending upon the direction (sign) of the given wave. The nonlinear limiter function itself is best chosen so that the magnitude of the second-order component of the numerical flux function is maximized, subject to the TVD constraints. For a TVD scheme to be second-order accurate, its limiter function must lie in the shaded region shown in Figure 2.5. The parameter space shown is that used by Sweby and Roe

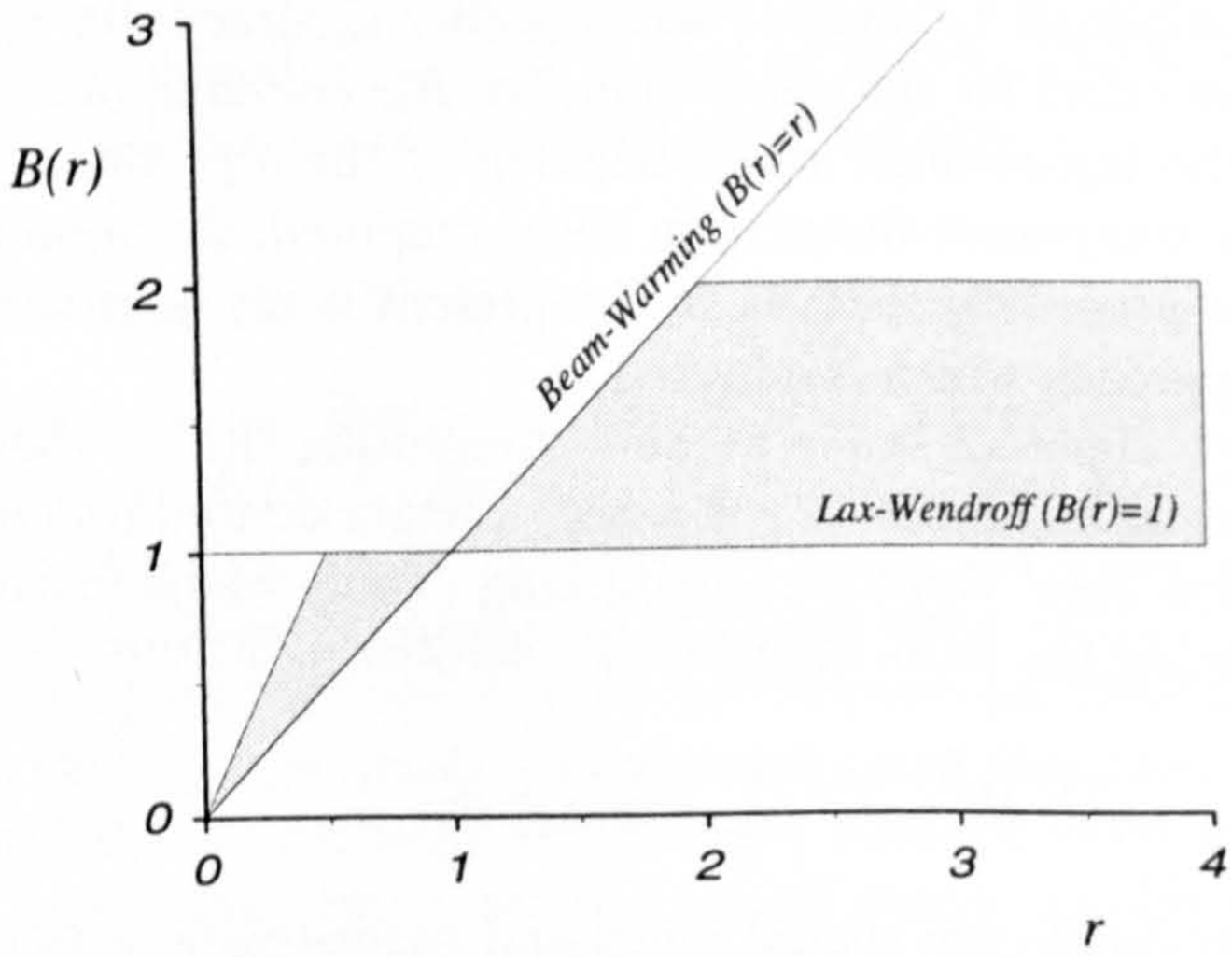


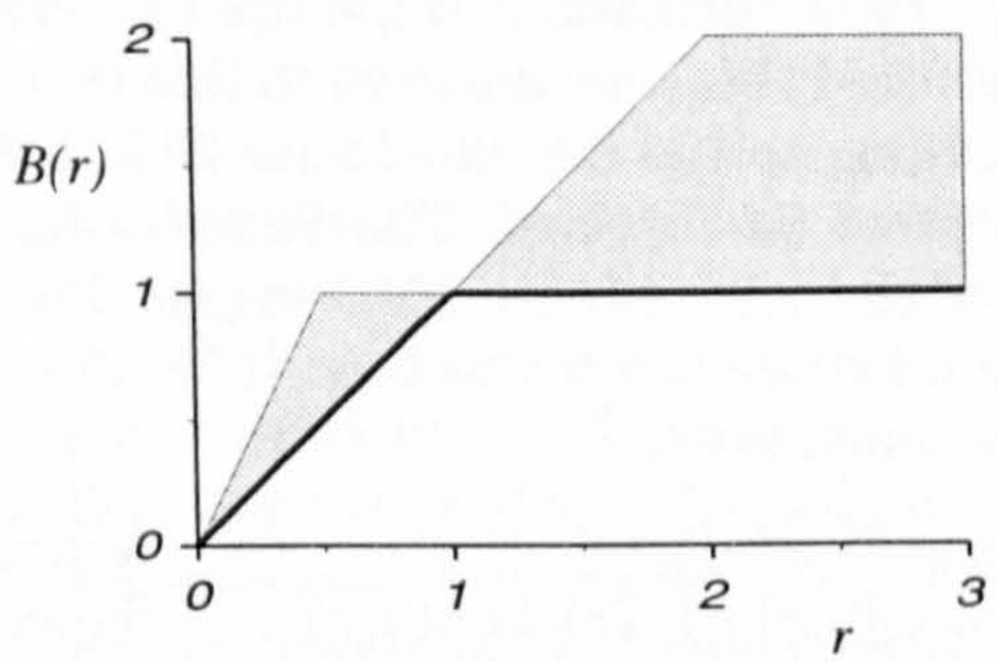
FIGURE 2.5. Second-order TVD region.

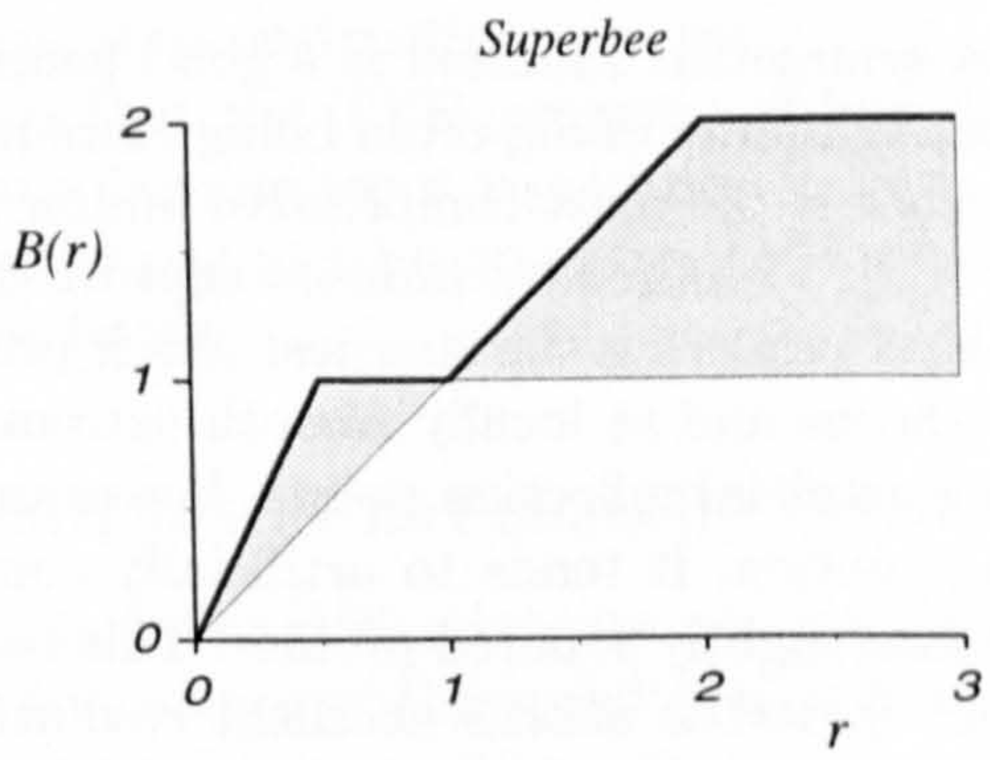
to construct limiter functions. The figure also shows where the classical Lax-Wendroff and the Beam and Warming schemes lie. It is apparent from the figure that a general requirement for second-order accuracy is that the function must pass through the point $B\,(r_k) = 1$. The symmetry property

$$B(r_k) = r_k B(1/r_k) \tag{2.82}$$

is also important for limiter functions. By this it is meant that the limiter function should return the same value when applied to consecutive gradients, irrespective of the flow direction.

Three common limiter functions are illustrated in Figure 2.6. The first two





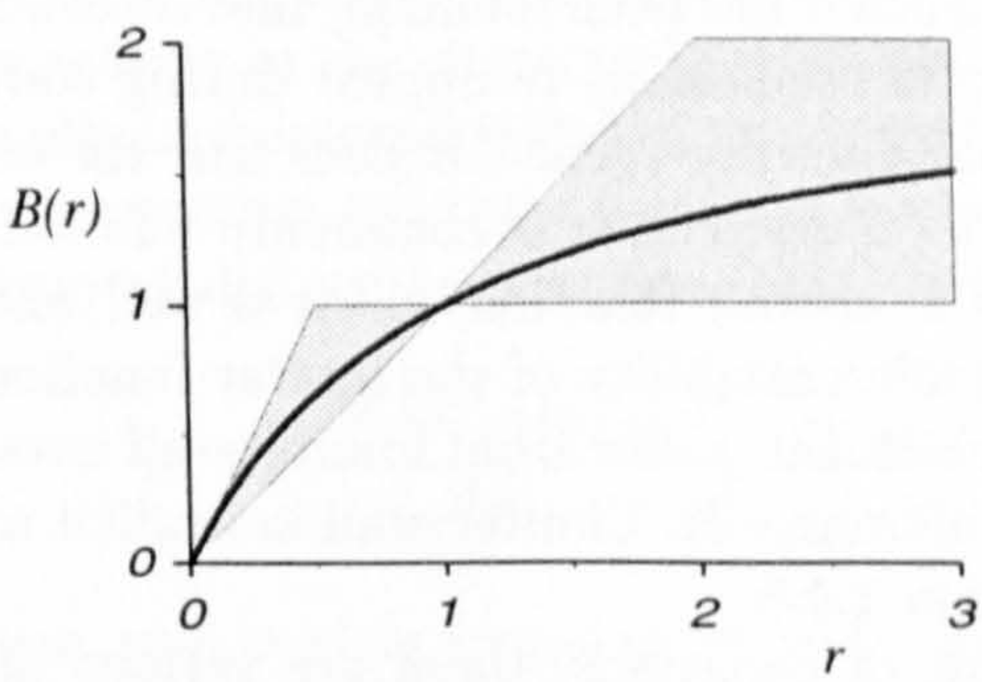


FIGURE 2.6. Common flux limiter functions.

were introduced by Roe [128] and are referred to as *Minmod/Minbee* and *Superbee*. Minmod corresponds to the lower boundary of the second-order TVD region of the previous figure. It may be recognized as being equivalent to the Beam and Warming scheme in one region ($B = r, \forall\ r \leq 1$) and equivalent to the Lax-Wendroff scheme in another ($B = 1, \forall\ r > 1$). Since the Beam and Warming scheme produces pre-shock oscillations while the Lax-Wendroff scheme produces post-shock oscillations, it became apparent to Roe that by combining both schemes adaptively according to the this particular B function, oscillation-free capture of shock waves could be obtained. The *Superbee* function is equivalent to the upper boundary of the second-order TVD region. The third limiter illustrated is a differentiable function, due to van Leer [162] after which it is named. The definitions for these functions are

$$\begin{aligned} B(r) &= \max\left[0, \min(1, r)\right] && \text{Minmod} \\ B(r) &= \max\left[0, \min(1, 2r), \min(2, r)\right] && \text{Superbee} \\ B(r) &= (r + |r|)/(1 + |r|) && \text{van Leer} \end{aligned} \tag{2.83}$$

Each of these limiters is symmetric. Minmod is a good general purpose limiter but it is the most dissipative limiter of all, often being more restrictive than necessary. In contrast Superbee is the most compressive limiter having the highest upper bound ($B(r)_{\max} = 2$). Compressive limiters capture discontinuities with fewer transition points thus rendering the sharpest shock profiles. However, for moving and dissipating shocks and at locally smooth extrema Superbee can be too compressive, creating artificial inflection points. For example, when applied to the linear advection equation, it tends to artificially compress the smooth peaks of sine wave data into slightly squared profiles. This behaviour is referred to as 'clipping'. However, Superbee affords excellent resolution of contact discontinuities.

Continuous differentiability of the limiter function is a necessary property for use in *implicit* schemes since it has been found [3] that discontinuous limiter functions can cause limit-cycle oscillations to appear during convergence. Although this behaviour cannot produce divergence, it does halt the expected drop in the trace of the l_2 norm. The phenomenon is commonly referred to as *limiter stall*. The van Leer limiter is a smooth function which is well suited to implicit and explicit schemes alike. Differentiability of the limiter function is less of an issue with explicit schemes which can suffer from limiter stall irrespective of whether the limiter function is differentiable. Limiter stall in explicit schemes is discussed in greater detail in section 2.4.8.

In the case of systems of equations, there are various plausible candidates for the limiter argument r_k, or more specifically the quantity $\Delta u(k)$. Choices include the conserved variables, the primitive variables or a characteristic variable. In this study, the wave flux components $\lambda_k \alpha_k r_{1,k}$, have been used, where $r_{1,k}$ represents the first element of each eigenvector k corresponding to density changes. In addition, the characteristic variables $\mathbf{R}^{-1}\Delta\mathbf{Q}$ have also been used. These particular limiter arguments give comparable flowfield resolution to one another and are believed to afford superior accuracy to either the conserved or primitive variables. Furthermore, it is not necessary to use the same limiter function throughout. For example, if the limiting is based on characteristic quantities, one could use Superbee on shear waves and contact discontinuities and a differentiable limiter for the acoustic fields.

Although geometric schemes are not considered in this study, they tend to be more popular in the literature than ones based on the algebraic approach. This is in part due to the fact that through truncation error analysis it can be shown that certain geometrically reconstructed schemes, for example ones based on van Leer's $\kappa = 1/3$ MUSCL reconstruction, are formally third-order accurate in a volume-averaged sense. Improved accuracy is also claimed for the latter on the basis that the reconstruction is based on the conserved variables which are smoother than the split fluxes used in post-processing approaches. However, whether these arguments translate into perceptible improvements in accuracy over algebraic approaches has yet to be quantifiably demonstrated. The available evidence, which includes comprehensive comparative studies by Yee [188] and by Yang and Przekwas [187] does not show this. In fact the latter suggests that the MUSCL scheme produces discontinuities that are a little more dissipative than the second-order Roe/Sweby TVD formulation. Perhaps the most pressing argument in favour of the present approach is that it is more computationally efficient than an equivalent MUSCL formulation which uses a comparatively expensive reconstruction process.

Finally, it is noted that the TVD property of flux limited methods do not necessarily apply to inhomogeneous equations. In fact, equations with source terms can admit oscillations in solutions from TVD schemes, particularly in the case of strong source terms.

2.4.7 Flux limiting in hypersonic flows

For flows where $\dot{M}_\infty > 4$, Yee [188] advocates scaling the argument of the limiter function $B(r_k)$, by the square of the speed of sound a^2. Yee reasons that as a consequence of the non-uniqueness of any given choice of eigenvectors, this further measure improves eigenvector scaling in high Mach number regions where scaling disparities become large. When this additional factor is included in the limiter argument, variations in the characteristic variables become proportional to pressure. This results in enhanced stability and convergence for these type of problems. Thus, for hypersonic solutions following the present limiter formulation, the limiter argument defined by equation (2.81) is replaced by

$$r_k = \frac{\Delta u(k)_{i+1/2-\sigma_k}\, \tilde{a}^2_{i+1/2-\sigma_k}}{\Delta u(k)_{i+1/2}\, \tilde{a}^2_{i+1/2}} \tag{2.84}$$

This limiter argument was used in conjunction with $\Delta u(k) \equiv \lambda_k \alpha_k r_{1,k}$ in the present study for the hypersonic solutions. Note that $\tilde{a}^2$ is a Roe-averaged value. In addition, special treatment of the entropy fix was also required for such computations, as discussed in section 2.4.5.

2.4.8 Limiter stall

Convergence is normally attained by first-order upwind schemes. However, when higher-order upwind schemes that rely on flux limiters are used, global convergence is not always obtained. The author has found that this occurs particularly in subsonic flows or mixed flows, viz. flows containing subsonic regions or pockets yet not in the solution of fully hyperbolic problems (cf. Figure 3.5). However,

Venkatakrishnan [174] has reported its occurrence in fully supersonic solutions using an unstructured grid upwind solver. It has also been observed that when there is a subsonic region in a solution, the higher the freestream Mach number, the lower the residual drop, prior to difficulties being encountered. Inspection of the residual trace (l_2 norm) of such solutions show a characteristically shallow drop, typically of $1-4$ orders of magnitude before bottoming, usually into high-frequency oscillations, often indicative of a limit cycle. This type of convergence pathology is commonly termed *limiter stall*.

Limiter stall is caused by the nonlinear process of limiting which hampers convergence to a steady-state. The problem builds up in smooth regions of flow where gradients are weak, well after the solution has attained its basin of attraction. In these regions, the field values of the limiter function which regulate the anti-diffusive flux develop small localized fluctuations. They affect neither the establishment of the evolving flow structure nor the global stability of the solution as they are small and bounded in magnitude. Physically, the phenomenon represents very low-amplitude, high-frequency oscillations in the solution which are akin to numerical noise and confined to regions of flow in which little is happening. This behaviour is driven by a nonlinear resonance between the individual values returned from the limiter function from smooth regions. Consequently, the limiting cannot produce a unique set of values throughout the solution domain. This does not imply that the solution is not adequately converged. However, more caution is needed in assessing whether a solution is sufficiently converged and in particular, other parameters must be checked for convergence. For example, it should be ascertained whether the number of subsonic/supersonic points have settled to a constant value and whether the force coefficients (lift and drag) are still changing.

Increasing grid resolution worsens the effect of limiter stall. In fine grid solutions the residual drop prior to stall is less than the drop encountered in coarse grid solutions. This has been noted by Kroll, Gaitonde and Aftosmis [80]. Ordinarily, the drop in the residual before stall is greater when a low Courant number is used. However, when the support of a scheme is increased, a high Courant number can be used without such adverse effect on convergence. As a consequence, implicit schemes tend to suffer less than explicit schemes [80]. Similarly, residual smoothing usually postpones the stall of explicit schemes by about an order of magnitude or so.

With explicit schemes, the standard limiter functions have little effect on limiter stall, though compressive limiters can cause the stall to occur slightly more prematurely than would be the case with more dissipative limiters. Likewise when more than one limiter is used, (for example, when limiter A is used for acoustic fields and limiter B for linear-degenerate fields), the stall usually occurs sooner. From the latter observation it is speculated that compounding the already nonlinear limiting process by more than one limiter increases the resonant behaviour that drives the stall. As noted above, differentiability of a limiter function does not appear to have much bearing on the stall characteristics of explicit schemes, though Anderson, Thomas and van Leer [3] have shown that it can prevent stall occuring in the convergence of implicit schemes.

Cures for limiter stall are known. The simplest is an empirical fix known as *limiter freezing* which has been used by De Zeeuw and Powell [29]. Once a global quantity, such as the lift coefficient has converged, the idea is to freeze

the limiter values using previously stored values rather than recompute new ones for subsequent timesteps. This generally results in the residuals dropping to machine zero. However, according to Venkatakrishnan [176], the effectiveness of the technique is problem-dependent and the transition from limiting to non-limiting situation should be smooth. Furthermore, if the limiters are frozen too early, the solution will lock into a spurious state rather than the true steady-state solution.

A more satisfactory remedy for limiter stall was introduced by Venkatakrishnan, both for structured [175] and unstructured [174, 176] upwind schemes. Following van Albada [160] who originally modified a differentiable limiter function for the purpose of preventing the clipping of smooth extrema, Venkatakrishnan applied the same modified limiter to the limiter stall problem, for which it has been found to work equally well. The modification effectively turns off the limiter when oscillations are below a certain threshold. This threshold is set by a parameter ϵ which is incorporated into the limiter function and taken to be $\epsilon^2 = \mathcal{O}(\Delta x^n)$ where Δx is an average measure of cell width, and $n > 2$. Venkatakrishnan also suggests that the differentiability of a limiter is a desirable property for devising stall-free limiters. Jameson [71] has also devised stall-free limiters for some new upwind/upwind-biased schemes which he calls *soft limiters*. Again they are based on including a threshold parameter in the limiter function. In Jameson's formulation, this threshold is made proportional to a representative local quantity in the far field, chosen for consistency with the dimensions of the eigenvectors. Jameson claims that as well as preventing limiter stall, these new limiters improve accuracy and result in a faster rate of convergence. Although the Jameson and Venkatakrishnan limiters are devised specifically for MUSCL schemes, it is likely that the ideas used in their construction can be applied to schemes such as the present one, which are based on algebraic limiting methods.

For present purposes, no mechanisms were used to prevent limiter stall: convergence was assessed by whether the number of supersonic points in the flowfield and the lift and drag coefficients had stabilized to near-constant values.

2.4.9 A SECOND-ORDER NUMERICAL FLUX FUNCTION

The form that a second-order numerical flux function takes depends on the type of time update to be used. For example, a Lax-Wendroff type scheme, sometimes referred to as an operator-split scheme, has a different flux function to a semi-discrete (method of lines) formulation. Here the approach adopted is the latter, for which a second-order generic flux is given by

$$\hat{\mathbf{E}}_{m+\frac{1}{2}}(\mathbf{Q}_m, \mathbf{Q}_{m+1}) = \frac{1}{2}\left[(\hat{\mathbf{E}}_m + \hat{\mathbf{E}}_{m+1}) - \{1 - B(r_k)\}\,\mathbf{R}\,|\Lambda|\,\mathbf{R}^{-1}\Delta\mathbf{Q}\right] \quad (2.85)$$

In smooth regions, $B(r_k) \rightarrow B(r_k)_{\max}$ and the scheme has the full accuracy associated with the underlying higher-order scheme which in this case is that of a biased central differencing approximation of three point support. In non-smooth regions of flow, $B(r_k) \rightarrow 0$ and the numerical flux reduces to first-order, ensuring monotonicity, cf. equation (2.57).

This form of interface flux is superficially similar to central differencing schemes based on scalar dissipation [76] in as much as the flux functions of both consist of a central difference and separate 'dissipation' term. However, scalar dissipa-

tion methods are typically based on second- and fourth-order differences of the conserved variables which are each factored adaptively by empirically-tuned constants. In contrast, the above flux function is based on a matrix valued decomposition of the flux Jacobian which is naturally dissipative and does not require any user-defined parameters. Finally, it should be noted that the above flux formulation is formally second-order accurate only in the case of uniformly spaced mesh. Nevertheless, it is well known that second-order accuracy is retained in the case of mesh that vary smoothly.

2.5 Discretization of Diffusive Terms in 3-D

2.5.1 Viscous flux discretization

The discretization of the viscous terms of the Navier-Stokes equations requires an approximation to the velocity derivatives in order to compute the stress tensor τ_{ij}, defined by equations (2.8) so that the viscous fluxes of equations (2.16–2.18) may be evaluated. For these terms, a central discretization is used. Several approaches to discretizing the diffusive terms may be found in the literature, [66, 21, 97]. The approach used here is a conservative technique, similar to that used in [97] which ensures second-order accuracy on smoothly varying mesh.

Using Gauss' theorem, the average value of the velocity gradient may be evaluated as a function of the boundary values of a control volume. Thus, for an arbitrary volume $\mathcal{V}$ with boundary $\mathcal{S}$,

$$\int_{\mathcal{V}} \nabla . u \, d\mathcal{V} = \oint_{\mathcal{S}} u \mathbf{n} \, d\mathcal{S} \tag{2.86}$$

where $\mathbf{n}$ is the outward normal. For a hexahedral cell this gives

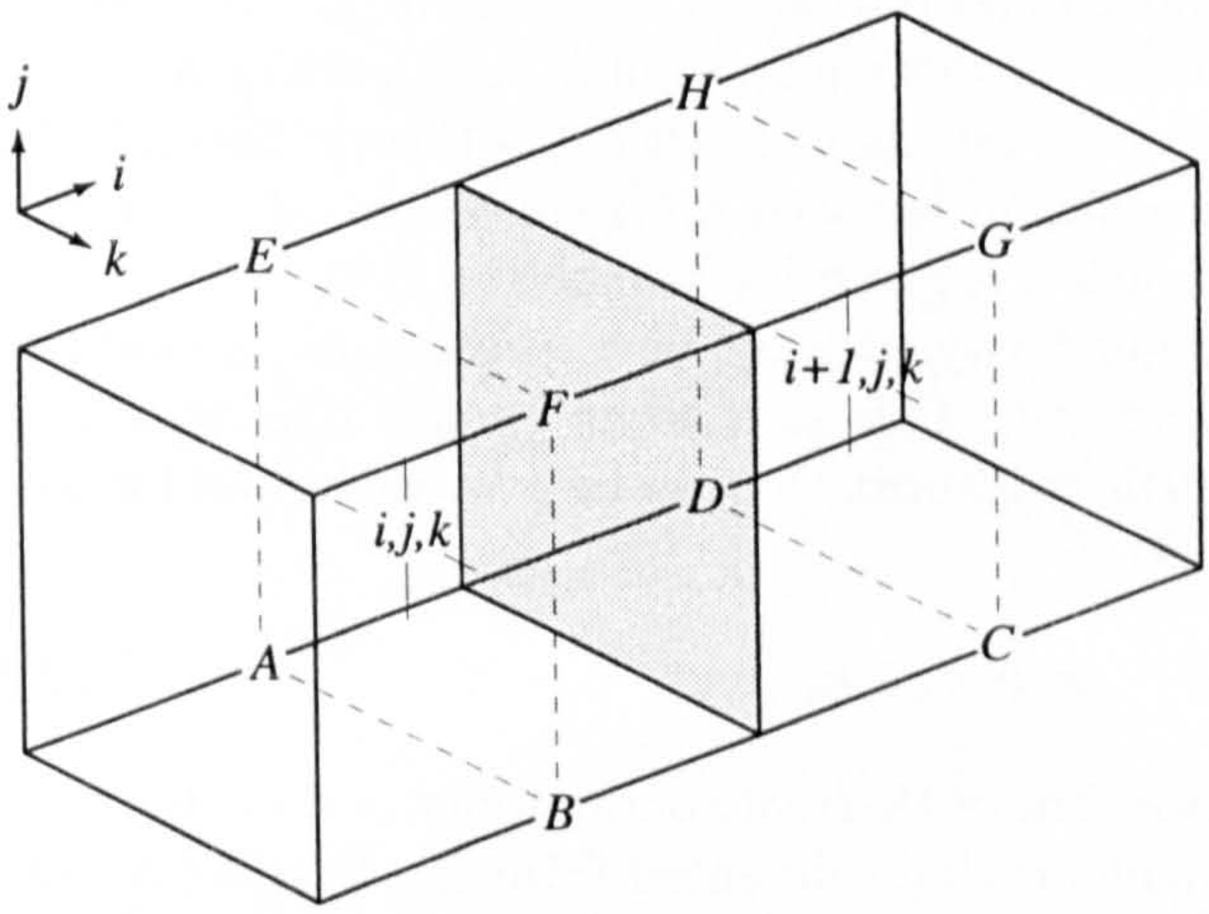
$$\nabla . \bar{u} = \frac{1}{\mathcal{V}} \sum_{k} \bar{u}_i \mathbf{n}_k \mathcal{S}$$

where $\bar{u}_i$ is an estimate of the average value of u_i over a face and, the summation is over the k faces bounding the control volume. The control volume is chosen to contain the interface at which a particular viscous flux is required. A convenient choice of control volume is given by the dual volume illustrated in Figure 2.7, which is shown enclosing a shaded interface in the ξ-direction. The interface is shared by the pair of cells with centres at i, j, k and $i+1, j, k$. The dual volume, to be denoted $\mathcal{V}_{dual}$, is bounded by the lettered corners $A - H$. The figure also shows the logical counters for the corner points. The locations of these corner points are determined by arithmetic averaging. For example, the coordinate of a generic point in the ξ-direction would be

$$\xi_{i,j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{2}\left(\xi_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \xi_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right) \tag{2.87}$$

with similar expressions for the other points. The surface integral may now be evaluated to obtain the necessary velocity gradients as

$$\nabla . \bar{u} \;=\; \frac{1}{\mathcal{V}_{dual}} \sum_{k=1}^{6} \bar{u}_i \mathbf{n}_k \mathcal{S}$$



A:	i,	j-1/2,	k-1/2
B:	i,	j-1/2,	k+1/2
C:	i+1,	j-1/2,	k+1/2
D:	i+1,	j-1/2,	k-1/2
E:	i,	j+1/2,	k-1/2
F:	i,	j+1/2,	k+1/2
G:	i+1,	j+1/2,	k+1/2
H:	i+1,	j+1/2,	k-1/2

FIGURE 2.7. The dual volume used in the computation of the diffusive fluxes.

$$
\begin{aligned}
= \quad & \Big[u_{i+1,j,k}\ \mathbf{i}.\vec{\mathcal{S}}_{DCGH} + u_{i,j,k}\ \mathbf{i}.\vec{\mathcal{S}}_{ABEF} + \\
& \bar{u}_{i+\frac{1}{2},j+\frac{1}{2},k}\ \mathbf{i}.\vec{\mathcal{S}}_{EFGH} + \bar{u}_{i+\frac{1}{2},j-\frac{1}{2},k}\ \mathbf{i}.\vec{\mathcal{S}}_{ABCD} + \\
& \bar{u}_{i+\frac{1}{2},j,k+\frac{1}{2}}\ \mathbf{i}.\vec{\mathcal{S}}_{BCGF} + \bar{u}_{i+\frac{1}{2}j,k-\frac{1}{2}}\ \mathbf{i}.\vec{\mathcal{S}}_{ADHE}\Big]\ /\mathcal{V}_{dual}
\end{aligned}
\tag{2.88}
$$

The interface values of $\bar{u}$ are approximated as

$$
\begin{aligned}
\bar{u}_{i+\frac{1}{2},j-\frac{1}{2},k} &= (u_{i,j-1,k} + u_{i+1,j-1,k} + u_{i,j,k} + u_{i+1,j,k})/4 \\
\bar{u}_{i+\frac{1}{2},j+\frac{1}{2},k} &= (u_{i,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i+1,j+1,k})/4 \\
\bar{u}_{i+\frac{1}{2},j,k-\frac{1}{2}} &= (u_{i,j,k-1} + u_{i,j,k} + u_{i+1,j,k-1} + u_{i+1,j,k})/4 \\
\bar{u}_{i+\frac{1}{2},j,k+\frac{1}{2}} &= (u_{i,j,k} + u_{i,j,k+1} + u_{i+1,j,k} + u_{i+1,j,k+1})/4
\end{aligned}
\tag{2.89}
$$

As an example, the gradient $\partial\bar{u}/\partial z$ would be evaluated from equation (2.88) as

$$
\begin{aligned}
\frac{\partial\bar{u}}{\partial z} = \quad & \big[u_{i+1,j,k}k_{z_{DCGH}} - u_{i,j,k}k_{z_{ABEF}} + \\
& \bar{u}_{i+\frac{1}{2},j+\frac{1}{2},k}l_{z_{EFGH}} - \bar{u}_{i+\frac{1}{2},j-\frac{1}{2},k}l_{z_{ABCD}} + \\
& \bar{u}_{i+\frac{1}{2},j,k+\frac{1}{2}}m_{z_{BCGF}} - \bar{u}_{i+\frac{1}{2}j,k-\frac{1}{2}}m_{z_{ADHE}}\big]\ /\mathcal{V}_{dual}
\end{aligned}
\tag{2.90}
$$

where $k_{x,y,z}, l_{x,y,z}, m_{x,y,z}$ are the surface normals defined in section 2.3.4. Thus, $l_{z_{EFGH}}$ represents the surface normal of the face $EFGH$ in the j-direction. The evaluation of all the other gradients follow analogously. Finally, it should be noted that the temperature gradients $\partial T/\partial x_i$, needed to evaluate the heat fluxes of equation (2.9), may be found in an equivalent fashion to the velocity gradients.

2.5.2 THE DIFFUSIVE COEFFICIENTS

The diffusive coefficients considered here are those introduced when the governing equations were first presented in section 2.2.2. They comprise the coefficient of thermal conductivity κ and the viscous coefficients λ^v and μ which are related to the thermodynamic variables by means of kinetic gas theory. Several of the assumptions on which their derivation is based are semi-empirical in nature. A thorough account of their derivation is given by Schlichting [139].

Beginning with thermal conductivity, it may be expressed as the sum of a laminar and a turbulent component. This is a common practice done to enable the mechanisms of turbulent transport to be correctly accounted for in the Reynolds-averaged equations. Thus

$$\kappa = \kappa_l + \kappa_t \tag{2.91}$$

in which κ_l is the laminar component of thermal conductivity and κ_t, the turbulent component. Each component may be calculated from the Prandtl Number, Pr which is defined as

$$\mathrm{Pr} = \frac{\mu c_p}{\kappa} \tag{2.92}$$

in which c_p is the specific heat at constant pressure. To do this, the Prandtl number is also expressed in terms of laminar and turbulent flow components. Furthermore, since the ratio c_p/Pr is approximately constant for most gases, the thermal conductivity coefficient κ may be calculated as

$$\kappa = \frac{c_p \mu}{\mathrm{Pr}_l} + \frac{c_p \mu}{\mathrm{Pr}_t} \tag{2.93}$$

For air, the laminar Prandtl Number Pr_l is usually assumed to have a value of 0.72 while its turbulent counterpart Pr_t is assumed to have a value of 0.9. These are the values adopted in the present work.

Now turning to the evaluation of the viscosity coefficients, it will be remembered that the viscous shear stresses, introduced in section 2.2.2, are obtained under the condition of flow equilibrium. They are rewritten here, this time in tensor notation as

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial u_j} + \frac{\partial u_j}{\partial u_i} - \lambda^v \delta_{ij} \Delta \right) \tag{2.94}$$

where

$$\Delta = u_x + v_y + w_z \quad , \quad \delta_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j \end{cases} \tag{2.95}$$

The viscous coefficients μ and λ^v are related to the Bulk viscosity μ' through the relationship

$$\mu' = \frac{2}{3}\mu + \lambda^v \tag{2.96}$$

With the aid of the Stokes hypothesis advanced by G.G. Stokes in 1845, the Bulk viscosity may be set to zero for flows in which the divergence of velocity Δ does not vanish identically. This allows the coefficient λ^v to be determined as

$$\lambda^v = \frac{2}{3}\mu \tag{2.97}$$

It is commonly accepted that provided $\Delta \neq 0$, the Stokes hypothesis is always satisfied.

Next, consider the viscosity coefficient μ. It will be recalled from section 2.2.2 that with Reynolds averaging this becomes the effective viscosity coefficient μ_{eff} and that it is broken down as

$$\mu = \mu_l + \mu_t \tag{2.98}$$

where the component μ_l is the laminar viscosity or *molecular viscosity* coefficient and μ_t is the turbulent viscosity coefficient, often referred to as the *eddy viscosity*. For laminar flows the turbulent viscosity coefficient is set to zero. The molecular viscosity is determined readily from the empirically derived Sutherland formula which is given by

$$\mu_l = \mu_0 \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_0 + S_1}{T + S_1} \tag{2.99}$$

where μ_0 denotes the viscosity at a reference temperature T_0, and S_1 is a constant which for air assumes the value $S_1 = 110.3$. The expressions needed for the final diffusive coefficient μ_t are given in section 2.5.3.

2.5.3 Closure for turbulent flow solutions

In this study, the Reynolds-averaged equations, already discussed in section 2.2.2, are the starting point for computing turbulent flow solutions. Their use implies the need for closure since the relationships or correlations between the mean flow quantities and the viscous stresses and turbulent heat diffusion are unknown. As such they need to be modelled. The information used in this modelling is based on theoretical considerations inescapably coupled with empirical assumptions. It is contained in what has become known as *turbulence models*. It is these models that are added to the Reynolds-averaged equations to effect closure of the system.

Turbulence modelling is an area of ongoing activity and debate. Many different models have been developed, ranging from simple algebraic ones to sophisticated second-order closure models. One of the most significant difficulties encountered with turbulence models is their limited range of applicability. For example, some afford excellent solutions for thin, attached shear layers but cannot cope with separation. Others model low Reynolds number flows accurately but are inadequate for high Reynolds number flows, etc. For this reason, the more sophisticated industrial CFD design codes have several turbulence model options so as to suit a range of diverse flow conditions. For a comprehensive overview of the issues involved in turbulence modelling together with a description of a wide range of models, refer to the text by Wilcox [184].

In the second-order closure models, transport equations for the second-order correlation terms are deduced from the Navier-Stokes equations while the third-order correlation terms are modelled as a function of the second-order correlations. The computational effort for these type of models is great and can obviated for some flows by adopting instead the simpler first-order models. These lower-order models often provide an acceptable approximation for the influence of turbulent transport and diffusion on the mean variables. First-order models may be classified according to the number of additional transport equations they require

for the turbulent quantities. Methods that use one or two additional differential equations are called one-equation and two-equation models. Zero-equation models, generally referred to as algebraic models, do not involve any differential equation for the turbulence quantities and are therefore the simplest to use. One of the most popular algebraic turbulence models is that due to Baldwin and Lomax [5]. In the present work, it is this turbulence model that has been implemented and used for the turbulent flow results presented.

2.5.4 The Baldwin-Lomax turbulence model

Most algebraic turbulence models are based on the original two-layer model developed by Cebeci and Smith [18], for boundary layer applications. A modified formulation of this model was adapted for Navier-Stokes solutions by Baldwin and Lomax [5], after whom it is named. The Baldwin-Lomax turbulence model has been used extensively in the computation of thin, attached shear layers in transonic flows, producing good results. It has been widely accepted as a standard model for both 2- and 3-D applications up to and including the onset of separation. However, like most algebraic models, as separation of the boundary layer is approached, its predictive accuracy can degrade.

The model considers the turbulent boundary layer to be composed of two layers or regions, an inner and an outer region, each having individual expressions for the eddy viscosity coefficient μ_t. The inner region viscosity is assumed to be contained within the attached boundary layer while the outer region viscosity is assumed to lie outside it. For every point within the flow, these two viscosities are calculated. The model then uses the logical criterion

$$\mu_t = \begin{cases} (\mu_t)_{inner} & y_n \leq (y_n)_{crossover} \\ (\mu_t)_{outer} & y_n > (y_n)_{crossover} \end{cases} \tag{2.100}$$

to chose which of the two viscosities $(\mu_t)_{inner}$ and $(\mu_t)_{outer}$, is appropriate to the point under consideration. In the above expression, y_n is the normal distance to the wall while $(y_n)_{crossover}$ represents the value of y_n at which the inner and outer viscosities crossover in value. The model assumes that the crossover position at which the viscosities are equal, corresponds to the edge of the inner region of the boundary layer. In practice, it is improbable that the two values are exactly equal and so $(y_n)_{crossover}$ is taken to be the first point at which $(\mu_t)_{inner}$ is greater than $(\mu_t)_{outer}$. The turbulent viscosity μ_t so found, is then added to its laminar counterpart μ_l, as prescribed by equation (2.98), to produce the effective viscosity μ to be used in the governing equations (2.1,2.3–2.5,2.8).

In the remainder of the section, the formulations for the inner and outer regions are given, after which the treatment of wakes will be considered. Finally, improvements to the basic model will be discussed briefly.

The Inner Formulation

The formulation in the inner region follows the Cebeci-Smith [18] model. For this region, it provides a semi-empirical expression for $(\mu_t)_{inner}$ through the mixing length l. This is given by

$$(\mu_t)_{inner} = \rho l^2 |\omega| \tag{2.101}$$

in which ρ is the density and $|\omega|$ is the magnitude of the vorticity, i.e.

$$|\omega| = \sqrt{\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right)^2} \tag{2.102}$$

In the mixing length formula (which can be explained as a model of the shear-stress transport equation neglecting all the transport terms) empirical information is inserted via the mixing length itself. Only if the transport terms really are small (as in the log-law region) is the mixing length related to a true eddy scale; otherwise, it is just a quantity to be correlated empirically. The mixing length formula used in the present model is given by the classical representation of Prandtl ($l = Ky_n$), corrected by van Driest through the addition of an exponential term:

$$l = Ky_n\left(1 - e^{-\frac{y^+}{A^+}}\right) \tag{2.103}$$

where K is the von Kármán constant and A^+ is known as the van Driest constant. The latter is determined through calibration with boundary layer data. The variable y^+ is variously termed the law of the wall coordinate and it is defined by

$$y^+ = \frac{\rho_w u_\tau y_n}{\mu_w} = \frac{y_n\sqrt{\rho_w \tau_w}}{\mu_w} \tag{2.104}$$

In this definition, u_τ is the friction velocity, τ is the shear stress and quantities subscripted by w correspond to wall values.

The Outer Formulation

The eddy viscosity in the outer region is defined by

$$(\mu_t)_{outer} = C\beta\rho\Gamma_{wake}F_{kleb}(y_n) \tag{2.105}$$

where C is the Clauser constant, β is another constant and Γ_{wake} and $F_{kleb}(y_n)$ are functions of the flow variables. The function Γ_{wake} is defined by

$$\Gamma_{wake} = \min\left\{y_{\max}\Gamma_{\max}\ ,\ \frac{C_{wake}y_{\max}U_{diff}^2}{\Gamma_{\max}}\right\} \tag{2.106}$$

in which C_{wake} is the wake constant while the value of U_{diff} is the difference between the maximum and minimum total velocity at the station in the profile that is under consideration. i.e.

$$U_{diff} = \left(\sqrt{u^2+v^2+w^2}\right)_{\max} - \left(\sqrt{u^2+v^2+w^2}\right)_{\min} \tag{2.107}$$

The second term $(\bullet)_{\min}$ in this expression is set to zero unless the function Γ_{wake} is being evaluated in a free shear-layer. The value $\Gamma_{\max}$ is determined from the function

$$\Gamma(y_n) = y_n|\omega|\left(1 - e^{-\frac{y^+}{A^+}}\right) \tag{2.108}$$

to be the first maximum of $\Gamma(y_n)$ in the boundary layer. Likewise, $y_{\max}$ is the corresponding value of y_n at which $\Gamma_{\max}$ is attained. Finally, $F_{kleb}(y_n)$ is defined

as the intermittency function which is given by

$$F_{kleb}(y_n) = \left[1 + 5.5\left(\frac{C_{kleb}y_n}{y_{max}}\right)^6\right]^{-1} \tag{2.109}$$

where C_{kleb} is the Klebanoff constant.

The constants appearing in each of the foregoing relations are prescribed to be consistent with the Cebeci formulation [17] for constant pressure boundary layers at transonic speeds. Their values (for air) are given in Table 2.1

Name	*Symbol*	*Value*
von Kármán constant	K	0.4
van Driest constant	A^+	26
Clauser constant	C	0.0168
Outer region constant	β	1.6
wake constant	C_{wake}	0.24
Klebanov constant	C_{kleb}	0.3

Table 2.1: Baldwin-Lomax turbulence model constants

Treatment of wakes

One of the drawbacks of all algebraic models and indeed the vast majority of higher-order turbulence models, is their reliance upon wall distance y_n. The existence of wall distance in the context of free shear-layers is clearly a misnomer. However, the effect of abandoning the turbulence model in the wake, particularly in the region $0.95\bar{c} - 1.05\bar{c}$, where $\bar{c}$ represents the mean chord length, would be to ignore turbulent diffusion in the free shear-layer where there is rapid change and strong flow interaction. This would undoubtedly compromise the accuracy of the computed solution. Indeed, A.M.O. Smith specifically identifies this region of flow as being of major importance to basic theoretical problems in high-lift aerodynamics [143].

For consistency with the turbulence model in regions of the domain in which walls are present, the wall distance must be maintained though with appropriate modification. It is also necessary to determine or at least approximate the path of the wake. Accepted practice suggests that accurate determination of the wake beyond a distance of about $1.05\bar{c} - 1.1\bar{c}$ aft of the trailing edge is not critical to the performance of such models. However, up to that distance downstream it certainly is important. In the methods developed in the present study, the mesh topologies adopted are C–H and C–O, in which the cut behind the wing trailing edge coincides with the wake for the critical distance downstream of the trailing edge, to within a zeroth-order approximation. Thus it is feasible to approximate the centre of the wake path by the grid boundary (the mesh cut) behind the wing. With this approximation, a pseudo wall distance in the wake y_n may be defined relative to the mesh cut. As already mentioned, the formulation must be modified for the wake. Such modification is quite trivial and merely involves dropping the exponential term from the expressions for both the mixing length l, equation (2.103), and the function $\Gamma(y_n)$, equation (2.108). Also, as mentioned above, all the terms in the equation for U^2_{diff} must be evaluated for free shear-

layers, unlike the true wall case in which the last term on the right-hand side of equation (2.107) is dropped.

Improvements to the basic model

Some modifications to the basic Baldwin-Lomax turbulence model have been developed which can offer improvements in solution accuracy. Typically they involve small changes to the model or tuning of some of the empirical constants.

For flows in which there is strong separation and for vortical flow applications, there may be strong vorticity outside the boundary layer region. In these cases the function $\Gamma(y_n)$ can have more than one maximum Γ_{max}. In this eventuality, a simple modification due to Degani and Schiff [32] may be introduced. It involves taking the first maximum Γ_{max} closest to the wall, to evaluate the wake function Γ_{wake}, equation (2.106). If this is not done and a maximum farther away from the wall is used, a value of y_{max} that is too large will be result, causing an over-prediction of the eddy viscosity and hence an artificially thick turbulent boundary layer. In the present study, the Degani-Schiff modifications were implemented but were not found necessary for the computations presented, all of which involve thin attached shear layers.

While most users of the Baldwin-Lomax model use the standard Cebeci-Smith empirical constants listed in Table 2.1, 'tuned' values for some of them have been put forward in the open literature for improving the accuracy of the model. One of the more recent such references is that by Haase et al. [65] which recommends modified values for C_{wake} and C_{kleb}. For the present study, the standard coefficients were used.

2.6 Time integration using explicit operators

The spatial discretization adopted for the convective and diffusive terms of the Euler and Navier-Stokes equations has already been described and formulations for their flux functions presented in semi-discrete conservation law form. The semi-discrete formulation (method of lines) is commonly used for the solution of time-dependant problems, reducing the original system of partial differential equations to a system of ordinary differential equations which can then be integrated in time by an independent computational procedure. This approach may also be used for steady-state problems and the time integration performed using several different implicit or explicit techniques. It has the additional advantage that it is simpler and easier to program than fully discrete schemes such as the Lax-Wendroff type [142]. Furthermore, since this approach facilitates convergence to the steady-state that is independent of the timestep Δt, it affords a greater choice of convergence acceleration techniques.

In the present study, three types of explicit time discretization are considered, namely the forward-Euler scheme, the multi-stage Runge-Kutta scheme and hybrid multi-stage schemes. The forward-Euler scheme is first-order accurate and undoubtedly the simplest time integration method. In fact, it may be considered as a first-order single-step Runge-Kutta scheme. Runge-Kutta schemes have been analyzed in some detail by Butcher [15] and van der Houwen [161]. They are a commonly used type of multi-stage time integration method which have been

extensively developed for solving systems of equations. These schemes were originally conceived for obtaining higher-order temporal accuracy in solutions to ordinary differential equations and were used for several years in the field of numerical oceanography before being introduced to CFD by Jameson [76] in 1981. Jameson was the first to demonstrate their use for obtaining efficient steady-state transonic solutions to the three-dimensional Euler equations. Subsequently he developed a variety of hybrid multi-stage schemes [69, 73] by modifying the stability characteristics of existing Runge-Kutta schemes through selective evaluation of the dissipative fluxes. The formulations for these methods are given below.

2.6.1 The forward-Euler scheme

Starting from the semi-discrete form of the conservation laws, a general flux function may be written as

$$\mathcal{V}\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{R}^n = \mathbf{0} \tag{2.110}$$

In a finite volume formulation $\mathcal{V}$ is the volume of the cell under consideration and the residual function $\mathbf{R}$ is defined at time level n in terms of the spatial operator $\mathcal{L}_s$ as

$$\mathbf{R}^n/\mathcal{V} = \mathcal{L}_s(\mathbf{Q}^n) \tag{2.111}$$

For the Euler equations, $\mathbf{R}^n$ is composed of the inviscid and dissipative fluxes while for the Navier-Stokes equations, it would also include the physically diffusive fluxes. With the forward-Euler time discretization, the time derivative of the above expression is approximated by a forward-difference operator as

$$\mathbf{Q}^{n+1} = [\mathbf{I} + \Delta t \mathcal{L}_s](\mathbf{Q}^n) \tag{2.112}$$

where $\mathbf{I}$ represents the identity matrix and $\Delta t = t^{n+1} - t^n$. This formula advances the solution of a system of ordinary differential equations from the state $\mathbf{Q}^n$ to $\mathbf{Q}^{n+1}$, based on the Taylor series truncated after the linear term in Δt. It therefore gives a first-order solution by a projection from t^n to $t^n + \Delta t$ along the tangent to the solution at t^n. As such, the formula is unsymmetrical: it advances the solution through an interval Δt, but uses derivative information only at the beginning of that interval.

This time integration is stable for upwind schemes using the first-order flux function given by equation (2.57). However, linear stability analysis of the scalar advection equation reveals that the second-order flux function of equation (2.85), is unconditionally unstable when integrated in time by the forward-Euler method. This is believed to be a consequence of the fact that the truncation error associated with equation (2.112), which is of $\mathcal{O}(\Delta t^2)$ cannot be compensated by a similar term arising from the spatial discretization. It can be shown that the scheme can damp the highest frequencies and that the amplification factor is larger than 1 only for the lowest frequencies. Thus the instability is weak.

In consequence, when the second-order flux function of equation (2.85) is used, a higher-order integration method such as a multi-stage scheme is necessary to ensure stability in regions of smooth flow.

2.6.2 RUNGE-KUTTA SCHEMES

The classical m-stage Runge-Kutta (R-K) scheme is a multi-stage time integration method for the semi-discrete equation (2.110), that takes the generic form

$$\begin{aligned} \mathbf{Q}^{(0)} &\stackrel{\text{def}}{=} \mathbf{Q}^{(n)} \\ \mathbf{Q}^{(k)} &= \mathbf{Q}^{(0)} + \alpha_k \Delta t \mathcal{L}_s[\mathbf{Q}^{(k-1)}] \qquad k = 1, m \\ \mathbf{Q}^{n+1} &\stackrel{\text{def}}{=} \mathbf{Q}^{(m)} \end{aligned} \tag{2.113}$$

in which

$$\alpha_k = \frac{1}{m-k+1} \tag{2.114}$$

This method is conditionally stable with its stability domain depending upon the set of coefficients $\{\alpha_k\}$, $k = 1, m$ used in the given spatial discretization. This may be illustrated by carrying out local Fourier analysis of model hyperbolic equations. For linear differential equations, an m-stage R-K scheme is m^{th}-order accurate. However, for nonlinear equations, the highest temporal accuracy attainable by a R-K scheme, regardless of the number of stages is second-order. In fact, the two-stage scheme obtained from equation (2.113), denoted as RK2 and given by

$$\begin{aligned} \mathbf{Q}^{*} &= \mathbf{Q}^{n} + \tfrac{\Delta t}{2} \mathcal{L}_s(\mathbf{Q}^{n}) \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^{n} + \Delta t \mathcal{L}_s(\mathbf{Q}^{*}) \end{aligned} \tag{2.115}$$

produces second-order time accuracy. For nonlinear systems, application of a standard R-K schemes may result in violation of the TVD condition. Consequently, R-K schemes have been developed by Shu and Osher [142] to conform to this constraint on boundedness of a solution. The two-stage TVD counterpart to equation (2.115) is

$$\begin{aligned} \mathbf{Q}^{*} &= \mathbf{Q}^{n} + \Delta t \mathcal{L}_s(\mathbf{Q}^{n}) \\ \mathbf{Q}^{n+1} &= \tfrac{1}{2}\left[\mathbf{Q}^{n} + \{\mathbf{Q}^{*} + \Delta t \mathcal{L}_s(\mathbf{Q}^{*})\}\right] \end{aligned} \tag{2.116}$$

However, for the two-stage R-K computations carried out in the present study, the form adopted is that given by equation (2.115). Use of this formula for three-dimensional Euler computations across the Mach number range did not produce any undesirable effects and it is likely that TVD R-K schemes are only really necessary for time accurate computations and perhaps other more demanding applications involving forcing terms.

Since each additional stage of a Runge-Kutta scheme increases the computational expense, the principal justification for using a scheme with more than two-stages in the computation of nonlinear problems, is to increase the Courant number and/or to enhance the damping of high-frequency error modes in order to improve the convergence of steady-state solutions. Generally the more the number of stages, the higher the permissible Courant number and the better the damping of high-frequency error modes. However, schemes with more than 6 stages become impractical due to their increased computational cost.

2.6.3 HYBRID MULTI-STAGE SCHEMES

Greater freedom in the design of multi-stage schemes may be gained by evaluating the dissipative fluxes selectively during a multi-stage cycle, rather than

at every stage. Moreover, the evaluation of dissipation may be weighted. This approach offers the additional benefit of lowering the operation count. Indeed, for algorithms that use sophisticated spatial operators such as upwind schemes, such a reduction in operation count can be significant since evaluation of the dissipative fluxes takes up the majority of floating-point operations. Recognition of the importance of improving the damping properties of a given m-stage scheme while at the same time lowering the operation count has resulted in the development of schemes known as *Weighted multi-stage schemes* or *Hybrid multi-stage schemes*. In fact, it was Jameson [69] who first devised such schemes, specifically for improving the efficiency of his multigrid scalar dissipation method.

The hybrid multi-stage scheme is obtained by re-writing the classical R-K scheme, given by equation (2.113), as

$$\begin{aligned} \mathbf{Q}^{(0)} &\stackrel{\text{def}}{=} \mathbf{Q}^{(n)} \\ \mathbf{Q}^{(k)} &= \mathbf{Q}^{(0)} + \alpha_k \tfrac{\Delta t}{\mathcal{V}} \mathbf{R}[\mathbf{Q}^{(k-1)}] \qquad k = 1, m \\ \mathbf{Q}^{n+1} &\stackrel{\text{def}}{=} \mathbf{Q}^{(m)} \end{aligned} \tag{2.117}$$

in which the residual is composed as

$$\mathbf{R}(\mathbf{Q}) = \mathbf{R}_\iota + \mathbf{R}_\upsilon + \mathbf{R}_d \tag{2.118}$$

Here, the subscripts on the right-hand side of the above equation refer to the components of inviscid flux ι, viscous flux υ, and dissipative flux d. However, the dissipative flux is treated distinctly from the other fluxes. Typically, it is weighted according to the dissipative residual function given by

$$\mathbf{R}_d^{(k)} = \beta^{(k)} \mathbf{R}_d \left[\mathbf{Q}^{(n+1,k)}\right] + \left(1 - \beta^{(k)}\right) \mathbf{R}_d^{(k-1)} \tag{2.119}$$

which is subject to a consistency condition on the coefficients β_k. A hybrid scheme having an even number of stages may, for example, evaluate the dissipation only during even-numbered stages. Similarly, schemes having an odd number of stages have been devised which require flux evaluations only at the odd-numbered stages. Other variations are also possible. Considering the complex Fourier symbol for the residual, the imaginary part corresponds to the centred difference approximation of the inviscid flux term, while the real part corresponds to the dissipative term. Thus the coefficients α_k are chosen to maximize the stability interval along the imaginary axis of the stability diagram while the coefficients β_k are chosen to increase it along the negative real axis so as to establish a good parabolic stability limit and so permit higher Courant numbers. Such schemes are referred to generically as m/n multi-stage schemes. For example, a 4/2 multi-stage scheme is a four-stage scheme with two evaluations of dissipation, while a 5/3 scheme is a five-stage scheme with three evaluations. The latter scheme, for example, provides a convergence rate which is commensurate with, if not better than that of a five-stage scheme yet with an operation count for the dissipative fluxes being that of a three-stage scheme. Compared to a 5/5 scheme (viz. the dissipative fluxes evaluated at every stage), this represents a potential saving per computation of 40% in floating-point operations expended on evaluating the numerical dissipation.

Finally, it should be noted that to improve the convergence of multi-stage methods further, procedures have been developed to determine their key parameters via rigorous optimization procedures and, to extend their stability by simple

scalar operators acting on the residuals. These developments are discussed later in the review of convergence acceleration methods, under *Optimal smoothing multi-stage methods* (Section 4.5) and, *Residual smoothing* (section 4.4).

2.6.4 Timestep definition

Local time stepping is actually a means of convergence acceleration, being the simplest and most commonly used method of accelerating steady-state computations. However, in using it any notion of time accuracy is clearly lost. The present algorithm relies on the use of local timesteps for steady-state solutions or global timesteps if time-accuracy is required.

Any timestep must conform to a linear stability constraint known as the CFL condition in order to maintain stability. This requires that the computational domain of the numerical method must lie within the domain of dependence of the governing PDE, at least in the limit as Δt, $\Delta x \rightarrow 0$, where Δt is the timestep and, Δx the mesh spacing. For a three-point differencing scheme applied to a one-dimensional linear system, the CFL condition requires that

$$\left|\lambda_k \frac{\Delta t}{\Delta x}\right| \leq 1 \tag{2.120}$$

for each eigenvalue λ_k $(k = 1, 3)$, of the flux Jacobian $\mathbf{A}$. From this expression, the local Courant number ν, sometimes referred to as the CFL number, is defined as

$$\nu \equiv \max_k \left|\lambda_k \frac{\Delta t}{\Delta x}\right| \tag{2.121}$$

where max (λ_k) is the spectral radius or maximum eigenvalue of the flux Jacobian. The timestep is therefore obtained as

$$\Delta t = \nu \frac{\Delta x}{\lambda_{\max}} \tag{2.122}$$

in which $\lambda_{\max} \equiv \max(\lambda_k)$. This expression restricts the magnitude of the timestep to ensure that a wave from any cell interface can only influence its immediate neighbours on either side. For a first-order method, a necessary condition for stability is that the Courant number be no greater than 1.

In multi-dimensional calculations, definition of the timestep is not unique and the CFL condition may only be expressed in an approximate sense. In the present algorithm, computation of the timestep proceeds as follows. An upper bound on the timestep is imposed for each cell as

$$\frac{1}{\Delta t_{i,j,k}} \geq \left(\frac{1}{\Delta t_\xi} + \frac{1}{\Delta t_\eta} + \frac{1}{\Delta t_\zeta}\right)_{i,j,k} \tag{2.123}$$

in which a generic component Δt_ϕ in a given coordinate direction is made up of a convective and a diffusive contribution as

$$(\Delta t_\phi)_{i,j,k} = \left(\Delta t_\phi^\iota + \Delta t_\phi^\upsilon\right)_{i,j,k} \tag{2.124}$$

The bounds for the convective components of the timestep are set to their respective spectral radii which are given by

$$\left(\lambda_\xi^\iota\right)_{i,j,k} = \left(|U| + a\sqrt{k_x^2 + k_y^2 + k_z^2}\right)_{i,j,k}$$

$$\begin{aligned}(\lambda_\eta^\iota)_{i,j,k} &= \left(|V| + a\sqrt{l_x^2 + l_y^2 + l_z^2}\right)_{i,j,k} \\ (\lambda_\zeta^\iota)_{i,j,k} &= \left(|W| + a\sqrt{m_x^2 + m_y^2 + m_z^2}\right)_{i,j,k}\end{aligned} \tag{2.125}$$

in which U, V, W represent the contravariant velocities in each respective coordinate direction, given by equation (2.20), and a is the speed of sound. The diffusion limit on the timestep is obtained in a similar manner as

$$\begin{aligned}(\lambda_\xi^\upsilon)_{i,j,k} &= \frac{\gamma}{\rho}\left(\frac{\mu_l}{\mathrm{Pr}_l} + \frac{\mu_t}{\mathrm{Pr}_t}\right)_{i,j,k}(k_x^2 + k_y^2 + k_z^2)_{i,j,k} \\ (\lambda_\eta^\upsilon)_{i,j,k} &= \frac{\gamma}{\rho}\left(\frac{\mu_l}{\mathrm{Pr}_l} + \frac{\mu_t}{\mathrm{Pr}_t}\right)_{i,j,k}(l_x^2 + l_y^2 + l_z^2)_{i,j,k} \\ (\lambda_\zeta^\upsilon)_{i,j,k} &= \frac{\gamma}{\rho}\left(\frac{\mu_l}{\mathrm{Pr}_l} + \frac{\mu_t}{\mathrm{Pr}_t}\right)_{i,j,k}(m_x^2 + m_y^2 + m_z^2)_{i,j,k}\end{aligned} \tag{2.126}$$

In the last two equations the surface normals k, l, m correspond to cell centred values which are approximated as simple averages of the interface values. Each component timestep is evaluated as

$$(\Delta t_\phi)_{i,j,k} = \left(\frac{\nu}{\lambda_\phi^\iota + \lambda_\phi^\upsilon}\right)_{i,j,k} \tag{2.127}$$

in which $\mathcal{V}_{i,j,k}$ is the cell volume. A single timestep for each cell, may now be formulated by introducing the local Courant number and using the convective and diffusive component timesteps given above, together with equation (2.123). It is given by

$$\begin{aligned}\Delta t_{i,j,k} &= \nu \frac{1}{\left(\frac{1}{\Delta t_\xi} + \frac{1}{\Delta t_\eta} + \frac{1}{\Delta t_\zeta}\right)_{i,j,k}} \\ &= \nu\left(\frac{\Delta t_\xi \Delta t_\eta \Delta t_\zeta}{\Delta t_\xi \Delta t_\eta + \Delta t_\xi \Delta t_\zeta + \Delta t_\eta \Delta t_\zeta}\right)_{i,j,k}\end{aligned} \tag{2.128}$$

It is this quantity which is referred to as a *local* timestep. With local time stepping, each cell is advanced to a different level in time by its own timestep $\Delta t_{i,j,k}$, yielding a uniform Courant number for all cells rather than a constant timestep for the whole domain. Alternative ways of formulating a local timestep for multi-dimensional problems may be found in the literature. For example, Tannehill *et al.* [152] define the diffusive limit as the maximum of the coefficient of viscosity and the coefficient of thermal diffusivity. See also Vasta *et al.* [173]. For viscous flows, the most restrictive timestep is in the boundary layer region near solid surfaces where the component terms Δt_η^ι and Δt_η^υ dominate. In addition, the diffusion limit becomes more important as Mach number is increased. Although the diffusion limit reduces the allowable timestep in the near wall region, its inclusion in the timestep facilitates computations across a wide range of Mach number, so enhancing the overall robustness of the present algorithm. For inviscid computations, the diffusive term λ_ϕ^υ is simply dropped from equation (2.127).

With unsteady or time-accurate solutions, each cell must be advanced to the same level in time for each solution update. Thus the timestep is fixed at a

constant value across all cells in the domain. This is known as a *global* or *fixed* time stepping. To use a constant timestep throughout and at the same time maintain stability in every cell, the global timestep must be defined as

$$\Delta t_{global} = \min_{i,j,k} \{\Delta t_{i,j,k}\} \tag{2.129}$$

in which $\Delta t_{i,j,k}$ may be given by equation (2.128). As a global timestep is based on the cell containing the fastest moving disturbance within the domain it is the most restrictive condition to impose and compared to local time stepping, it results in a very slow rate of convergence.

2.7 Boundary Treatment

Recently, Mazaheri and Roe [99] developed a unique new boundary condition for two-dimensional inviscid flows. Called a *soft wall* boundary condition, it aims to attenuate acoustic disturbances at solid surfaces which would otherwise reflect from the boundary causing the regeneration of residuals. It acts analogously to a mass-less spring-damper, by damping flow transients as though the surface were a soft wall. In the steady-state, the soft wall boundary coincides with a normal rigid wall. In fact, its principle advantage is that it significantly improves convergence. Unfortunately, the analysis on which the boundary condition is based is two-dimensional and it may not be possible to extend it to three-dimensions. Another study by Dadone and Grossman [27] also considered inviscid wall boundary conditions for the two-dimensional flux-difference splitting method of Roe. They developed another new boundary condition which shows marked advantages over several other methods, particularly with regard to spurious entropy generation, total pressure loss, drag and grid convergence. Although details are not given for three-dimensional wall boundaries, it should not be difficult to extend Dadone and Grossman's two-dimensional wall boundary condition.

Of particular note in far-field boundary condition development, Giles [48] has produced a unified linear theory for the construction of non-reflecting boundary conditions for the Euler equations and the boundary conditions derived from this analysis have been shown to be second-order accurate. They are based on characteristic theory and have been used to produce excellent results for two- and three-dimensional transonic turbo-machinery computations [48, 138].

For the present study, the boundary condition procedures that have been adopted are fairly standard ones for structured cell-centred algorithms. Their implementation is facilitated by the use of fictitious *ghost cells*, two rows deep on each boundary of the computational domain. This is illustrated schematically in Figure 2.8 in which cell index notation is given for two ghost cells abutting the first two interior ones. For clarity, only indices counting normal to the boundary are used. This convention is also adopted in writing much of the formulae that follow. Irrespective of the boundary type, boundary condition procedures are used to compute each of the conserved variables as well as pressure in each of the fictitious cells. This facilitates simple and consistent application of the Riemann solver across each boundary without the need for special logic or fixes which would otherwise be needed to accommodate the flux limiting procedure. This strategy also simplifies the programing and improves code vectorization.

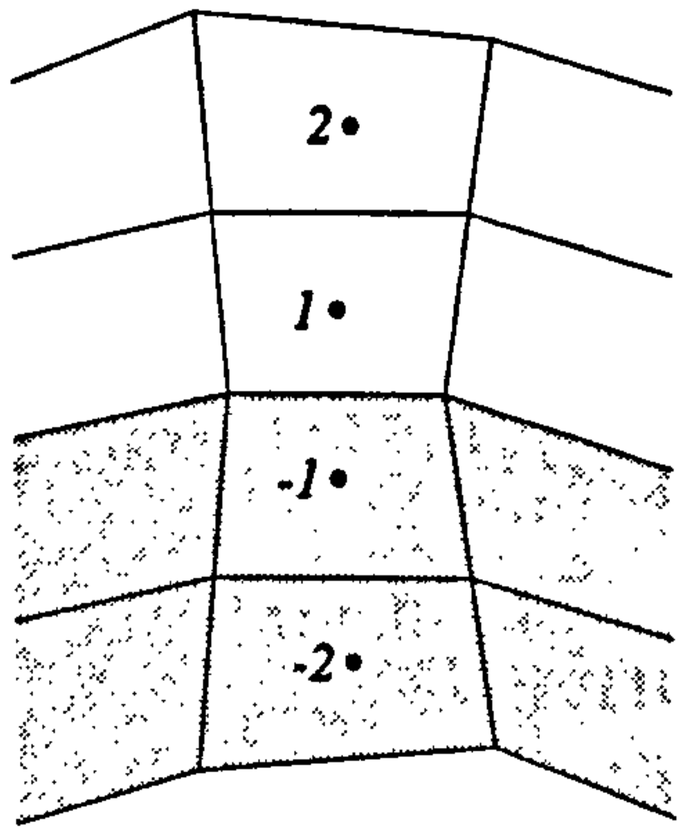


FIGURE 2.8. Schematic diagram of boundary cells. Shading denotes fictitious ghost cells.

All the farfield boundary condition procedures are automated. Thus, for known topologies it is not necessary to specify whether a boundary is inflow or outflow, subsonic or supersonic. This improves the robustness and ease of use of the flow solver. In the following sections, each boundary condition type is catalogued with a brief explanation.

2.7.1 Surface boundaries

Inviscid walls

The boundary condition procedures for inviscid walls ensure that flow does not pass through solid surfaces and that flow tangency at these surfaces is maintained. In addition, the pressure gradient normal to a wall must be zero. Boundary values of each component of velocity u, v and w are obtained by reflection, assuming that the mesh spacing in the first two interior cells is equidistant. In ξ, η, ζ coordinates, the reflected velocities are obtained as

$$\begin{aligned} u_{-j} &= u_j - 2V_j\hat{n}_{x_b} \\ v_{-j} &= v_j - 2V_j\hat{n}_{y_b}, \qquad (j = 1, 2) \\ w_{-j} &= w_j - 2V_j\hat{n}_{z_b} \end{aligned} \tag{2.130}$$

Here V denotes the velocity normal to the boundary which is determined using equation (2.20). Quantities subscripted by b are evaluated on the boundary itself. Thus $\hat{n}_{x_b,y_b,z_b}$ represent the unit normals on the surface of the wall, given by equation (2.30).

To obtain wall pressure, the technique adopted follows Rizzi [122] using the so-called normal momentum boundary condition. It is derived from the steady-state form of the momentum equation in the coordinate direction normal to the

surface, coupled with the requirement of an impermeable wall. The interested reader is referred to the original paper for its derivation. This renders the pressure at the wall, denoted as p_b. To determine the pressure in the fictitious cells, linear extrapolation is used, again assuming equidistant mesh spacing. Furthermore, to prevent negative pressure which may arise in the course of convergence due to numerical transients, a lower bound on the pressure is set via a tolerance ϵ. This results in the expressions

$$p_{-1} = \max[\epsilon, (2p_b - p_1)] \tag{2.131}$$
$$p_{-2} = \max[\epsilon, (4p_b - 3p_1)] \tag{2.132}$$

The density can be obtained most simply via extrapolation as

$$\rho_{-j} = \rho_j, \qquad (j = 1, 2) \tag{2.133}$$

Alternatively, it may be obtained by making the assumption that Entropy s is constant in the wall region and using the pressure already found from equations (2.131–2.132). i.e.

$$s_{-j} = s_j, \qquad (j = 1, 2)$$

and since $s \propto p/\rho^\gamma$, this implies that

$$\frac{p_{-j}}{\rho_{-j}^\gamma} = \frac{p_j}{\rho_j^\gamma}$$

Hence

$$\rho_{-j} = \rho_j \left(\frac{p_{-j}}{p_j}\right)^{1/\gamma}, \qquad (j = 1, 2) \tag{2.134}$$

Finally, the total energy e is computed from the equation of state using the values of $\rho_{-j}, u_{-j}, v_{-j}, w_{-j}$, and p_{-j} now available, as

$$e_{-j} = \frac{p_{-j}}{\gamma - 1} - \rho_{-j}(u_{-j}^2 + v_{-j}^2 + w_{-j}^2)/2, \qquad (j = 1, 2) \tag{2.135}$$

Viscous walls

A no-slip boundary condition is imposed to ensure that the velocity at the wall is identically zero. In addition, a boundary condition is required for temperature, which may be set in one of two ways. The first assumes the wall to be adiabatic or insulating, allowing no heat flux through it. The second assumes an isothermal wall, in which case the temperature at the wall is held constant at a specified temperature. In the computations presented in section 5.2, an adiabatic wall is assumed. The adiabatic wall boundary condition is implemented as follows. A zero gradient of heat flux is specified by imposing constant temperature as

$$T_{-j} = T_j, \qquad (j = 1, 2) \tag{2.136}$$

Since temperature may be calculated from the thermodynamic relation

$$T = \frac{\gamma p}{c_p(\gamma - 1)\rho} \tag{2.137}$$

equation (2.136) may be satisfied simply by ensuring constant density and pressure across the wall. i.e.

$$\begin{aligned} \rho_{-j} &= \rho_j, \\ p_{-j} &= p_j, \end{aligned} \qquad (j = 1, 2) \tag{2.138}$$

Considering a boundary that is aligned with constant j-planes, for example, this effectively sets the conductive heat flux terms of equation (2.9) which appear in equations (2.16–2.19) to zero. Finally the no-slip condition is imposed as

$$\begin{aligned} u_{-j} &= -u_j \\ v_{-j} &= -v_j, \\ w_{-j} &= -w_j \end{aligned} \qquad (j = 1, 2) \tag{2.139}$$

with the total energy being specified consistently as

$$e_{-j} = e_j, \qquad (j = 1, 2) \tag{2.140}$$

The above formulae are first-order accurate. Generally, they are deemed to be adequate for high Reynolds Number flows which use densely clustered computational mesh normal to solid boundaries, in order to properly resolve the usually thin boundary layers. Second-order accurate formulae are not usually deemed necessary at wall boundaries when such fine mesh are used since their truncation error would be proportional to the square of the mesh size and thus, particularly small. However, gradients can be steep in wall regions such as at stagnation points and there the accuracy of the above first-order formulae is less satisfactory.

2.7.2 Farfield boundaries

Subsonic inflow/outflow

Both the rate of convergence and accuracy of solution is likely to be impaired if outgoing waves are reflected back into the flow from farfield boundaries. In an attempt to devise non-reflecting boundary conditions, Jameson [72] has advanced the use of boundary conditions based on the notion of Riemann Invariants. These are derived from one-dimensional wave propagation theory assuming that at a boundary, the flow is locally normal to it. Although this assumption has been shown theoretically by Mazaheri and Roe [99] to be poor, surprisingly the deficiencies of the approach are not as bad in practice. In fact Riemann invariants are commonly used in numerous industry-standard CFD design codes.

Assuming isentropic flow, the continuity and momentum equations for inviscid flow may be integrated and using standard polytropic gas relations, the expressions known as Riemann invariants are obtained, see Courant and Friedrichs [25]. They are given by

$$R_+ = u_n + \frac{2a}{\gamma - 1} \tag{2.141}$$

$$R_- = u_n - \frac{2a}{\gamma - 1} \tag{2.142}$$

The R_+ Riemann invariant is associated with $u+a$ waves and has the property of being constant along C_+ characteristics, while R_- is associated with $u-a$ waves and is constant along C_- characteristics. These useful properties allow Riemann Invariants to be used to establish appropriate boundary condition procedures for subsonic flow which has both incoming and outgoing characteristics. In subsonic flow, the incoming Riemann invariant R_- propagates information from the freestream to the interior domain and is therefore evaluated as a function of freestream conditions. Conversely, the outgoing Riemann invariant R_+ which propagates information from the interior to the farfield, is evaluated as a function of the interior conditions. Thus, for a three-dimensional calculation in which the velocity normal to the boundary is V, the Riemann invariants are

$$\begin{aligned} R_+ = \quad R_e = \quad & \left[V + \tfrac{2a}{\gamma-1}\right]_e \\ R_- = \quad R_\infty = \quad & \left[V - \tfrac{2a}{\gamma-1}\right]_\infty \end{aligned} \tag{2.143}$$

Here the quantities subscripted by e and ∞, represent extrapolated and freestream conditions respectively. Adding and subtracting the two above equations at the boundary yields

$$V_b = \frac{1}{2}(R_e + R_\infty) \tag{2.144}$$

and

$$a_b = \frac{\gamma-1}{4}(R_e - R_\infty) \tag{2.145}$$

where the subscript b denotes quantities on the boundary interface. The sign of the normal velocity V_b indicates whether the flow is inflow or outflow and the velocities needed in the ghost cells are determined as

$$\begin{aligned} u_{-1} &= u_\infty + (V_b - V_\infty)\hat{n} \\ v_{-1} &= v_\infty + (V_b - V_\infty)\hat{n} \quad , \qquad V_b < 0 \\ w_{-1} &= w_\infty + (V_b - V_\infty)\hat{n} \end{aligned} \tag{2.146}$$

and

$$\begin{aligned} u_{-1} &= u_e + (V_b - V_e)\hat{n} \\ v_{-1} &= v_e + (V_b - V_e)\hat{n} \quad , \qquad V_b \geq 0 \\ w_{-1} &= w_e + (V_b - V_e)\hat{n} \end{aligned} \tag{2.147}$$

where $\hat{n}$ is the unit normal at the boundary. The boundary condition procedures are completed by assuming constant entropy. For subsonic inflow

$$s = s_\infty, \qquad V_b < 0 \tag{2.148}$$

while for subsonic outflow

$$s = s_e, \qquad V_b \geq 0 \tag{2.149}$$

Thus the density is found as

$$\rho_{-1} = \rho_\infty \left(\frac{p_{-1}}{p_\infty}\right)^{1/\gamma}, \qquad V_b < 0 \tag{2.150}$$

$$\rho_{-1} = \rho_e \left(\frac{p_{-1}}{p_e}\right)^{1/\gamma}, \qquad V_b \geq 0 \tag{2.151}$$

In the above equations, the quantities denoted by subscript e are evaluated in terms of cell counters using the association $e = j = 1$. Using the above density ρ_{-1} and speed of sound a_b, the pressure is determined from the standard relation

$$p_{-1} = \frac{\rho_{-1} a_b^2}{\gamma} \tag{2.152}$$

while the total energy is found using the perfect gas relation of equation (2.135). Finally, the second row of halo cells are obtained via replication of the first row, i.e. $\rho_{-2} = \rho_{-1}, u_{-2} = u_{-1}$, etc.

Clearly the use of Riemann invariants has the advantage of not requiring specification of whether the flow is entering or leaving the domain and they are able to deal with unsteady flow features such as vortex shedding. Also, the procedure avoids the need to specify pressure at the downstream boundary, where it does not necessarily revert to the freestream value. This can happen in instances where trailing vortex sheets reach the boundary.

Supersonic inflow

For supersonic inflow, all characteristics are incoming and consequently all quantities must be specified in the fictitious boundary cells by a physical boundary condition. This is done as

$$\begin{array}{lllllll} \rho_{-j} & = & \rho_\infty & & u_{-j} & = & u_\infty \\ e_{-j} & = & e_\infty & , & v_{-j} & = & v_\infty \quad , \quad (j = 1, 2) \\ p_{-j} & = & p_\infty & & w_{-j} & = & w_\infty \end{array} \tag{2.153}$$

Supersonic outflow

When supersonic flow leaves the domain, all characteristics are outgoing. Hence, all quantities are obtained by numerical boundary procedures and are extrapolated from the interior as

$$\begin{array}{lllllll} \rho_{-j} & = & \rho_j & & u_{-j} & = & u_j \\ e_{-j} & = & e_j & , & v_{-j} & = & v_j \quad , \quad (j = 1, 2) \\ p_{-j} & = & p_j & & w_{-j} & = & w_j \end{array} \tag{2.154}$$

2.7.3 Other boundaries

Planes of symmetry

Frequently the symmetry of a problem can be exploited to reduce the size of a computational domain and hence reduce the cost. To ensure that a flow solver will return the same solution for the half-plane problem that it would for an equivalent full-plane problem, the domain is truncated along a plane of symmetry. The boundary conditions needed for symmetry boundaries are similar to the inviscid wall boundary conditions. Flow tangency and impermeability across the boundary are enforced using equations (2.130) for the velocities. The remaining quantities of density, total energy and pressure, are obtained by first-order extrapolation as

$$\rho_{-j} = \rho_j$$

$$\begin{aligned} e_{-j} &= e_j, \qquad (j = 1,2) \\ p_{-j} &= p_j \end{aligned} \tag{2.155}$$

Continuity boundaries

Continuity boundaries often arise in structured grid computations. In the generation of wing and wing-body C-H and C-O topologies, they are formed where the grid wraps back on itself, aft of the wing in the region of wake flow. It is important to ensure that the flow acts as though it were a continuum across these boundaries. To achieve this, the following procedure is applied:

$$\begin{array}{rclcrcl} (\rho_{-j})_{S_A} &=& (\rho_j)_{S_B} & & (\rho_{-j})_{S_B} &=& (\rho_j)_{S_A} \\ (u_{-j})_{S_A} &=& (u_j)_{S_B} & & (u_{-j})_{S_B} &=& (u_j)_{S_A} \\ (v_{-j})_{S_A} &=& (v_j)_{S_B} & & (v_{-j})_{S_B} &=& (v_j)_{S_A} \\ (w_{-j})_{S_A} &=& (w_j)_{S_B} & , & (w_{-j})_{S_B} &=& (w_j)_{S_A} \quad , \qquad (j = 1,2) \\ (e_{-j})_{S_A} &=& (e_j)_{S_B} & & (e_{-j})_{S_B} &=& (e_j)_{S_A} \\ (p_{-j})_{S_A} &=& (p_j)_{S_B} & & (p_{-j})_{S_B} &=& (p_j)_{S_A} \end{array} \tag{2.156}$$

Here the quantities $(\bullet)_{S_{A,B}}$ denote values at points along the regions A and B of the boundary S, at which region A is immediately adjacent to region B. The first column of equations inject the fictitious cells along S_A with flowfield values from the interior cells along S_B. Likewise the second column of equations inject the fictitious cells along S_B with values from the interior cells along S_A.

Degenerate boundaries

When a three-dimensional grid is generated by rotation about a line, a degenerate surface is formed along the axis of rotation. Cells along the degenerate line have only five sides, resembling a wedge of cheese. Grids that are generated in this matter can be problematic for a flow solver, as discussed later in section 3.1.3 in reference to high speed, blunt-body solutions. Nonetheless, they are commonly used for computations involving bodies of revolution due to their ease of generation. The boundary conditions that are applied to degenerate boundaries are similar to those used for continuity boundaries. They are

$$\begin{array}{rclcrcl} \rho_{-j} &=& \rho_j & & u_{-j} &=& u_j \\ e_{-j} &=& e_j \quad , & & v_{-j} &=& v_j \quad , \qquad (j = 1,2) \\ p_{-j} &=& p_j & & w_{-j} &=& w_j \end{array} \tag{2.157}$$

2.8 Closing Discussion

In this chapter some necessary background and theory for the development of an explicit upwind algorithm for the computation of steady inviscid and viscous flows has been given. The algorithm that has been outlined is based on the use of existing ingredients that have been selected and assembled to produce a flow solver that is capable of accurate solution of the governing equations. The particular formulation discussed permits the temporal and spatial discretization to be treated separately. The discretization in space is carried out using the flux difference splitting method of Roe. To obtain second-order accuracy, algebraic

flux limiting is used, following the post-processing approach. This offers some advantage in terms of computational expense over the geometric, pre-processing approach to achieving higher-order accuracy. The system of ordinary differential equations that is obtained by the spatial discretization is advanced in time by a choice of multi-stage time stepping schemes. Thus, the steady-state solution obtained is independent of the timestep. This makes the scheme more amenable to the use of convergence acceleration techniques. This is a necessary pre-requisite to the main thrust of the study –the development of efficient upwind methods for the Euler and Navier-Stokes equations.

3

Numerical Results for Inviscid Flow

Based on the theory of Chapter 2, a three-dimensional Euler upwind flow solver was written as part of the present study. The programming language used is Fortran 77. Once the coding was verified for correctness, a series of test cases were undertaken to validate the new solver. For the solutions presented in this chapter, a two stage Runge-Kutta time integration scheme was used. In addition, the minmod limiter function was adopted. At this stage, the only convergence acceleration device used was local time stepping.

The first section of this chapter contains the results of six test cases conducted at flow conditions ranging from Mach 0.675 to Mach 25. They comprise transonic channel flow over a bump, supersonic flow about a slender forebody and mixed high speed flows about a sphere at different Mach numbers including hypersonic flow. Four further inviscid test cases are presented in the second section of the chapter in which the results from the new solver are compared to ones from other flow solvers. In each of these test cases, the solutions from all solvers are obtained from the same grid. The comparative tests involve supersonic flow over a blunt-nose body, supersonic and transonic flow over a wing-body, and vortex flow over a delta wing.

3.1 Inviscid Flow Solutions

3.1.1 Transonic channel flow

This test case is the two-dimensional GAMM channel which has a symmetric bump in the mid-section of the lower wall. The bump height is 10% that of the channel opening. At inflow, the freestream Mach Number is $M_\infty = 0.675$. Two different grids were used in this case, both of dimensions $64 \times 16 \times 1$. Although a 2-D problem, the computed solutions use the present 3-D solver by imposing a zero flow gradient across a single lamina of cells in the third dimension. The first grid used is a regular body conforming grid, while the second grid has been adapted to the flow features. At inflow and outflow the flow is subsonic. From the (left) inflow boundary the flow expands as it encounters the bump and becomes critical. At about 70% bump length, the critical region terminates in a shock wave. Experimentally, the shock has been found to have a value of $M = 1.4$. After the shock, the flow expands back to freestream conditions.

Figure 3.1(a) and (b) show the regular grid and the corresponding numerical solution in terms of isomach contours ($\Delta M = 0.04$). The expansions at both ends of the bump are clearly represented with the lowest Mach numbers occurring at

inlet and exit. The shock is captured at the correct location with a Mach number of $M = 1.385$, which is in reasonable agreement with the experimental value.

The adapted grid and the isomach solution obtained from it are given in Figure 3.1(c) and (d). With the adapted grid there is some improvement in the initial expansion but marked improvement in shock resolution. The acoustic disturbance emanating around the start of the bump attenuates further upstream than in the regular grid computation, reflecting the improved grid distribution in that area of the flow. The shock clearly benefits from the high density cell clustering which accounts for its sharpened profile. The peak Mach number at the shock also increases slightly above the regular grid solution. Conversely, the mesh adaptation slightly degrades the acoustic disturbance at the downstream end of the bump due to the lower mesh density provided in that region of the flow.

The convergence histories of both regular and adapted mesh computations are given in Figure 3.2 in terms of the log of the l_2 norm, $\log_{10} \|d\rho/dt\|_2$. The regular grid solution achieves a slightly lower residual than the adapted one while the latter converges more erratically. This possibly reflects increased high-frequency errors associated with grid clustering in the shock region. It is likely that the rapid skewing of cells at the shock accounts for the slight degradation in convergence.

3.1.2 Supersonic flow over a slender forebody

The configuration used in this test is a slender tangent ogive forebody. The specified flow conditions are $M_\infty = 1.6$, and $\alpha = 5°$. The grid ($89 \times 29 \times 9$), is shown in Figure 3.3. Figure 3.4(a) and (b) illustrate the solution in terms of isomach contours ($\Delta M = 0.01$) and lines of constant density ($\Delta\rho = 0.01$), respectively. The solution domain is wholly supersonic, the principal features of which are the shock wave attached to the body at the nose, and the expansion about the shoulder of the ogive. The disturbance from the nose attenuates strongly into the flowfield on both leeward and windward sides as far as the outflow boundary. The shock is monotone and sharp while around the shoulder of the nose the expansion is well defined. The convergence history is plotted in Figure 3.5.

3.1.3 Supersonic & hypersonic flow over a sphere

Test cases were conducted of flow past a sphere at Mach numbers $M_\infty = 2.0, 8.0$ and 25.0. The lowest Mach number case uses a different grid to the other test cases, although in each case the grid dimensions are $40 \times 20 \times 20$. As each case is based on a perfect gas model, the results of the latter two cases are not representative of the true hypersonic conditions. In true hypersonic flow, chemical reactions and viscous effects are present and the ratio of specific heats γ varies. Therefore, in any realistic simulation these additional effects need to be included. Nevertheless, the tests have been carried out to demonstrate the robustness and quality of solution of the upwind scheme in some of the more taxing high Mach number flow conditions that a shock capturing algorithm is likely to encounter. In addition, the numerical results from these cases demonstrate the suitability of this type of algorithm for future development of a hypersonic, real gas capability.

The solutions for each of the test cases are fundamentally similar. Each con-

sists of a detached bow shock wave separating the supercritical flow region from the subcritical region. At zero incidence, the subsonic region is disposed symmetrically with respect to the stagnation streamline. In the computations, the stagnation streamline corresponds to the centre-line of the body about which the grid is generated. (With axial grid generation, cells along the centre-line have five sides, two of which are joined by a degenerate surface/line. The shape is similar to a wedge of cheese). The subsonic flow behind the shock expands as it moves around the body to facilitate freestream recovery. The effect of increasing Mach number is to increase the magnitude of gradients in the flow variables, particularly across the shock and, to press the position of the shock closer towards the body. As the shock position changes with increasing Mach number, so the extent of the subsonic region expands outwards away from the stagnation point, increasing in its radial extent. In a full hypersonic simulation, real gas effects would press the shock closer towards the body but the gross flow features would evolve in a similar fashion to the inviscid computation.

In several of the contour plots presented for this test case, in instances where individual contours meet the shock, a small blip or zig-zag is evident. This misleading effect is a manifestation of the contour plotting algorithm, rather than any solution anomaly. It is produced when points on a contour are found in the wrong order.

The entropy fix used in the present computations is given by equations (2.75–2.76). In the $M_\infty = 2.0$ case the values used were $\delta_k^* = 0.25 \; \forall \, k$. The numerical results for this computation are given in Figure 3.6 in which (a) is a view of the grid in the symmetry plane, (b) the isomach contours, (c) lines of constant density and, (d) isobar contours. From the solution contours, the shock appears sharp and monotone, being spread over a single cell across the stagnation streamline. The sonic region behind the shock is well defined and the expansion around the body clearly evident.

The subsonic contours in front of the body that straddle the centre-line show a slight dimpling. This would not appear in real flow, being a consequence of the degenerate cells that form the singularity in the grid discretization along the stagnation streamline, normal to the shock. This singular line prevents evaluation of flux across it. The same effect has been noted in the literature by others including Eliasson and Rizzi [37], who attribute it to a local loss of conservation. They present results for a similar test to the present one, only using a central difference scheme. Unlike the present solution, their central solution demonstrates a severe local distortion of the flow features about the stagnation streamline. In fact, the effect is so severe that it corrupts the shape of the bow shock. Eliasson and Rizzi prove that the effect is caused the degenerate cells, by rotating their grid through 90°. In the new orientation there are no longer any degenerate cells about the stagnation streamline and the problem disappears. Upwind algorithms have been shown to be far more robust than central differencing schemes for hypersonic flows [80], and with the present algorithm the effect of the degenerate cells is present but it is not significant. It does not appear to influence the position or shape of the shock to any noticeable extent.

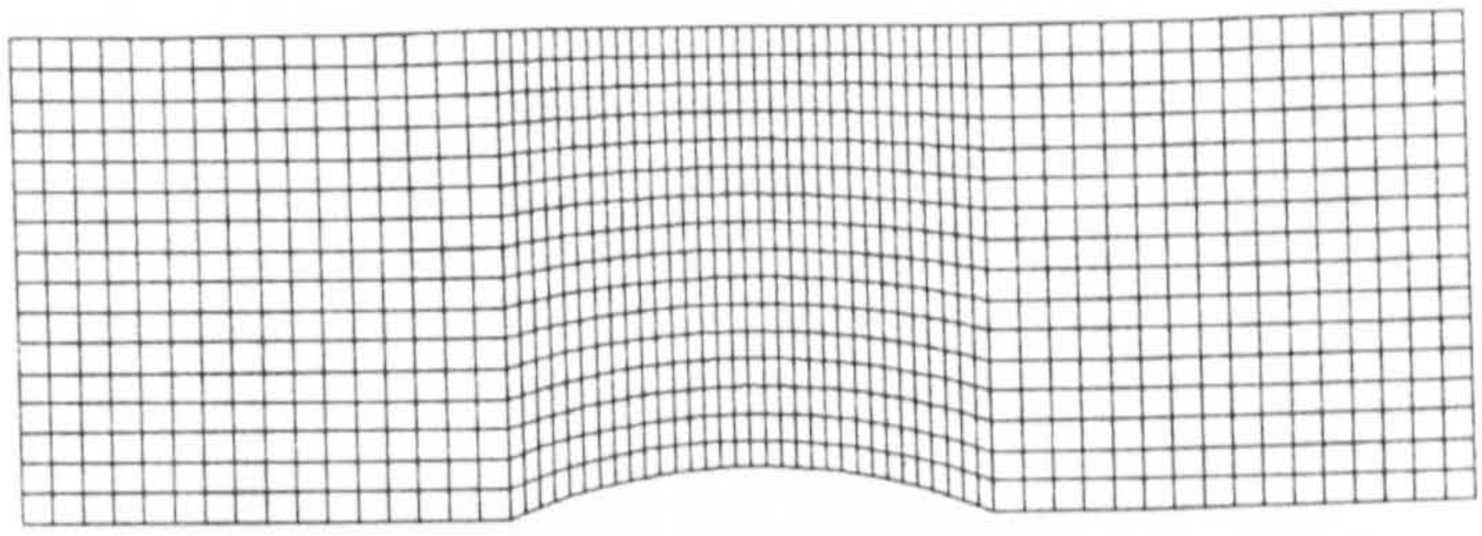
The convergence history for the $M_\infty = 2.0$ case is given in Figure 3.7 in terms of the l_2 norm and the number of supersonic points. The l_2 norm decreases sharply for the first 1000 iterations after which the convergence trace exhibits high-frequency low amplitude oscillation as it bottoms out. At that point, it has

decreased by nearly two orders of magnitude. The number of supersonic points decreases monotonically. Since the solution quality is high, and the number of supersonic points is constant, the low drop in the residual is indicative of convergence stall. (Section 2.4.8 gives a comprehensive discussion of this phenomenon). This type of convergence pathology is repeated in several of the remaining results which exhibit a characteristically shallow drop in the l_2 norm. In order to judge whether an adequate level of convergence has been achieved when convergence stall is evident from the residual trace, it is necessary to monitor other convergence parameters such as the force coefficients and the number of supersonic points in the solution.

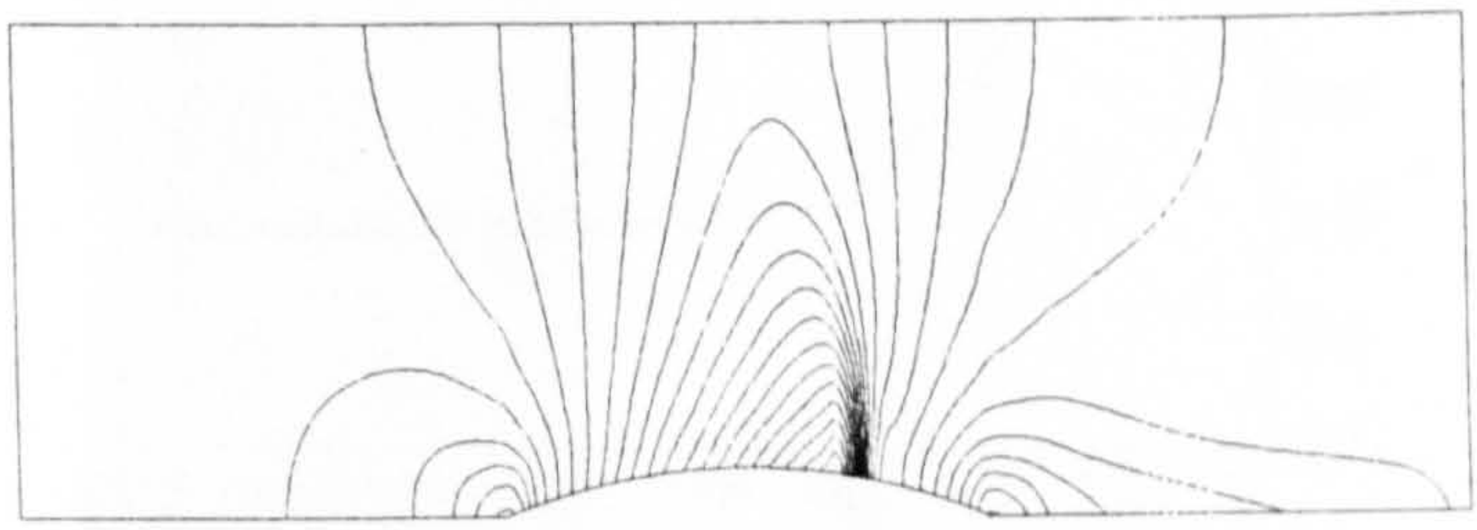
As the present algorithm is based on Roe's numerical flux formulae, it will suffer from the blunt-nose instability known as the *carbuncle phenomenon*, reported by Peery and Imlay [110], if measures are not taken to prevent it in the case of higher Mach number flows. Here it is avoided by tuning the entropy fix to increase dissipation in the vicinity of the shock where instability would otherwise develop. In addition, for $M_\infty > 4$, the limiting is modified to improve eigenvector scaling. The modification used here is described in section 2.4.7.

The grid and flowfield solutions for the $M_\infty = 8.0$ test case are presented in Figure 3.8, following the established format. Again the shock is well defined as is the subsonic region and flow expansion approaching the outflow boundary. The stand off distance from the body to the shock has decreased as expected and the subsonic region behind the shock has advanced further outwards in the radial direction, compared to the lower Mach number test case. For the present flow condition, the entropy fix parameters were set according to the characteristic field, see section 2.4.5. For the acoustic waves δ_k^* was set to $\delta^*_{acoustic} = 0.25$ while for the linear degenerate waves it was set to $\delta^*_{linear} = 0.5$. Figure 3.9 displays the convergence histories.

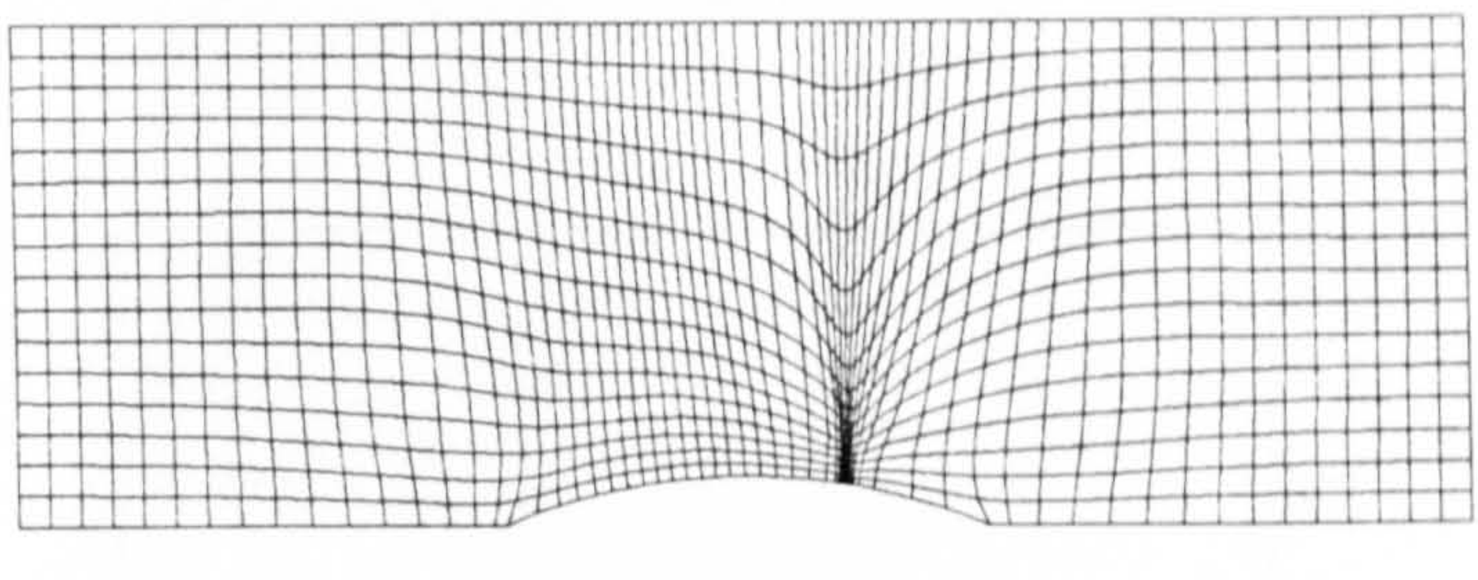
In Figure 3.10 the grid and flowfield solutions for the $M_\infty = 25.0$ test case are presented. For this case, the entropy fix parameters used were $\delta^*_{acoustic} = 0.25$ and $\delta^*_{linear} = 0.8$. The shock is crisp and all remaining flow features are similarly well represented despite the extreme gradients present. Across the shock, the pressure ratio is over 800:1. While contour plots give some indication of a scheme's ability to capture strong shocks, a more precise examination is obtained from line plots of Mach number along radial grid lines/rays. Such plots are presented in Figure 3.11. In part (a) of the figure, the chosen ray is coincident with the stagnation streamline while in part (b) the ray is inclined at 34.5° from the vertical. The plots reveal the high resolution of the shock wave, particularly where the grid is in closest alignment with the discontinuity. Along the stagnation streamline, it can be seen that the shock is resolved with only one interior point. Theoretically, a single point in the shock wave is the optimal solution attainable by a conservative shock capturing algorithm unless the shock lies *exactly* on a cell interface. The convergence histories are presented in Figure 3.12.



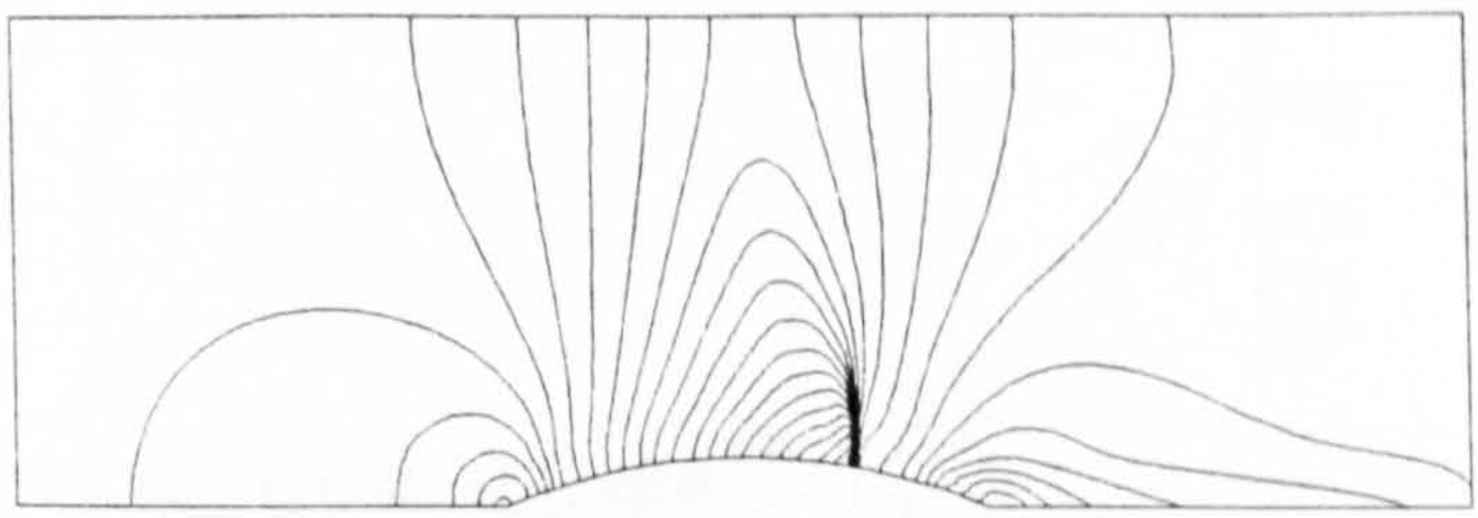
(a) Regular grid (64 x 16 x 1)



(b) Regular grid solution -Lines of constant Mach number (ΔM=0.04)

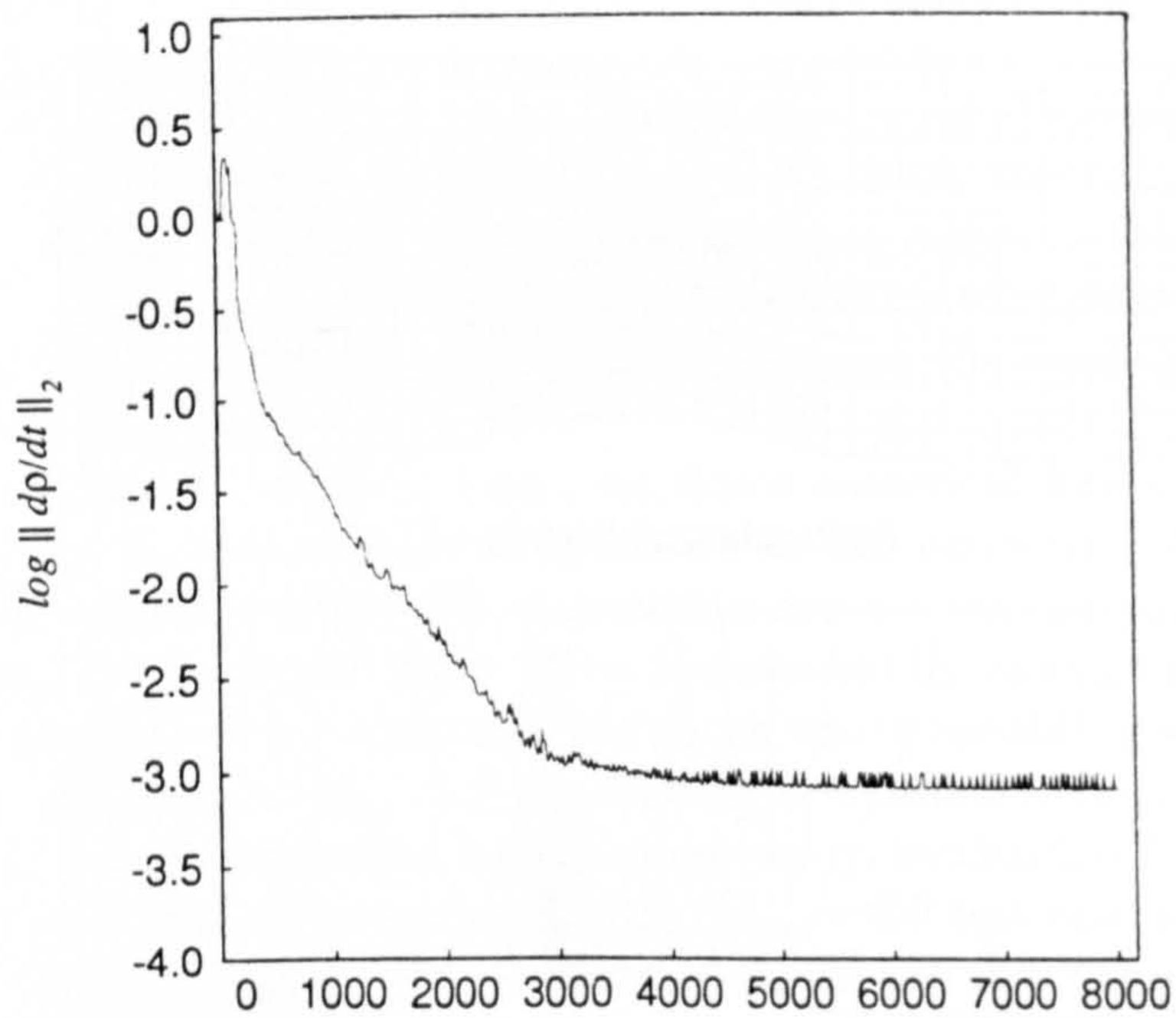


(c) Adapted grid (64 x 16 x 1)

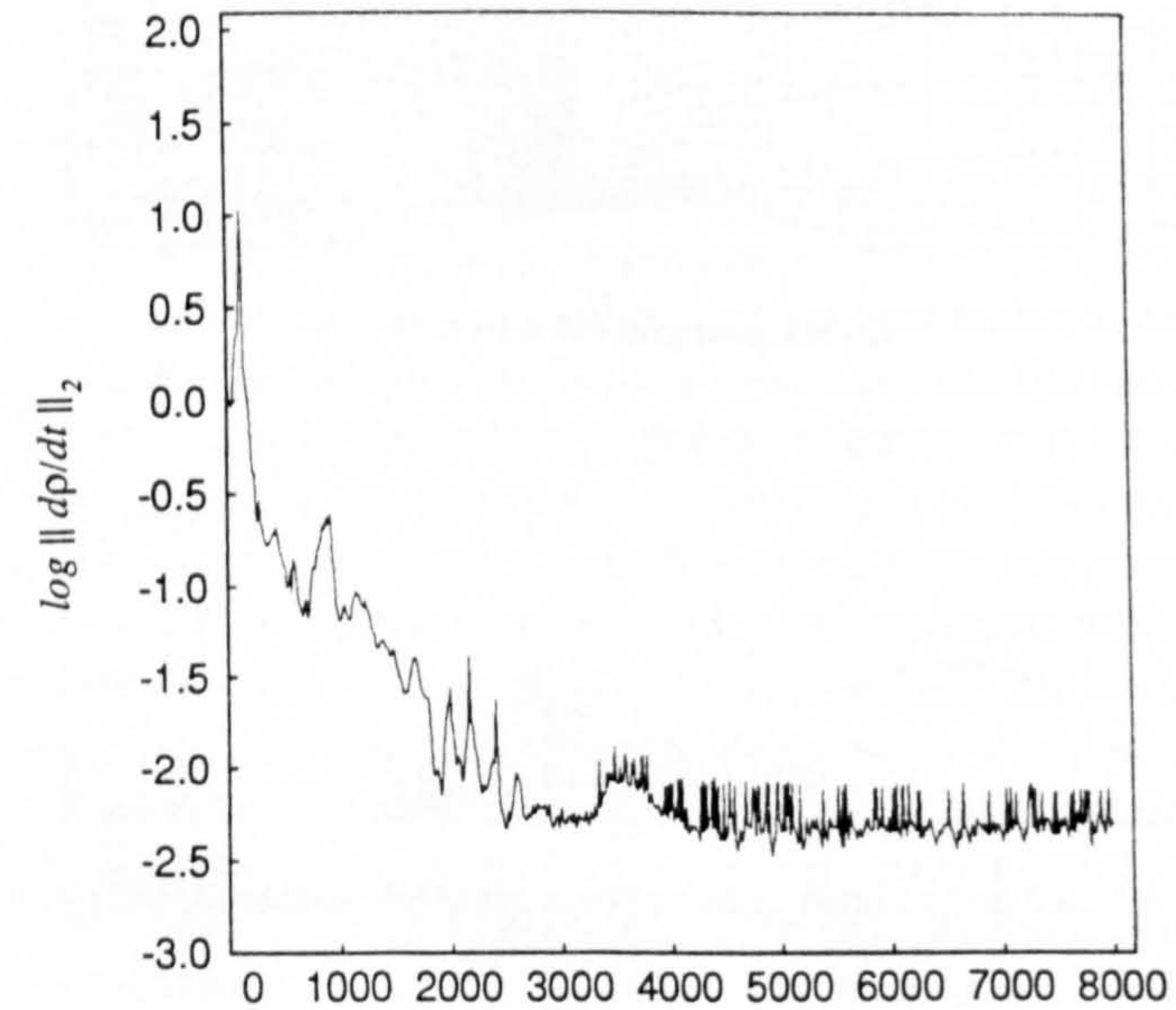


(d) Adapted grid solution -Lines of constant Mach number (ΔM=0.04)

FIGURE 3.1. GAMM channel transonic flow. Grids and isomach contours.

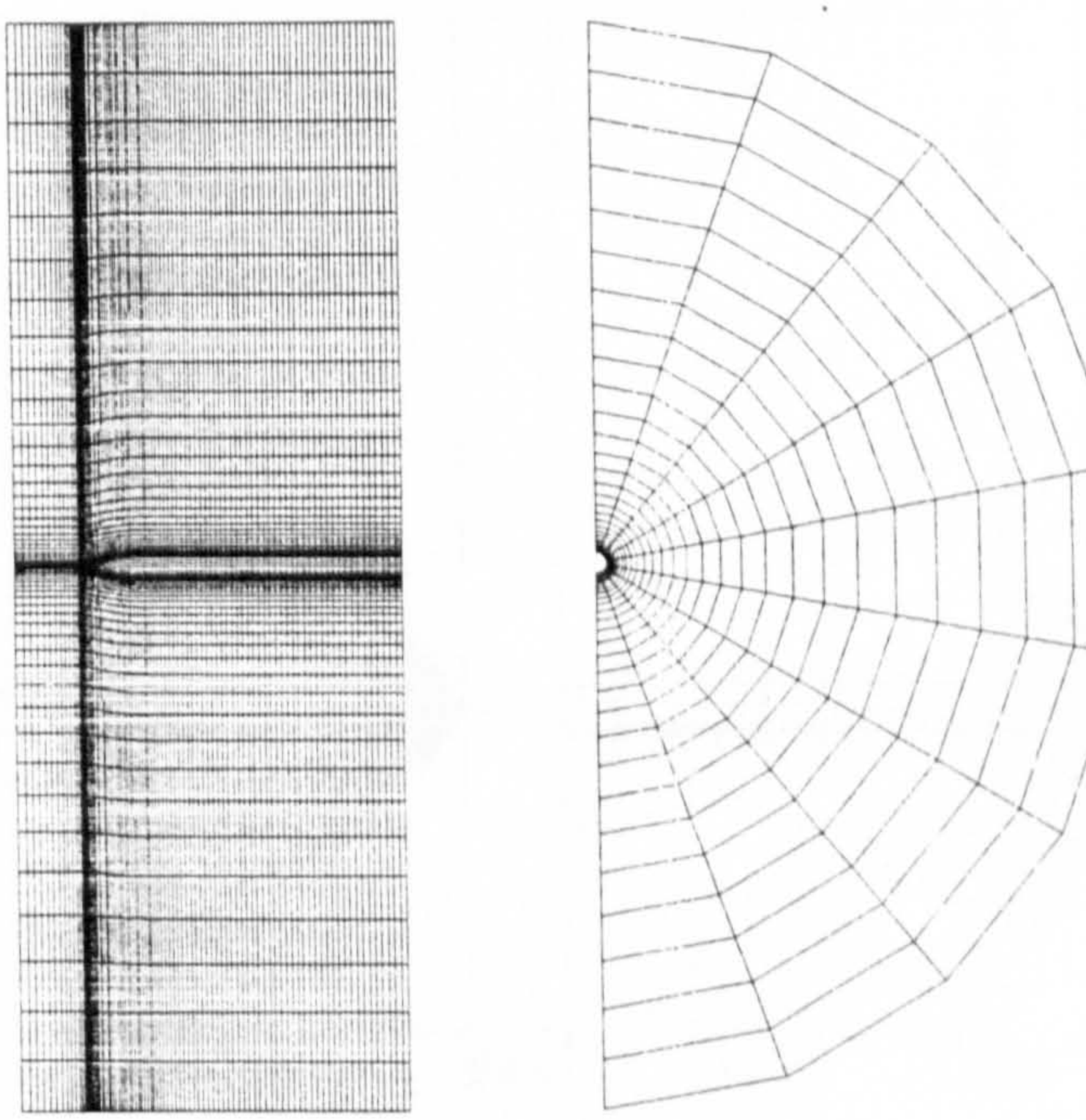


(a) Regular grid



(a) Adapted grid

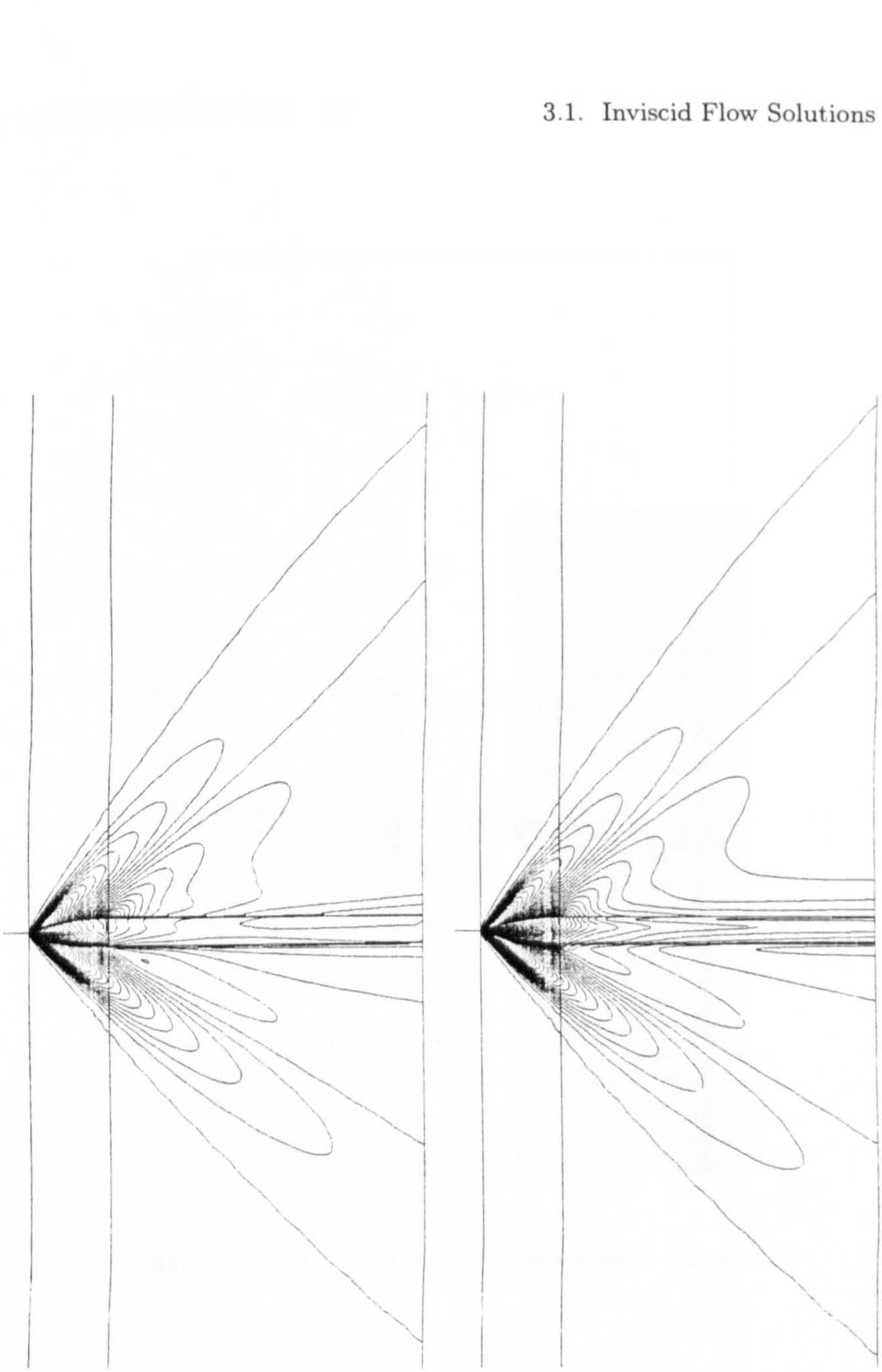
FIGURE 3.2. GAMM channel transonic flow. Convergence histories.



(a) Side view -complete grid

(b) Front view -farfield grid

FIGURE 3.3. Tangent-ogive foreboby grid, $(89 \times 29 \times 9)$.



(a) Lines of constant Mach number ($\Delta M = 0.01$)

(b) Lines of constant density ($\Delta \rho = 0.01$)

FIGURE 3.4. Tangent-ogive supersonic flowfield solutions.

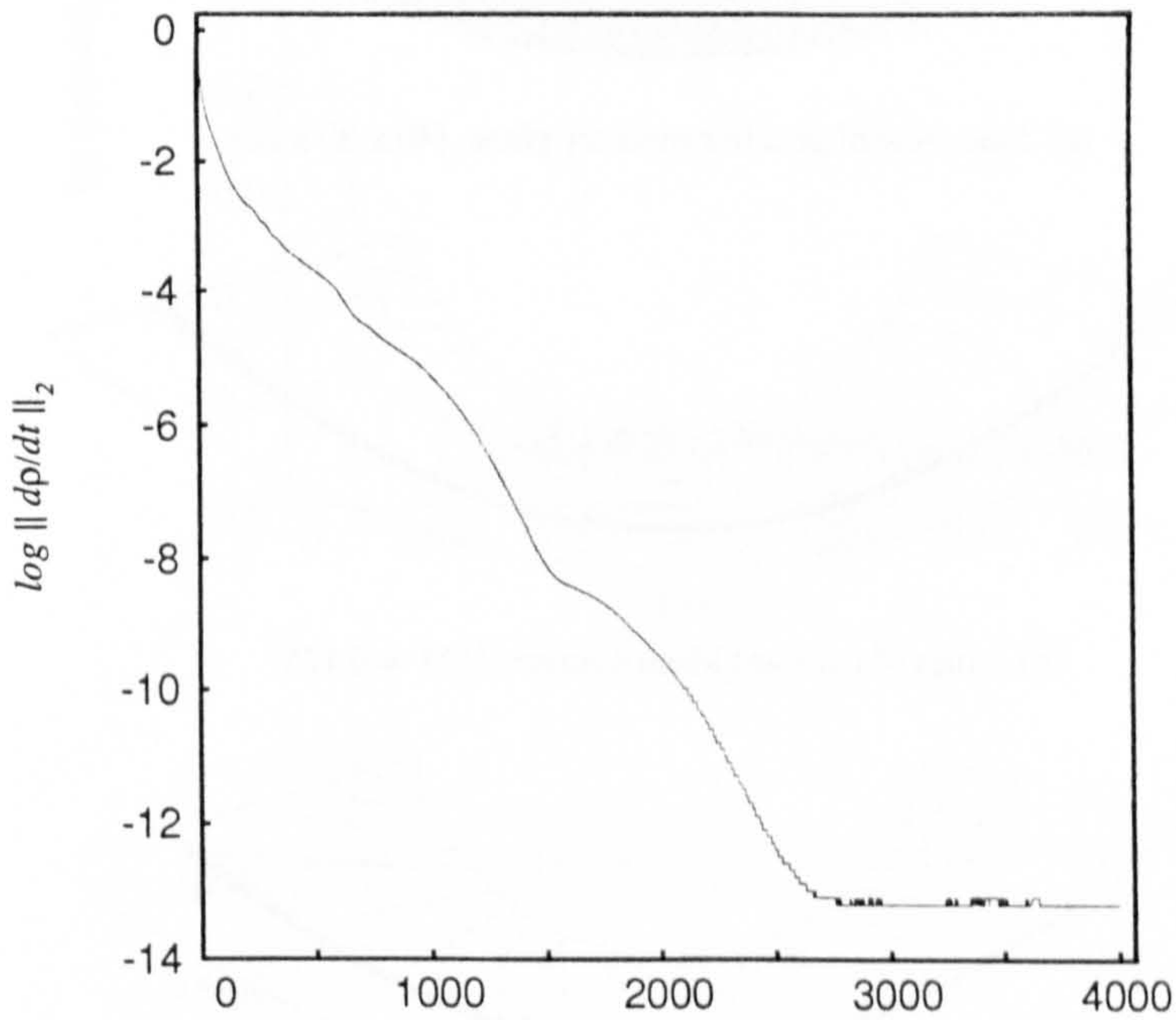
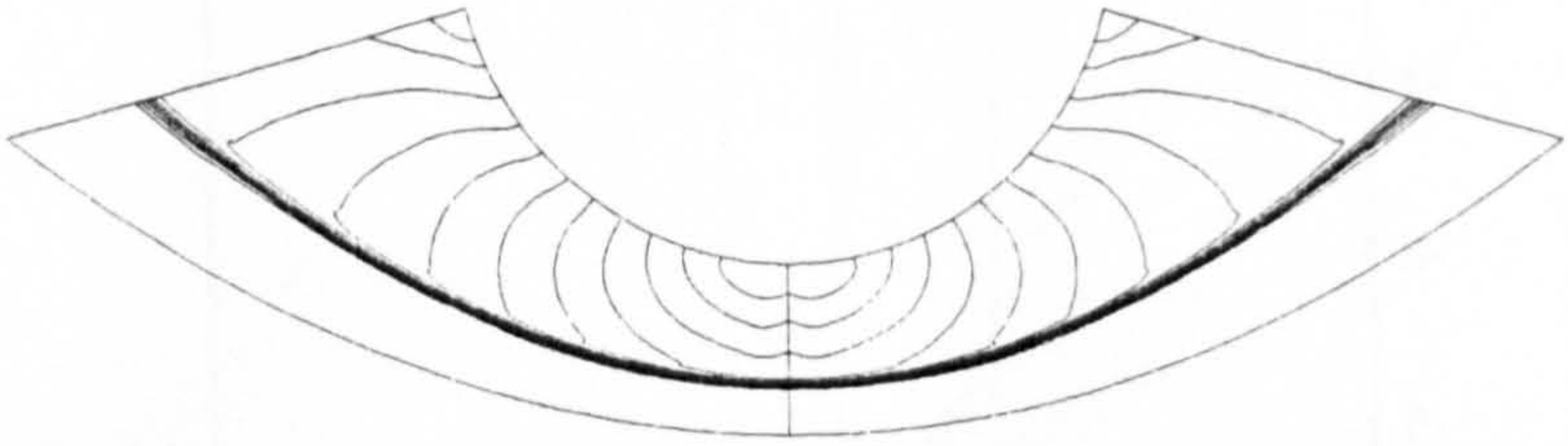


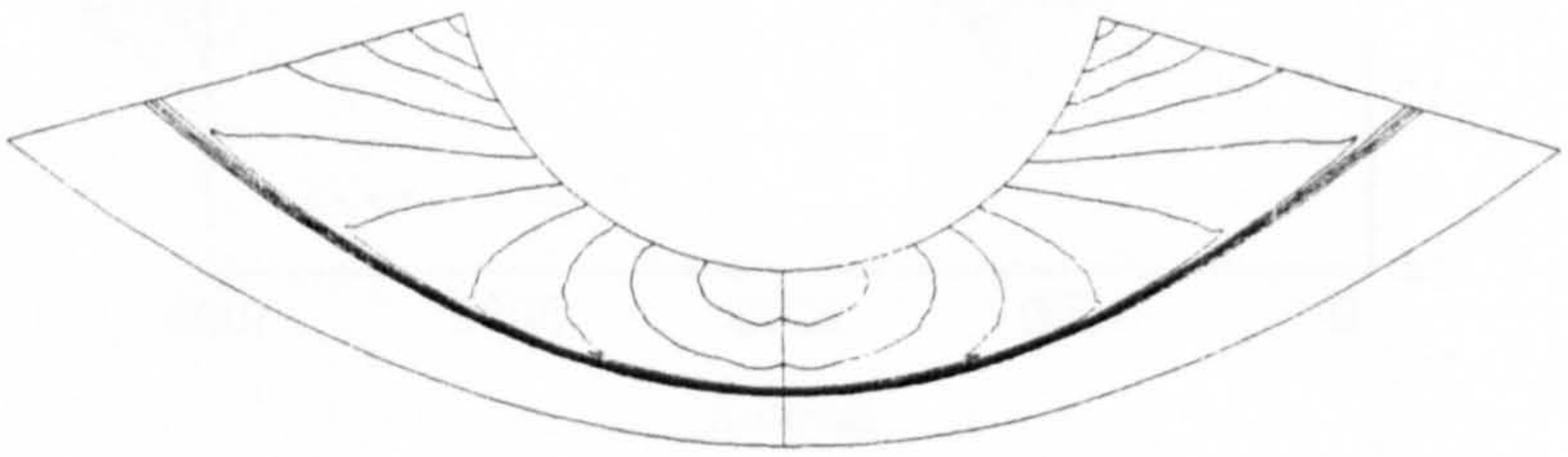
FIGURE 3.5. Tangent-ogive supersonic flow convergence history.



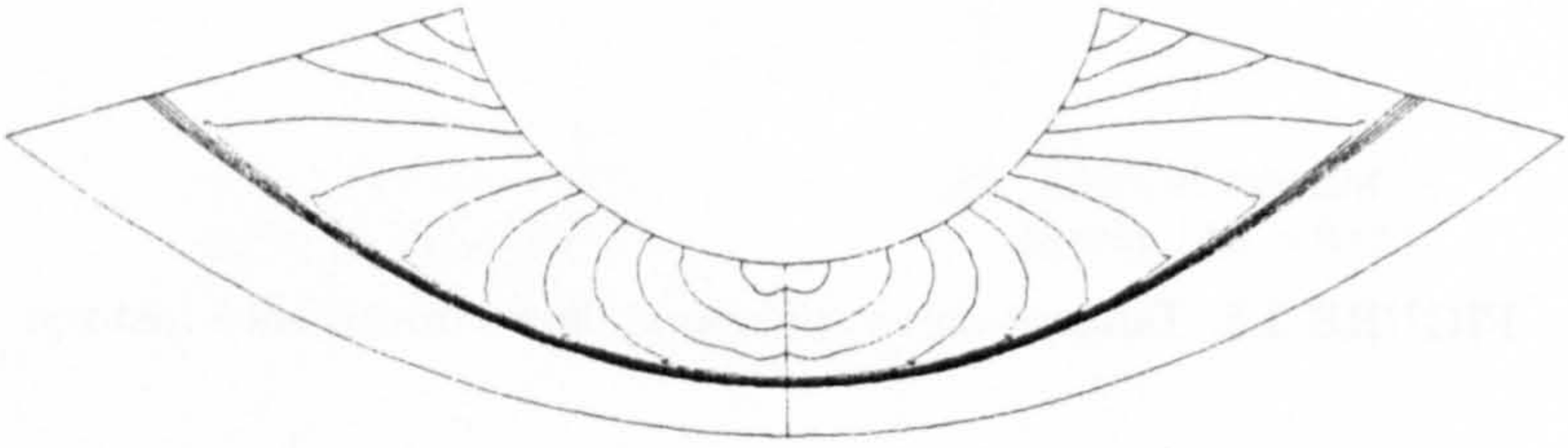
(a) Side view of grid in symmetry plane (40 x 20 x 20)



(b) Lines of constant Mach number ($\Delta M = 0.125$)

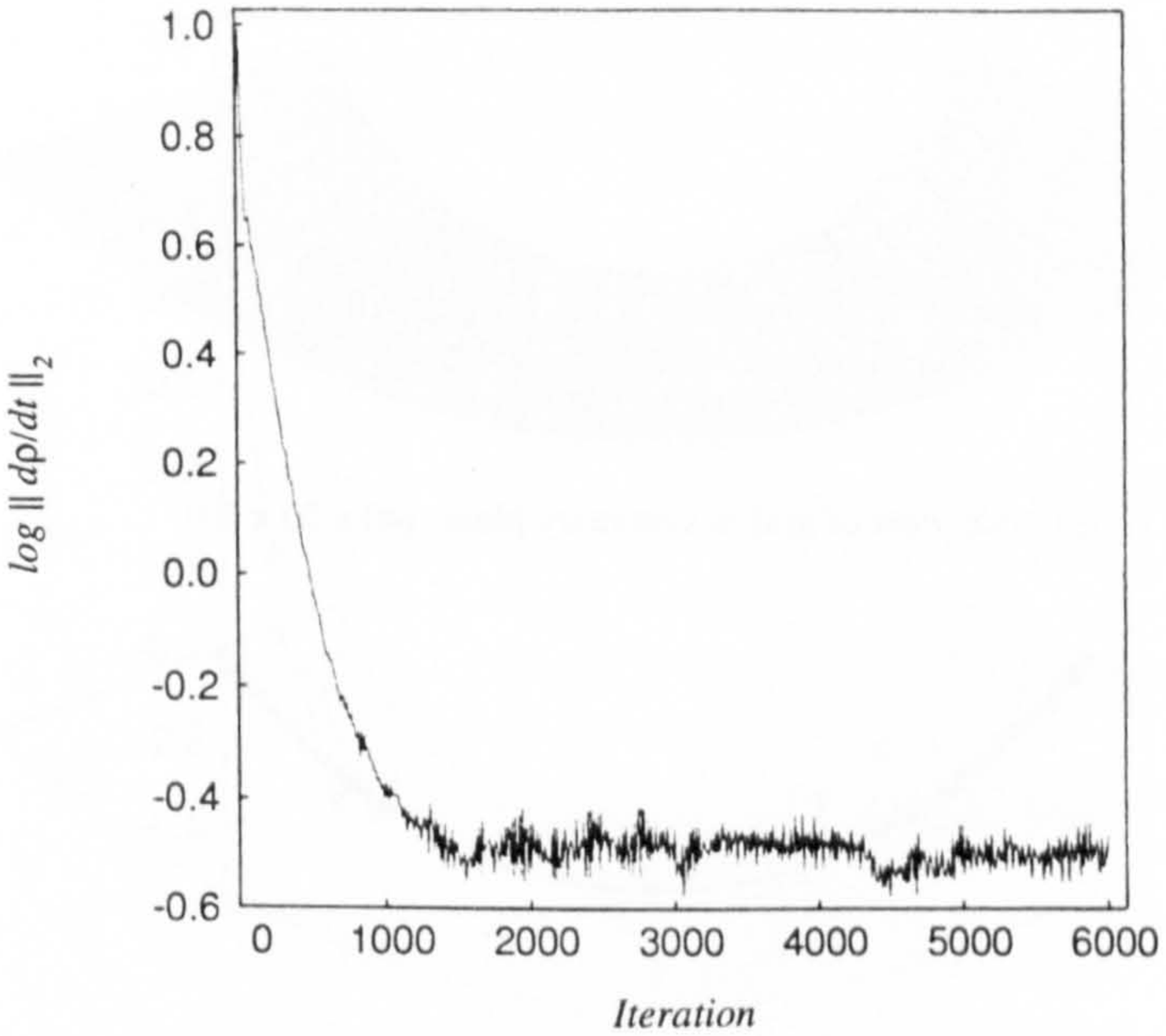


(c) Lines of constant density ($\Delta\rho = 0.2$)



(d) Lines of constant pressure p/p_∞, ($\Delta p/p_\infty = 0.4$)

FIGURE 3.6. Sphere flow at $M_\infty = 2.0$. Grid and flowfield solutions.



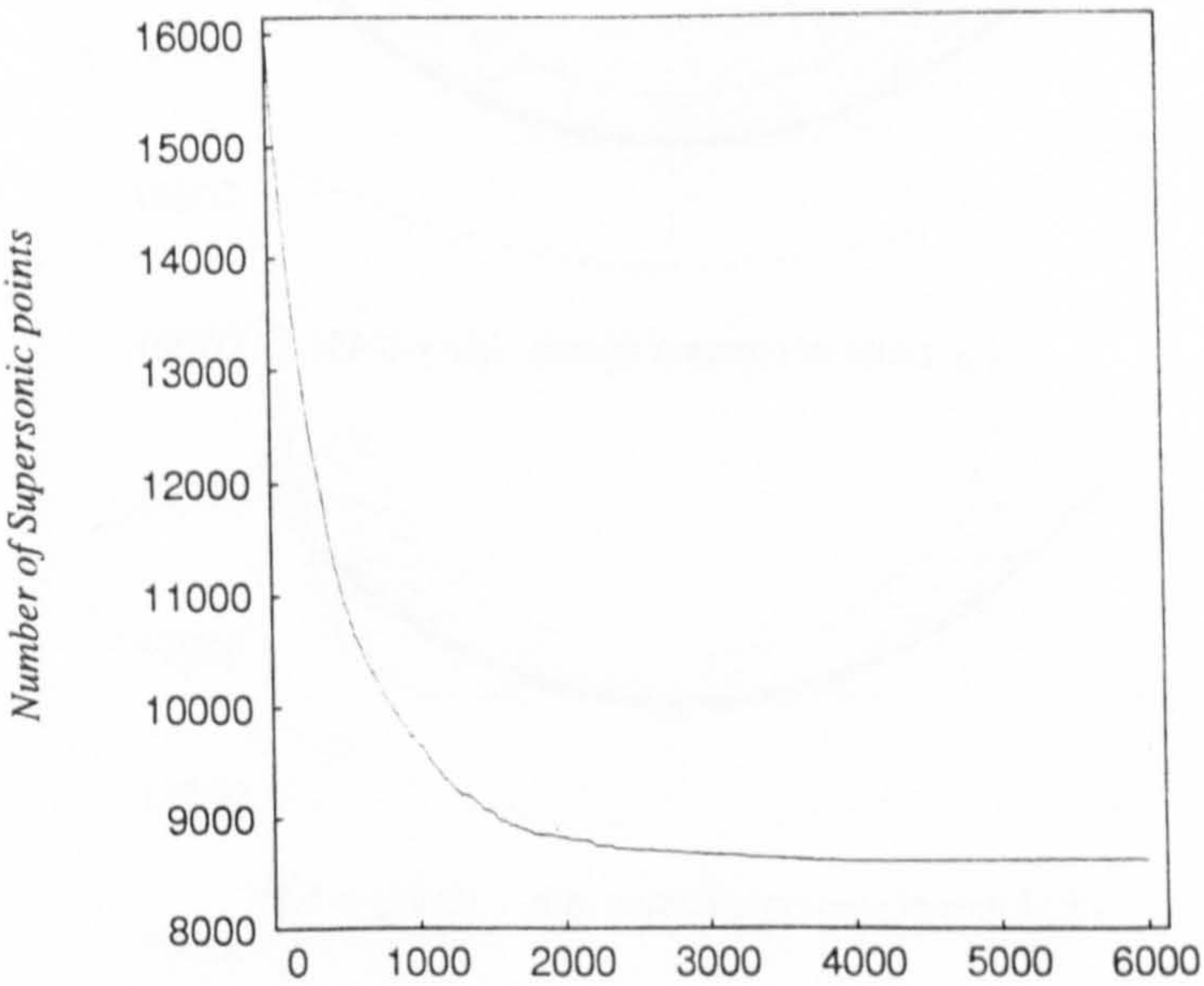
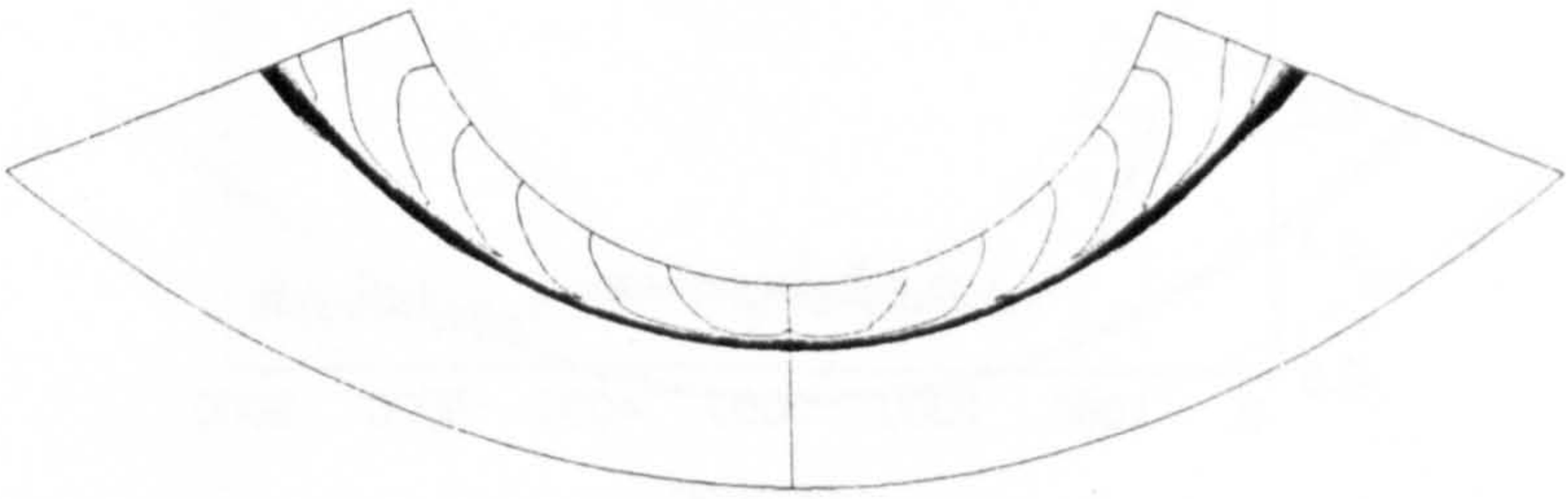


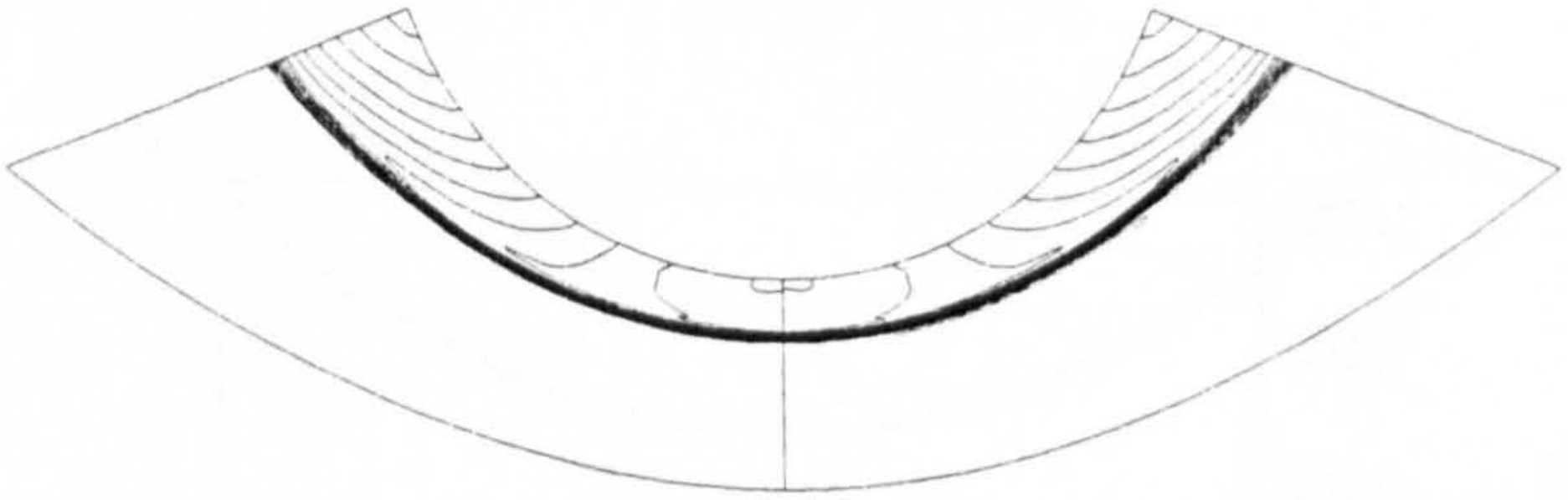
FIGURE 3.7. Sphere flow at $M_\infty = 2.0$. Convergence histories.



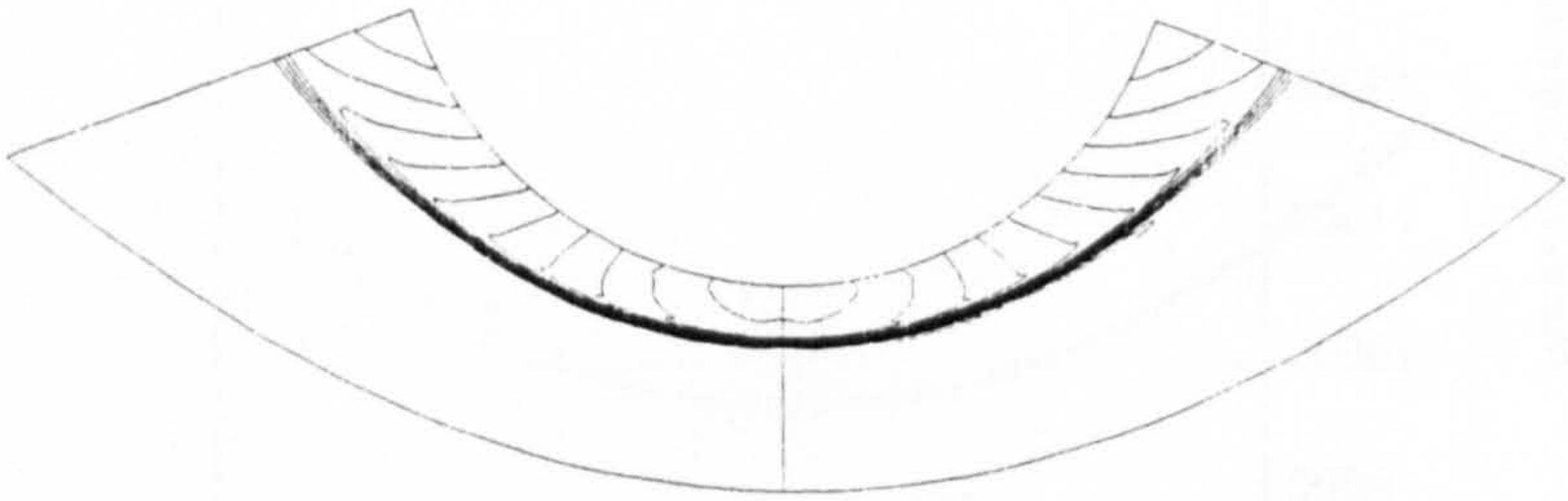
(a) Side view of grid in symmetry plane (40 x 20 x 20)



(b) Lines of constant Mach number ($\Delta M = 0.3$)



(c) Lines of constant density ($\Delta\rho = 0.45$)



(d) Lines of constant pressure p/p_∞, ($\Delta p/p_\infty = 6.5$)

FIGURE 3.8. Sphere flow at $M_\infty = 8.0$. Grid and flowfield solutions.

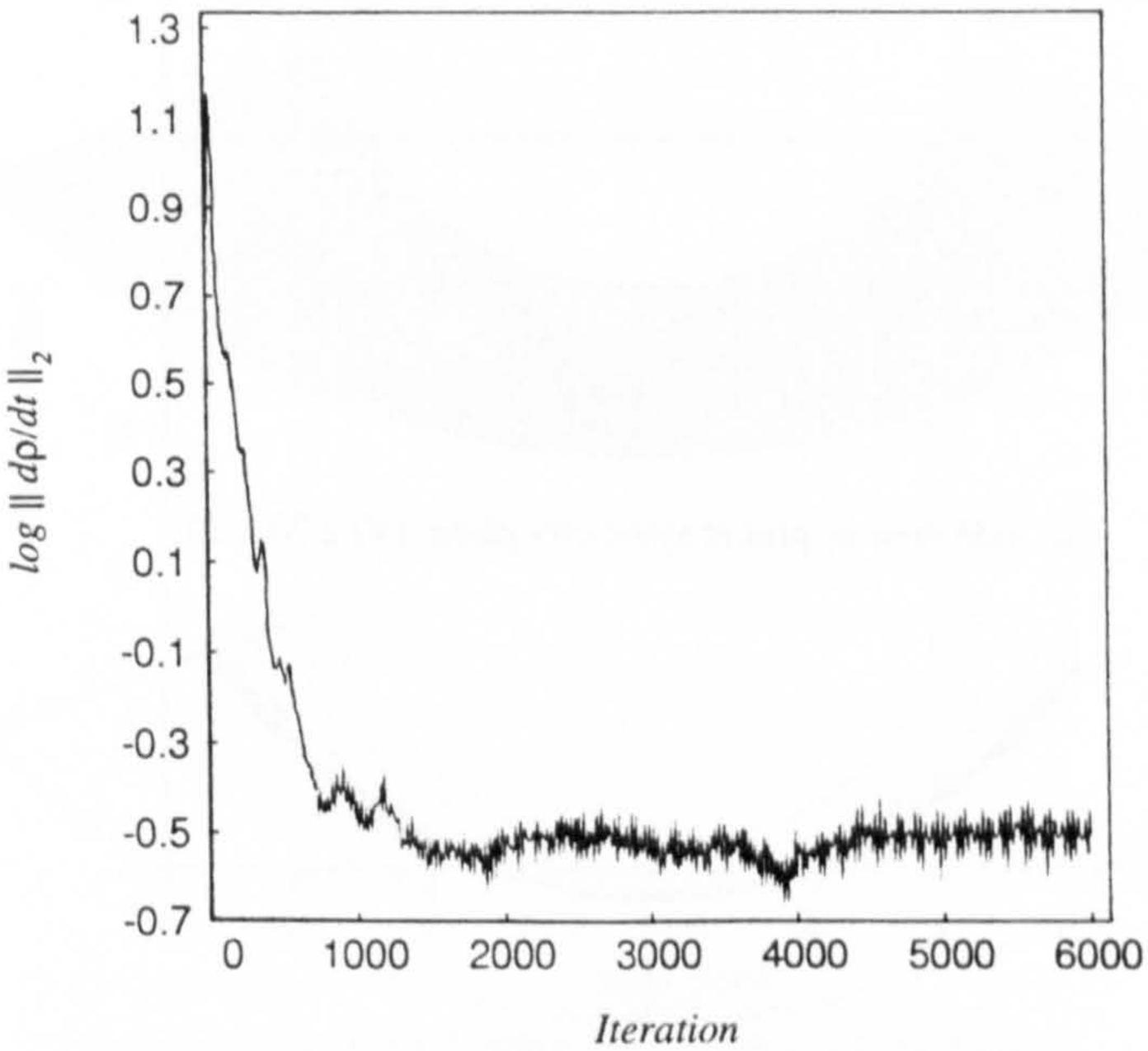
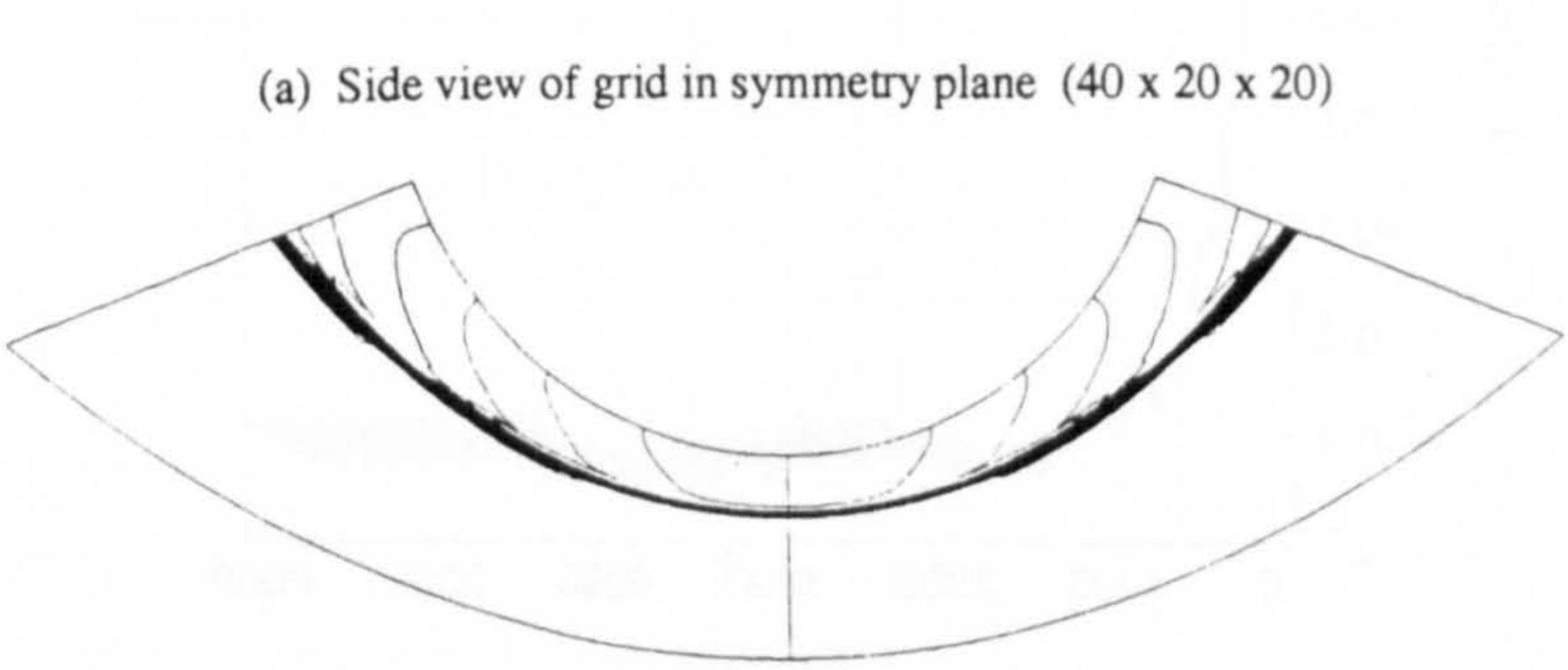


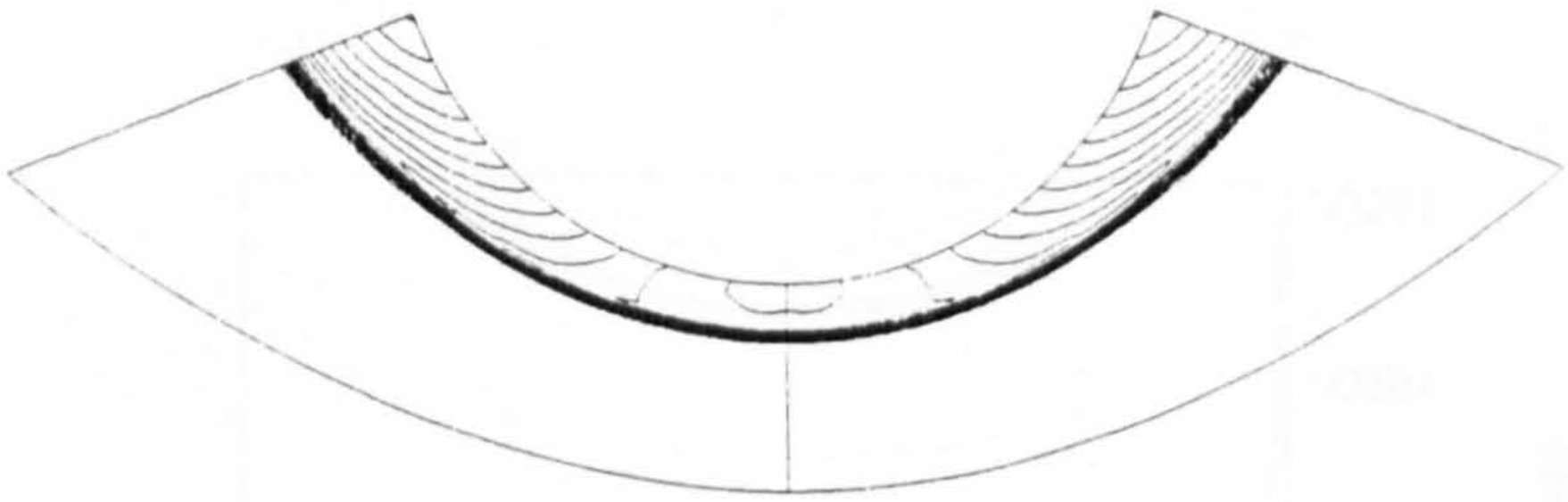
FIGURE 3.9. Sphere flow at $M_\infty = 8.0$. Convergence histories.



(a) Side view of grid in symmetry plane (40 x 20 x 20)



(b) Lines of constant Mach number ($\Delta M = 0.4$)

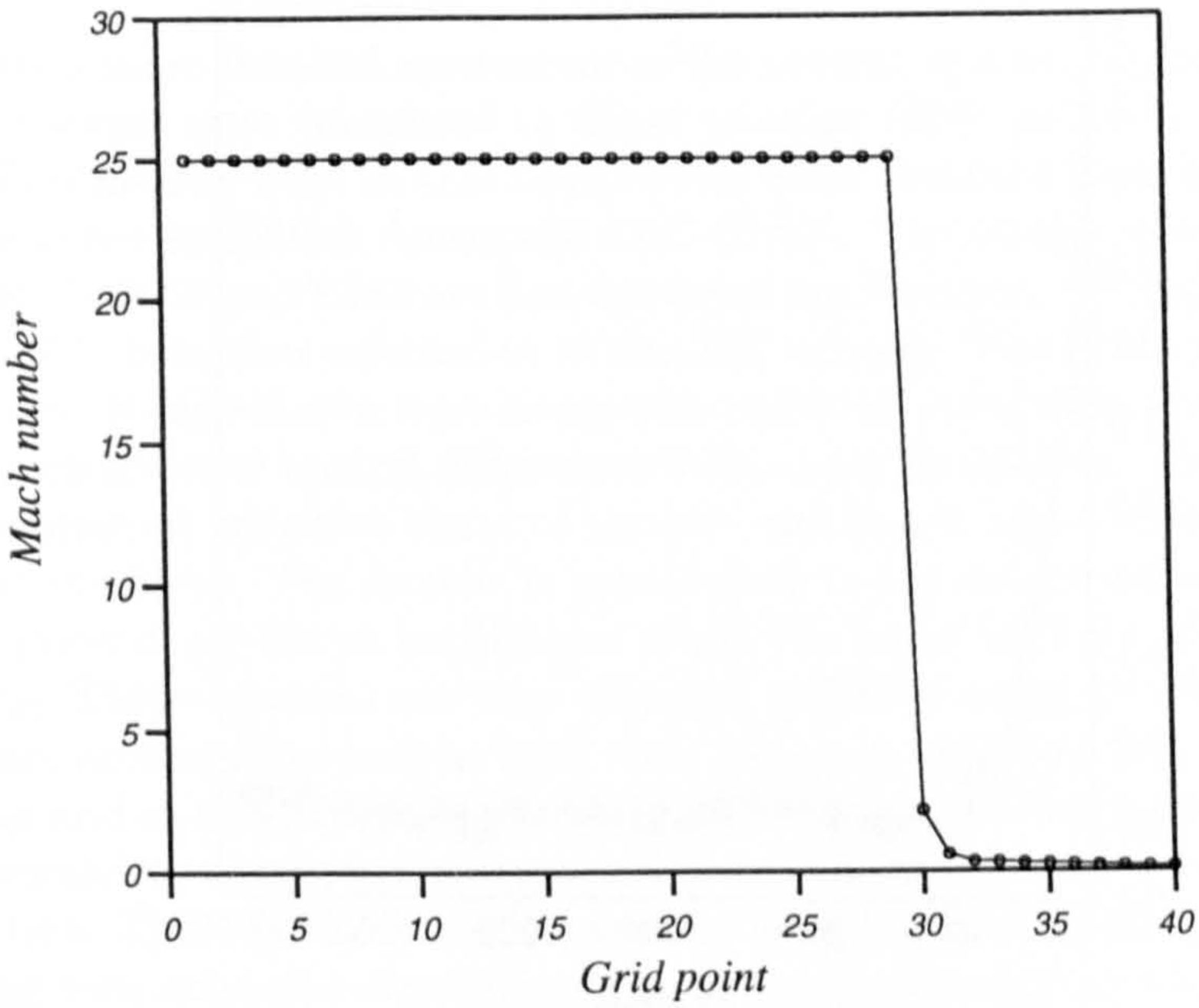


(c) Lines of constant density ($\Delta\rho = 0.4$)

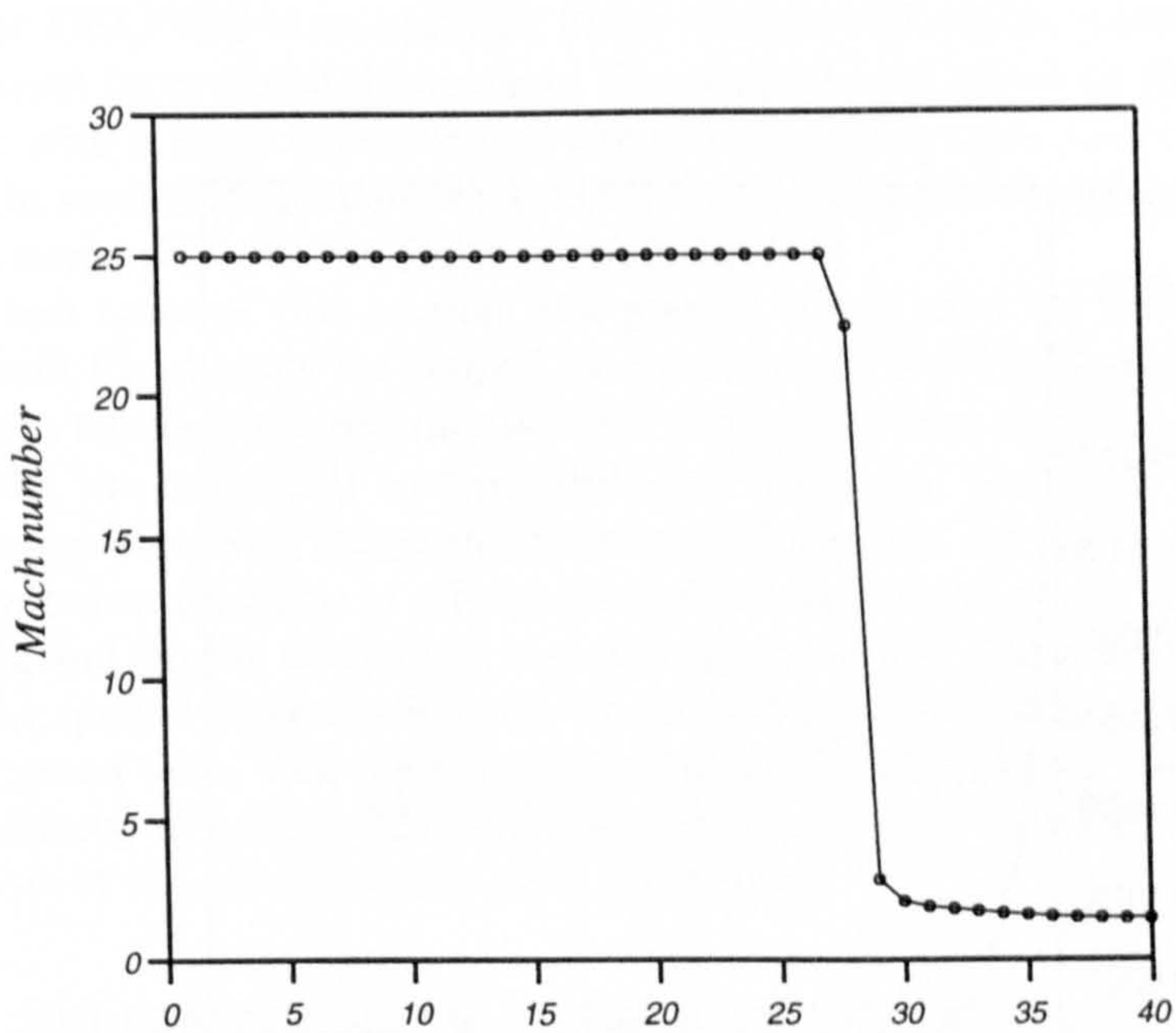


(d) Lines of constant pressure p/p_∞, ($\Delta p/p_\infty = 30.0$)

FIGURE 3.10. Sphere flow at $M_\infty = 25.0$. Grid and flowfield solutions.

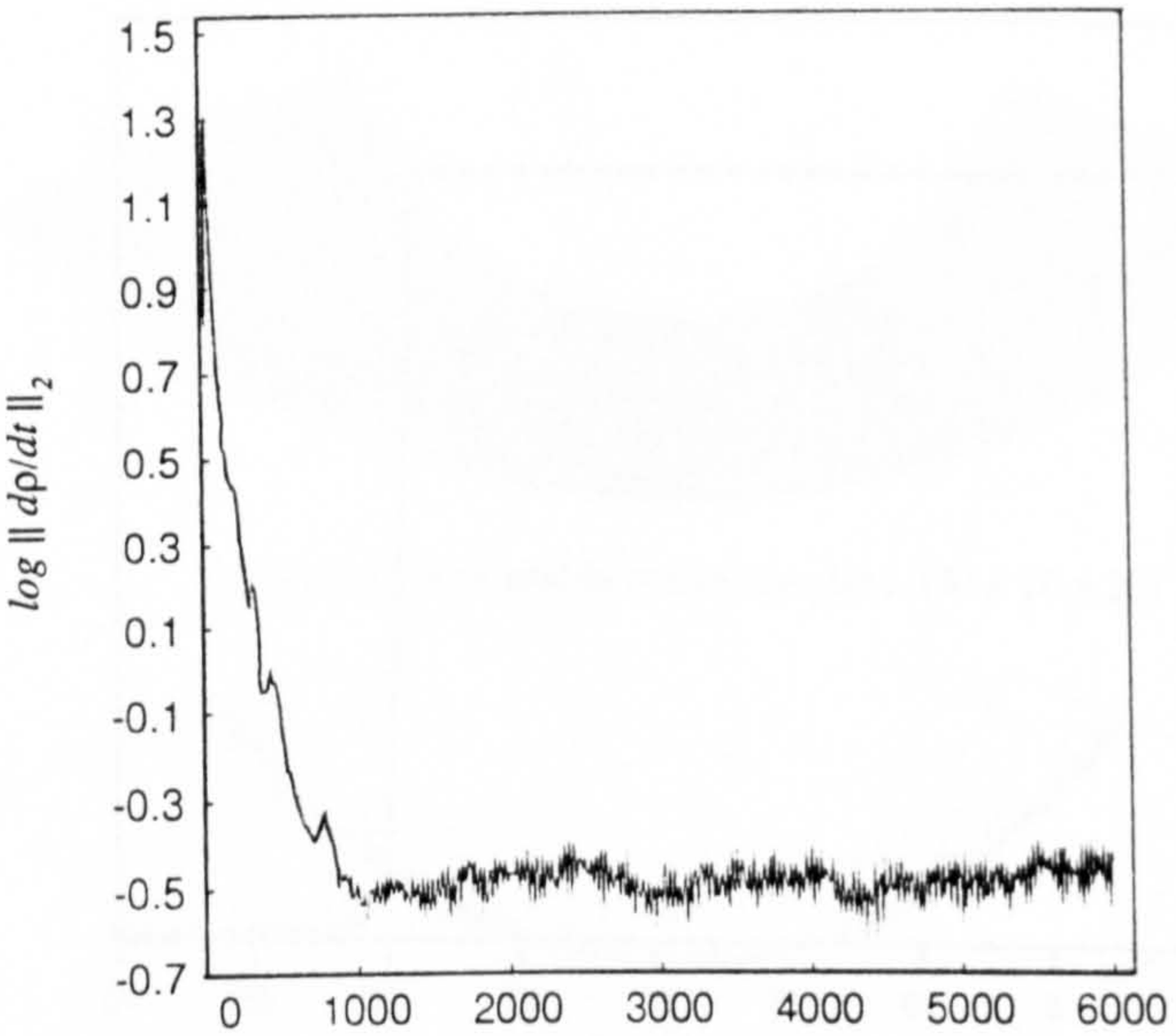


(a) Plot along the stagnation streamline.



(a) Plot along the ray at 34.5° to the vertical.

FIGURE 3.11. Sphere flow at $M_\infty = 25.0$. Plot of Mach number vs. grid point along rays normal to the surface.



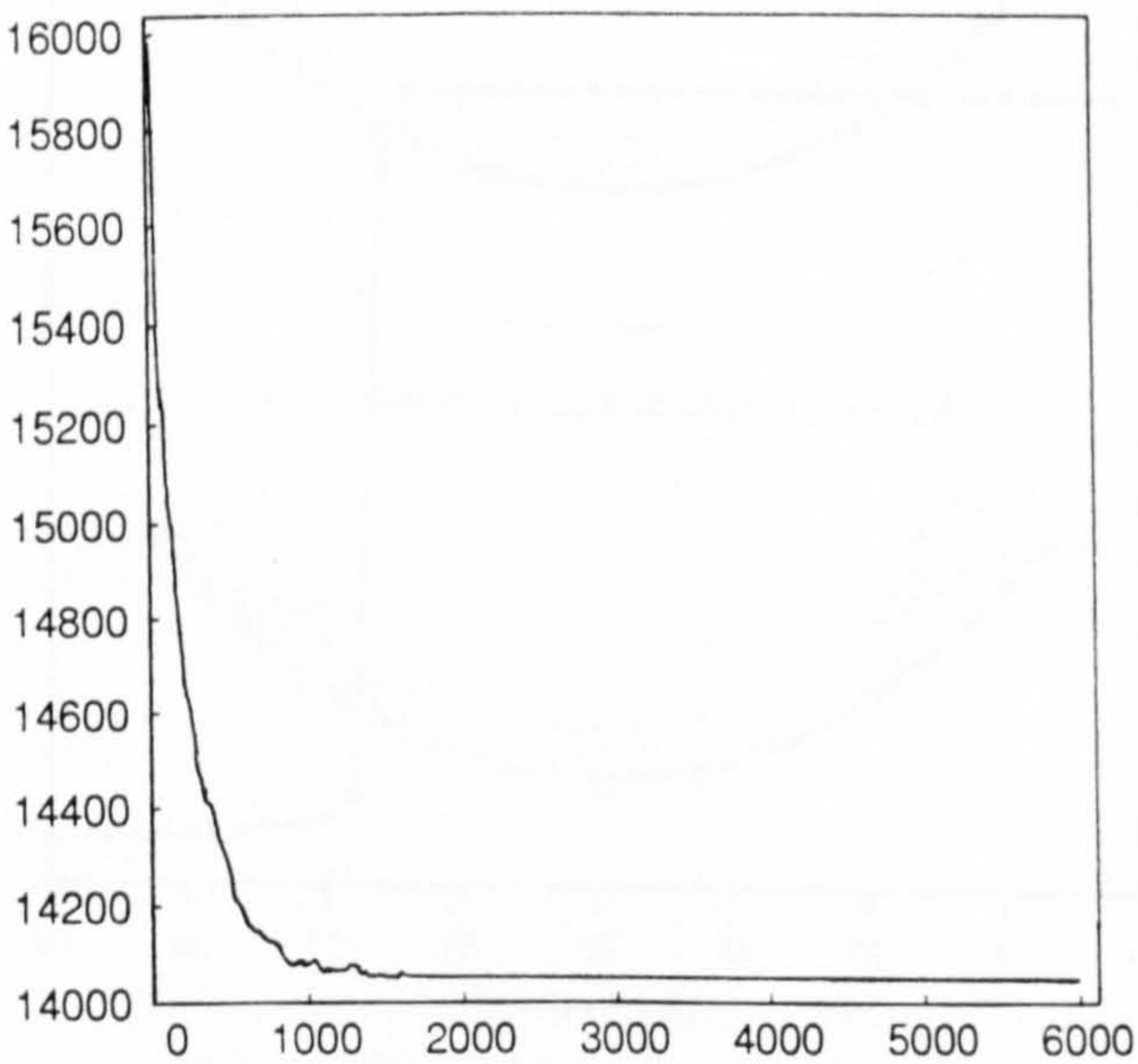


FIGURE 3.12. Sphere flow at $M_\infty = 25.0$. Convergence histories.

3.2 Comparison with Inviscid Solutions from Other Methods

To facilitate a more detailed assessment of the present upwind algorithm, further inviscid solutions were compared to those of other CFD methods. Most of the other CFD solutions used in this comparison were obtained from industrial design codes in use at British Aerospace PLC (BAe). The solvers referred to below as NS87, RANSMB and EJ83 are derivatives of the Jameson, Schmidt and Turkel algorithm [76], hereafter referred to as the JST scheme. The latter is an explicit, finite volume, Runge-Kutta time integration method. Its spatial discretization is based on second-order central differences with scalar dissipation. The dissipation terms comprise an adaptive blend of second- and fourth-order differences of the dependent variables. The former is introduced in the neighbourhood of shock waves to prevent pre-shock oscillations while the latter inhibits odd-even point decoupling. These schemes are very efficient, typically using several devices for convergence acceleration such as local time stepping, enthalpy damping, residual smoothing and multigrid. Refinements to these solvers include highly tuned dissipation modules, convergence acceleration enhancements and a high degree of vectorization. These CFD methods represent a mature industrial design capability, having seen extensive development over many years during which they have been validated on a wide variety of flow conditions and complex geometries. The final solver DELPHIS is an explicit, finite volume, cell vertex scheme. It is also supplemented by artificial dissipation. The algorithm is based on the Ni scheme [106], but with a more sophisticated dissipation model than the original Ni algorithm. In several of the plots presented, the present upwind scheme is referred to by the name given to the Euler variant, ER3D.

In the test cases of this section, the present solver used the minmod limiter function and the entropy fix defined by equation (2.72–2.74). It did not use any convergence acceleration techniques other than local time stepping. Where data is available, the numerical results from each solver are compared for accuracy and efficiency. However, comprehensive comparisons are not always possible due to the limited availability of off-the-shelf flowfield solutions and the expense of generating and storing the large data sets associated with fine grid computations. For all the quoted force coefficients, the reference area is taken to be the half-wing projected area. The figures quoted for CPU and memory usage refer to a single processor of a *Cray YMP-4E* supercomputer with 6 nanosecond clock, on which most of the computations were carried out.

3.2.1 Supersonic flow over a blunt-nose body

This test case was originally considered by the GARTEUR AG-15 Action Group for Supersonic Euler flows [46] in 1991. Participants of that exercise were given a series of demanding 3-D inviscid supersonic test cases to undertake with the aim of comparing solution quality of their flow solvers. The exercise attracted participants from European industry and the research establishments comprising ONERA, DLR, NLR, BAe Dynamics, Aerospatiale, Matra and MBB. Each participant was supplied the flow conditions and grids for several test cases. One of these cases was the ONERA blunt-nose body for which the flow conditions

are $M_\infty = 2.96$ and $\alpha = 10°$. Three grids were provided, coarse, medium and fine. The present computation was carried out after the original exercise, but the mesh used is the medium mesh ($64 \times 32 \times 32$) provided by GARTEUR. Views of the mesh are shown in Figure 3.13.

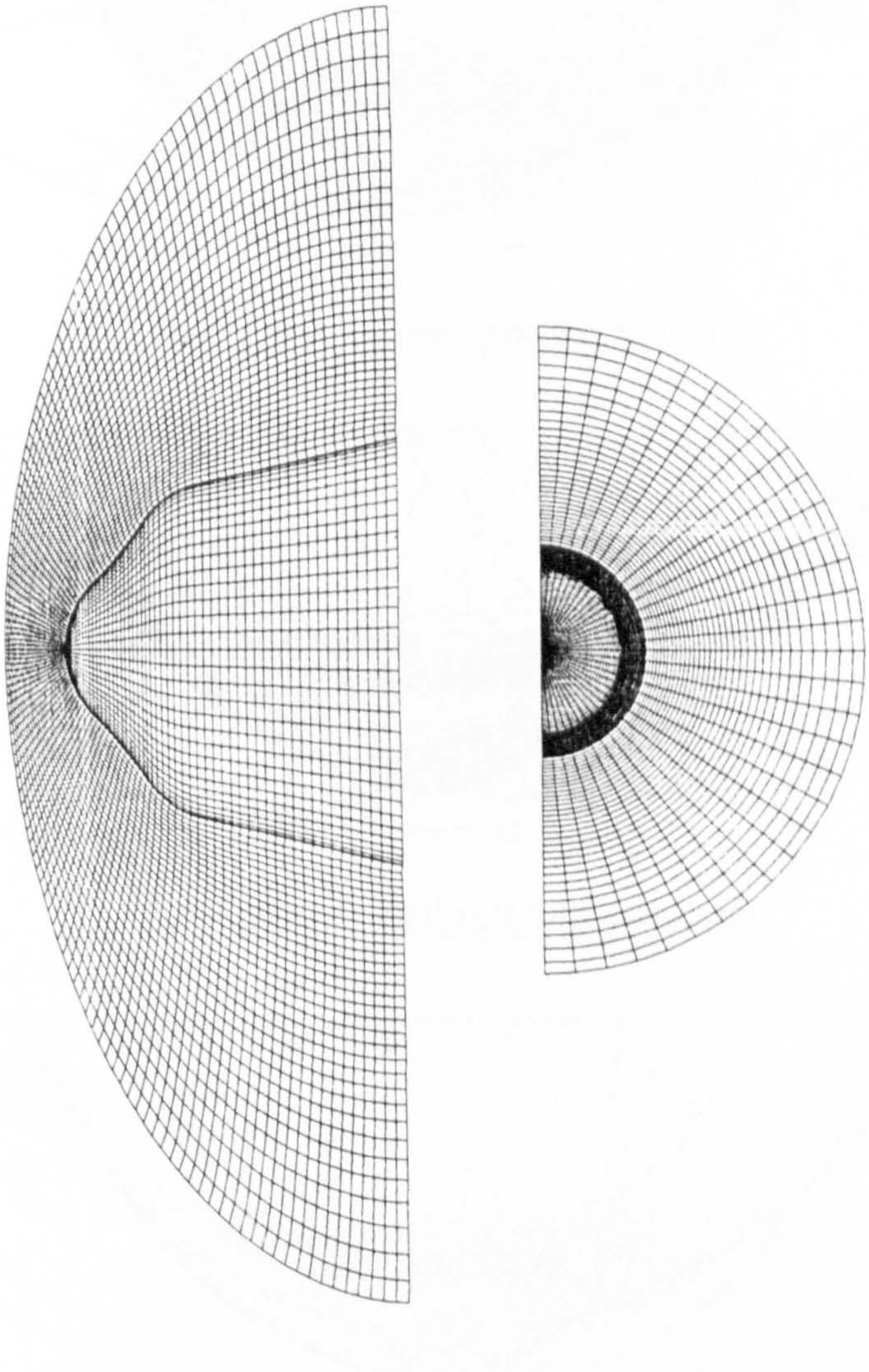
Here, the solution of the present upwind code is presented along with solutions produced by two GARTEUR participants. Figure 3.14 shows the three numerical solutions in terms of isomach contours ($\Delta M = 0.1$) in the symmetry plane. Part (a) of the figure is a cell vertex upwind solution computed by DLR, part (b) a scalar dissipation central difference solution from a JST algorithm, while part (c) is the solution from the present solver. Each of these solutions capture the essential features of the flow: the outer bow shock orientation and position, the expansions over each shoulder of the body and, the stagnation point location. The DLR solution was of the highest quality presented at GARTEUR while the JST solution was typical of several of the lesser quality solutions. The solution of the present solver is of comparable if not higher accuracy than the DLR solution, having marginally better shock resolution. Both solutions render sharp discontinuities and clean surface contours. In contrast, the JST solution suffers from wiggles in the surface contours and to some extent in some of the flowfield features. Towards the outflow plane, the shape of the outer bow shock of the JST solution curves closer to the body than the other solutions presented, suggesting a tendency of the shock to align itself with the grid. There is also an unexpected irregularity in the isomach contours ahead of the nose, slightly above the stagnation region.

Figure 3.15 shows isobar contours ($\Delta p/p_\infty = 0.5$) in the same format as the preceding figure. Again the present solution is of comparable quality to the DLR one. In the JST solution, wiggles appear in the expansion about the second shoulder of the nose on both leeward and windward sides. Figures 3.16 and 3.17 compare the DLR solution with that of the present solver, this time in the crossflow plane, again in terms of isomach and isobar contours. The increment between contours is the same as that of the preceding plots. The flow features are remarkably similar and comparison of the bow shock contours indicate that the present solution is of higher resolution than the DLR one.

In the GARTEUR exercise only Aerospatiale and Matra elected to present convergence data for this test case [46]. The best Aerospatiale code reduced the residual by only two orders of magnitude using a first-order implicit solution. Apparently, they were unable to obtain a second-order solution. Matra's convergence data also indicated a drop in residual of only two orders of magnitude. This suggests that the convergence of numerical solutions to this particular test case was problematic for at least some of the participants' solvers. Convergence histories from the present upwind solver are given in Figure 3.18 in terms of the log of the l_2 norm and the number of supersonic points in the computational domain. The former exhibits a drop of over three orders of magnitude before levelling out.

3.2.2 Supersonic flow over a wing-body

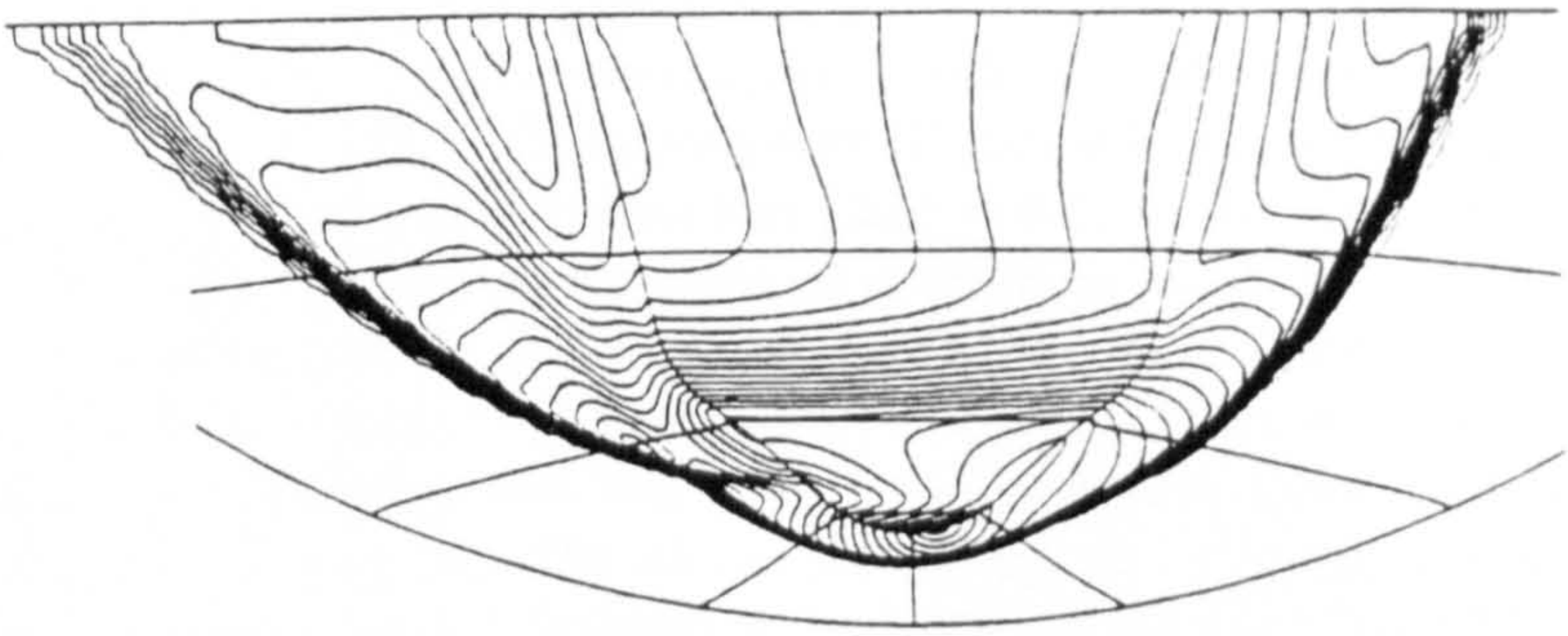
This test case is based on a combat aircraft wing-body research configuration known as M165. The body is flat-sided along the length of the wing root. The wing is low-mounted with a leading-edge sweep of 58° and an aspect ratio of 2.3.



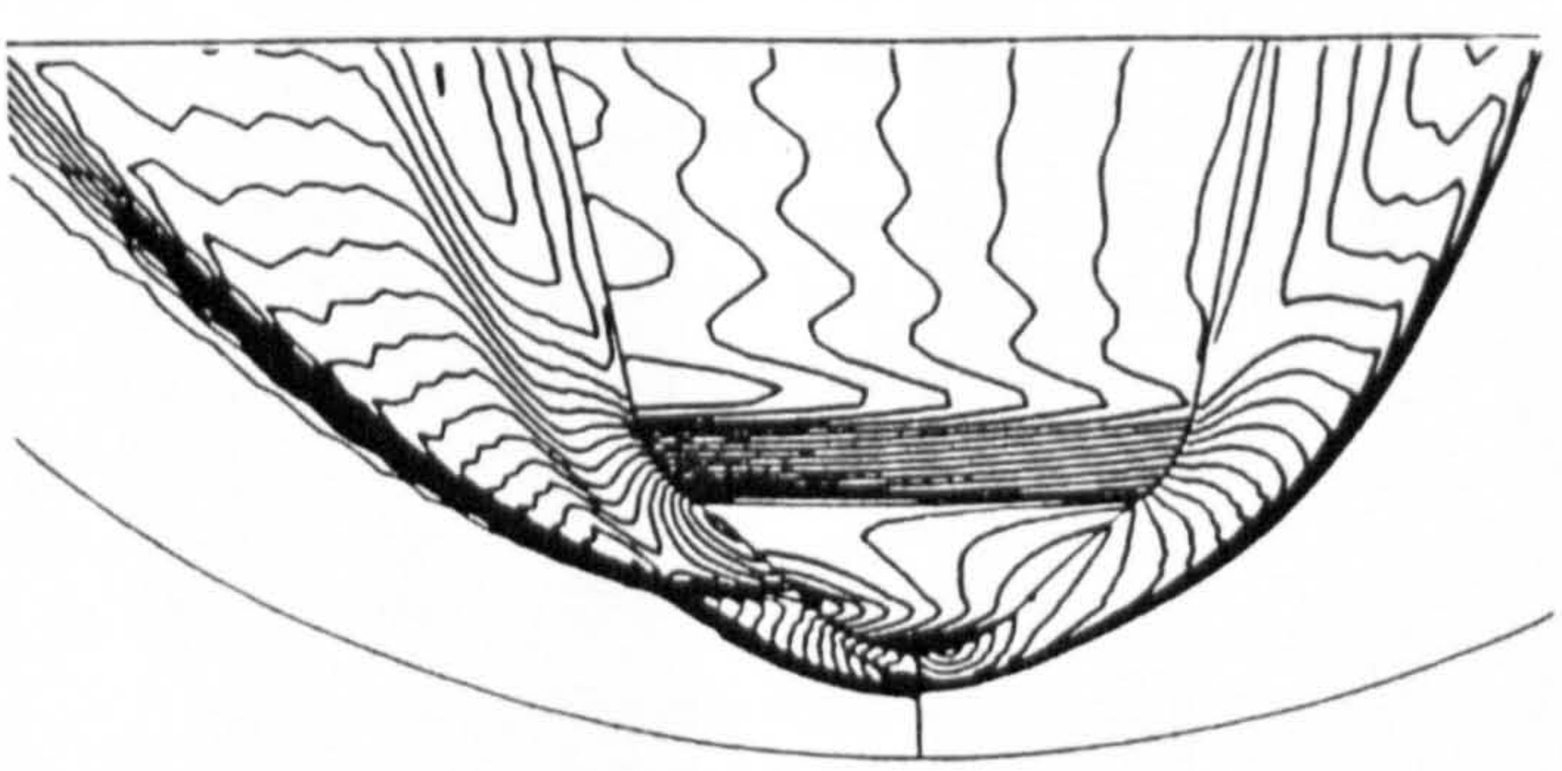
(a) Side view

(b) Front view -not to same scale as side view

FIGURE 3.13. ONERA blunt-nose body medium grid, $(64 \times 32 \times 32)$.



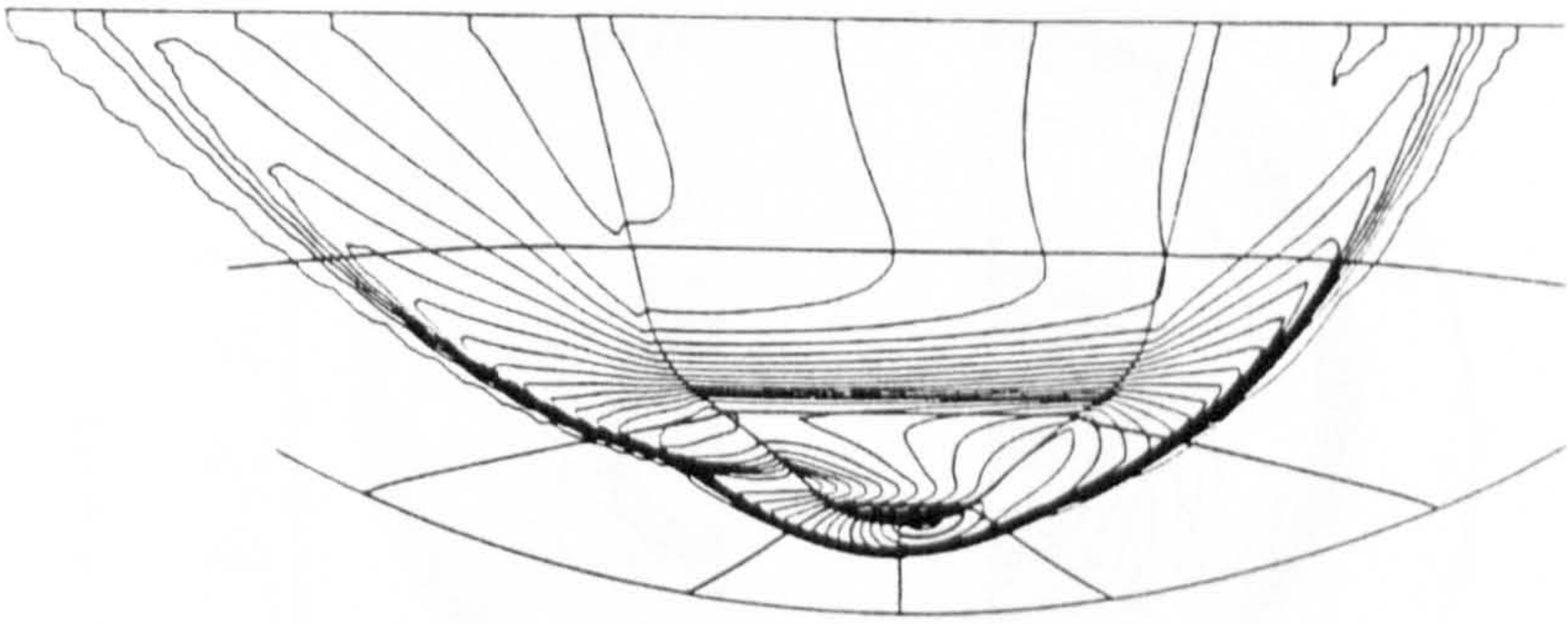
(a) DLR cell-vertex upwind scheme solution



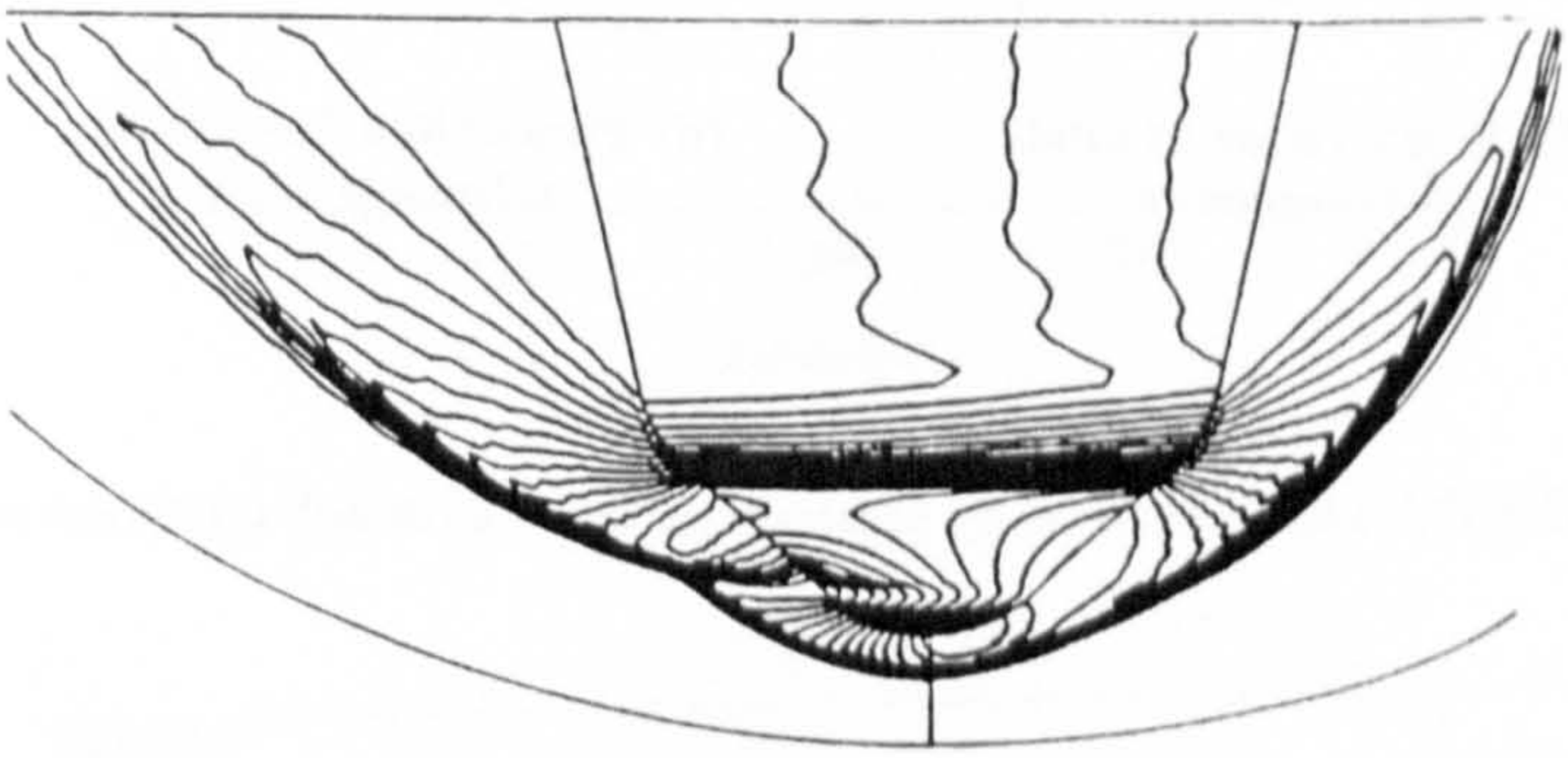
(b) Central difference scheme (Jameson) solution

(c) Present Roe flux upwind scheme solution

FIGURE 3.14. ONERA blunt-nose body supersonic flow. Isomach contours.



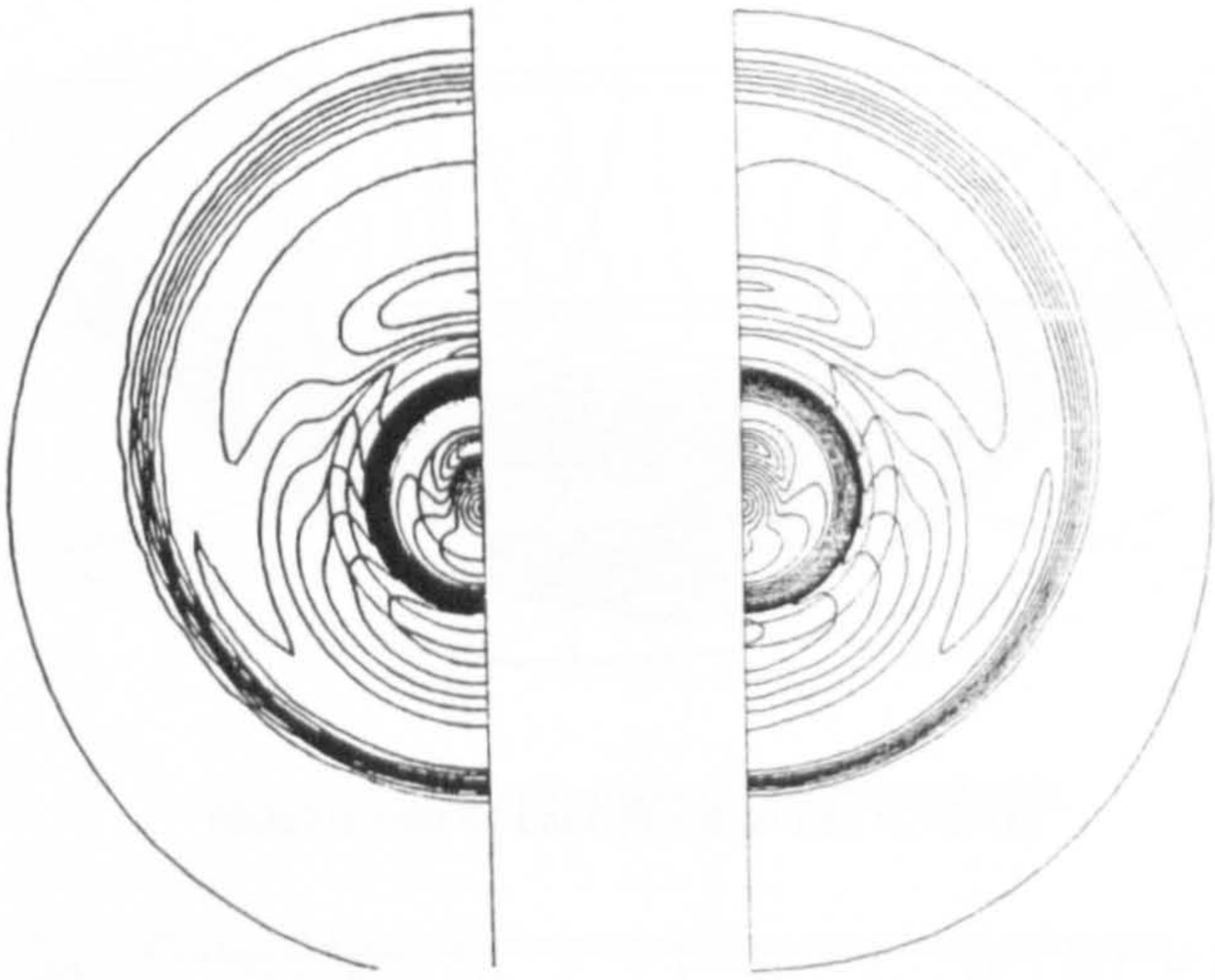
(a) DLR cell-vertex upwind scheme solution



(b) Central difference scheme (Jameson) solution

(c) Present Roe flux upwind scheme solution

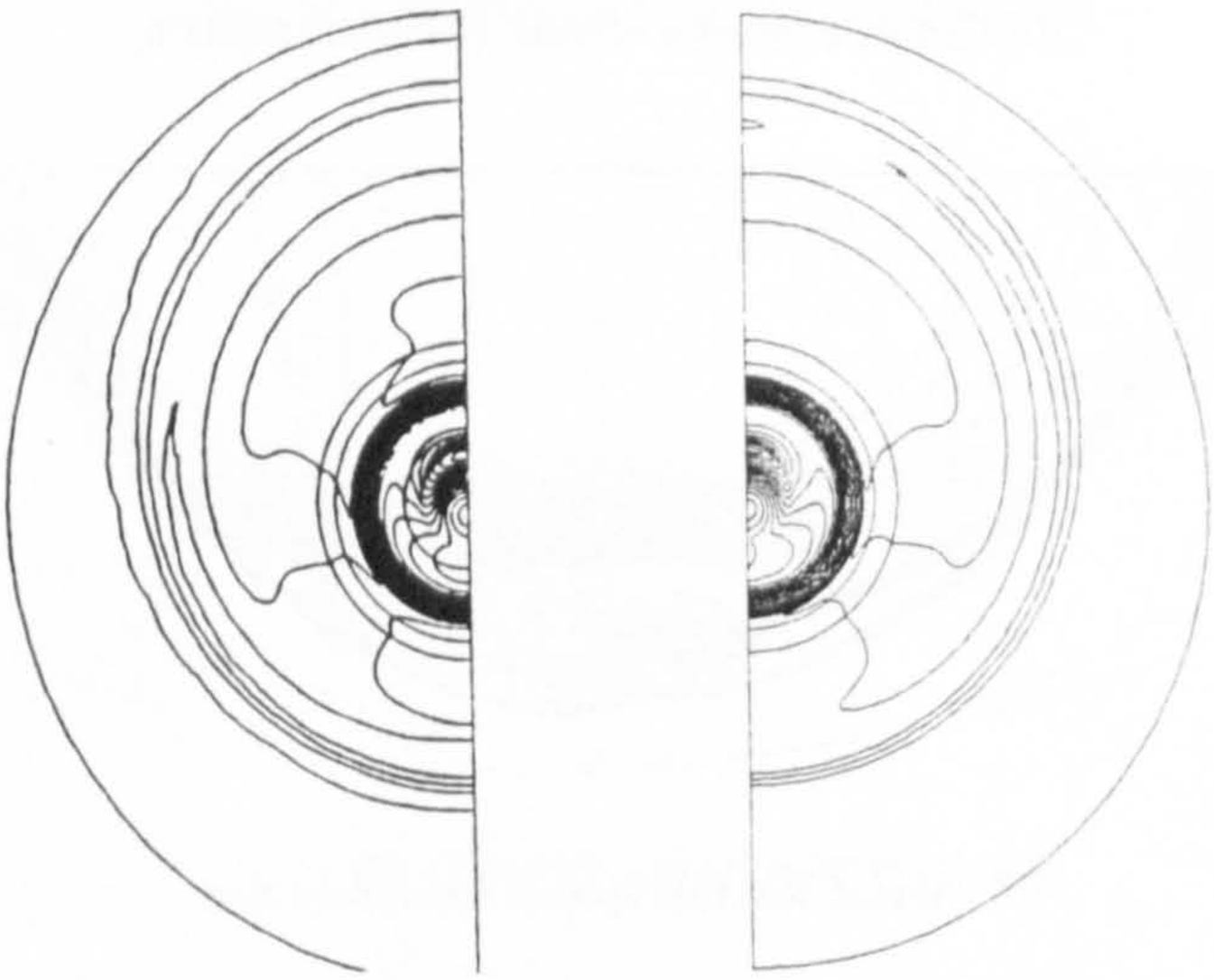
FIGURE 3.15. ONERA blunt-nose body supersonic flow. Isobar contours.



(a) DLR cell-vertex upwind scheme solution

(b) Present Roe flux upwind scheme solution

FIGURE 3.16. ONERA blunt-nose body supersonic flow. Isomach contours in the cross-flow plane.



(a) DLR cell-vertex upwind scheme solution

(b) Present Roe flux upwind scheme solution

FIGURE 3.17. ONERA blunt-nose body supersonic flow. Isobar contours in the cross-flow plane.

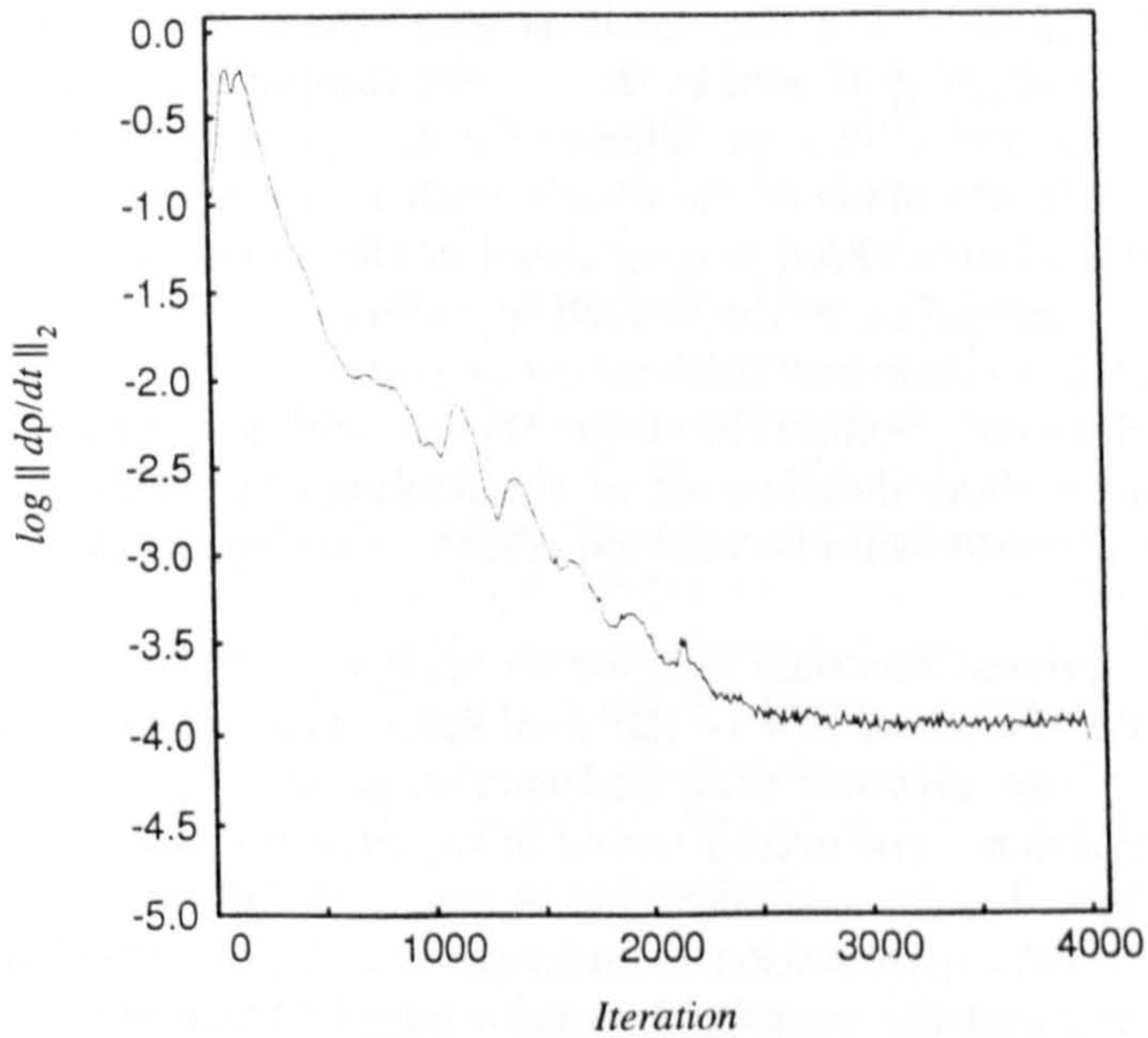


FIGURE 3.18. ONERA blunt-nose body supersonic flow. Convergence histories.

The sectional profile has both camber and twist. The specified flow conditions are $M_\infty = 1.2$ and $\alpha = 2.27°$. Datum experimental data derived from wind-tunnel test were available in terms of sectional C_p profiles. The grid was produced from a multi-block grid generator and the resulting grid blocks were subsequently amalgamated into the single-grid used in the present computation. The mesh is a C-H topology of size $144 \times 32 \times 64$. Views of it are given in Figure 3.19. A plan-view of the grid in the plane of the wing reveals fairly severe distortion of cells in the nearfield about a chord length ahead of the wing, and in the cells abutting the wing tip extending to the farfield boundary.

Under the present flow conditions there would be a detached bow shock ahead of the rounded leading-edge. Behind the shock, there would be a subsonic region which would extend a short distance aft of the leading-edge. Moving further downstream, the flow would again become supersonic. Another shock is likely at the trailing edge.

Figures 3.20–3.22 present sectional C_p's for six span-wise stations, located at 19%, 30%, 49%, 61%, 74% and 86% of the semi-span. The numerical solution predicts the leading-edge pressure peak and suction peak in good agreement with the experimental data. The overall trends in the experimental data at each section are reproduced by the computation, however, the dip on the upper-surface C_p up to the 49% span station is under-predicted. Differences between the numerical solution and the experimental data may be attributable in part to viscous effects absent from the current solution. Grid quality in the nearfield may also account for some degradation in solution accuracy, though the grid over the wing surface is well defined.

The lift and drag coefficients and the number of supersonic points ($NSUP$) in the solution from one of the industrial codes [76, 97], are listed together with those from the present upwind solver in Table 3.1.

	Upwind	NS87
C_L	0.1080	0.1103
C_D	0.0183	0.0179
$NSUP$	293139	292827

Table 3.1: Comparison of force coefficients and number of supersonic points for supersonic M165 wing-body test case.

The lift and drag from both methods are in fairly close agreement, creating confidence in the solution from the present algorithm. In terms of $NSUP$, the solution from upwind solver has a marginally higher number of supersonic points than that from the other solver. Usually in mixed flow problems, a larger value of $NSUP$ is indicative of crisper flow features and by implication, a higher degree of accuracy. Since error bands are not provided for the experimental data and as flowfield data for the other numerical method was not available either, it is difficult to comment further.

Figure 3.23 shows the convergence histories for the upwind solver in terms of the log of the l_2 norm, the number of supersonic points and the force coefficients. For engineering purposes, an acceptable level of convergence is reached in 2000 iterations, when the force coefficients have settled sufficiently. The CPU usage was 12200 seconds while the memory requirement was 29 Mwords (in-core).

3.2.3 TRANSONIC FLOW OVER A WING-BODY

The M165 wing-body research configuration was also used in a transonic case. The computation used the same grid as the supersonic case but here the test conditions are $M_\infty = 0.9$ and $\alpha = 4.0°$. Numerical solutions were obtained from the solvers NS87, RANSMB, DELPHIS and EJ83. Sectional C_p's for the spanwise stations located at 30% and 74% semi-span are presented for each solver in Figure 3.24. All the numerical solutions suggest a shock location towards the trailing edge along the entire wing span. However, the experimental data indicates a shock on the upper-surface for only the last 20% of span. This discrepancy between experiment and the inviscid computation is believed to be due to viscous effects. Each of the solutions show qualitatively similar behaviour, with no solution being significantly closer to the experiment than another at the stations presented. The experimental lift coefficient and the computed force coefficients from most of the solvers are listed in Table 3.2 together with the total number of supersonic points in the computed solutions. Compared to the experimental lift coefficient, the lift from the upwind solution is too low while that from each of the other solvers is over-predicted. The poor upwind lift prediction is due to a minor coding error which was not present in any of the other computations presented.

	Expt	Upwind	NS87	RANSMB	DELPHIS
C_L	0.25	0.202	0.2841	0.2843	0.2885
C_D	n/avail	0.0109	0.0149	0.0161	0.0211
$NSUP$	-	15591	14743	16438	n/avail

Table 3.2: Comparison of force coefficients and number of supersonic points for transonic M165 wing-body test case.

The convergence histories are given in Figures 3.25 and 3.26. (Note that in comparing the l_2 norms of different solvers, the absolute drop in residual is a more meaningful measure of convergence that the minimum level achieved since the latter is dependant in part on the flowfield non-dimensionalization. For example, the l_2 norm of one solver commences at -3.08 while that of the present upwind solver starts at $+0.65$. Also, caution is urged when comparing computational efficiency between multigrid and non-multigrid solutions. At the time of the inviscid test cases, the present upwind solver used only local time stepping for acceleration whereas most of the other solvers employed local time stepping, enthalpy damping, residual smoothing and multigrid. The CPU and memory usage for most of the numerical solutions are compared in Table 3.3, along with the number of cycles/iterations needed to achieve convergence.

	Upwind	NS87	RANSMB	DELPHIS
CPU seconds	16900	4530	10590	58010
Memory (Mw)	29	25	16	2
No of cycles	2000	400	400	1500

Table 3.3: Comparison of CPU and in-core memory usage for M165 transonic wing-body test case.

Since RANSMB is a multi-block solver, it has a lower memory requirement than both the present upwind solver and NS87. Unlike the other solvers, DELPHIS is an out-of-core solver and hence its requirement for in-core memory is negligible, though its CPU usage is significantly higher than the other codes.

3.2.4 VORTEX FLOW OVER A DELTA WING

The configuration for this test case is the IEPG W1 wing, a sharp-edged, cropped delta wing with a leading-edge sweep of 65° and a taper ratio of 0.15. The wing section is a symmetric biconvex. The wind-tunnel data for this case was obtained at $M_\infty = 0.85,\ \alpha = 10°$ with a Reynolds number based on mean root chord of $Re_{\bar{c}} = 9 \times 10^6$. A description of the experimental study and analysis is given in reference [38]. (In addition, an earlier comparative study of various numerical methods using this test case may be found in [185]). Under the specified flow conditions, the sharp leading-edge produces flow separation along its length. The resulting shear layer rolls up to form a primary vortex that dominates the upper-surface flow. This is evident from the wing upper-surface pressure distribution which reveals a pronounced suction peak with large span-wise pressure gradient. Shortly outboard of the suction peak, the adverse pressure gradient causes further flow separation producing a secondary vortex. While Euler methods are capable of modelling flow separations from sharp edges they cannot model separation from smooth surfaces such as the upper-surface of the present geometry from which the secondary vortex emanates. This must be taken into account when comparing the present inviscid solutions with the experimental data. To predict secondary separation usually a viscous algorithm is used.

Views of the W1 wing mesh are shown in Figure 3.27. It has a C-H topology consisting of $144 \times 32 \times 48$ cells. This topology produces a chisel-shaped wing tip which deviates slightly from the true geometry and which will account for a slight difference between the computed solutions and experiment in the tip region.

Figure 3.28 is the isobar plot ($\Delta C_p = 0.05$) on the wing upper-surface produced by the present upwind code. The well-defined footprint of the primary vortex is evident. Figure 3.29 shows isobar contours ($\Delta C_p = 0.05$) taken normal to the plane of the wing, along the mesh line which is coincident with the trailing edge. Grid lines are superimposed and the viewing plane is illustrated by the corner inset. Since the viewing plane follows a grid line, it bends away backwards after the trailing edge. The eye of the primary vortex is superimposed on the mesh. Aside from the core of the vortex, two smaller vortices are just discernible outboard of the wing tip. To magnify the view of these tip vortices, a blow-up view with greater isobar density ($\Delta C_p = 0.05$), but without the grid present is given in Figure 3.30. The qualitative results are concluded with Figure 3.31 which presents a velocity vector plot taken from the same viewing plane as the preceding figures.

Sectional C_p profiles at three chord-wise stations denoted $s1 - s3$ and corresponding to 30 %, 60% and 80% of the root-chord are presented in Figure 3.32. The station at which each C_p profile is taken is illustrated in the figure by inset. Experimental data is presented together with data from the computed solutions of the present upwind code ER3D as well as three other BAe solvers. From analysis of these results, the following observations are made:

1. All the numerical solutions predict a concentrated leading-edge primary vortex indicated by the pronounced suction peak. As the vortex core broadens with downstream distance so the suction peak in the numerical solutions increases in height;

2. The magnitude of the suction peak varies between solutions. At station $s1$ all codes with the exception of DELPHIS, underpredict the suction peak; at the remaining stations, all of them overpredict it. It may appear surprising that the upwind solution does not render the highest suction peak, as upwind methods generally exhibit lower levels of numerical dissipation than other methods. In fact, it does not necessarily follow that the upwind solver should render less dissipation than the other solvers in *all* parts of the flow. Considering the span-wise position of the suction peak, it is apparent that the upwind solution consistently predicts the primary vortex core farther inboard than any of the other solutions, in closer agreement with experiment. Using the experiment as a datum, the errors in the magnitude of the computed suction peak and, the errors in the computed span-wise position of the suction peak are given for each numerical solution in Tables 3.4 and 3.5 respectively. The smallest errors are underlined;

3. It is to be expected from inviscid solutions that the location of the primary vortex will be farther outboard than evident in real flow. This is due to the absence of the secondary vortex from the inviscid solutions. In real flow, the secondary vortex is located just outboard of the primary vortex core. Its presence has the effect of displacing the primary vortex laterally inboard. It is conjectured that the upwind solution produces a more accurate lateral position of the primary vortex as a result of another influence, namely the capture of the tip vortices. From the sectional C_p plots, it is evident that the tip vortices in the upwind solution increase suction just inboard of the wing tip, being in closer agreement with the experimental data than the other solutions. This has the knock-on effect of displacing the primary vortex core slightly inboard. In contrast, none of the other solutions appear to capture the tip vortices. This is possibly due to excessive numerical dissipation of the non-upwind solutions in the tip region. Naturally in the real flow, the tip vortices are farther away from the primary vortex than the secondary vortex and they would exert less influence on it than the secondary vortex;

4. All the computed pressure distributions on the wing lower-surface are in close agreement. Predictions differ most from experiment in moving upsteam towards the wing apex; and,

5. A better appreciation of this flow would necessitate the use of more accurate tip geometry modelling, the use of viscous solvers and, further detailed flowfield analysis.

The lift and drag coefficients for each of the numerical solutions are given in Table 3.6. The convergence histories for the present scheme are given in Figure 3.33. The l_2 norm traces a shallow drop before levelling out into high-frequency oscillations of bounded magnitude. The force coefficients suggest a converged solution in 4000 iterations by which time there little change in lift and drag. The

CPU usage was approximately 26500 seconds and the memory usage was 21.7 Mwords (in-core).

Code	% Error relative to experimental data		
	Station s1	*Station s2*	*Station s3*
Upwind	−3.2	+19.2	+14.6
NS87	−6.4	+14.3	+22.5
RANSMB	−4.3	+11.7	+17.9
DELPHIS	+1.1	+22.9	+30.8

Table 3.4: Comparison of error in computed suction peak magnitude for IEPG W1 delta wing.

Code	% Error relative to experimental data		
	Station s1	*Station s2*	*Station s3*
Upwind	+6.3	+7.2	+5.0
NS87	+10.9	+11.9	+8.4
RANSMB	+15.0	+8.6	+8.4
DELPHIS	+15.0	+10.6	+8.4

Table 3.5: Comparison of error in computed suction peak position for IEPG W1 delta wing.

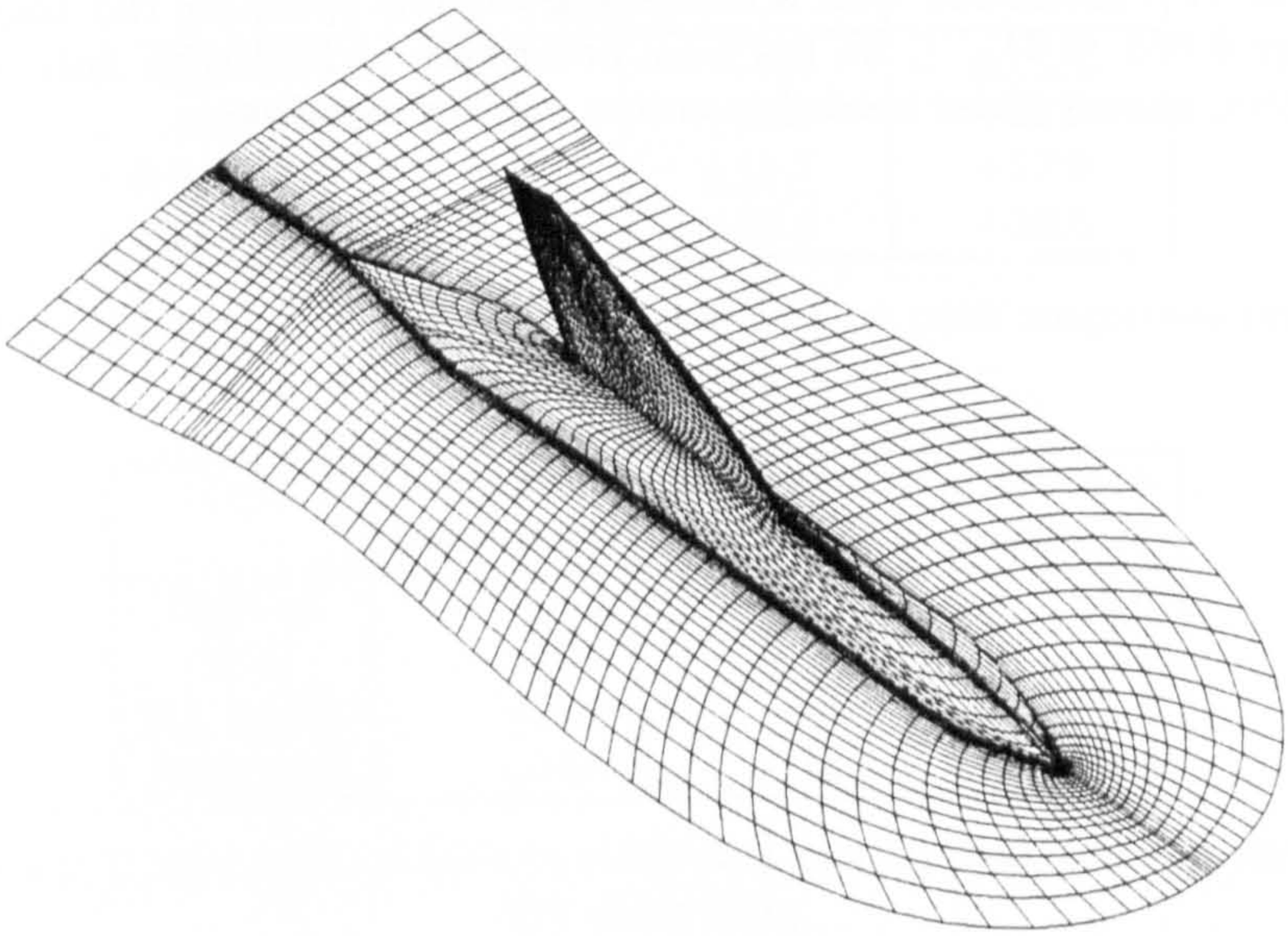
	Upwind	NS87	RANSMB	DELPHIS
C_L	0.5010	0.5076	0.5212	0.5207
C_D	0.0814	0.0831	0.0856	0.0847

Table 3.6: Comparison of computed force coefficients for IEPG W1 delta wing.

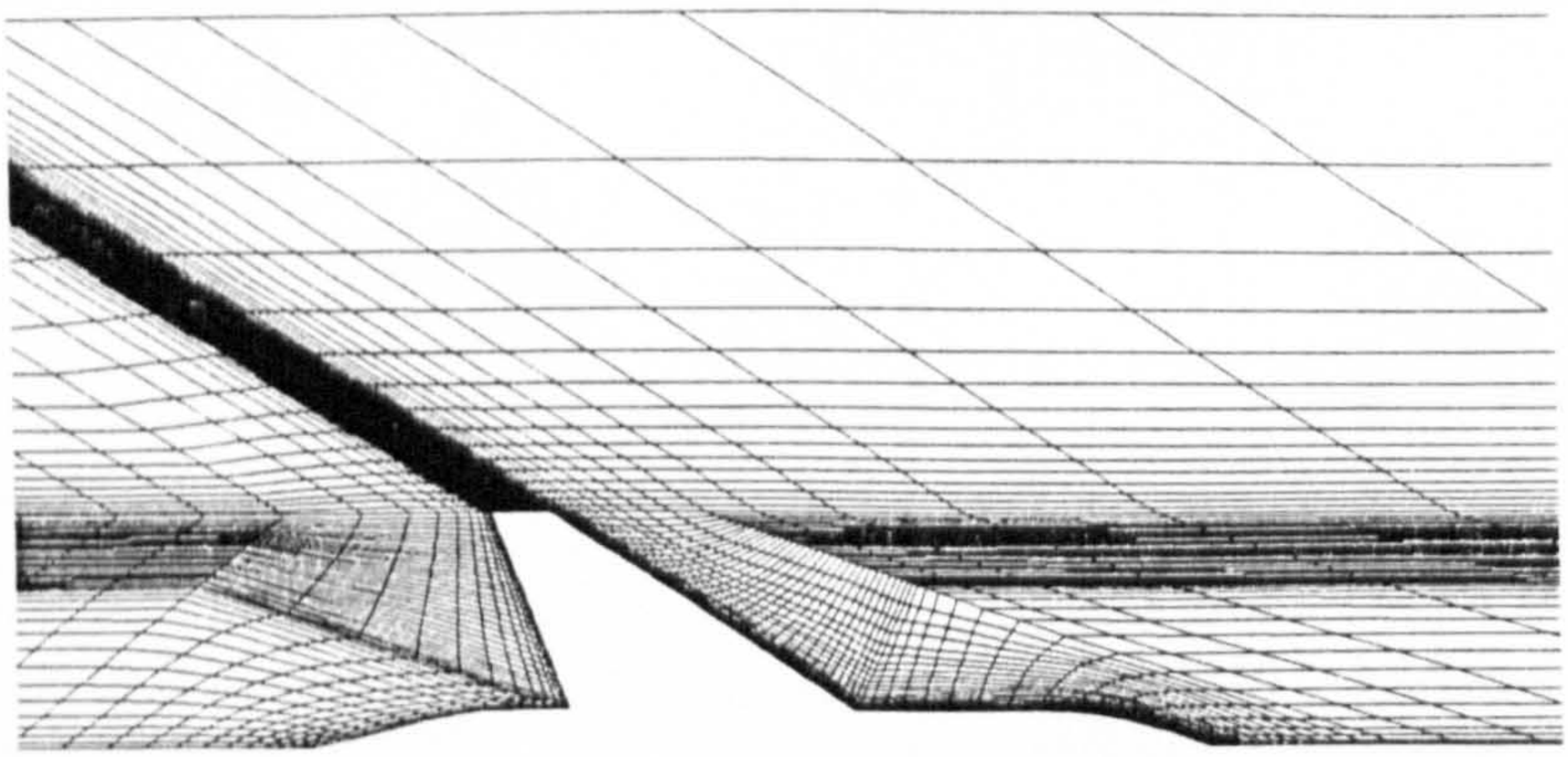
3.3 Closing Discussion

The results from a number of numerical computations for inviscid flows have been presented and discussed in this chapter. The emphasis has been on obtaining accurate inviscid flow solutions over a wide range of test conditions of practical interest. Where possible, results have been compared with those obtained from other flow solvers and experimental data. These comparisons afford a perspective on the performance of the present upwind solver relative to other CFD solvers, most of which represent industrial state-of-the-art design tools. Particularly noteworthy computations which have been presented include the ONERA blunt-nose body case, flow over a sphere at a Mach number of 25 and, vortex flow over a delta wing. The ONERA test case is one which has been attempted by several of the leading European research institutes and aerospace companies in a controlled exercise in which all participants used the same mesh. Significantly, the present upwind solver produces results which are equal if not better than the best result presented in that exercise. For the $M_\infty = 25$ test case in which the pressure ratio across the shock is 800:1, the present upwind solver captured the bow shock wave crisply with a single transition point. Theoretically, a discrete shock structure with one transition point is the best that

a conservative shock capturing method can achieve. In the vortex flow case, the present upwind solver captured wing-tip vortices that several other industrial design codes could not resolve using the same grid. An undercurrent running through all the computations presented is the robustness of the present upwind algorithm. It is significant that a wide range of flows spanning the Mach number range $0.675 \leq M_\infty \leq 25$ has been accurately tackled with only two free parameters, namely those needed to ensure entropy satisfaction.



(a) Near-field mesh over wing-body and in plane of symmetry



(b) Plan cross-section

FIGURE 3.19. M165 wing-body C-H grid, $(144 \times 32 \times 64)$.

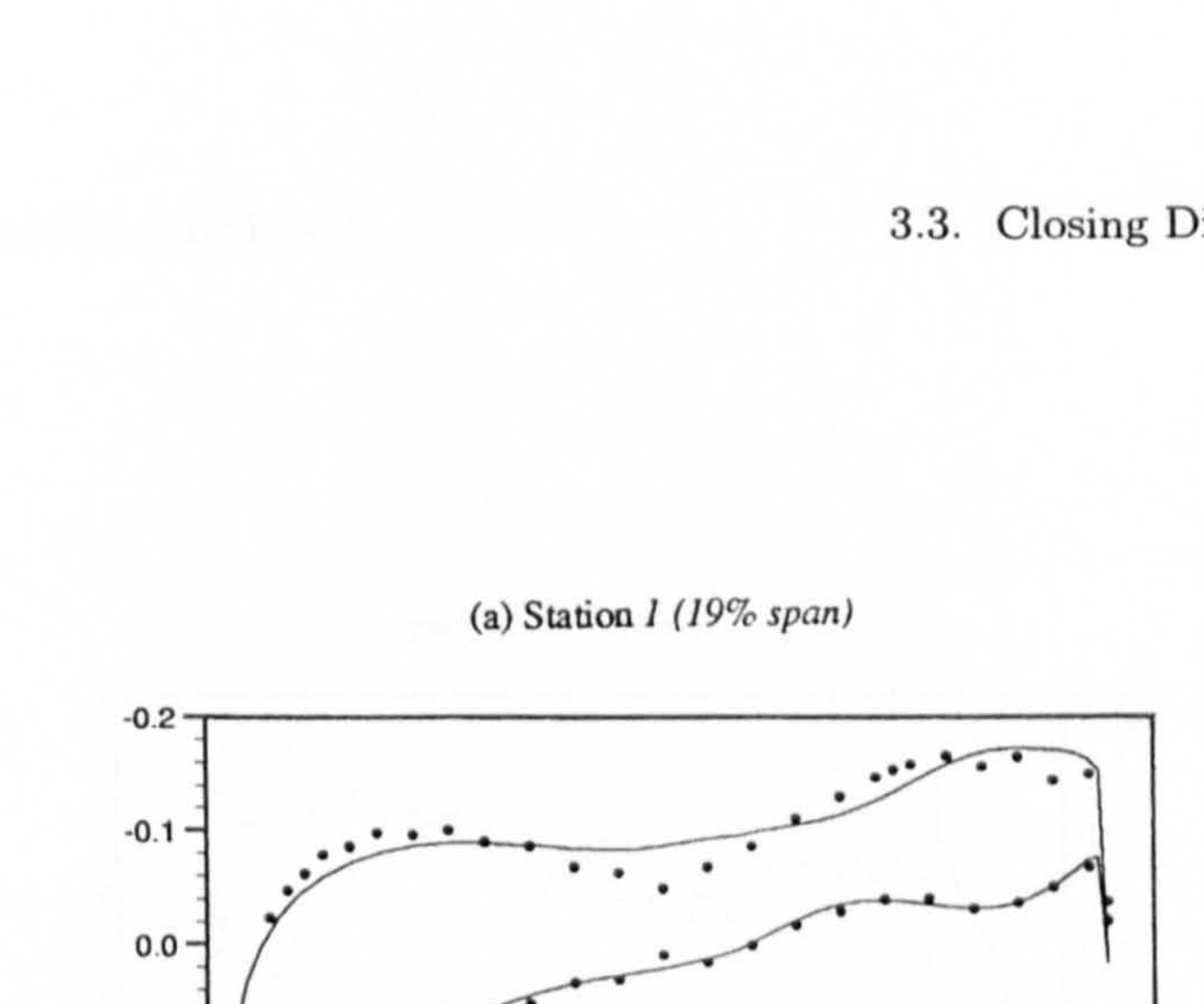
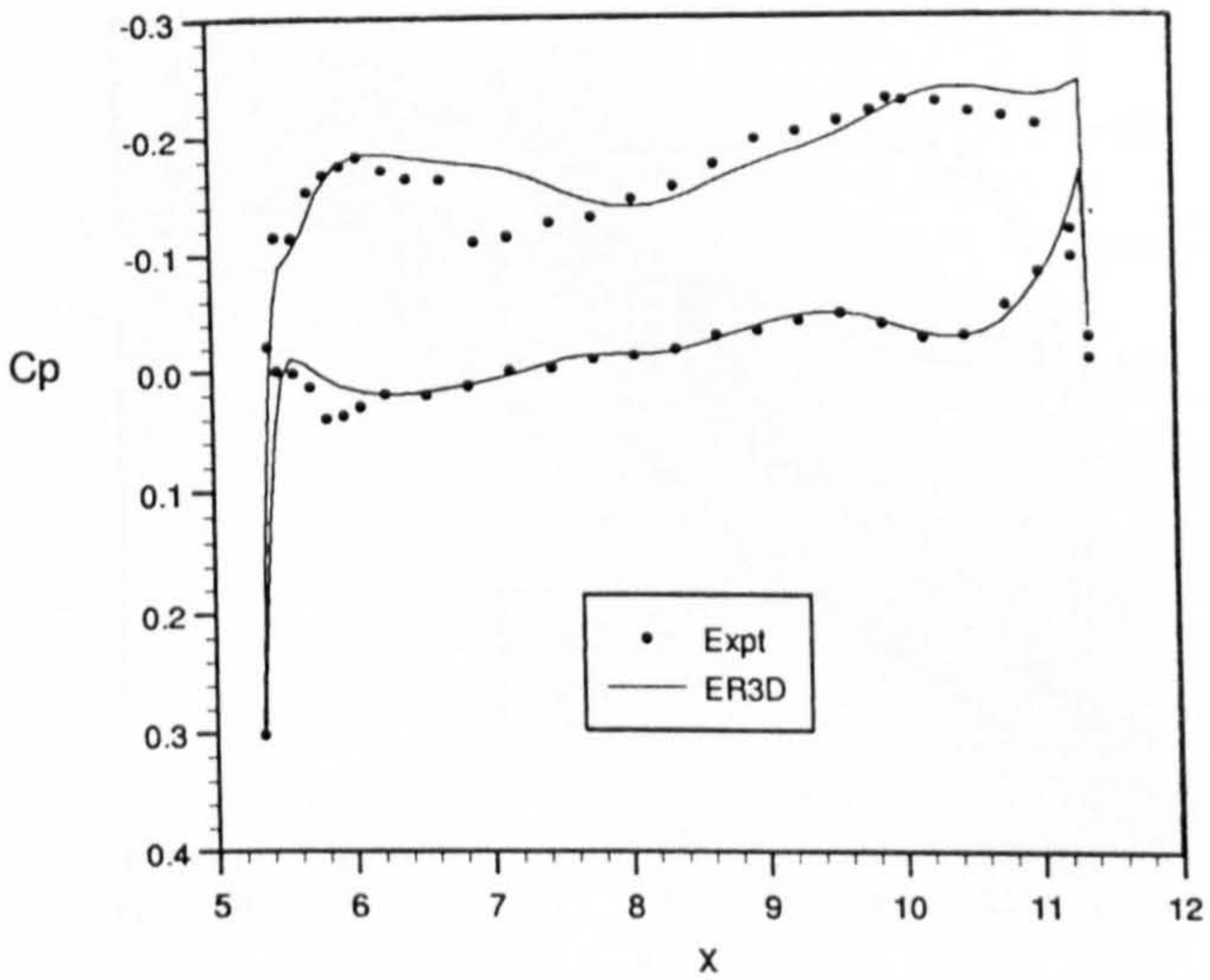


FIGURE 3.20. Supersonic M165 wing-body sectional C_p profiles at 19% and 30% span.

(a) Station *3 (49% span)*



(b) Station *4 (61% span)*

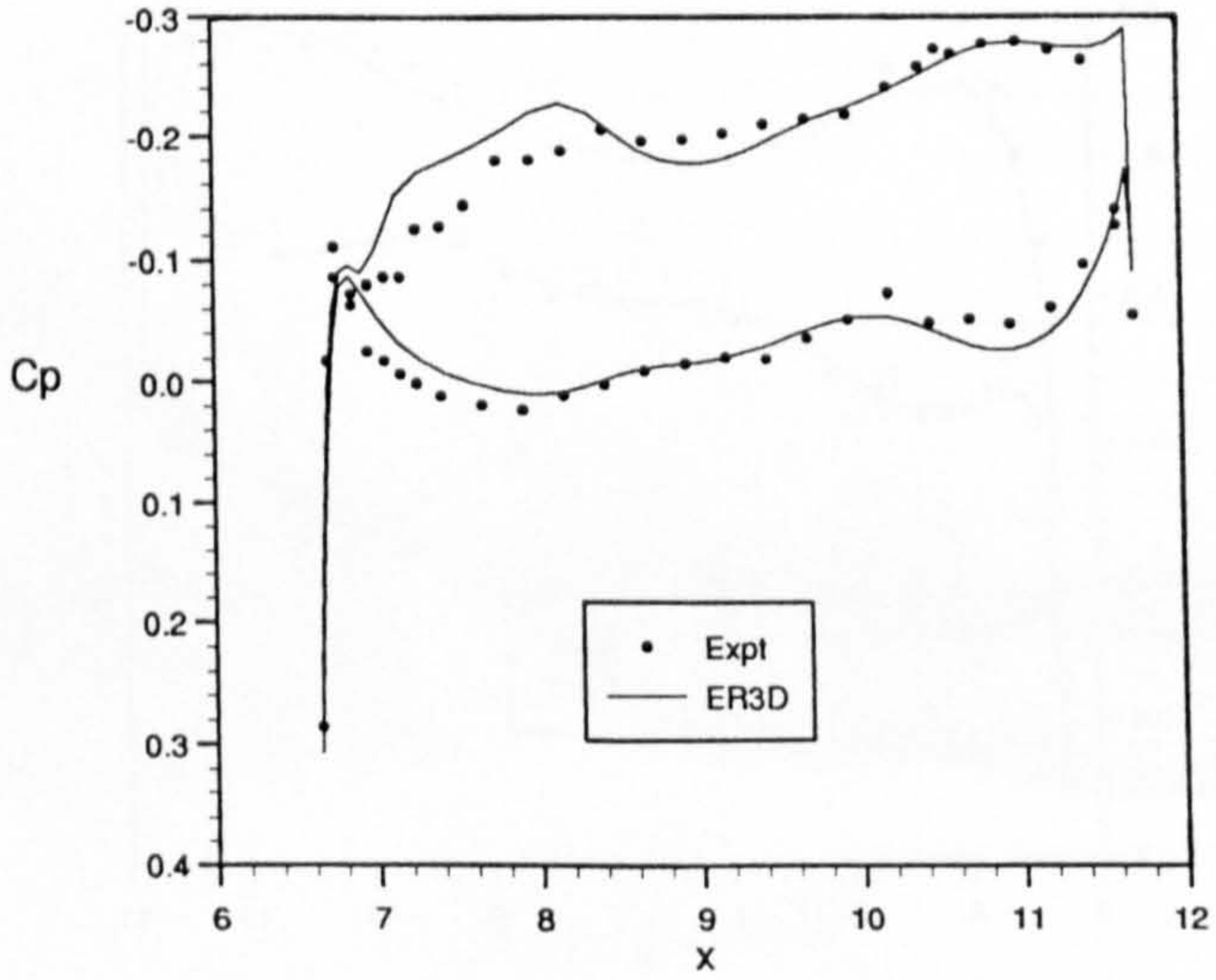
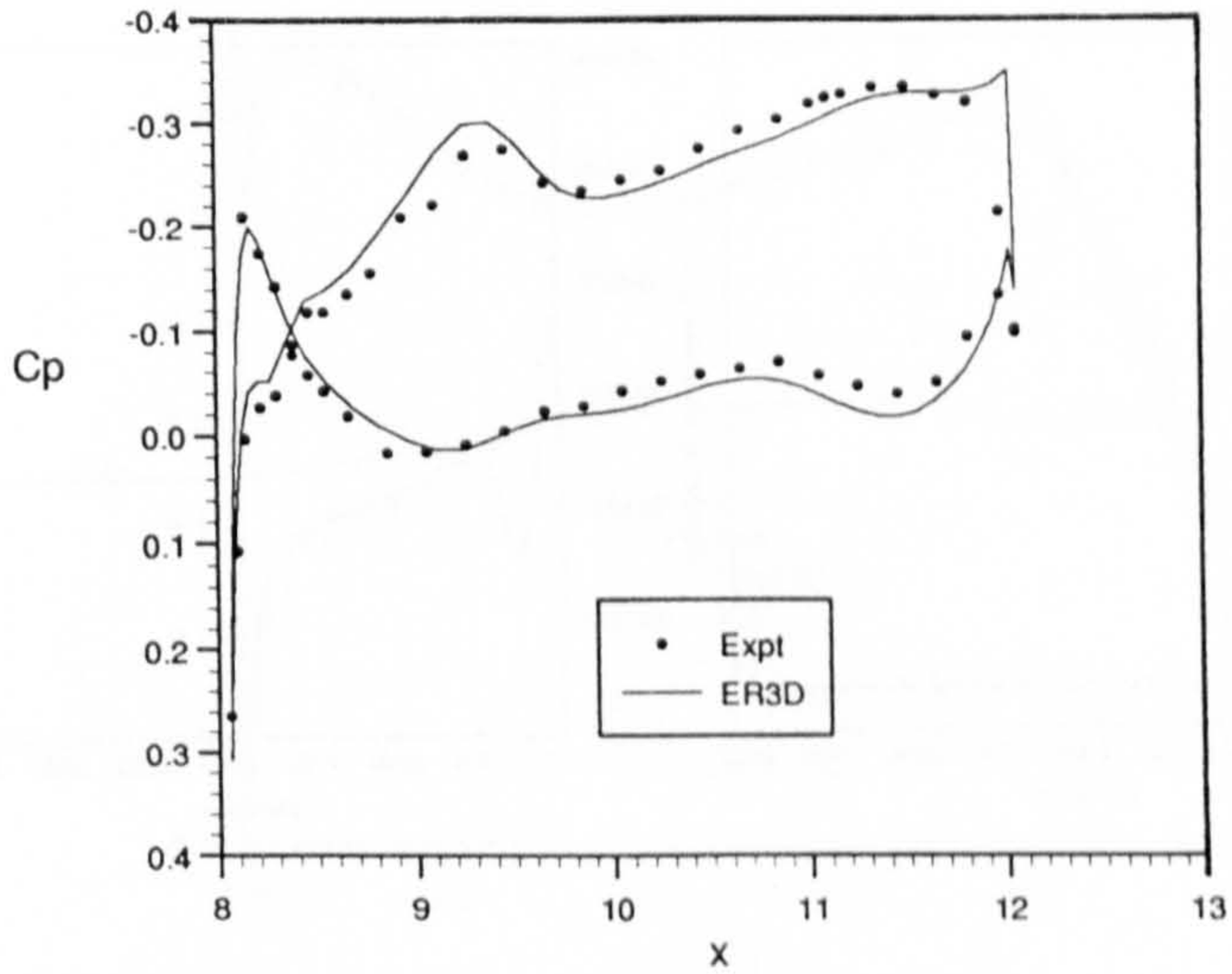


FIGURE 3.21. Supersonic M165 wing-body sectional C_p profiles at 49% and 61% span.

(a) Station *5 (74% span)*



(b) Station *6 (86% span)*

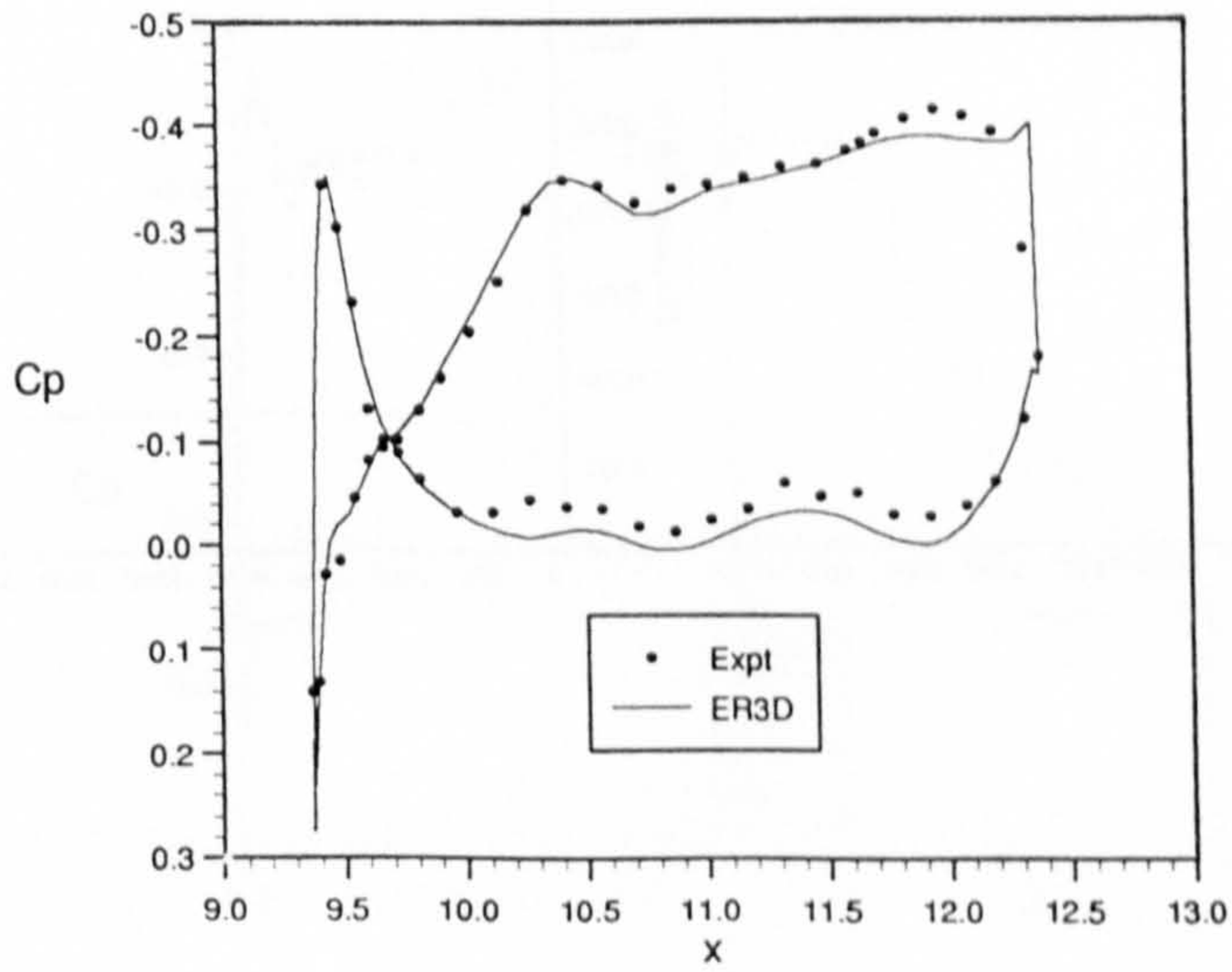


FIGURE 3.22. Supersonic M165 wing-body sectional C_p profiles at 74% and 86% span.

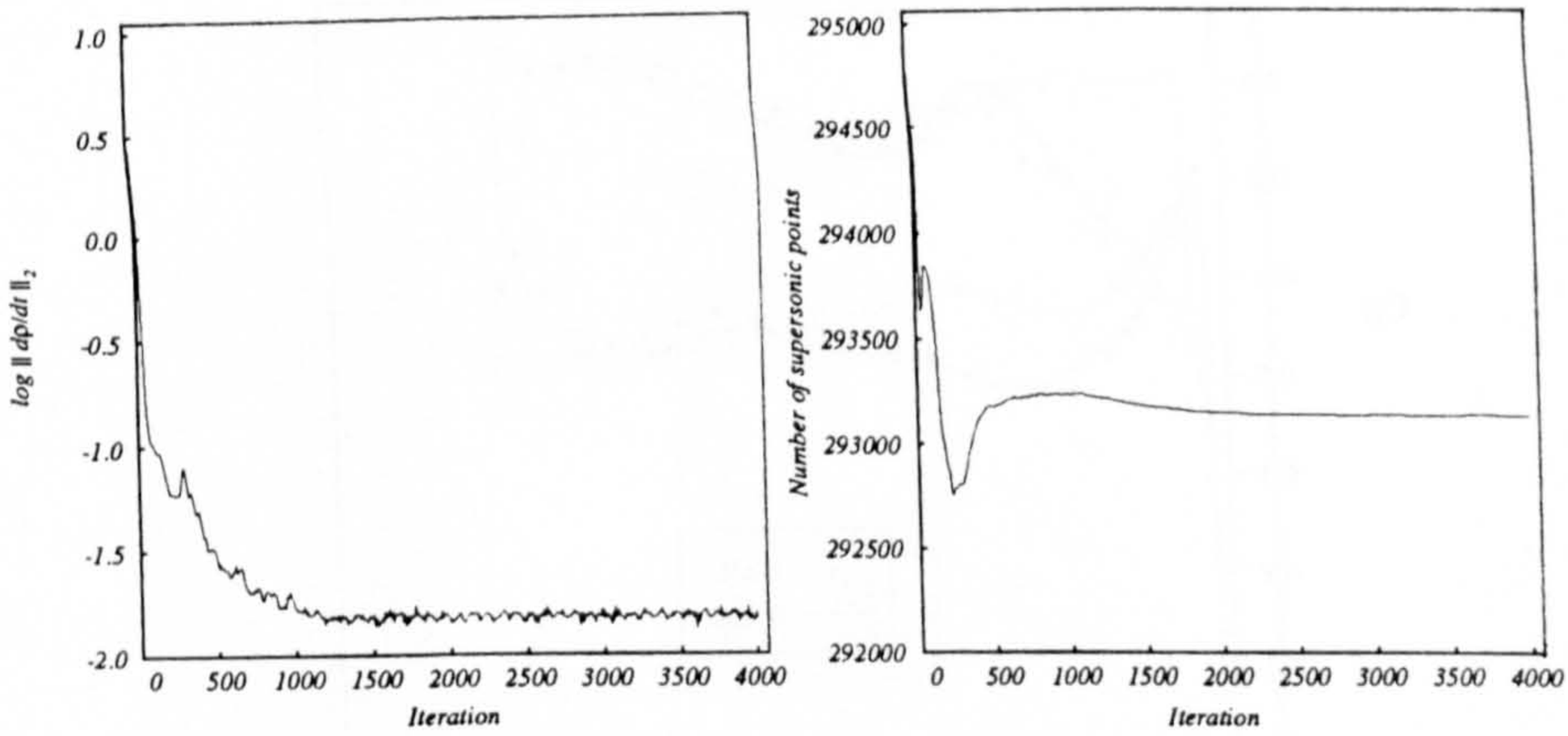
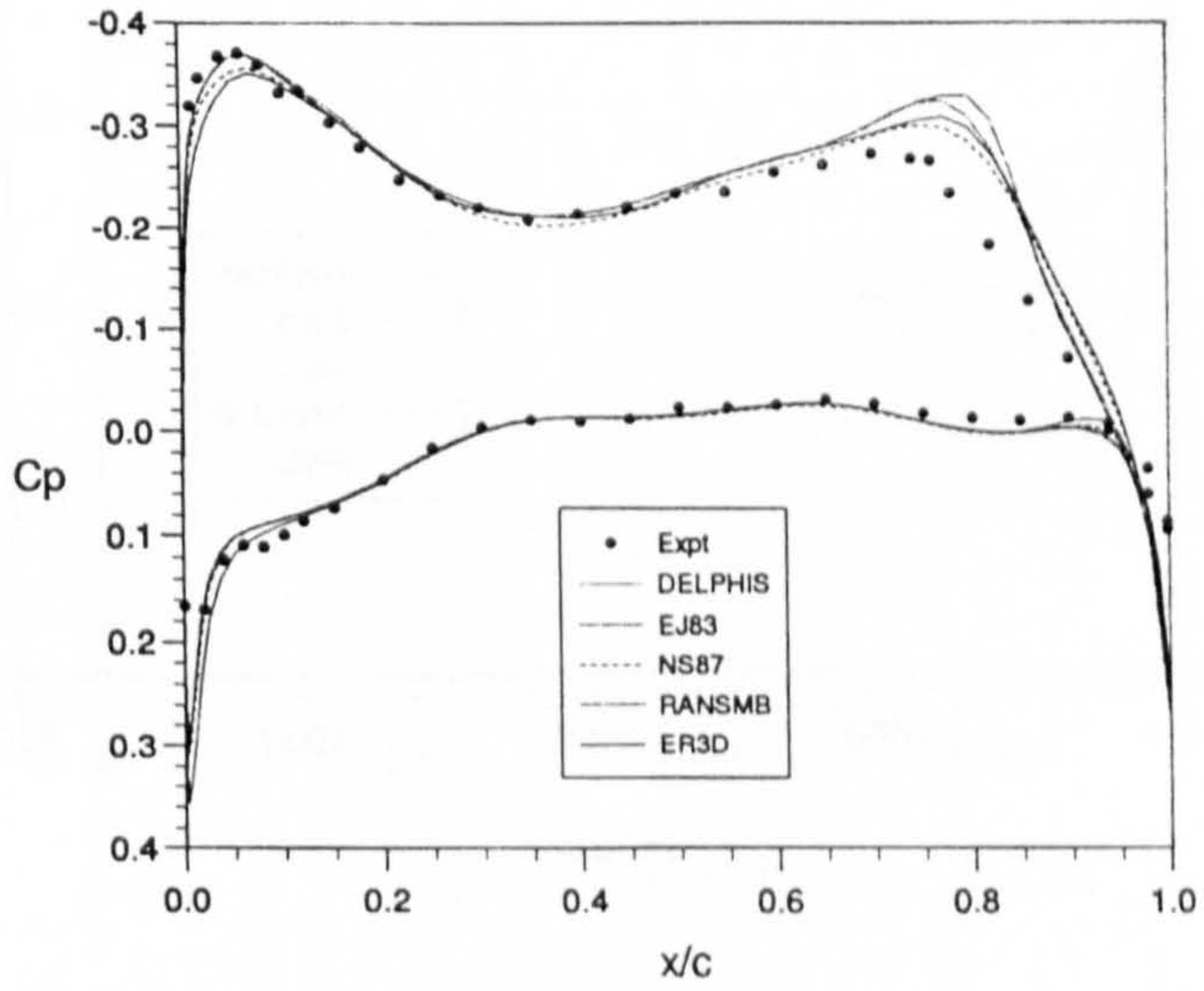


FIGURE 3.23. Supersonic M165 wing-body convergence histories.

(a) Station *2 (30% span)*



(b) Station *5 (74% span)*

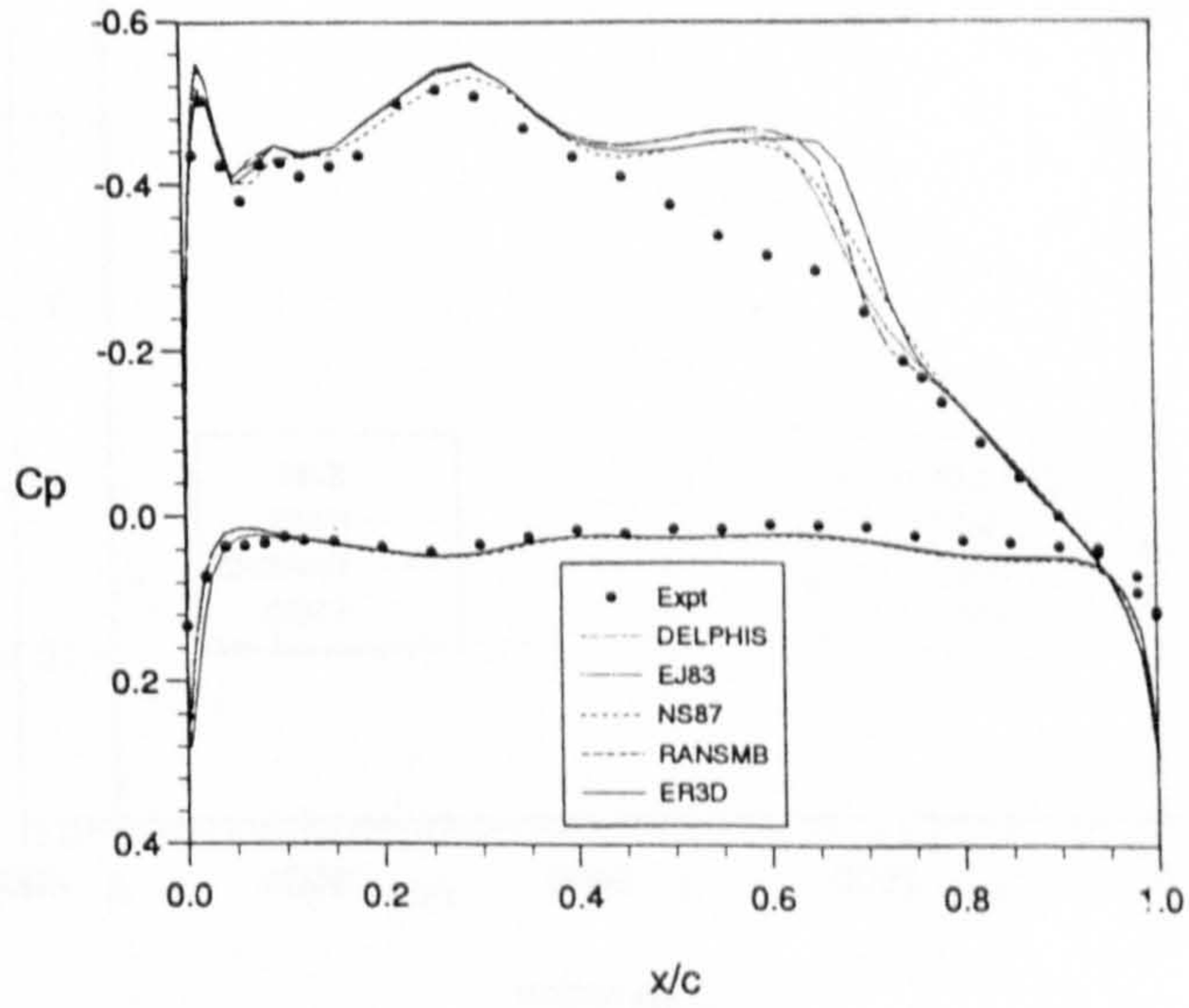


FIGURE 3.24. Transonic M165 wing-body sectional C_p profiles at 30% and 74% span for several flow solvers.

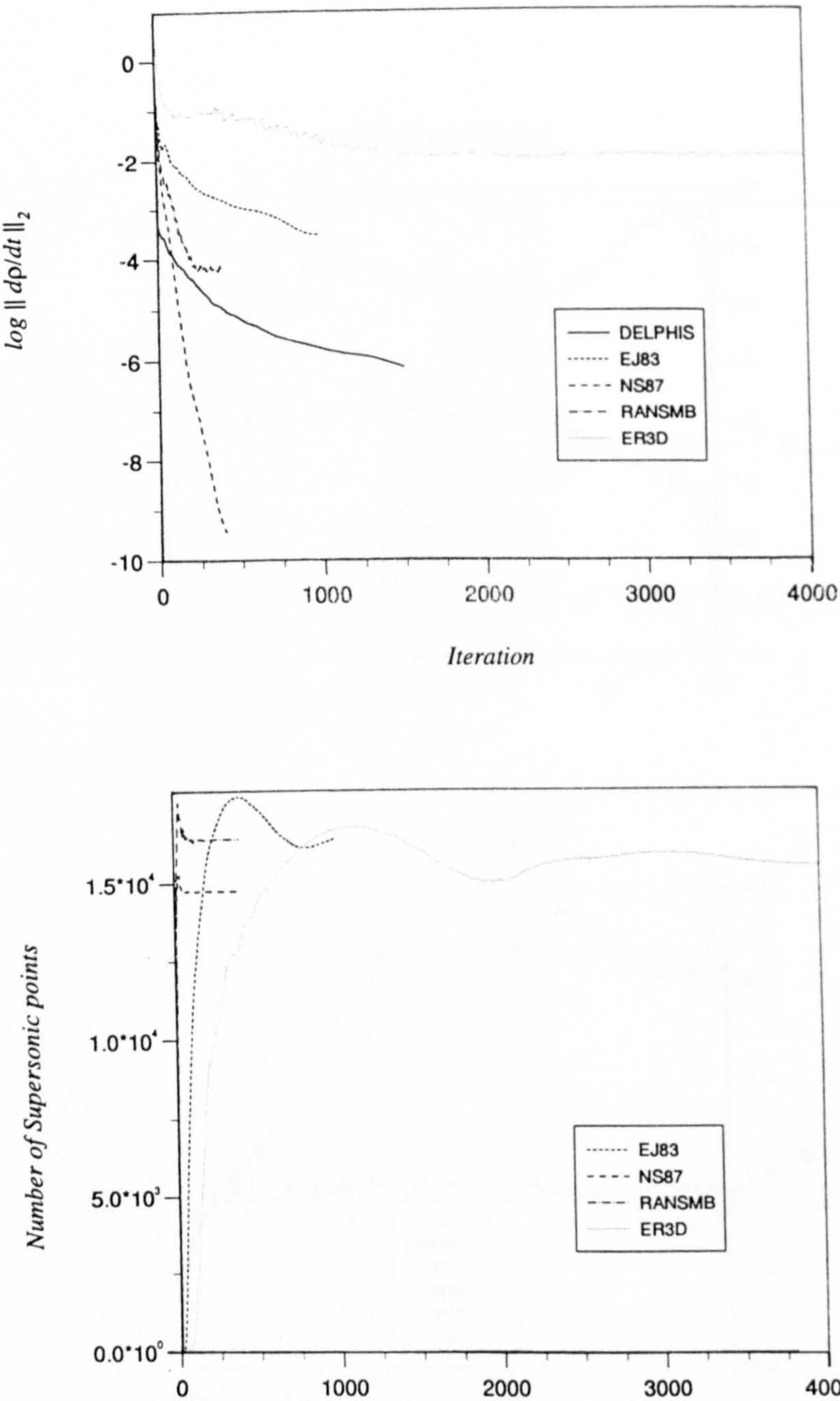
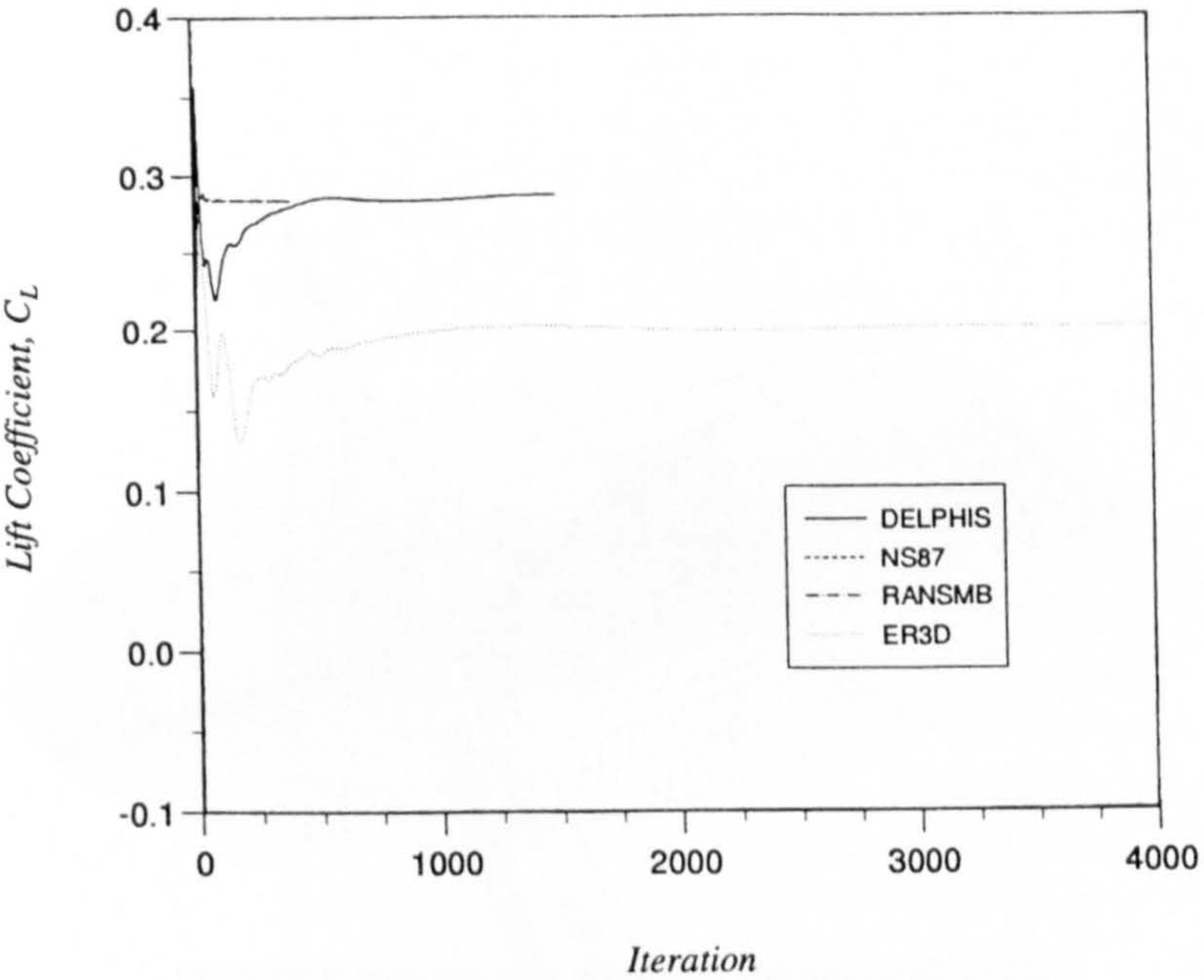


FIGURE 3.25. Comparison of transonic M165 wing-body convergence histories from several flow solvers.



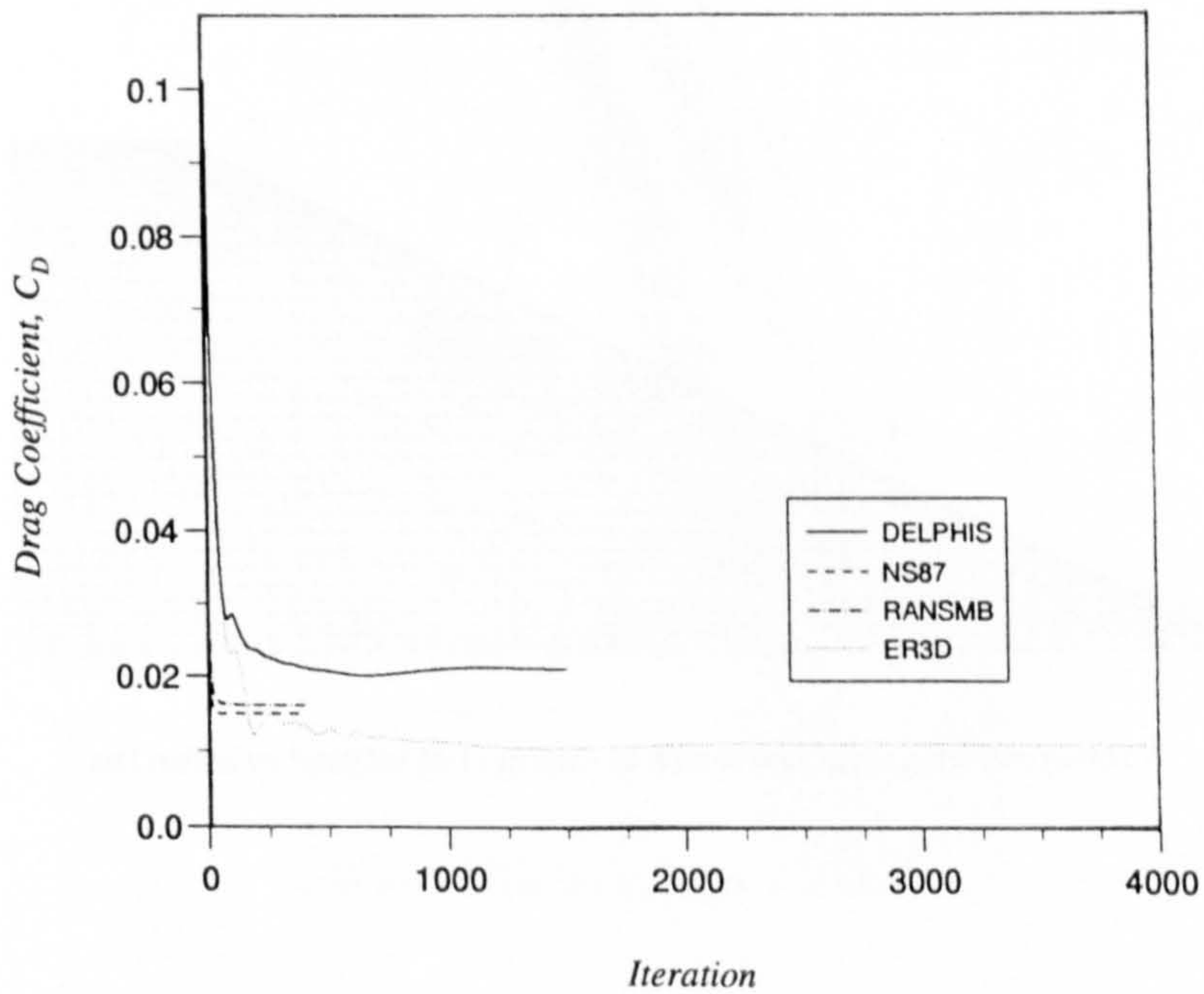
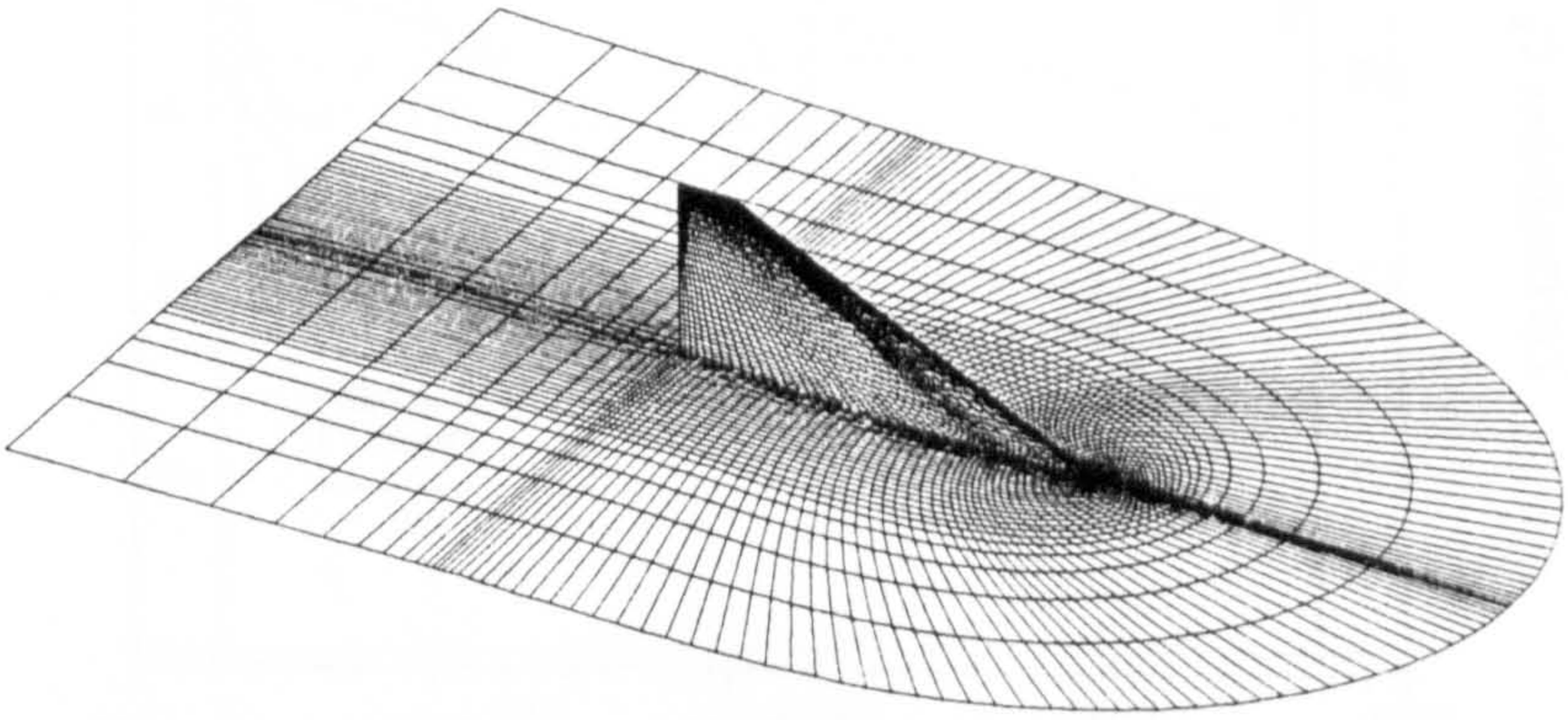
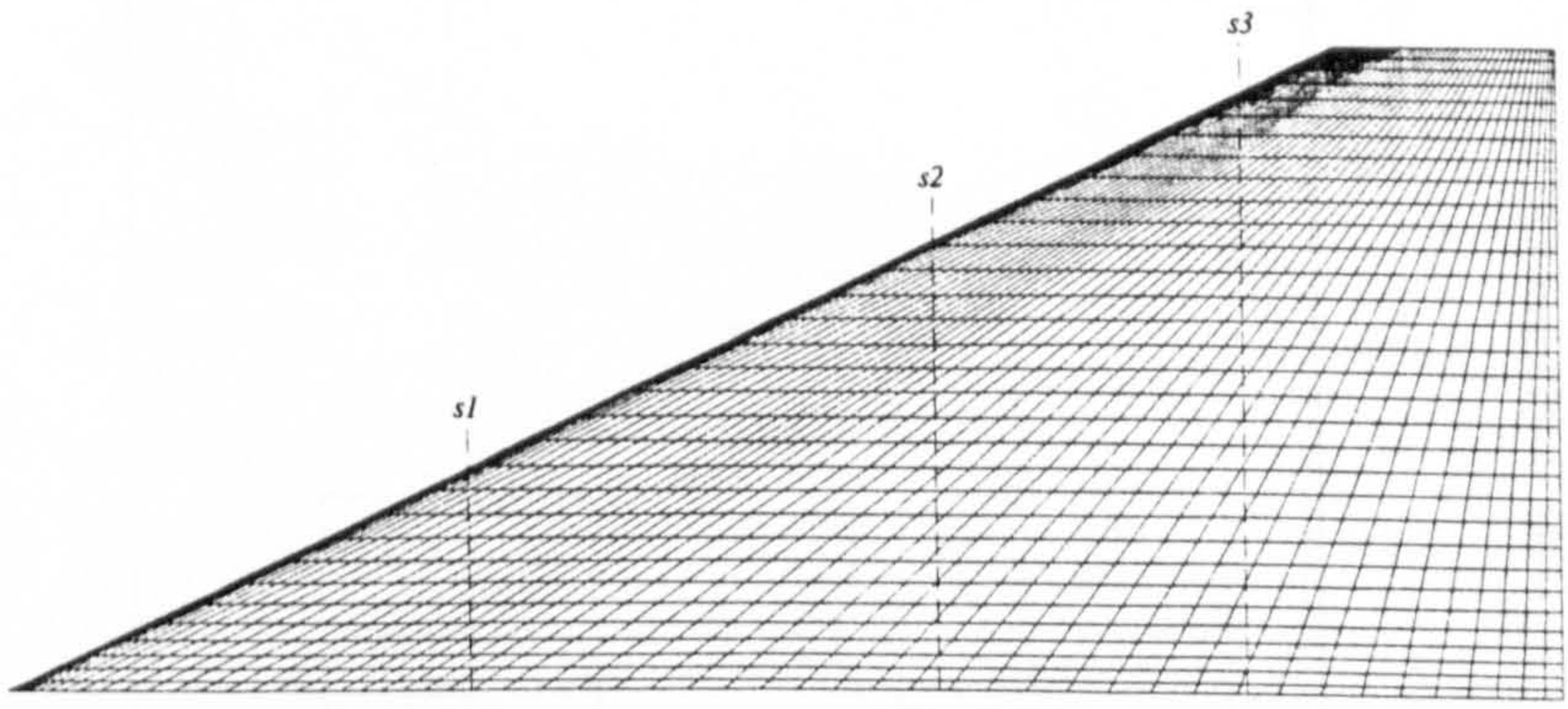


FIGURE 3.26. Comparison of transonic M165 wing-body convergence of force coefficients from several flow solvers.



(a) Near-field mesh over wing and in wing root plane of symmetry



(b) Mesh over wing upper surface with the stations s1-s5 indicated by broken line

FIGURE 3.27. IEPG W1 delta wing C-H grid, (144 × 32 × 48).

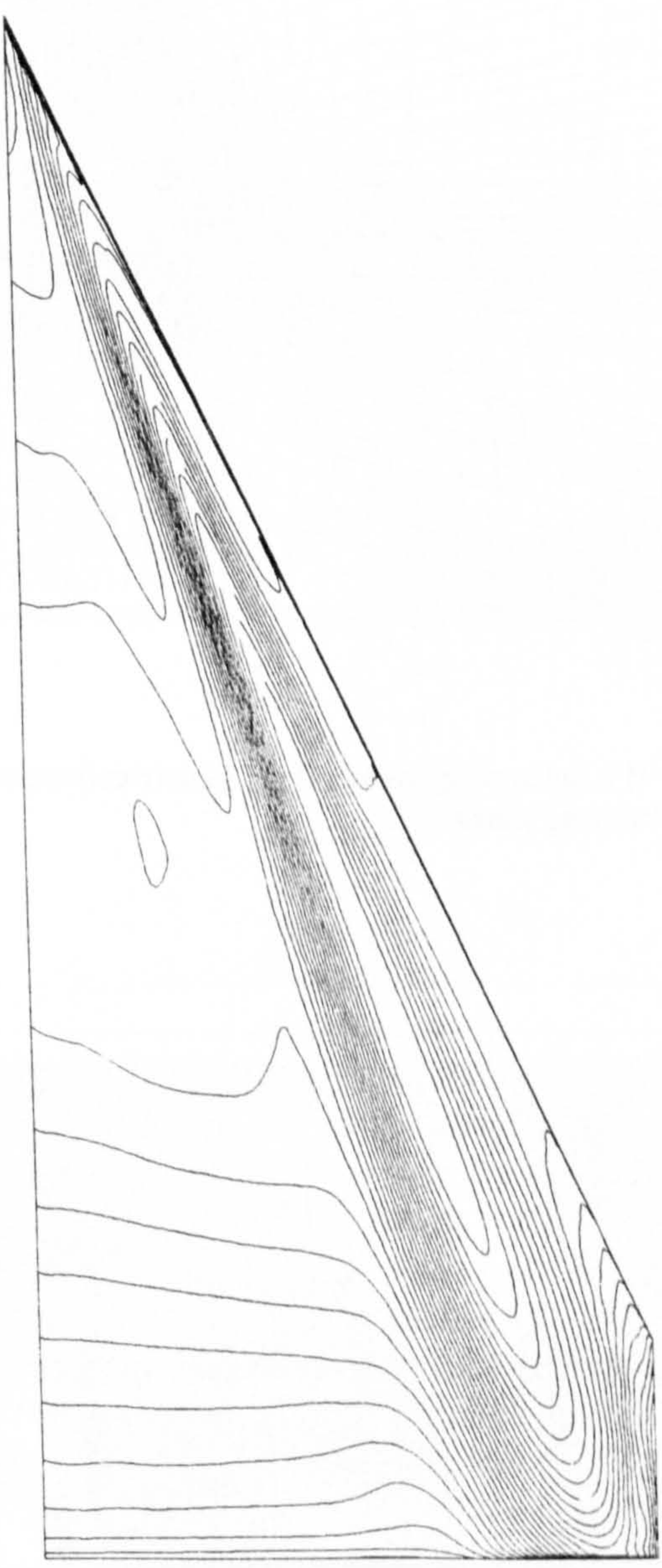


FIGURE 3.28. IEPG W1 delta wing vortex flow. Upper-surface isobar contours.

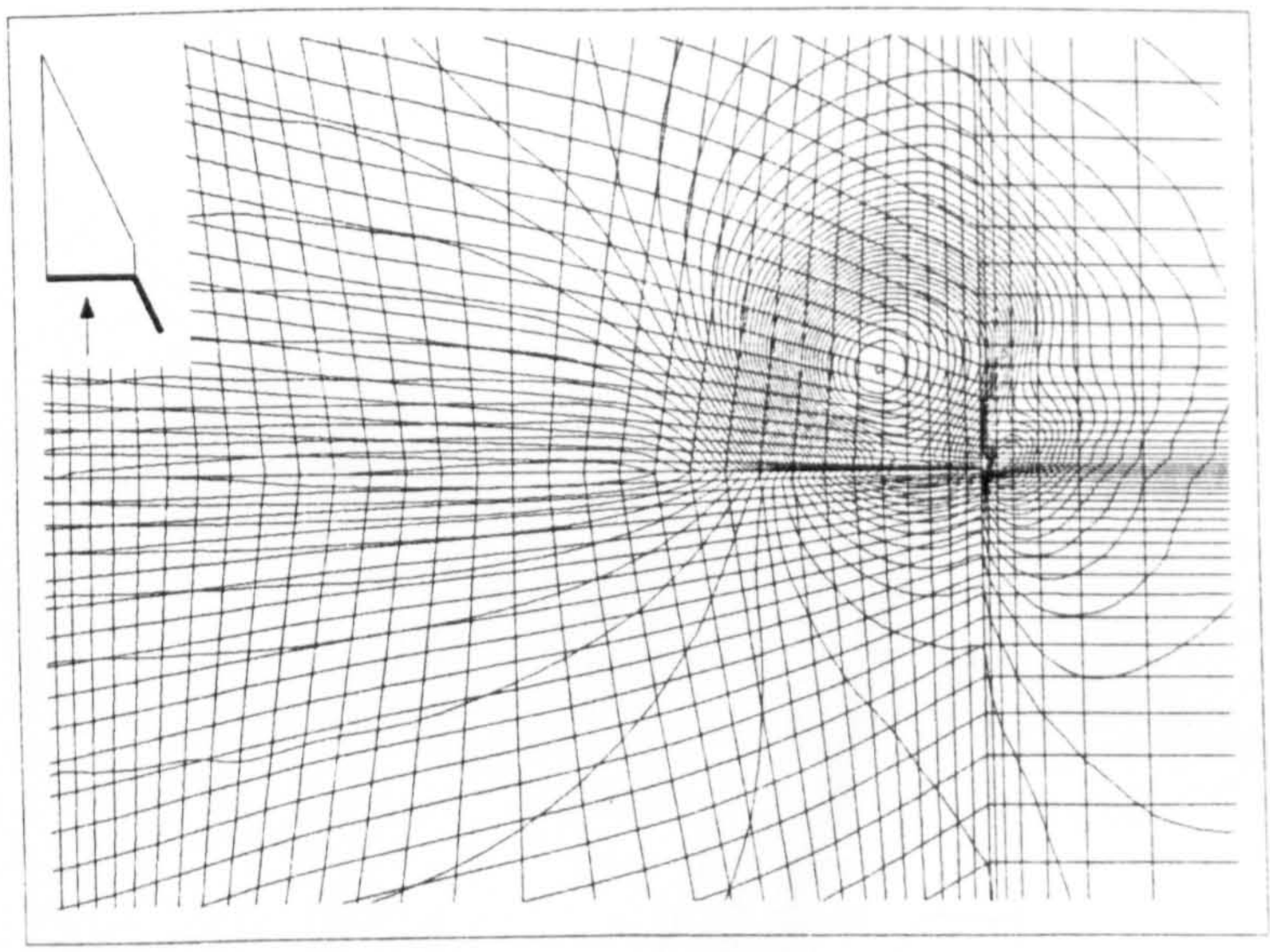


FIGURE 3.29. IEPG W1 delta wing vortex flow. Isobar contours normal to the trailing edge. Inset indicates viewing plane.

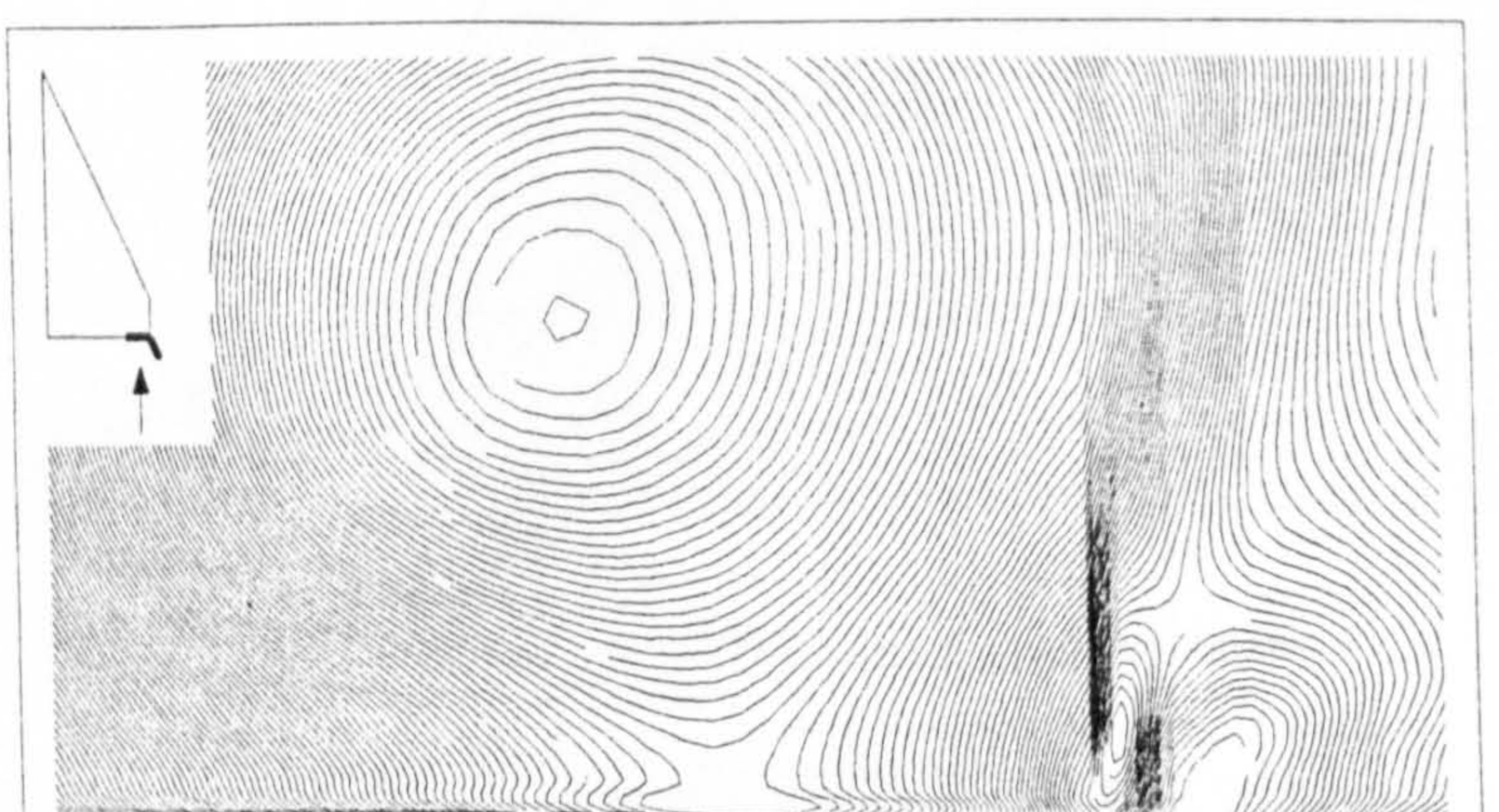


FIGURE 3.30. IEPG W1 delta wing vortex flow. Blow-up view of isobar contours normal to the trailing edge. Inset indicates viewing plane.

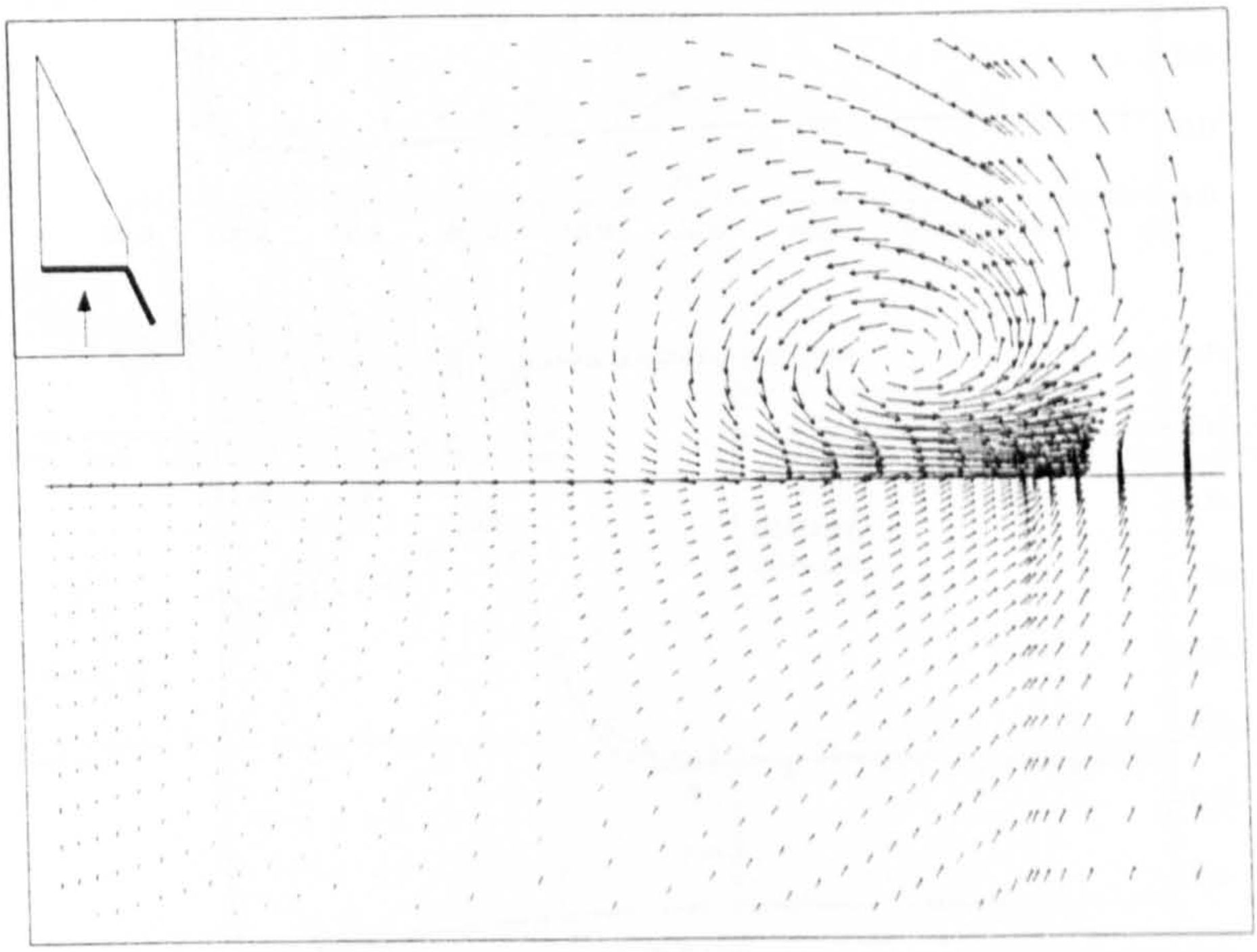


FIGURE 3.31. IEPG W1 delta wing vortex flow. Velocity vectors normal to the trailing edge. Inset indicates viewing plane.

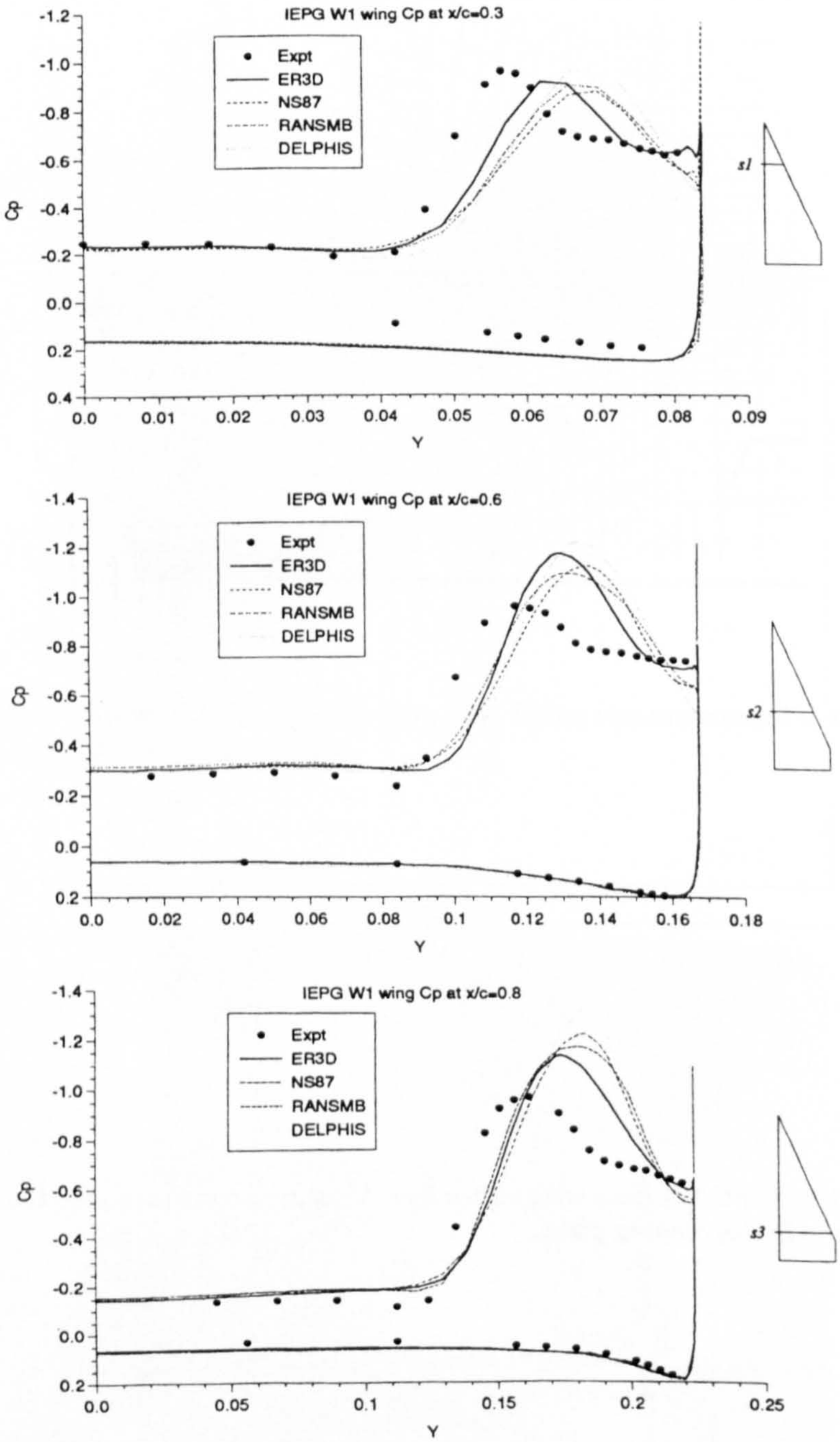


FIGURE 3.32. IEPG W1 delta wing vortex flow. Sectional C_p profiles. Insets indicate each chord-wise station.

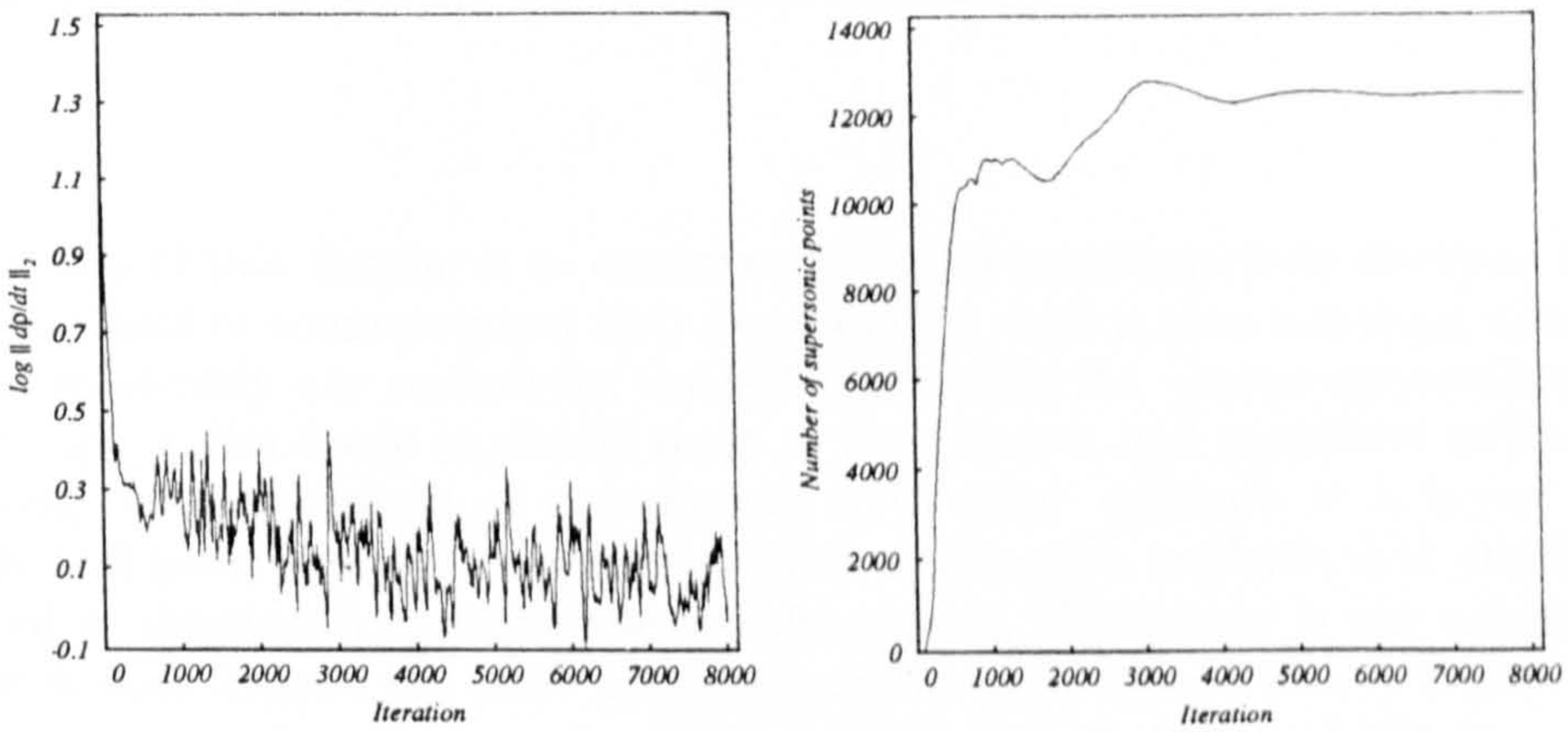


FIGURE 3.33. IEPG W1 delta wing vortex flow. Convergence histories.

4

Review of Convergence Acceleration Methods

The aim of this chapter is to examine the major convergence acceleration techniques used in computational fluid dynamics for *explicit* time marching methods and to identify any underlying concepts that unify the various approaches. An attempt is also made to clarify some of the physical and numerical difficulties facing the development of convergence acceleration methods. It is hoped that this will place the particular convergence acceleration methods that are developed in chapters 5 and 6 in the broader setting. The review is not exhaustive but it does include the more important methods in current use. It begins with some general observations concerning the background of these methods.

4.1 Background

Explicit time marching methods are commonly used in CFD to solve hyperbolic problems containing weak solutions. When such methods are applied to the computation of solutions that in time become asymptotically steady, viz. steady-state solutions, various convergence acceleration techniques may be applied to the marching process to reduce the computational time to arrive at the solution and hence to reduce its expense. When this is done, the transient solution is modified such that it no longer corresponds to any well defined physical experiment. As time accuracy is compromised, any method that is accelerated in this manner is formally referred to as a Pseudo-unsteady or False transient method, see Viviand [178]. There is a close connection between relaxation methods used for solving time-independent problems and pseudo-unsteady methods in which the time variable plays the rôle of an iteration parameter. Nevertheless, from a practical point of view, each approach constitutes a different way of constructing an iterative method, especially in the case of nonlinear equations of mixed type. In what follows, it is the pseudo-unsteady methods applied to the compressible fluid equations that are of interest.

Apart from their simplicity, explicit schemes offer several advantages over implicit schemes. For example, they require less storage, they are readily implemented on vector and parallel architectures and they are naturally suited to local grid refinement techniques and to use with unstructured grid methods. However, their efficiency is limited by the permissible timestep size which must satisfy the CFL condition to ensure stability. Consequently, most of the convergence acceleration methods and techniques advanced for explicit schemes have as their prime objective enlargement of the stability region. Those that may be classified in this way include local time stepping, multi-stage methods, residual smoothing, multi-

grid methods and Krylov methods. Additional gains in algorithmic efficiency are often achieved through combining some of these techniques in a single algorithm. However, these particular methods cannot deal with the problem of stiffness in the governing equations, also referred to as eigenvalue stiffness.

Hyperbolic systems of equations become stiff if the speeds of propagating waves become vastly disparate, i.e. the *characteristic condition number*, σ (defined by the ratio of the largest eigenvalue in the system to that of the smallest), becomes large. Typically, this happens in low Mach number flow, stagnation regions and transonic flow. In particular, flow singularities occurring as $M \rightarrow 0$ and $M \rightarrow 1$ cause stiffness that impedes convergence, often severely. Upwind methods tend to suffer more from this problem than other more dissipative methods. With very stiff problems, poor convergence may be accompanied by a deterioration in the accuracy of the computed solution. Fortunately there are some preconditioning techniques that can accelerate the convergence of solutions to stiff problems, one of which is described here. An alternative approach is to write the governing equations in canonical form which separates the equations describing acoustic waves from those describing convection. This allows optimized numerical procedures to be applied to the separate subsystems. Unlike most other methods, with this approach performance is not impaired by the singularity $M \rightarrow 0$.

Before moving on to review existing convergence acceleration methods, mention is made of a new convergence acceleration technique termed *shock acceleration*, which has been devised as part of the present work. This method, which is presented in chapter 6, specifically addresses the convergence difficulties arising from stiffness caused by the $M \rightarrow 1$ singularity. Consequently, it may be classified along with the preconditioning techniques devised to cure the problem associated with eigenvalue stiffness.

4.2 Geometric and Algebraic Subspace Methods

Referring to steady transonic flows, van Leer [166] remarks that the inherent nonlinearity of computational fluid dynamic problems raises two separate but related questions, namely:

1. how to efficiently march from an initial guess to within the range of attraction of the steady solution; and,

2. how to quickly converge to the steady solution from a nearby state.

Using the terminology of dynamical system theory, the answer to the first problem assumes the property of *global convergence* while the answer to the second problem requires the property of *local convergence.* van Leer states that vector sequencing strategies answer the second problem, which is the easier of the two owing to the validity of the linearization and, that the only current methodology that addresses the first problem is full multigrid. Both multigrid and vector sequencing methods belong to the generic class known as *subspace methods.*

Often error modes associated with the transients of an unconverged solution can be better represented in a functional subspace of the original computational domain. Where this is possible, the errors may be expelled or eliminated more

efficiently than would be possible from the original domain. In this context, a subspace method may be defined as one which exploits this property to accelerate the convergence of an iterative scheme. As such, subspaces are very important in the study of many iterative methods. However, it should be noted that the term subspace is rather general and that the primary purpose of some subspace methods, such as spectral type methods which are based on Fourier or wavelet expansions, is to afford a more accurate representation of the continuum problem at the discrete level, rather than to reduce computational expense through convergence acceleration.

The most common subspaces employed for the purpose of convergence acceleration are geometric or algebraic in nature. The classical multigrid method is an example of the former. To solve a problem on a particular geometric discretization (a fine grid), it uses a series of different discretizations (coarser grids) on which to solve or simply approximate the original problem. This has two important advantages. The first is that solutions obtained on coarse discretizations are cheaper than those obtained on fine ones. The second advantage is that long wave length error modes are better represented on the coarser grids and therefore solutions on those grids remove long wavelength errors more effectively than could solutions on the fine mesh. The solutions or near solutions from each geometrical representation are transferred typically in predetermined sequences from one grid to another and eventually back to the original discretization via grid transfer operators. Hence, the work involved in eliminating long wavelength error modes exploits what is in effect the geometric subspace of the original problem.

Krylov methods come under the category of algebraic subspace methods. A Krylov subspace is formally defined as one which is generated by powers of a matrix operating on a single vector [54]. Each iterate of a numerical solution can be represented by a large vector and the aim of Krylov methods is to converge this vector to a final solution. Consequently these methods are also referred to as vector sequencing methods [144] or sequence transformation methods [12] and they are normally derived using minimization principles. The Generalized Minimum Residual Method (GMRES) is one such method that is gaining widespread use. It is a conjugate gradient like method that applies directly to non-symmetric linear systems of equations but which has also been extended to nonlinear equations. Thus, it is essentially a method of steepest descent, in which a number of orthogonal search directions are explored before the definitive update is made. As the subspace that is exploited in the minimization process may be considered to be algebraic in nature, it is classified as an algebraic subspace method.

To close this discussion, mention should be made of algebraic multigrid methods [136]. Their development may be viewed as an attempt to abstract and formalize the ideas inherent in grid based multigrid to algebraic sets of equations. Hence the principles of multigrid may be considered as being united both in geometric subspace and algebraic subspace representations. The development of algebraic multigrid methods has lagged that of geometric multigrid methods and though the former are particularly attractive for unstructured grid methods, at present they are only applicable to linear problems. As such, they are not considered further in this study. They are mentioned principally to demonstrate the connection that can exist between convergence acceleration methods which have different subspace representations. The basic principles of the major subspace

methods in current use, namely those of the geometric multigrid method and the GMRES method, are considered next in finer detail.

4.2.1 The Multigrid method

The first *true* multigrid publication was by Fedorenko [44] in 1964, in which he formulated a multigrid algorithm for a classical five-point finite difference discretization of the Poisson equation, proving that the work required to reach a given precision is $\mathcal{O}(N)$ in which N is the total number of grid points in a given problem. Other early theoretical estimates of the method were pessimistic and the theory was not implemented in a practical algorithm until a pioneering paper by Brandt [8] in which the first numerical results were presented. In a later paper, Brandt [9] outlined the main principles and the practical utility of multigrid methods, drawing wider attention to the method. Further important contributions to the development of the multigrid method have been made by Hackbusch [61, 62].

The application of a multigrid method to transonic Euler computations was first carried out by Ni who used it to accelerate his time-marching scheme [106], a Lax-Wendroff type method. Called the *multiple grid* method, Ni's original implementation of multigrid was restricted to Courant numbers less than one, severely restricting its effectiveness. This restriction was subsequently relaxed by Chima, Turkel and Schaffer [22] and others. Later Jameson [70] devised one of the most efficient multigrid methods for hyperbolic problems based on his own finite volume multi-stage formulation [76]. Indeed, it is largely due to the work of Jameson that multigrid has been popularized as a convergence acceleration technique for aerospace applications. The Jameson approach is in fact equivalent in form to the FAS scheme of Brandt (described below), but differs in the grid transfer operators. (The multigrid work contained in the present study follows this approach and is described in detail in chapter 5). An excellent introduction to multigrid which devotes some attention to computational fluid dynamic applications is that by Wesseling [181].

Multigrid Principles

Multigrid methods have met with great success in nonlinear elliptic equations for which substantial theory has been developed. In elliptic problems the fundamental idea behind the multigrid method is to accelerate the solution of a problem that requires the resolution given by a fine grid, by computing corrections on a coarser grid. The motivation for this approach comes from consideration of the errors of the numerical solution in the frequency domain. High-frequency errors associated with local variations in the solution are eroded by the application of a time marching procedure. The low-frequency errors are more global in nature and they are relatively insensitive to the fine grid iterative procedure. Once the high-frequency errors in a solution have been eliminated, further fine grid iterations are inefficient at tackling the remaining low-frequency errors for which one receives diminishing returns. With multigrid, the solution from the fine grid is transferred to a coarser grid on which the low-frequency (fine grid) errors are better represented and therefore eliminated more efficiently. The coarse grid corrections computed in this manner are interpolated back to the fine grid to update

the solution. The procedure may be applied recursively on a sequence of coarser and coarser grids. Each such grid level is responsible for eliminating a particular error bandwidth.

Much of this theory, however, is not strictly applicable to hyperbolic equations though heuristic reasoning suggests that it should be possible to accelerate the convergence of hyperbolic systems by using large timesteps on coarse grids as this will enable disturbances to pass more rapidly to and from the outer boundary. This reasoning is based on the fact that the permissible timestep of an explicit scheme is larger for a coarse grid than for a fine grid and therefore for a given number of timesteps a wave travels further on a coarse grid.

Work by Eriksson and Rizzi [43] sheds some light on why multigrid methods are less effective with hyperbolic problems containing discontinuous solutions than they are for elliptic problems. To do this they calculated the eigenvalues and eigenvectors of the iteration matrix associated with an explicit multistage central differencing scheme that uses artificial viscosity and found that there are only a few dominant eigenvalues but more importantly, the corresponding eigenvectors are highly oscillatory. To get rid of the high-frequency component of these dominant error modes, many iterations on the fine mesh were needed.

In addition, it is known that a small perturbation of a discontinuous solution produces delta-like high-frequency error modes. Therefore when there is an error in the shock position, it is not possible to correct it using coarser grids. Thus a solution which has a sharp shock requires more iterations on the finest grid in order for it to attain its correct position. The use of multigrid in isolation is not effective on these type of high-frequency error components as they cannot be represented on coarse mesh.

Consequently, in order to use multigrid methods effectively for hyperbolic problems, more emphasis must be placed on developing efficient smoothing algorithms for damping the high-frequency errors. (In the multigrid jargon, the smoothing algorithm or *smoother* is the iterative method employed to obtain the solution on the finest grid level). By careful matching of a smoother that is effective at damping high-frequency errors to a suitable multigrid method inherently efficient at expelling the lower-frequency errors, the convergence of discontinuous flows can be accelerated more effectively. (For a description of the ideas behind improving the smoothing properties of an iterative scheme for use with multigrid, refer to the section on *optimal smoothing methods* (Section 4.5). Another approach to improving the high-frequency damping of error modes which are specifically associated with errors in shock position, is termed *shock acceleration*. This is a new technique proposed in Chapter 6 of the present study).

The greatest achievement that a properly formulated multigrid algorithm can achieve is referred to as *multigrid convergence*. Assuming that any grid used permits sufficient resolution of flow features, the term multigrid convergence implies convergence in a fixed number of iterations, regardless of the size of the finest grid (viz. grid independent convergence). For elliptic problems multigrid convergence has been realized in several instances, however, for hyperbolic problems it has been achieved only for a simple channel flow problem by Mulder [102]. The reason for this is the loss of information during grid coarsening when long waves in one direction are coupled to short waves in another direction. This problem is discussed further in a later section titled *semi-coarsening*.

The general mathematical formulation of a multigrid algorithm is now pre-

sented. This is done without reference to any particular set of equations, the discretization or grid type or the dimensionality of the problem. In this way, it is the intention to demonstrate the basic principles of the method. Initially, a well known multigrid strategy for nonlinear problems which was proposed by Brandt [9] and known as the Full Approximate Storage (FAS) algorithm, is described. After this, the strategies used to cycle between different grid levels will be discussed.

Full Approximate Storage (FAS) Algorithm

Consider the solution of the discrete problem

$$L_h u_h = f_h \tag{4.1}$$

in which the subscripts refer to the discretization of the continuous problem on a mesh of spacing h. The current estimate of the solution u_h is denoted as $\bar{u}_h$, which is determined by solving the above *fine grid* equation using an iterative technique (i.e. the smoothing scheme). Since $\bar{u}_h$ does not satisfy this equation exactly, its substitution into equation (4.1) produces

$$L_h \bar{u}_h - f_h = r_h \tag{4.2}$$

Here r_h is the residual which vanishes only when the exact solution to the discrete problem is found. The object of the scheme is to compute a correction v_h such that the exact solution is given by

$$u_h = \bar{u}_h + v_h \tag{4.3}$$

Clearly this correction is in fact the fine grid error. The coarse grid solution variable $\bar{u}_H$ is next introduced. It is defined as

$$\bar{u}_H = \bar{I}_h^H \bar{u}_h + v_H \tag{4.4}$$

where v_H is the coarse grid correction and $\bar{I}_h^H$ represents an operator that interpolates *solution variables* from the fine grid h to the coarse grid H. In multigrid terminology $\bar{I}_h^H$ is termed a *restriction operator*. From the above definition, the correction v_H may be expressed as

$$v_H = \bar{u}_H - \bar{I}_h^H \bar{u}_h \tag{4.5}$$

Now providing the high-frequency errors in the solution have been eliminated by sufficient smoothing iterations on the fine grid, the correction v_H must be smooth, containing only the lower-frequency errors. Therefore it should be represented well on a coarser grid. The coarse grid equation for *linear* problems is given by

$$L_H v_H = -I_h^H r_h \tag{4.6}$$

It may be re-expressed using equation (4.5) to obtain the nonlinear coarse grid equation:

$$L_H \left(\bar{u}_H - \bar{I}_h^H \bar{u}_h \right) = -I_h^H r_h \tag{4.7}$$

Here I_h^H represents the restriction operator which transfers *residuals* from fine to coarse grids. Note that the operators $\bar{I}_h^H$ and I_h^H may in principle be different from one another. The *nonlinear* coarse grid equation may be rewritten as

$$L_H \bar{u}_H = P_H \tag{4.8}$$

where

$$P_H = L_H \bar{I}_h^H \bar{u}_h - I_h^H r_h \tag{4.9}$$

Written in this form, it is apparent that the structure of the coarse grid equation is similar to that of the original fine grid equation (4.1), but with a rather different source term. The coarse grid source term P_H is also referred to as a *residual forcing function* as it drives the solution on the coarse grid by the residuals obtained from the fine grid. The residual forcing function also ensures that the original nonlinear problem of the fine mesh is correctly approximated on the coarse grid. (In contrast, if equation (4.6) were used instead of equation (4.8), the coarse grid algorithm would solve a linearized problem, rather than the original nonlinear problem). In addition, use of the nonlinear coarse grid equation permits the use of similar solution strategies for solving both the coarse and fine grid problems. If the grid is coarse enough, equation (4.8) may be solved exactly. If this is not feasible in the event that there are yet lower-frequency errors in the solution that are not represented sufficiently well on the current coarse grid, the existing procedure may be performed recursively on coarser grids. Once the coarse grid equations have been solved, exactly or approximately, the fine grid variables are updated as

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_H^h(\bar{u}_H^{new} - \bar{I}_h^H \bar{u}_h^{old}) \tag{4.10}$$

which can also be expressed as

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_H^h v_H \tag{4.11}$$

where I_H^h is called a *prolongation operator* and represents the interpolation of the coarse grid variables to the fine grid. Note that in the final update, equation (4.10), it is the difference between the initial and final coarse grid variable which is used, since this constitutes the definition of the correction given by equation (4.4).

Now by further manipulation of the above equations, the forcing function may be rewritten as

$$P_H = f_H + \tau_H \tag{4.12}$$

in which

$$f_H = I_h^H f_h \tag{4.13}$$

and

$$\tau_H = L_H \bar{I}_h^H \bar{u}_h - I_h^H (L_h u_h) \tag{4.14}$$

The term τ_H is often termed the *defect correction* and loosely represents the difference between the coarse grid discretization and the interpolation of the fine grid discretization onto the coarse grid. Its presence in equation (4.8) ensures that the fine grid problem is represented by the coarse grid discretization and, that both the coarse and fine grid equations converge to the same solution. This may be appreciated by considering the case where the fine grid equations have been solved exactly. In this instance, the fine grid residuals go to machine zero as does their interpolated result on the coarse grid. Thus the right-hand side of the coarse grid equation (4.7), vanishes and the solution interpolated from the fine grid onto the coarse grid (i.e. $u_H = \bar{I}_h^H u_h$) satisfies the coarse grid equation exactly. Therefore, no further corrections are generated from the coarse grid.

One of the principal advantages of the FAS algorithm is its ability to handle nonlinear problems directly. This obviates the need to linearize the problem, with all the difficulties and complications that such approaches usually entail. Furthermore, by utilizing this ability to handle nonlinear problems in the context of a time marching method, global convergence can be attained. This offers a potential advantage over Krylov and Newton-like methods which are locally convergent.

Multigrid schedules

The term *multigrid schedule* or *multigrid cycling strategy* refers to the order in which grids are visited (viz. the recursive structure which organizes the switching between the various different grid levels). Cycling strategies may be divided into two basic types: fixed schedules and adaptive schedules.

With a fixed schedule, an invariant pattern of coarse and fine grid iterations is prescribed. The most prevalent fixed schedules are the called the V-cycle, the Saw-tooth cycle and the W-cycle. Each is illustrated with four grid levels in Figure 4.1. The multigrid V-cycle begins on the finest grid in the sequence

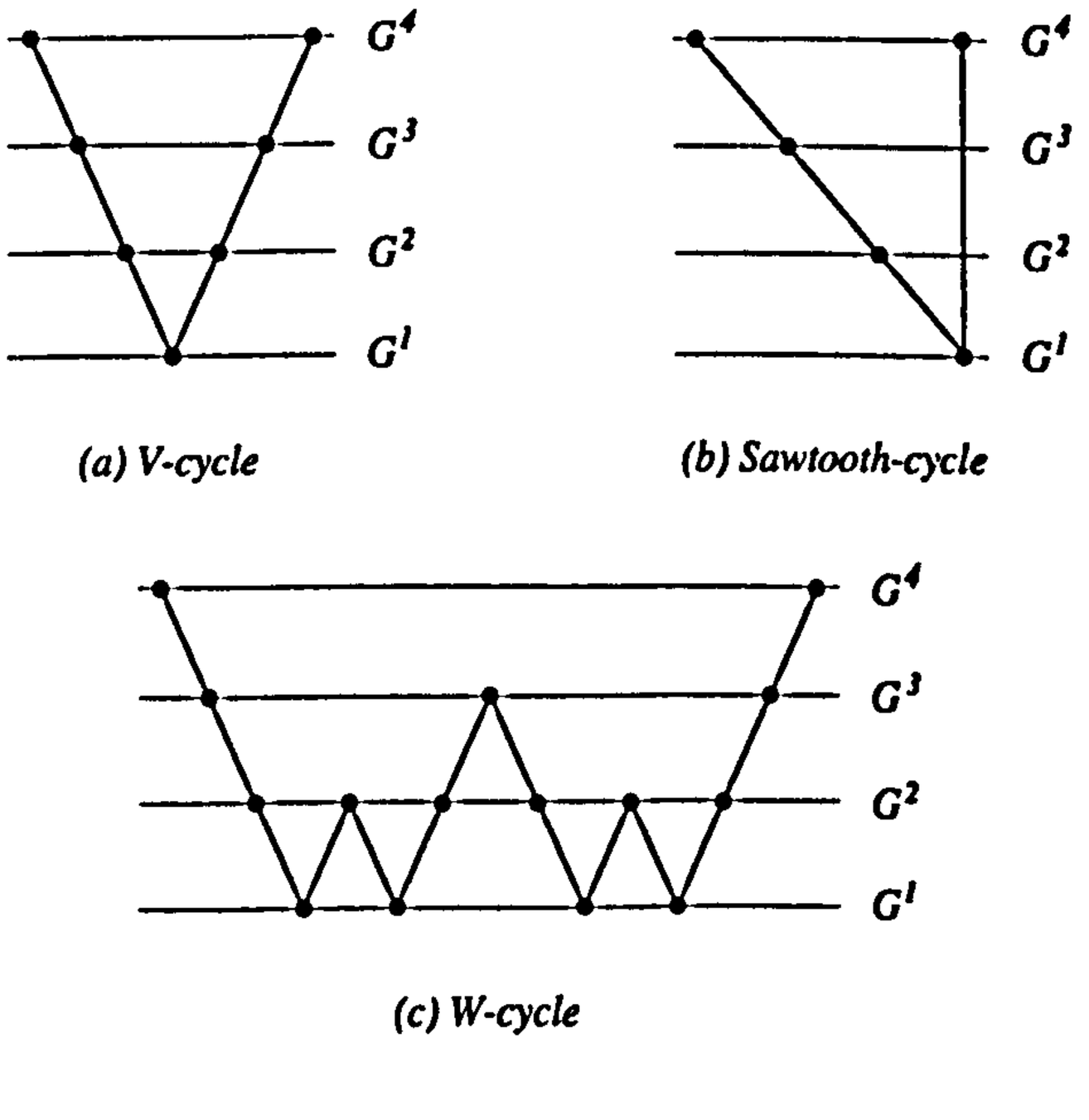


FIGURE 4.1. Common Multigrid schedules.

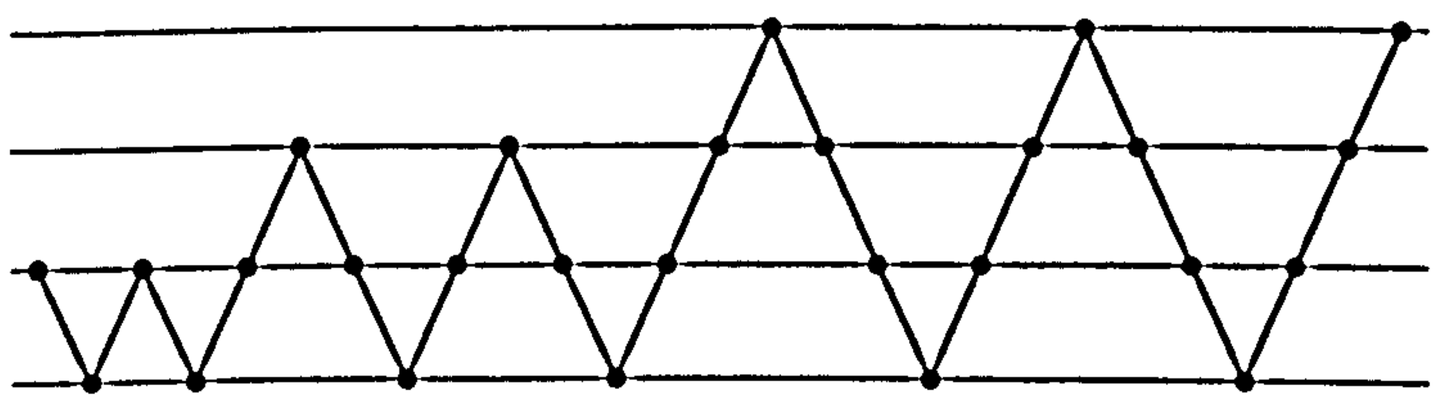
(grid level G^4 in the diagram) where one or more pre-smoothing iterations (i.e. time-steps or relaxation sweeps) are performed before coarse grid correction. The solution and residuals are then interpolated (restricted) to the next coarser grid level on which further pre-smoothing is performed. This procedure is repeated on each successively coarser grid of the sequence until the coarsest grid (level G^1), is reached. Then, the correction to the solution is calculated and interpolated (prolongated) to the next finer grid level where one or more post-smoothing iterations are preformed. The process is repeated until the finest grid (level G^4), is attained. This constitutes a single multigrid V-cycle. It is repeated successively a prescribed number of times or until a specified tolerance of the fine grid residual is attained. The saw-tooth cycle is sometimes loosely referred to as a V-cycle. In fact it is a variant of the V-cycle in which no smoothing is performed on the coarse-to-fine phase of the cycle. In the W-cycle, the coarser grid levels are visited more frequently and it is sometimes said to be more robust than the V-cycle. In practice the W-cycle is not often found to be necessary, and it is clearly more computationally intensive.

The idea of an *adaptive schedule* is to avoid any unnecessary computational effort inherent in a fixed schedule by monitoring the convergence process and tailoring the multigrid strategy to suit the pathology of the attempted problem. With this type of schedule, when it is observed that the high-frequency errors on the current grid have been effectively eliminated, usually by observing a sharp drop in the convergence rate, the algorithm automatically switches to a coarser grid. Although adaptive schedules may appear to be more efficient, they are harder to program and their efficacy does not always outweigh their increased sophistication to any significant extent. Consequently, the practical considerations of simplicity and robustness favour fixed schedules in preference to adaptive ones. Algorithms for several multigrid schedules are given in pseudo-code form by Wesseling [181].

Full multigrid

Full Multigrid (FMG), also referred to as *nested iteration*, is a procedure that was introduced by Brandt [10, 11] based on the following idea. When no *a priori* information about a solution is available to assist in the choice of the initial guess for the finest grid, it is often wasteful to commence the computation on the finest grid. Since coarse grid computations are so much cheaper, it is frequently beneficial to use coarse grids in order to provide a starting solution for the fine grid. With FMG, the solution is started on a coarse grid (sometimes the coarsest grid), on which the solution is iterated. If the starting grid is not the coarsest in the sequence, the algorithm cycles between lower level (coarser) grids a prescribed number of cycles. The solution is then interpolated to the next finer grid for which it provides a good initial guess. The process is repeated at this grid level and recursively at other levels until the finest grid is reached by which time a well-conditioned starting solution is available for the fine grid solution. At that point, normal cycling based on all grids in the sequence is commenced.

Thus using nested iteration, the normal multigrid schedule is modified only at start-up such that the fine grid starting solution is a prolongation of a coarser grid solution. This strategy is illustrated in Figure 4.2 in which FMG is implemented in the context of a V-cycle with four grid levels. As shown in the diagram, the



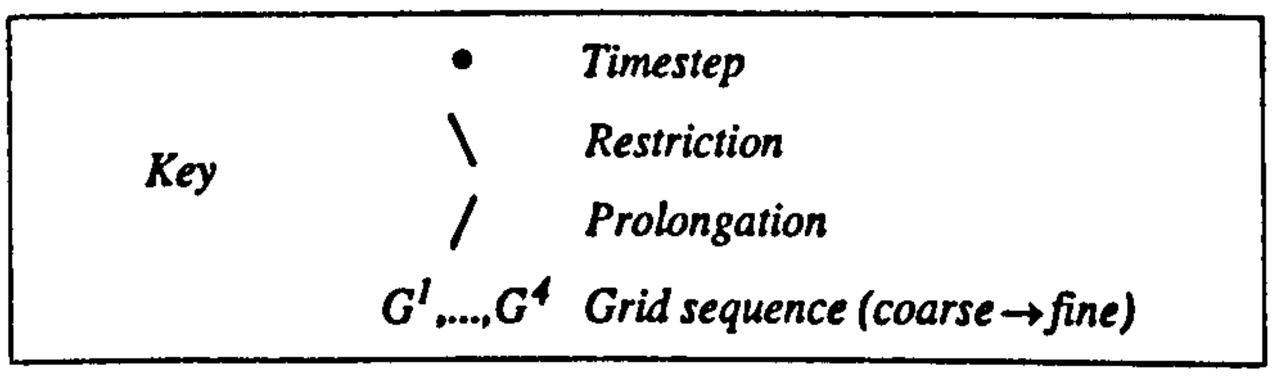


FIGURE 4.2. An example of Full Multigrid (FMG) using V cycles.

computation commences on level G^2. The parameters that vary are the number of timesteps/relaxations performed at each grid level and the number of times a given level is visited during the FMG phase.

Semi-coarsening

Explicit iterative methods can have difficulty in efficiently eliminating the highest-frequency errors in problems where flow-alignment or anisotropic diffusion occurs. A example of this is high Reynolds number flow in which the unknown solution parameters are strongly coupled in one direction due to high mesh aspect ratios in the boundary layer and wake. This is because explicit methods have difficulty in eliminating the highest-frequency errors in both the strongly coupled and weakly coupled directions simultaneously. Similarly when the standard multigrid grid coarsening procedure (i.e. doubling the mesh size in each space dimension), is applied to such problems its efficacy diminishes and poor convergence may result. This may be compensated for by the use of a technique known as *semi-coarsening*. In this type of grid coarsening, the frequencies that are not being smoothed effectively must be identified. Then, a sequence of mesh is introduced that allows these errors to be represented as dominant high-frequency modes. This is accomplished by coarsening the grid selectively in the direction of strong coupling. This ensures that the additional coarse levels relieve any stiffness associated with spatial anisotropy. Semi-coarsening techniques have been developed for anisotropic problems by Mulder [103] and Koren and Hemker [79]. Recent work by de Zeeuw [30] considers these techniques from a fundamental level of understanding and suggests a new way for further development of a generalized and robust grid coarsening strategy for flow alignment problems, known as *Sparse-grid multigrid*.

4.2.2 KRYLOV METHODS

Most classical iteration schemes employ information exclusively from time level n to advance to level $n+1$. Consequently all data obtained from the first $n-1$ iterates are discarded, even though they could contain useful information on how best to proceed beyond the current iterate. In contrast to this approach, Krylov or vector sequencing methods exploit the use of information at several different iterates, making them significantly more memory intensive than other acceleration methods. However, this disadvantage assumes less importance as computer memory becomes cheaper. Among the more important Krylov or vector sequencing methods that have recently been applied to problems in computational fluid dynamics are those of Saad and Schultz [137], Wigton *et al.* [183], Hafez *et al.* [63] and Celestina and Sidi [19]. In what follows the Generalized Minimum Residual (GMRES) method of Saad and Schultz [137] is described after which two variants of the method, α–GMRES and mGMRES are discussed. However, it must be stressed that the methods described here constitute only one class of Krylov method, albeit perhaps the most successful class, and that other Krylov methods such as the Minimum Polynomial Extrapolation (MPE) method of Smith, Ford and Sidi [144] which are not discussed, also hold much promise.

GMRES

As its name implies the *Generalised Minimum Residual* (GMRES) method is based on minimization of the residual. It is really a method of steepest descent that uses several orthogonal search directions to determine a definitive update. As such the basic method is founded on grounds of mathematical justification rather than any in-depth physical perception. The original algorithm only applies to linear systems of equations. Wigton *et al.* [183] modified it for nonlinear equations and it is with reference to their variant of the method that it is described below.

Consider a differentiable system

$$\mathbf{F}(\mathbf{u}) = 0 \tag{4.15}$$

of N nonlinear equations in N unknowns. The differential of $\mathbf{F}$ at $\mathbf{u}$ in the direction of $\mathbf{p}$ is defined by the finite difference approximation

$$\mathbf{F}'(\mathbf{u};\mathbf{p}) = \lim_{\epsilon \to 0} \frac{\mathbf{F}(\mathbf{u}+\epsilon\mathbf{p}) - \mathbf{F}(\mathbf{u})}{\epsilon} \tag{4.16}$$

For practical purposes, $\mathbf{F}'(\mathbf{u};\mathbf{p})$ is estimated by taking ϵ to be a small number and ensuring that the variables $\mathbf{u}$ together with the component values of $\mathbf{F}(\mathbf{u})$ are reasonably scaled to permit accurate evaluation. Given an approximate solution $\mathbf{u}^n$ to equation (4.15), a single cycle of GMRES advances the solution by first choosing k orthonormal search directions $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k$ as follows:

1. Compute and normalize $\mathbf{p}_1$

$$\mathbf{p}_1 = \mathbf{F}(\mathbf{u}^n) \tag{4.17}$$

$$\mathbf{p}_1 = \mathbf{p}_1 / \|\mathbf{p}_1\| \tag{4.18}$$

(Here the notation $\|\bullet\|$ represents the Euclidean or l_2 norm).

2. For $j = 1, 2, \ldots, k-1$ take

$$\mathbf{p}_{j+1} = \mathbf{F}'(\mathbf{u}^n; \mathbf{p}) - \sum_{i=1}^{j} b_{ij}\mathbf{p}_i \tag{4.19}$$

where

$$b_{ij} = (\mathbf{F}(\mathbf{u}^n; \mathbf{p}_j), \mathbf{p}_i)$$

such that

$$(\mathbf{p}_{j+1}, \mathbf{p}_i) = 0 \text{ for } i = 1, 2, \ldots, j$$

(Note that $(\circ, \circ)$ *denotes the inner product).*

3. Normalize $\mathbf{p}_{j+1}$

$$(\mathbf{p}_{j+1}, \mathbf{p}_i) = 0 \text{ for } i = 1, 2, \ldots, j$$

$$\mathbf{p}_{j+1} = \mathbf{p}_{j+1} / \|\mathbf{p}_{j+1}\| \tag{4.20}$$

4. Update $\mathbf{u}^n$ using

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^{k} a_j \mathbf{p}_j \tag{4.21}$$

in which the coefficients a_j are chosen to minimize the function $\mathbf{F}(\mathbf{u})$ as

$$\min_{a_j} \|\mathbf{F}(\mathbf{u}^{n+1})\|^2$$

$$= \min_{a_j} \left\| \mathbf{F}(\mathbf{u}^n + \sum_{j=1}^{k} a_j \mathbf{p}_j) \right\|^2$$

$$\cong \min_{a_j} \left\| \mathbf{F}(\mathbf{u}^n) + \sum_{j=1}^{k} a_j \mathbf{F}'(\mathbf{u}^n; \mathbf{p}_j) \right\|^2 \tag{4.22}$$

To ensure the overall success of this algorithm, this least squares problem must be solved by a numerically stable procedure such as the QR algorithm which may be found in standard numerical analysis textbooks such as [112].

A cycle of the GMRES algorithm is an approximation to a Newton iteration cycle which uses

$$\mathbf{F}(\mathbf{u}^n + \mathbf{p}) \cong \mathbf{F}(\mathbf{u}^n) + \mathbf{F}'(\mathbf{u}^n; \mathbf{p}) \tag{4.23}$$

to estimate a value of $\mathbf{p}$ which ensures $\mathbf{F}(\mathbf{u}^n + \mathbf{p}) = \mathbf{0}$. Thus GMRES effectively solves the following linear equation for $\mathbf{p}$:

$$\mathbf{F}(\mathbf{u}^n) + \mathbf{F}'(\mathbf{u}^n; \mathbf{p}) = \mathbf{0} \tag{4.24}$$

by finding the best possible solution over the k dimensional linear subspace spanned by the search directions $< \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k >$. Theoretically with $k = N$, GMRES would search the entire subspace and therefore compute the exact solution to the minimization problem –at enormous expense. In practice the key to efficiency is to use only a small number of search directions k, but sufficient

enough to provide a good solution to equation (4.24). An efficient implementation of GMRES(k) (GMRES with k search directions) has a total memory requirement of essentially $(k+4)N$. Clearly there is also an increased operations count which varies linearly with k. However, all the common operations are vectorizable and as they may be carried out over long one-dimensional loops, equal to the number of unknowns in the problem, it runs extremely efficiently on vector computers.

For many problems the efficiency of GMRES may be improved substantially through suitable preconditioning. It is known that the rate of convergence of the method, measured by the value of k required to achieve a given level of accuracy, depends on the eigenvalue distribution. Like most iterative methods, the more the eigenvalues are clustered together, the faster GMRES will converge. Thus the objective of the preconditioner is to replace the given problem with an equivalent problem (i.e. one with the same solution), in which the clustering of eigenvalues is improved. In fact, standard iterative methods such as implicit schemes or explicit time marching schemes can be used together with GMRES and in this context, they are viewed by exponents of the method as fulfilling the function of a preconditioner. To this end, GMRES is often used as a smoother for the multigrid method. Like the other vector sequencing methods, GMRES can only speed up the latter stages of convergence –once the solution is well within its basin of attraction. Hence preconditioning is often essential for highly nonlinear problems in order to ensure a robust solution strategy. Finally, to get the most out of the algorithm, fine-tuning a few free parameters is usually necessary.

α-GMRES

The application of both fully implicit and Newton-like methods to nonlinear systems of steady-state equations produces a sparse non-symmetric linear system given by

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.25}$$

in which the structure of $\mathbf{A}$ depends on the chosen spatial discretization. For instance, with a second-order accurate two-dimensional Navier-Stokes problem, the linear system would typically be a block 13 point diagonal matrix of sparse structure. Xu *et al.* [186] have reported that using GMRES directly with different Newton-like methods can produce non-convergent solutions. Furthermore, they discovered that applying a simple block-diagonal preconditioning to equation (4.25) as

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{x} = \mathbf{D}^{-1}\mathbf{b} \tag{4.26}$$

where $\mathbf{D}$ is the block-diagonal matrix of $\mathbf{A}$, gives little improvement. Based on these observations, they have devised a new preconditioner in which a damping parameter α is added to the diagonal of equation (4.26). The new preconditioned system then becomes

$$(\alpha\mathbf{I} + \mathbf{D}^{-1}\mathbf{A})\mathbf{x} = \mathbf{D}^{-1}\mathbf{b} + \mathbf{x}, \quad (\alpha > 0) \tag{4.27}$$

To solve this new system, an outer loop is introduced such that

$$(\alpha\mathbf{I} + \mathbf{D}^{-1}\mathbf{A})\mathbf{x}^{k+1} = \mathbf{D}^{-1}\mathbf{b} + \alpha\mathbf{x}^{k} \tag{4.28}$$

For α in the range $(0 < \alpha < 1)$, it is proven [186] that the new iterative procedure is convergent. Thus, given $\mathbf{x}^k$, equation (4.28) may be solved for $\mathbf{x}^{k+1}$ using an inner iterative strategy. With GMRES chosen as the inner algorithm, the new multilevel iterative scheme is referred to as the α-GMRES method. In practice, the value of α is selected by balancing the convergence of the outer iterative procedure (equation (4.28)) with that of the inner scheme (GMRES). For very small values of α, it is noted that non-convergent solutions may still occur.

Some impressive improvements in the convergence of hypersonic problems have been obtained by Qin *et al.* [115] using the α-GMRES method together with Newton-like methods (Refer to section 4.3 for further details). In [115], an approximate eigenvalue analysis is also conducted which demonstrates that the block-diagonal preconditioning improves the clustering of the eigenvalue distribution while the damping displaces the entire distribution to the right by a distance equal to the damping parameter α. This indicates that the combined effect of preconditioning and damping improves the convergence of the GMRES procedure and indeed this is confirmed in the results presented. Furthermore, it is reported that the algorithm is fully parallelizable on both shared and distributed-memory computers, as the results presented [186] from tests using 4 processors of a distributed-memory architecture confirm.

mGMRES

An interesting variant of the GMRES algorithm that was introduced for a model hyperbolic problem by Gustafsson and Lötstedt [60] and subsequently extended to the Euler equations by Lötstedt [91], is now described. It is referred to as the modified GMRES algorithm or mGMRES(k). To formulate their modification, Gustafsson and Lötstedt drew on the similarities shared by GMRES and time marching iterative schemes. Like the authors of the α-GMRES method, their motivation was the poor convergence of the standard method for first-order hyperbolic systems.

For a first-order system such as the Euler equations in a bounded domain, convergence takes place as a consequence of two mechanisms –wave propagation of smooth error modes through the open boundaries and damping of the amplitude of oscillatory error modes. Most iterative schemes may be considered as approximations of a time-dependent differential equation and for smooth solutions the approximation should behave essentially as the differential equation. Newton's method may be interpreted similarly as an explicit approximation of the time-dependent problem

$$\frac{du}{dt} = \alpha L(u) \tag{4.29}$$

in which L is an approximation for a first-order differential operator, while here the coefficient α is the Courant number –a measure of the wave speed. Now since a cycle of the GMRES algorithm is equivalent to a Newton iteration cycle, it is argued [60] that the GMRES algorithm may be thought of as resembling an explicit marching procedure and in particular a Runge-Kutta method. The difference between them is that in GMRES, the coefficients a_j in equations (4.21) and (4.22) are recalculated for each new timestep and the scheme remains nonlinear even for linear problems, while for the Runge-Kutta scheme, the Runge-Kutta coefficients remain constant throughout.

Finding that the standard GMRES produced poor convergence in first-order problems, Gustafsson and Lötstedt [60] attempt to exploit the wave propagation attributes of Runge-Kutta schemes in a modified GMRES formulation which they term mGMRES(k) in which k represents the number of search directions. To accomplish this, the first coefficient a_1 in equations (4.21) and (4.22) is fixed while the remaining coefficients are determined from a suitably modified minimization procedure. This leaves the choice of a_1 to the user in an analogous fashion to choosing the timestep for a traditional explicit marching scheme where the stability limit must be kept. Furthermore, its choice does not appear to be problem dependent. Rather, it appears to be related to the number of search directions that are to be adopted in the mGMRES(k) algorithm and in turn the particular Runge-Kutta scheme that produces the wave propagation properties that are desired. One advantage claimed is that the remaining coefficients a_j ($j \neq 1$) which are found automatically in the algorithm, provide optimal damping of the residual, given that a_1 is appropriately selected. Numerical evidence from the standard and modified GMRES algorithms applied to a model hyperbolic problem [60] suggests that poor convergence from the standard implementation is caused by a_1 varying rapidly and being often close to zero. Further evidence is presented for the Euler equations [91] from which it is apparent that a stall in the convergence history from the standard algorithm corresponds to the condition $a_1 \rightarrow 0$. As a_1 has a constant value in the modified algorithm, this problem is obviated and indeed the convergence improves significantly. This suggests that by imposing the condition $a_1 = const \neq 0$, good wave propagation properties are guaranteed by the modified algorithm so resulting in an improvement over the standard method for first-order partial differential equations. To improve convergence further, the basic iterative algorithm is preconditioned by *implicit-explicit residual smoothing* [40] (see section 4.4.3), in tandem with multigrid. Substantial improvements in convergence rate are demonstrated using only 3 search directions and it is claimed that the overall robustness of the algorithm is substantially improved [91].

4.3 Newton-like Methods

In solving the Euler or Navier-Stokes equations for steady flow computations using a finite difference or finite volume formulation with a suitable discretization of convective and diffusive terms together with correct treatment at the boundaries of the domain, a large sparse nonlinear system results which can be written as

$$\mathbf{R}(\mathbf{Q}) = \mathbf{0} \tag{4.30}$$

To solve this system, an iterative or time-dependent approach may be used, which adds a time or pseudo-time derivative to the system as

$$\mathbf{q}_t + \mathbf{R}(\mathbf{Q}) = \mathbf{0} \tag{4.31}$$

This system could be solved using a fully implicit scheme such as the backward Euler implicit method

$$\left[\frac{1}{\Delta t}\mathbf{I} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right)\right]^n \Delta \mathbf{Q}^n = -\mathbf{R}(\mathbf{Q}^n) \tag{4.32}$$

for which unconditional stability can be achieved. If the linearization is exact, as $\Delta t \rightarrow \infty$ the method reduces to the Newton method

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right)^n \Delta \mathbf{Q}^n = -\mathbf{R}(\mathbf{Q}^n) \tag{4.33}$$

which offers quadratic convergence for the solution of equation (4.30). However, such rapid convergence can only be obtained if an appropriate procedure exists for the exact linearization, viz. for evaluation of the Jacobian. Unfortunately, in most practical applications of interest within computational fluid dynamics it is very difficult to obtain the analytical Jacobian for nonlinear systems, particularly so with sophisticated higher-order upwind schemes involving complicated physical modelling (e.g. turbulence and chemical reactions). It is also difficult to solve the resulting large sparse non-symmetric linear system efficiently. To circumvent these difficulties, two common practices have evolved. The first is to construct simplified implicit operators based on first-order inviscid approximations while the second is to implement an approximate factorization of the implicit operator (i.e. an AF scheme). Both approaches produce inconsistency between the implicit operator and the right-hand side of the system (the explicit operator) which restricts the size of the timestep and generally degrades convergence. An alternative approach to overcoming these difficulties has resulted in the development of Newton-like methods.

Newton-like methods are essentially based on finding an approximation for the Jacobian which is then updated using Newton's iterative method (equation (4.33)). One approach to this, suggested by Curtis, Powell and Ried [26] is to approximate the Jacobian using finite differences. Another approach is the quasi-Newton method of Broyden [13] which was originally developed for nonlinear equations having fully populated Jacobians. The method was subsequently extended to sparse systems by Schubert [141] and Broyden [14]. The basic idea of the quasi-Newton method is to approximate the Jacobian of the nonlinear system using function evaluations that have already been calculated. However, when the matrix is sparse, its inversion does not result in a sparse matrix and hence sparse systems must receive special treatment. For the sparse case, the resulting procedure depends only on the sparse structure of the Jacobian and the solution from the previous iteration. By itself, the method does not have the property of global convergence and hence it may be combined with an explicit time marching scheme which is used in the initial stages of convergence after which the calculation is switched to the quasi-Newton method to exploit its property of rapid local convergence.

Qin and Richards [113, 114] have further developed both approaches for steady-state solutions of the Euler and Navier-Stokes equations and call their variants of these methods the Sparse Finite Difference (SFDN) method and the Sparse Quasi-Newton (SQN) method. The simplicity and generality of these procedures suggests their use as efficient methods for rapid steady-state solutions in Computational Fluid Dynamics. Edwards and McRae [34] also use a quasi-Newton formulation for improving the convergence of Navier-Stokes computations. The SFDN and SQN methods of Qin and Richards are each described below.

4.3.1 Sparse finite difference Newton method

Consider the Navier-Stokes equations in a locally conical coordinate system which are solved with a higher-order spatial discretization. In the *Sparse Finite Difference Newton* (SFDN) method of Qin and Richards [113], the structured sparsity of this system is exploited while the Jacobian of the nonlinear system is calculated numerically by a finite difference method. Because the discretization has a 13 point stencil, the number of calculations of $\mathbf{R(Q)}$ can be minimized through a local perturbation $h^l_{i,j}$ of one of the five state variables at every fifth point in both coordinate directions. Thus for a single evaluation of $\mathbf{R(Q)}$,

$$\mathbf{R}\left(\mathbf{Q}+\sum_{\substack{i=m,I,5\\ j=n,J,5}} h^l_{i,j}\mathbf{e}^l_{i,j}\right) \quad for \begin{cases} l=1,5 \\ m=1,5 \\ n=1,5 \end{cases} \tag{4.34}$$

is calculated, where $\mathbf{e}^l_{i,j}$ is the unit vector at the point i,j for the lth state component. Thus the finite difference approximation of the Jacobian is obtained through a total of $5\times5\times5$ evaluations of $\mathbf{R(Q)}$. Correctly choosing $h^l_{i,j}$, according to machine zero and the rounding errors in calculating $\mathbf{R(Q)}$, the method can produce quadratic convergence.

To use the method, a time-dependent marching scheme is used to provide a starting solution for the initial phase of convergence. For this Qin and Richards adopt an explicit, Runge-Kutta method with local time stepping in which the numerical fluxes are obtained by Osher's scheme. Once the solution attains global convergence the time marching is switched to the SFDN iteration procedure. As theory does not presently exist for estimating the basin of attraction for such iterative procedures, the switching point is determined heuristically (e.g. depending on test conditions a 2–4 order of magnitude reduction in the residual may suffice). The nonlinear SFDN method has been combined with the α-GMRES linear method (described above) to produce compound efficiency. Results for the SFDN–α-GMRES method, as it is called, are presented in [115] for a hypersonic test case, indicating the rapid convergence that can be achieved.

4.3.2 Sparse quasi-Newton method

The *Sparse Quasi-Newton* (SQN) method of Qin and Richards [113, 114] updates an approximation to the Jacobian from the solution of the linear system and the available value of $\mathbf{R(Q)}$. To maintain the sparsity structure of the Jacobian, a matrix projection operator $\mathbf{P_J(M)}$ maps a matrix $\mathbf{M}$ to a matrix retaining only non-zero entries of the Jacobian. The updating procedure then becomes:

$$\begin{aligned} \mathbf{A}^n &= -\mathbf{R}(\mathbf{Q}^n) \\ \mathbf{Y}^n &= \mathbf{R}(\mathbf{Q}^{n+1}) - \mathbf{R}(\mathbf{Q}^n) \\ \Delta\mathbf{A}^n &= \mathbf{P_J}\left[\mathbf{D}^+(\mathbf{Y}^n - \mathbf{A}^n\Delta\mathbf{Q}^n)(\Delta\mathbf{Q}^n)^T\right] \\ \mathbf{A}^{n+1} &= \mathbf{A}^n + \Delta\mathbf{A}^n \end{aligned} \tag{4.35}$$

in which $\mathbf{A}$ is the approximated Jacobian, and $\mathbf{D}^+$ is a diagonal matrix maintaining the sparsity structure of the original Jacobian, which is determined from the linear solution $\Delta\mathbf{Q}^n$.

The method has been formulated by Qin and Richards for nonlinear systems with sparse block-structured Jacobian matrices arising from the Euler and Navier-Stokes equations and it has been proven to render superlinear convergence. However, like the SFDN method described immediately above, the convergence of the method is a local property and therefore to use it, a time-dependent marching scheme is necessary to provide a starting solution for the initial phase of convergence. Once global convergence is achieved the scheme is switched to the SQN iteration procedure. The nonlinear SQN method has also been used with the α-GMRES linear method and named the SQN–α-GMRES method. Results for the method are presented in [115] where it is compared to the SFDN–α-GMRES method for a hypersonic test case.

The SQN method is demonstrated by Qin *et al.* [115] to produce slower (superlinear) convergence to the SFDN method which demonstrates quadratic convergence, though it is noted that the SQN method requires less computational effort in generating the Jacobian approximation than the SFDN method. In the test cases presented in the latter reference, the CPU usage was similar for the two methods. However, in practice the difference between quadratic and superlinear convergence may be significant due to the computational intensity involved in solving the large linear systems for each iteration. In some instances this could give an advantage to the SFDN method.

4.4 Residual Smoothing

To maintain stable solutions, explicit time marching schemes for hyperbolic problems must obey the Courant-Friedrichs-Lewy (CFL) condition which requires that the domain of dependence of the numerical scheme must at least contain the domain of dependence of the governing partial differential equations. This places a restriction on the allowable size of the timestep. However, if this stability restriction can be relaxed, then the permissible timestep may be increased to accelerate convergence. To achieve this, the support of the numerical scheme must be increased. One way of increasing the support is to average (or smooth) each cell residual with neighbouring residuals, a technique known as *residual smoothing*. Conceptually it may be thought of as introducing a moderate degree of 'implicitness' into a basic algorithm. An additional benefit of residual smoothing is that it can improve the efficiency of multigrid by enhancing the smoothing properties of the time marching procedure. Consequently it is frequently used in combination with multigrid. Residual smoothing techniques may be classified into two broad classes, namely ones based on centred operators and ones that are upwind-biased. However, both centred residual smoothing methods and upwind residual smoothing methods may be applied to upwind methods.

The simplest of these methods is *Explicit Residual Smoothing* (ERS). It was first advanced by Jameson [68] and is based on an explicit centred operator. It produces fairly modest though worthwhile improvements in convergence and is particularly suited to algorithms intended for parallel computing and to unstructured grid methods. (In this context, the term unstructured covers both unstructured grid solvers as well as structured solvers that use unstructured data-structures. The latter are sometimes used for adaptive mesh algorithms).

Centred *Implicit Residual Smoothing* (IRS), using fixed smoothing coefficients

was first introduced by Lerat [87] for the Lax-Wendroff scheme, and later developed by Jameson for Runge-Kutta schemes [72, 68]. IRS is more effective than ERS and may be implemented with either fixed or variable smoothing coefficients. A variable coefficient formulation was first introduced by Martinelli [96, 97], based on Fourier analysis of a model hyperbolic equation, for two-dimensional viscous flows involving extreme mesh stretching. An improved formulation was subsequently advanced by Wigton and Swanson [182]. Variable coefficient IRS has also been extended to three-dimensions by Radespiel, Rossow and Swanson [118] and several others. However, the development of these formulations is based on hyperbolic considerations only, and they do not directly address the restrictions imposed by the diffusive effects found in viscous flow problems, although they can improve the convergence of viscous flows slightly in problems containing cells of high aspect ratio. To deal more effectively with viscous problems in which the restriction on the timestep is imposed by diffusive limits rather than convective ones, the variable coefficient formulation has been extended by Turkel, Swanson *et al.* [157] and its efficacy demonstrated in two-dimensional applications. Extension to three-dimensions should be straightforward.

In theory, IRS is unconditionally stable and should allow for arbitrarily large Courant numbers. In practice, the higher the Courant number, the poorer the damping of oscillatory high-frequency error modes and hence this imposes a practical limit on the permissible Courant number. Typically the smoothed scheme will be able to run at a Courant number that is two to three times that of the unsmoothed scheme. To increase the Courant number significantly higher, without detriment to the damping properties of high-frequency errors, *Implicit-Explicit Residual Smoothing* (IERS) was introduced by Enander [40, 41], in which implicit residual smoothing is modified by the inclusion of a specially derived explicit operator. With this variant of the technique, it is reported that two-dimensional algorithms may employ Courant numbers that are more than five times that of their unsmoothed counterparts.

Blazek *et al.* [6, 7] introduced residual smoothing methods that use upwind operators rather than centred ones. Since these smoothing methods are implicit in nature, they are known as *Upwind Implicit Residual Smoothing* (UIRS) methods. Three variants of UIRS have been presented. The first is based on projecting the Mach number in each coordinate direction while the second is based on splitting the residual according to the sign of the eigenvectors. The last variant is a smoothing based on a characteristic transformation of the residuals. All the UIRS methods appear to demonstrate superior convergence to the centred smoothing methods, particularly when implemented in upwind algorithms, with efficacy improving with the level of approximation used for the upwind operator. However, the higher the level of approximation of the upwind operator used, the higher the operation count. Clearly this has to be balanced against the speed-up in convergence that is achievable, in order to determine the overall computational efficiency of the method. For two-dimensional schemes which utilize UIRS, it is reported that Courant numbers as high as 10–15 may be used effectively.

Each of the residual smoothing techniques that have been discussed are described below in greater detail.

4.4.1 Explicit residual smoothing

For multistage time-marching methods, Jameson [72, 68] proposed using *Explicit Residual Smoothing* (ERS) to replace the residual at each cell in the computational domain by a weighted average of residuals at neighbouring points. This is carried out for each coordinate direction in turn. Note that throughout this section the subscripts ξ, η, and ζ refer to the coordinate directions. In the ξ-direction it may be written as

$$\bar{R}_i = SR_i \tag{4.36}$$

with similar expressions for the η- and ζ-directions. Here S is the averaging operator. Jameson replaces the residual R_i by the average

$$\begin{aligned} \bar{R}_i &= \varepsilon R_{i-1} + (1-2\varepsilon)R_i + \varepsilon R_{i+1} \\ &= (1+\varepsilon\delta_\xi^2)R_i \end{aligned} \tag{4.37}$$

where δ_ξ^2 is the second-order central difference operator in the ξ-direction, and ε is a positive constant, referred to as the smoothing coefficient. This procedure is carried out at each stage of the time stepping scheme, increasing the support of the spatial operator and so relaxing the restriction on the timestep imposed by the CFL condition. However, if there is an oscillation in R_i at the mesh frequency, smoothing with $\varepsilon = 1/4$ can produce $\bar{R}_i = 0$ for $R_i \neq 0$. In other words, the smoothing is destroying information at the mesh frequency. To prevent this occuring, the coefficient ε must be restricted to the range

$$0 < \varepsilon < 1/4 \tag{4.38}$$

ERS is both simple and cheap to implement but the fixed smoothing coefficients must be determined empirically. Its efficacy is restricted by the limited range of values that the smoothing coefficient(s) can assume. As a result, the Courant number limit can only be increased by a factor of about one half. This restriction may, however, be relaxed by smoothing the residuals in an implicit fashion as described next.

4.4.2 Implicit residual smoothing

Fixed Coefficient Smoothing

Both the range of stability and the robustness of the basic time stepping scheme can be extended by smoothing the residuals implicitly. Centred *Implicit Residual Smoothing* (IRS), was introduced by Jameson for Runge-Kutta schemes [72, 68]. Using a k stage Runge-Kutta scheme, the residual R_m, defined by

$$R_m = \alpha_m \frac{\Delta t}{\nu}(\mathcal{L}_{c_m} + \mathcal{L}_{v_m} - \mathcal{L}_{nd_m}) \qquad m = 1, k \tag{4.39}$$

is computed at each stage m. $\mathcal{L}_{c_m}$ and $\mathcal{L}_{v_m}$ are the differencing operators associated with the convection and physical diffusion terms respectively, while $\mathcal{L}_{nd_m}$ is the numerical dissipation operator. With the (unsmoothed) residual R_m known, IRS may be applied in the form

$$(1-\varepsilon_\xi\delta_\xi^2)(1-\varepsilon_\eta\delta_\eta^2)(1-\varepsilon_\zeta\delta_\zeta^2)\bar{R}_m = R_m \tag{4.40}$$

In the standard form of the method, the smoothing coefficients $\varepsilon_{\xi,\eta,\zeta}$ are each fixed at a constant value for all coordinate directions. $\bar{R}_m$ is the final smoothed residual at the mth stage of a Runge-Kutta cycle after a sequence of residual averaging in the ξ, η, and ζ directions. The form of equation (4.40) is particularly convenient as it requires the inversion of only three tri-diagonal matrices.

Assuming an infinite interval, each tri-diagonal component of equation (4.40) may be inverted to give the explicit solution

$$\bar{R} = \frac{1-k}{1+k}\left[R_i + k\left(R_{i-1} + R_{i+1}\right) + k^2\left(R_{i-2} + R_{i+2}\right) + \ldots\right] \tag{4.41}$$

in which k represents a decay parameter ($k < 1$), which is a solution of

$$\varepsilon = \frac{k}{(1-k)^2} \tag{4.42}$$

The scheme so defined is stable for all Courant numbers ν provided that the smoothing parameter satisfies the condition

$$\varepsilon \geq \frac{1}{4}\left[\left(\frac{\nu}{\nu^*}\right)^2 - 1\right] \tag{4.43}$$

where ν^* is the Courant number limit of the unsmoothed scheme. If IRS is operated at the stability limit, its effect is to collect information from the residuals at all points in the field with an influence coefficient that decays by the factor k for each separation interval away from the point of interest. This is apparent from inspection of equation (4.41). The benefit gained from IRS more than offsets the amount of extra work it entails. With the use of constant smoothing coefficients, IRS can extend the Courant number by a factor of between two and three, provided the mesh used does not contain cells of very high aspect ratio. There are two drawbacks to constant coefficient IRS. Firstly, it cannot significantly improve the convergence of computations in which highly stretched mesh are used, for example, in viscous flows where diffusive effects dominate stability. Secondly the values chosen for the smoothing parameters must be empirically tuned.

Variable Coefficient Smoothing

For problems involving highly stretched mesh and indeed in viscous flow computation, additional support is required to sustain the benefit of accelerated convergence. For central differencing schemes, such as the Jameson, Schmidt and Turkel scheme [76] and its derivatives, this additional support may be obtained through the use of *enthalpy damping*, provided heat transfer effects are negligible [118]. However, most upwind schemes cannot maintain solutions of constant enthalpy throughout the flowfield and therefore they are unable to use enthalpy damping. Instead, the additional support needed to obtain maximum benefit from residual smoothing on highly stretched mesh may be obtained by using locally varying coefficients $\varepsilon_{\xi,\eta,\zeta}$, that account for changes in cell aspect ratio.

Variable smoothing coefficients are derived through the following reasoning. Consider a cell which has sides that are significantly shorter in the η direction

than the other directions. The timestep is then limited by the characteristic wave speed in the direction of the short cell edge. To extend the local stability of the marching scheme in the cell under consideration, it is obvious that the support needs to be increased in the η direction (viz. residual smoothing is needed in this direction). Now if residual smoothing is also applied in the other directions where the characteristic wave speeds are much smaller, then the damping behaviour of the scheme (which is optimal for wave speeds near the stability limit), is impaired. To overcome this problem, the residual smoothing that is applied in each coordinate direction should be suitably factored. To achieve this, the smoothing coefficients are formulated on the basis of hyperbolic considerations using Fourier analysis. Martinelli [96, 97] was the first to give formulae for two-dimensional smoothing coefficients which are functions of the characteristic wave speeds. Several extensions of Martinelli's formulae to three-dimensions have been reported in the literature [118, 157, 4]. The coefficients of the Turkel, Swanson *et al.*[157] formulation are given by

$$\varepsilon_l = \max\left\{\frac{1}{4}\left[\left(\frac{\nu}{\nu^*}\Phi_l\right)^2 - 1\right], 0\right\}, \qquad l = \xi, \eta, \zeta \tag{4.44}$$

where

$$\Phi_\xi = \frac{1}{1+\psi(r_{\eta\xi}+r_{\zeta\xi})} \tag{4.45}$$

$$\Phi_\eta = \frac{1}{1+\psi(r_{\eta\xi}^{-1}+r_{\zeta\eta})} \tag{4.46}$$

$$\Phi_\zeta = \frac{1}{1+\psi(r_{\zeta\xi}^{-1}+r_{\zeta\eta}^{-1})} \tag{4.47}$$

Here ν/ν^* is the ratio of the Courant number of the smoothed scheme to that of the basic (unsmoothed) explicit scheme and the quantities $r_{\eta\xi}, r_{\zeta\xi}$ and $r_{\zeta\eta}$ are ratios of characteristic speeds defined as

$$r_{\eta\xi} = \frac{\lambda_\eta}{\lambda_\xi}, \qquad r_{\zeta\xi} = \frac{\lambda_\zeta}{\lambda_\xi}, \qquad r_{\zeta\eta} = \frac{\lambda_\zeta}{\lambda_\eta} \tag{4.48}$$

in which $\lambda_\xi, \lambda_\eta$, and λ_ζ are the spectral radii. When the coordinate system is body aligned, the spectral radii are defined consistently with the definition used to evaluate the timestep, equation (2.125). Since the variable smoothing coefficients are effectively functions of local mesh cell aspect ratio, smoothing is not activated in directions where it is not needed and where it would otherwise impede convergence. (Recall that damping of high-frequency errors is optimal at wave speeds near the stability limit). From linear stability analysis, the scheme with the smoothing coefficients presented above (equation 4.44) is stable for all mesh cell aspect ratios when the three-dimensional parameter $\psi \approx 0.0625$ and N/N^* is sufficiently large. There is however, a practical limitation on the Courant number which is placed by the need for effective high-frequency damping. Indeed, for large ν/ν^* the high-frequency damping of the marching scheme vanishes.

To utilize variable aspect ratio smoothing coefficients such as those derived above as a basis for new coefficients that will remove the diffusion limit on the timestep in viscous flows, Turkel, Swanson *et al.* [157] consider the thin-layer Navier-Stokes equations in two-dimensions. In this case, a 2-D variable

smoothing coefficient ε_ξ would be used in the streamwise-like ξ-direction while in the normal η-direction, the smoothing coefficient ε_η would be of a diffusive type in the boundary layer and of a convective type in other regions where viscous effects are negligible. Accordingly, a diffusive smoothing coefficient for the η-direction is derived from Fourier analysis which may be approximated as

$$(\varepsilon_d)_\eta = \frac{1}{4}\bar{C}_1 \frac{(\lambda_d)_\eta}{\lambda_\xi + \lambda_\eta} \tag{4.49}$$

in which λ_d is the diffusive component of the timestep defined as $\lambda_d = \mu\Delta t/\Delta y^2$ and $\lambda_{\xi,\eta}$ are the spectral radii in the directions denoted by subscript. The constant $\bar{C}_1$ is set using numerical experiment. Turkel, Swanson *et al.* note that the variable coefficient of equation (4.49) cannot be used in isolated regions the flow. For example, in flow about an aerofoil, $(\varepsilon_d)_\eta \rightarrow 0$ too fast at the leading-edge, resulting in a zero value in the inviscid region. The difficulty is overcome by calculating ε_η as

$$\varepsilon_\eta = \max\left[(\varepsilon_d)_\eta, \varepsilon_\eta\right] \tag{4.50}$$

in which ε_η is defined by the 2-D equivalent expression to equation (4.44). For 2-D computations at Mach 10 and Mach 20 Turkel, Swanson *et al.* [157] present results in which removal of the diffusion limit by their variable coefficient IRS formulation allowed the residual to be reduced by an additional order of magnitude in 200 multigrid cycles. It is thought that the method can be extended to three-dimensional computations without much difficulty.

Smoothing Strategy for Multi-stage Time Integration Schemes

It has been pointed out by Jameson [75] in reference to Euler solvers that stable schemes for large timesteps can be constructed without using residual smoothing on every stage in a multi-stage cycle. In general it is sufficient to apply smoothing on alternate stages. For a scheme with an even number of stages, smoothing may be applied on stages $2, 4, 6, \ldots$ while for schemes with an odd number of stages, smoothing may be applied on stages $1, 3, 5, \ldots$ This reduces the cost of smoothing without impairing efficiency. Furthermore, in instances when a hybrid multi-stage scheme is used, it is more consistent to apply residual smoothing only on those stages when the dissipation is evaluated. This prevents over-smoothing on stages when the dissipation is not evaluated.

In order to sustain good high-frequency damping and at the same time increase the Courant number further, Implicit-Explicit Residual Smoothing has been developed, the topic of the next sub-section.

4.4.3 Implicit-explicit residual smoothing

It will be remembered from the above discussions that as the Courant number is increased to high values, so damping of oscillatory high-frequency error modes reduces. In practical applications, effective damping of oscillatory modes is crucial since such disturbances are emitted from discontinuities in the solution and from the boundaries of the domain. As a result, the damping requirements of a marching scheme impose a practical limit on the permissible Courant number that may be used in conjunction with IRS since high-frequency damping of the marching scheme vanishes for large ν/ν^*. Indeed, if high-frequency disturbances

are not damped effectively, the numerical scheme will converge slower or even become unstable. To overcome the deteriorated damping of highly oscillatory modes for large ν/ν^*, Enander advanced *Implicit-Explicit Residual Smoothing* (IERS) [40, 41]. In this variant, implicit residual smoothing is combined with an explicit operator. The latter has the effect of sustaining good damping of the most oscillatory modes in a solution. Using familiar notation, in two-dimensions, IERS takes the general form

$$(1-\varepsilon_\xi\delta_\xi^2)(1-\varepsilon_\eta\delta_\eta^2)\bar{R}_m = \mathcal{L}_{e_m}R_m \tag{4.51}$$

being applied at each stage m of a Runge-Kutta cycle. $\mathcal{L}_{e_m}$ is a 2-D explicit operator which is given by

$$\mathcal{L}_{e_m} = \frac{\nu}{\nu^*}\left[1+\gamma\delta_\xi^2\delta_\eta^2-\varphi\delta_\xi^2-\varphi\delta_\eta^2\right] \tag{4.52}$$

After inversion of the implicit operator, the smoothed residual is

$$\bar{R}_m = (1-\varepsilon_\xi\delta_\xi^2)^{-1}(1-\varepsilon_\eta\delta_\eta^2)^{-1}\mathcal{L}_{e_m}R_m \tag{4.53}$$

The parameters γ and φ are chosen to recover the same damping properties as the unsmoothed scheme while permitting high Courant numbers (viz. large ν/ν^*). Both damping parameters are determined from analysis as functions of ν/ν^* and are given by

$$\varphi = \frac{1}{4}\left(\frac{1+4\varepsilon}{\nu/\nu^*}-1\right) \tag{4.54}$$

$$\gamma = \frac{1}{16}\left(\frac{(1+4\varepsilon)^2}{\nu/\nu^*}-8\varphi-1\right) \tag{4.55}$$

Here the implicit smoothing parameters appearing in the equations above are the same for both ξ and η directions. (i.e. $\varepsilon = \varepsilon_\xi = \varepsilon_\eta$). The choice of the φ and γ damping parameters given above, ensures that the smoothed scheme retains similar damping properties to that of the original unsmoothed scheme. However, other choices could be adopted that would change the behaviour of the basic scheme. An important advantage of IERS over IRS is that it is much more robust for non-optimal choices of ν/ν^*. For some cases, choices of ν/ν^* which are $\pm 50\%$ off the optimal value return gains in convergence acceleration that are comparable to the optimal value.

IERS has also been extended to cope with highly stretched grids in a similar fashion to IRS. In this instance, the smoothing parameters appearing in equations (4.51) and (4.53) are different for each direction and are given by

$$\varepsilon_\xi = \max\left\{0, \frac{1}{4}\left[\left(\frac{\nu}{\nu^*}\cdot\frac{1+a^\theta}{1+a}\right)^2-1\right]\right\} \tag{4.56}$$

$$\varepsilon_\eta = \max\left\{0, \frac{1}{4}\left[\left(\frac{\nu}{\nu^*}\cdot\frac{1+\frac{1}{a^\theta}}{1+\frac{1}{a}}\right)^2-1\right]\right\} \tag{4.57}$$

with the range for the exponent θ being $0 < \theta < 1$. A typical value used is $\theta = 0.5$. The explicit damping parameters needed for stretched grid computations depend

on the implicit smoothing parameters and can be chosen as

$$\varphi_\xi = \frac{1}{4}\left(\frac{1+4\varepsilon_\xi}{\nu/\nu^*}-1\right) \tag{4.58}$$

$$\varphi_\eta = \frac{1}{4}\left(\frac{1+4\varepsilon_\eta}{\nu/\nu^*}-1\right) \tag{4.59}$$

$$\gamma = \frac{1}{16}\left(\frac{(1+4\varepsilon_\xi)(1+4\varepsilon_\eta)}{\nu/\nu^*}-4\varphi_\xi-4\varphi_\eta-1\right) \tag{4.60}$$

In 3-D IERS is applied at the mth stage of a Runge-Kutta cycle as

$$(1-\varepsilon_\xi\delta_\xi^2)(1-\varepsilon_\eta\delta_\eta^2)(1-\varepsilon_\zeta\delta_\zeta^2)\bar{R}_m = \mathcal{L}_{e_m}R_m \tag{4.61}$$

with the explicit operator

$$\mathcal{L}_{e_m} = \frac{\nu}{\nu^*}\left[1-\gamma\delta_\xi^2\delta_\eta^2\delta_\zeta^2+\varphi\left(\delta_\xi^2\delta_\eta^2+\delta_\xi^2\delta_\zeta^2+\delta_\eta^2\delta_\zeta^2\right)-\sigma\left(\delta_\xi^2+\delta_\eta^2+\delta_\zeta^2\right)\right] \tag{4.62}$$

The smoothing parameters of the 3-D implicit operator $\varepsilon_\xi, \varepsilon_\eta$ and ε_ζ are chosen for stability while the 3-D damping parameters of the explicit operator γ, φ and σ are tuned for good damping.

Concerning implementation, IERS is applied only inside the computational domain (not overlapping into any *ghost cells*), with the explicit operator imposed one further cell inside the domain than its implicit counterpart. Treatment of the implicit operator at boundaries requires special attention, the details of which are given in [41].

The numerical results that have so far appeared in the open literature for IERS are for 2-D computations [40, 41]. Optimal Courant numbers that are up to twelve times that of an unsmoothed central scheme and 3.3 times that of an unsmoothed upwind scheme have been used effectively with IERS. For transonic and supersonic computations, it is claimed that IERS is typically 20% more efficient than IRS, while in the case of subsonic flow its improvement in efficiency over IRS is significantly higher.

4.4.4 Upwind implicit residual smoothing

Upwind-biased forms of the Implicit Residual Smoothing (IRS) operators have been developed by Blazek *et al.* [6, 7] for cell vertex upwind schemes. In contrast to the central IRS methods, *Upwind Implicit Residual Smoothing* (UIRS) methods use a one-sided operator based on the direction from which hyperbolic information propagates. UIRS allows considerably higher smoothing coefficients than IRS and thereby, higher Courant numbers. This leads to significant improvements of both the damping and convergence behaviour of multi-stage time stepping schemes, particularly for upwind spatial differencing operators. Three variants of UIRS have been developed, namely *Simplified Upwind Smoothing* (SUS), *Improved Upwind Smoothing* (IUS) and, *Full Upwind Smoothing* (FUS). Each method differs in the level of approximation of the upwind smoothing operator, and it is this which determines the computational expense which must be offset against the potential gains in improved convergence. For multi-dimensional problems, residual smoothing is first applied in the ξ-direction and then independently in the remaining direction(s). Each of the three UIRS techniques is now described in turn.

The SUS method is the most basic upwind approximation of UIRS. In the ξ-direction, the two-dimensional smoothing equations takes the form

$$\begin{aligned} -\varepsilon\bar{R}_{i-1,j} + (1+\varepsilon)\bar{R}_{i,j} &= R_{i,j}, \quad M > 1 \\ -\varepsilon\bar{R}_{i-1,j} + (1+\varepsilon)\bar{R}_{i,j} - \varepsilon\bar{R}_{i+1,j} &= R_{i,j}, \quad |M| \le 1 \\ (1+\varepsilon)\bar{R}_{i,j} - \varepsilon\bar{R}_{i+1,j} &= R_{i,j}, \quad M < -1 \end{aligned} \tag{4.63}$$

with similar expressions for the η-direction with the j-counter active. Here the upwind direction is determined simply by the sign of the Mach number M.

The next level up in approximating an upwind-biased operator results in the IUS method, which takes into account the direction of wave propagation separately for each equation. The semi-discrete representation of the 2-D Euler equations may be written as

$$\mathcal{V}_{i,j}\frac{d}{dt}Q_{i,j} + R_{i,j} = 0 \tag{4.64}$$

in which $\mathcal{V}_{ij}$ represents the area of the control volume denoted by index i, j and $Q_{i,j}$ is the vector of dependent variables. The residual $R_{i,j}$ is split to yield

$$\mathcal{V}_{i,j}\frac{d}{dt}Q_{i,j} + R^+_{i,j} + R^-_{i,j} = 0 \tag{4.65}$$

where $R^+_{i,j}$ and $R^-_{i,j}$ respectively represent the residual components corresponding to the positive and negative eigenvalues of the inviscid flux Jacobian A. The positive eigenvalue component is evaluated as

$$R^+_{i,j} = \left(T^{-1}_{i,j}\Lambda^+_{i,j}T_{i,j}\right) R_{i,j} \tag{4.66}$$

with $T^{-1}_{i,j}$ the matrix of left eigenvalues of A, $T_{i,j}$ the corresponding matrix of right eigenvalues, while $\Lambda^+_{i,j}$ is a diagonal matrix representing only positive eigenvalues which have been set to a value of unity. i.e.

$$\Lambda^+ = \text{diag}\left[\lambda^+_1 \quad \lambda^+_2 \quad \lambda^+_3 \quad \lambda^+_4\right] \tag{4.67}$$

where

$$\lambda^+_k = \frac{1+\text{sgn}(\lambda_k)}{2} \tag{4.68}$$

and sgn $(\lambda_k) \equiv \pm 1$. The negative eigenvalue component is evaluated simply as $R^-_{i,j} = R_{i,j} - R^+_{i,j}$. The smoothing is applied independently to each split residual component as

$$-\varepsilon\bar{R}^+_{i-1,j} + (1+\varepsilon)\bar{R}^+_{i,j} = R^+_{i,j} \tag{4.69}$$

$$(1+\varepsilon)\bar{R}^-_{i,j} - \varepsilon\bar{R}^-_{i+1,j} = R^-_{i,j} \tag{4.70}$$

Solution of the above two equations are obtained by backward and forward substitution respectively. Finally, once the positive and negative components of the residual have each been smoothed, they are summed as

$$\bar{R}_{i,j} = \bar{R}^+_{i,j} + \bar{R}^-_{i,j} \tag{4.71}$$

to render the final smoothed residual.

The highest level of approximation used to obtain an upwind-biased operator produces the FUS method. With this method, the conserved variable residuals are transformed into characteristic ones. The characteristic transformation of equation (4.64) in the ξ-direction produces

$$\mathcal{V}_{i,j}\frac{d}{dt}\left(T_{i,j}^{-1}Q_{i,j}\right)+P_{i,j}=0\,,\qquad P_{i,j}=T_{i,j}^{-1}R_{i,j} \tag{4.72}$$

in which $T^{-1}_{i,j}$ is the matrix of left eigenvalues of the flux Jacobian. A similar expression results for the η-direction. The transformed residual is now smoothed using backward or forward one-sided difference operators according to the sign of the eigenvalues λ as

$$-\varepsilon\bar{P}_{i-1,j}+(1+\varepsilon)\bar{P}_{i,j} = P_{i,j}\,,\qquad \lambda\geq 0 \tag{4.73}$$

$$(1+\varepsilon)\bar{P}_{i,j}-\varepsilon\bar{P}_{i+1,j} = P_{i,j}\,,\qquad \lambda<0 \tag{4.74}$$

The resulting tri-diagonal system of equations are solved by LU decomposition. The smoothed residual $\bar{P}$ is then transformed back by

$$\bar{R}_{i,j}=T_{i,j}\bar{P}_{i,j} \tag{4.75}$$

The same procedure is repeated in the η-direction. In each of the UIRS variants, variable smoothing coefficients may be used, derived along the lines adopted in IRS.

The efficacy of each UIRS method increases with the level of approximation of the upwind smoothing operator. Thus the operator based on the eigen-directions (FUS) is superior to the other upwind smoothing methods. However, the additional expense of computing the FUS operator,which involves many more operation counts than the simpler upwind smoothing operators, must be offset against the speed-up in convergence acceleration that it produces. When this is done, the FUS operator is not necessarily the best choice in terms of overall computational efficiency. Two-dimensional numerical results for transonic, supersonic and hypersonic flows using upwind smoothing operators have been reported by Blazek *et al.* [6, 7], and Rossow [135] for schemes employing both scalar and matrix based dissipation. In addition, Grasso and Marini [58] have presented numerical results for hypersonic flows obtained using SUS and FUS with schemes based on scalar dissipation models. It is apparent from these results that UIRS methods are more beneficial to the convergence of methods using upwind spatial discretizations than ones using centred discretizations. With upwind methods that utilize UIRS, Courant numbers as high as 10-15 have been reported. Furthermore, UIRS has been used together with multigrid to produce substantially accelerated convergence [135, 58].

4.5 Optimal Smoothing Multi-stage Time Stepping Schemes

Design of a time discretization by tailoring it to match the chosen spatial discretization can significantly enhance the rate of convergence of an explicit time marching method in steady-state computations. This is the motivation behind

the development of optimal smoothing multi-stage time stepping schemes. To formulate these schemes, optimal coefficients and Courant numbers are determined which maximize high-frequency damping and which increase the permissible timestep of a given m-stage scheme. The optimal parameters are determined via optimization procedures based on Fourier methods applied to model hyperbolic equations. Since optimal smoothing schemes are designed to tackle high-frequency error modes, they are ideally suited for use in conjunction with multigrid since the latter is efficient at eradicating the lower-frequency errors. However, such methods are sufficiently efficacious to be used in their own right to accelerate the convergence of single-grid methods. Furthermore, they are readily implemented in existing computer codes that use Runge-Kutta schemes.

van Leer, Tai and Powell [171, 111] and Lallemand [82] have demonstrated how to develop multi-stage time integration methods that yield optimal damping of high-frequency modes. Their schemes are derived for scalar convection equations in one space dimension, and have been used to solve the Euler equations in both one-dimensional and higher dimensional problems. A two-dimensional analysis for optimizing the smoothing properties of multi-stage schemes was presented by Catalano and Deconinck [16] for a scalar convection equation. Recently, a more extensive two-dimensional scalar analysis was carried out by Lynn and van Leer [92] which was extended to the preconditioned system of Euler equations based on the local preconditioning of van Leer *et al.* [170]. This approach became possible as a consequence of the breakthrough in preconditioning algorithms by van Leer, Lee and Roe [169] which makes the system behave more like a scalar equation by reducing the spread among the characteristic speeds as much as possible, see section 4.7.2. However, the optimal smoothing schemes so obtained are suitable only for marching methods which use that particular preconditioning.

In what follows, the one-dimensional van Leer, Tai and Powell [171, 111] optimization will be examined and the smoothing coefficients and Courant numbers obtained from it will be given for some second-order upwind multi-stage schemes. After that, the Lallemand [82] optimization will be discussed briefly and the optimal parameters obtained from it will also be presented for two second-order upwind schemes. Then optimal multi-stage schemes which have been developed specifically for use with residual smoothing will be discussed. Finally an optimal hybrid multi-stage scheme will be given.

4.5.1 The van Leer–Tai–Powell optimization

The procedure for optimizing the high-frequency damping in a one-dimensional convection scheme adopted in this optimization approach [171, 111], has been described as a geometry exercise in the complex plane. It consists of placing the zeros of the multi-stage amplification factor on the locus of the Fourier transform (the Fourier footprint) of the discrete spatial operator. This is done for one specific value of the timestep. Finding this value of the timestep is also part of the design process. The analysis begins with consideration of the linear ordinary differential equation

$$\frac{du}{dt} = \lambda u, \qquad \lambda \in C \tag{4.76}$$

A predictor-corrector integration method is used to discretize the temporal term as

$$\begin{aligned} \bar{u} &= u^n + \alpha\lambda\Delta t u^n \\ u^{n+1} &= u^n + \lambda\Delta t\bar{u} \\ &= \left[1 + \lambda\Delta t + \alpha(\lambda\Delta t)^2\right] u^n \end{aligned} \tag{4.77}$$

where the time-step ratio α is a free parameter. The stability and damping properties of the discretization are determined by its complex polynomial

$$P_2(z,\alpha) = 1 + z + \alpha z^2, \qquad z = \lambda\Delta t \tag{4.78}$$

The polynomial has two complex-conjugate roots (zeros), namely $z_1(\alpha)$ and $z_2(\alpha) = z_1^*(\alpha)$, with

$$z_1(\alpha) = -\frac{1}{2\alpha} - \frac{i}{2\alpha}\sqrt{4\alpha - 1} \tag{4.79}$$

These roots may be moved along the circle

$$[\Re(z_0 + 1)]^2 + [\Im(z_0)]^2 = 1 \tag{4.80}$$

by varying α.

If a partial differential equation is interpreted by the method of lines, λ represents the Fourier transform of the spatial differencing operator and depends on the spatial frequency ξ, or more specifically, on the spatial wave number $\beta = 2\pi\Delta x$. For instance, when solving the convection equation

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}, \qquad c > 0 \tag{4.81}$$

discretization of the spatial derivative by first-order upwind differencing produces

$$\Delta t\frac{\partial u}{\partial t} = -\nu\left[u(x,t) - u(x - \Delta x, t)\right] \tag{4.82}$$

in which the non-dimensional timestep

$$\nu = c\frac{\Delta t}{\Delta x} \tag{4.83}$$

is the customary Courant number. Insertion of the harmonic data

$$u(x) = u_0 e^{2\pi i\xi x} \tag{4.84}$$

into equation (4.82) renders

$$\Delta t\frac{\partial u}{\partial t} = -\nu(1 - e^{-i\beta})u_0 e^{-2\pi i\xi x} \tag{4.85}$$

This equation has the same form as equation (4.76) with

$$\lambda\Delta t \equiv z(\beta,\nu) = -\nu\left(1 - e^{-i\beta}\right) \tag{4.86}$$

Furthermore, all the information about the spatial differencing operator is included in the above function.

At this point it should be observed that for any β_0 in the high-frequency range $[\pi/2, \pi]$, it is possible to make $z(\beta_0, \nu) \equiv \lambda(\beta_0)\Delta t$ coincide with a zero of $P_2(z, \alpha)$, by choosing a particular combination of α and Δt. This results in perfect damping of the wave of wave number β_0 in one application of the predictor-corrector scheme. Through the use of strings of predictor-corrector schemes, tuned to damp different frequencies, the entire high-frequency range can be damped to arbitrarily low levels.

Clearly strings of predictor-corrector schemes generate multi-stage methods with an even number of stages. For an odd number of stages, a single application of the forward-Euler scheme

$$\begin{aligned} u^{n+1} &= u^n + \Delta t \lambda u^n \\ &= (1 + \lambda \Delta t) u^n \end{aligned} \tag{4.87}$$

is added to the string. The forward-Euler step has an amplification factor

$$P_1(z) = 1 + z \tag{4.88}$$

This polynomial has one zero at

$$z = -1 \tag{4.89}$$

It should also be observed that, for any fixed number of stages, there is an optimum scheme, in the l_∞ sense, that reduces all of the high frequencies to an amplitude not exceeding a unique minimum threshold level.

In the case of a general spatial-differencing operator, whether convective, diffusive or a combination of both, the Fourier transform can be written as

$$\lambda \Delta t = z(\beta, \nu) = \nu \left[a(\beta) + i b(\beta) \right] \tag{4.90}$$

where ν is a non-dimensional timestep. (In the framework of convective equations ν is the Courant number; for present purposes it shall continue be referred to as such). Considering the two-stage scheme associated with the polynomial (4.78), for a particular frequency β_0 to be perfectly damped, it is necessary to set $z_1(\alpha) = z(\beta_0, \nu)$, i.e.

$$-\frac{1}{2\alpha} - \frac{i}{2\alpha}\sqrt{4\alpha - 1} = \nu \left[a(\beta) + i b(\beta) \right] \tag{4.91}$$

This has the solution

$$\alpha_0 = \frac{a_0^2 + b_0^2}{4a_0^2} \tag{4.92}$$

$$\nu_0 = \frac{2\,|a_0|}{a_0^2 + b_0^2} \tag{4.93}$$

where $a_0 \equiv a(\beta_0)$, $b_0 \equiv b(\beta_0)$. Now the single-step forward-Euler scheme can only damp the frequency for which the Fourier transform, equation (4.90), is real-valued. For any finite-difference operator this means

$$\beta_0 = \pi \tag{4.94}$$

with the corresponding Courant number

$$\nu = \frac{1}{|a_0|} \tag{4.95}$$

For a first-order upwind differencing operator, optimization of the high-frequency damping in a string of predictor-corrector and single-step operators can be done analytically [171]. However, for more complex differencing operators this is not feasible and it is then necessary to adopt an iterative method to minimize the maximum of the amplification factor in the high-frequency range [111]. This has been done to find the first six multi-stage schemes for van Leer's family of upwind-biased *kappa schemes* [165]. The Fourier transform of the kappa spatial differencing operator is

$$\lambda\Delta t = z(\beta) = \nu\left(1-e^{-i\beta}\right)\left\{1+\frac{1}{4}\left[(1-\kappa)\left(1-e^{-i\beta}\right)+(1+\kappa)\left(e^{-i\beta}-1\right)\right]\right\} \tag{4.96}$$

Here, the parameter κ regulates the upwind-bias as follows:

$\kappa = 1$	central differencing
$\kappa = -1$	second-order accurate –full upwinding
$\kappa = 1/3$	third-order accurate upwind-biased

The key parameters that specify each of the optimally smoothed kappa schemes are given in the literature [171, 111]. Those for the fully upwinded second-order scheme are listed in Table 4.1. The quantities α_k given in the table are not the timestep ratios of the constituent predictor-corrector schemes described above, but rather the timestep ratios arising from the practical implementation of an m-stage scheme such as that given by equations (2.113), viz.

$$\begin{array}{lcll} \mathbf{Q}^{(0)} & \stackrel{\text{def}}{=} & \mathbf{Q}^{(n)} & \\ \mathbf{Q}^{(k)} & = & \mathbf{Q}^{(0)} + \alpha_k \frac{\Delta t}{\mathcal{V}}\mathbf{R}[\mathbf{Q}^{(k-1)}] & k = 1, m \\ \mathbf{Q}^{n+1} & \stackrel{\text{def}}{=} & \mathbf{Q}^{(m)} & \end{array}$$

Also given in the table are the optimal Courant numbers ν_{opt}, obtained from the analysis. Note that the mth step always spans the full time interval Δt, such that $\alpha_m = 1$. The amplification factor of the above scheme can be written as

$$P_m(z) = 1 + z(1 + \alpha_{m-1}z(1 + \alpha_{m-2}z(\ldots(1 + \alpha_2 z(1 + \alpha_1 z))\ldots))) \tag{4.97}$$

where z now corresponds to the Courant number for the full time interval. The coefficients are thus found by multiplying out the string of polynomials of the form of equations (4.78) and (4.88), and re-scaling z such that the linear term has a coefficient of unity.

Stage	ν_{opt}	α_1	α_2	α_3	α_4	α_5	α_6
2	0.4693	0.4243	1				
3	0.6936	0.1919	0.4930	1			
4	0.9214	0.1084	0.2601	0.5051	1		
5	1.1507	0.0694	0.1603	0.2898	0.5067	1	
6	1.3806	0.0482	0.1085	0.1884	0.3049	0.5062	1

Table 4.1: Tai multi-stage coefficients and Courant numbers for optimal second-order upwind schemes.

It may be observed from Table 4.1 that the higher the number of stages in a scheme, the greater its Courant number, and the better its damping of high-frequency errors. However, it is noted that the optimal Courant number for an m-stage scheme is considerably lower than its maximum stable Courant number. This makes optimal multi-stage schemes very robust.

4.5.2 The Lallemand optimization

Optimal smoothing schemes have also been devised by Lallemand and their efficiency compared with standard Runge-Kutta schemes [82] on unstructured grids. (The optimal parameters so determined are equally applicable to structured methods). The optimizations devised differ depending upon whether the spatial discretization is to be used for a single-grid method or for a multigrid method. The criterion adopted for optimization of a single-grid scheme is simply to find the coefficients α_k such that the stability domain is as large as possible, i.e. so as to maximize the magnitude of the Courant number. The criterion for multigrid methods is more severe and in addition to satisfying the maximum Courant number criterion, it is also necessary to optimize the high-frequency damping properties. Such optimizations have been implemented for four-stage schemes, namely first- and second-order upwind algorithms, used as single-grid and multigrid solvers. The Lallemand coefficients for the second-order schemes are given in Table 4.2.

Scheme	ν_{opt}	α_1	α_2	α_3	α_4
single-grid	2.0763	0.12	0.26	0.5	1
multigrid	1.9185	0.11	0.2766	0.5	1

Table 4.2: Lallemend multi-stage coefficients and Courant numbers for optimal four-stage second-order upwind schemes.

In each of the Lallemand optimizations for both single-grid and multigrid, the α_3 coefficient is fixed in advance to be $\alpha_3 = 1/2$. This ensures that the corresponding R-K method is second-order accurate in time both for linear and nonlinear problems. In contrast, the van Leer-Tai-Powell optimization does not impose this restriction. This is probably because the notion of time accuracy is not deemed important for steady-state computations and, by allowing the optimization of each coefficient with the exception of the last coefficient which is always set at $a_m = 1$ for consistency, there is a greater degree of freedom to achieve optimal high-frequency damping.

4.5.3 Optimal schemes for residual smoothing

Optimal smoothing multi-stage schemes have also been devised for solution of the two-dimensional Euler equations specifically for use with Implicit and Explicit Residual smoothing operators based on *fixed* smoothing coefficients, by Tai, Sheu and van Leer [151] . Originally, Tai *et al.* applied the van Leer-Tai-Powell optimization procedure to a one-dimensional IRS upwind operator. Subsequently they discovered that although the maximum of the amplification factor in the high-frequency range was indeed minimized for each scheme devised, its magni-

tude in the low-frequency range lay outside the stability region for smoothing coefficients $\epsilon > 0.3$. In fact, the instability became worse as the IRS smoothing coefficient was increased. This prompted the development of a modified procedure, namely, to keep the coefficients from the one-dimensional van Leer-Tai-Powell optimization without residual smoothing and then search for the optimal Courant number that minimizes the area under the high-frequency amplification factor curve while satisfying the stability condition for all low-frequencies. This approach produced the desired result of consistently maintaining stability, with the Courant number increasing with increasing smoothing coefficient. In this manner, the optimal Courant number becomes a function of the smoothing coefficient. In order to extend the procedure to two-dimensional problems, the practical strategy adopted is to maintain the coefficients obtained from the modified one-dimensional analysis and to redefine the Courant number for two-dimensional equations. This is accomplished using simple geometric analysis and it has been shown to be very robust in producing good damping properties for two-dimensional problems.

4.5.4 Optimal hybrid multi-stage schemes

Hybrid multi-stage schemes have already been discussed in section 2.6.3. Radespiel and Swanson [119] report that Tai has carried out a multi-stage optimization procedure to derive an *optimal hybrid multi-stage scheme* for higher-order upwind operators. This particular scheme is a 5/3 one with a Courant number, $\nu_{opt} = 2.4$. Its coefficients α_k and weightings β_k, are listed in Table 4.3.

Stage(k)	1	2	3	4	5
α_k	0.2742	0.2067	0.5020	0.5142	1
β_k	1	0	0.56	0	0.44

Table 4.3: Tai multi-stage coefficients and weightings for optimal 5/3 hybrid upwind scheme.

Note that this scheme requires only three evaluations of the dissipative fluxes which take the majority of floating-point operations in upwind algorithms.

4.6 Time Inclining

Originally Giles [47] formulated the technique of *time inclining* for the Euler equations in problems involving unsteady flows in turbo-machinery. The motivation arose from the need to satisfy a lagged periodic boundary condition encountered in rotor/stator interactions. However, the technique demonstrated an incidental benefit of accelerating convergence, and was subsequently applied to accelerating convergence of time marching methods for more general application [28]. The technique can be applied to most time marching algorithms. So far it has been used with Ni's Lax-Wendroff algorithm and van Leer's Flux Vector Splitting scheme. It will be described below with reference to one-dimensional problems after which some explanation will be given as to how it may be formulated in higher dimensions.

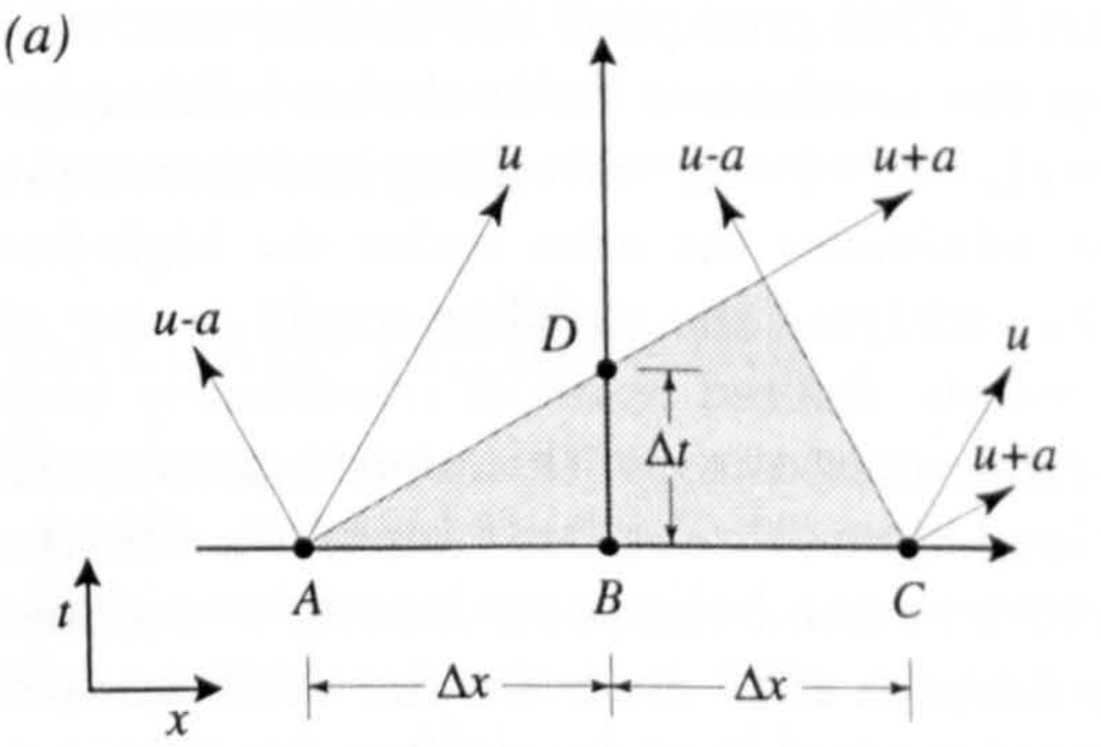


FIGURE 4.3. Time inclining. Geometric interpretation of the CFL condition. (a) Normal non-inclined computational plane and (b) Inclined computational plane.

The steady-state solution of a time marching scheme is obtained when all the errors associated with the initial conditions have propagated out of the solution domain. Since all waves in a hyperbolic problem do not propagate at the same rate, convergence to the steady-state is restricted by the slowest moving waves and the errors associated with them. From a physical perspective, time inclining may be viewed as a local inclination of time in the space-time (x-t) plane. By inclining each time level, nodes at the same computational time level are actually at different physical times. The effect is to modify the relative speed of the characteristics such that given an appropriate inclination, convergence may be enhanced. The argument is illustrated geometrically in Figure 4.3. Part (a) of the figure shows the normal (non-inclined) computational plane for given subsonic flow conditions. Three computational nodes A, B and, C are shown along the horizontal axis at time level n. Emanating from nodes A and C are the characteristic trajectories of each wave. According to the CFL condition, any point within the shaded region may be stably computed, given just the conditions between A

and C. The maximum stable timestep Δt allowable for updating the solution to the next time level corresponds to point D where the shaded region intersects the abscissa. During this time increment, the downsteam $(u+a)$ pressure wave moves one computational cell Δx whereas the entropy wave (u) and upsteam $(u-a)$ pressure wave each move $\Delta x/3$. The distance travelled by each being in direct proportion to the ratio of its speed to that of the fastest wave speed $(u+a)$. For the same flow conditions, but with the computational time plane now inclined as shown in part (b) of the figure, it is apparent that the $u-a$ wave propagates at only a slightly slower rate than the $u+a$ wave. Hence, the disparity between the fastest and slowest wave speeds is reduced, the CFL restriction improves and, a larger timestep may be chosen. A similar benefit in convergence can be demonstrated for supersonic flow. From the aforesaid, it is apparent that time inclining has much in common with preconditioning methods, the subject of section 4.7. In fact it has been described as a simplified form of preconditioning [23].

Mathematically the process implies a coordinate transformation from the physical (x,t) space to that of computation (x',t'). That is

$$x' = x \qquad t' = t - \lambda x \tag{4.98}$$

with the inverse given by

$$x = x' \qquad t = t' + \lambda x' \tag{4.99}$$

The inclination parameter λ is defined as $\lambda \equiv dt/dx$. Recall that the unsteady one-dimensional Euler equations in physical coordinates may be written in conservation form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{4.100}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad \mathbf{F(w)} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ e(u+p) \end{bmatrix}$$

In the transformed coordinates, equation (4.100) becomes

$$\frac{\partial \mathbf{Q}}{\partial t'} + \frac{\partial \mathbf{F}}{\partial x'} = 0 \tag{4.101}$$

in which $\mathbf{Q} \equiv \mathbf{U} - \lambda \mathbf{F}$. For a perfect gas, closed form relations between $\mathbf{U}$ and $\mathbf{Q}$ are readily obtained. Implementation of time inclining varies from method to method. Some implementation details for Ni's scheme are given in the literature [47, 28]. Since the domain of dependance changes through inclination of the time planes, the numerical stability of the modified scheme must be reconsidered. The speeds of the three characteristic waves are now governed by

$$\left.\frac{\Delta t}{\Delta x}\right|_{u+a} = \frac{1}{u+a} - \lambda \tag{4.102}$$

$$\left.\frac{\Delta t}{\Delta x}\right|_{u} = \frac{1}{u} - \lambda \tag{4.103}$$

$$\left.\frac{\Delta t}{\Delta x}\right|_{u-a} = \frac{\mp 1}{u-a} \pm \lambda \tag{4.104}$$

where the upper and lower signs of the last equation refer to subsonic and supersonic conditions, respectively. It follows that the maximum permissible timestep is

$$\Delta t_{max} = \Delta x \min\left\{\frac{1}{u+a} - \lambda, \frac{\mp 1}{u-a} \pm \lambda\right\} \tag{4.105}$$

For subsonic flow, the stability limit depends on the fastest signal speeds in both upstream and downstream directions and hence the maximum or optimal value of the inclination parameter λ occurs when the magnitude of signal speeds are equal. This yields

$$\lambda_{opt}\,|_{subsonic} = \frac{1}{a}\left[\frac{M}{M^2-1}\right] \tag{4.106}$$

while

$$\begin{aligned}\Delta t_{opt}\,|_{subsonic} &= \frac{1}{2}\Delta x\left[\frac{1}{u+a} - \frac{1}{u-a}\right] \\ &= \frac{\Delta x}{a}\left[\frac{1}{1-M^2}\right]\end{aligned} \tag{4.107}$$

In supersonic flow, all the characteristics travel in the same direction and the timestep is always limited by the fastest $(u+a)$ wave. By considering the supersonic branch of equation (4.105), it may be seen that the maximum timestep occurs in the limit as

$$\lambda_{opt}\,|_{supersonic} \to -\infty \tag{4.108}$$

yielding

$$\Delta t_{opt}\,|_{supersonic} \to \infty \tag{4.109}$$

Thus, no matter how large the timestep, in the supersonic case it is possible to find a value of λ which will result in stable integration.

Time inclining may be applied to higher dimensions by extending the concepts already considered. In two-dimensions, characteristic analysis indicates that the timestep is restricted both by pressure waves propagating in the streamwise direction and, pressure waves propagating normal to the flow direction. As a consequence, the two inclination parameters which are necessary for time inclination in two-dimensions need to be determined in terms of intrinsic orthogonal coordinates (s,n) where s is the local streamwise direction and n its normal direction. This results in the following expressions for the inclination parameters in the new coordinate system

$$\begin{aligned}\lambda_s &= (+u\lambda_x + v\lambda_y)/q \\ \lambda_n &= (-v\lambda_x + u\lambda_y)/q\end{aligned} \tag{4.110}$$

in which subscripts refer to coordinate direction, $\lambda_x = \Delta t/\Delta x, \lambda_y = \Delta t/\Delta y$ and, $q = (u^2+v^2)^{\frac{1}{2}}$. Restrictions on the size of timestep are more complicated than for the one-dimensional case. For further details, refer to [28]. In addition, a problem arises in the case of transonic flows, due to the large disparity between the magnitude of the optimum inclination parameter for adjacent subsonic and supersonic regions of flow. This can yield very warped pseudo-time planes, which may result in errors of the order of Δx. To overcome this, the inclination parameter λ_s should be smoothed to produce a less distorted distribution over the

domain. Details of the smoothing used are not given but it is reported that it does not affect the steady-state solution and that it actually improves convergence further.

One-dimensional results demonstrate that computation using optimal time inclining requires less than half the number of iterations to converge compared to the unmodified scheme, but with an apparent increase in computational work per timestep of approximately 15%. In two-dimensions, the benefits to be gained from time inclining depend on the Mach Number and the grid aspect ratio. It works best for flows in the range ($0.5 < M < 2$) and for flows in which the timestep is limited by streamwise waves rather than those in crossflow. However, the effectiveness of time inclining diminishes in discretizations with very high aspect ratio cells such as those typically used within the boundary layer of Navier-Stokes computations. It is suggested by Giles that this problem could be circumvented by using other techniques near wall regions such as semi-implicit formulations, in tandem with time inclining. The method may be used to accelerate time accurate flows provided the inclination parameter and timesteps are held constant throughout the domain. Naturally this restriction reduces the benefit of the method and clearly the advantage of an improved rate of convergence should be balanced against the increased computational work. Giles suggests that extension of the technique to three-dimensions should be straightforward and that it is easy to retrofit into existing explicit and implicit codes.

4.7 Local Preconditioning

Hyperbolic systems of equations become stiff when the speeds of different waves in the solution become significantly disparate. When this happens, the characteristic condition number becomes large, viz. $\sigma >> 1$. The occurrence of stiffness depends on the local flow conditions. However, it is particularly a problem in regions where the singularities $M \rightarrow 0$ or $M \rightarrow 1$ exist. The former can arise in low Mach number flow which is almost incompressible, and in stagnation flow. The latter arises particularly in transonic flow containing shock waves and sonic expansions. In such instances the backward moving acoustic wave $(u - a)$ can be very small in magnitude causing very slow shock motion. In these situations, the convergence and in some instances the accuracy of a numerical solution may deteriorate. Indeed, should one try an acceleration technique such as multigrid, it is found that the same disparity in wave speeds slows down the multigrid process. Stiffness may occur with both explicit and implicit schemes, but it is more of a problem with the former. To improve the convergence of stiff problems it is necessary to change the transient nature of the system to remove or reduce the disparities in wave speeds. Methods which are designed to do this are known as preconditioning methods since their aim is to reduce the condition number.

The artificial compressibility method of Chorin [24] is not strictly classed as a preconditioning method, but it was out of this method that preconditioning methods for the steady incompressible flow were developed. In Chorin's method an artificial derivative of pressure is added to the continuity equation together with a multiplicative variable β. With this artificial term the resultant scheme becomes a symmetric hyperbolic system . As the system is well-posed, numerical methods for hyperbolic systems may be applied to advance the system in time.

The parameter β may then be adjusted to optimize convergence. Turkel [155] advanced the idea that similar terms should be added to the momentum equations. The resulting system is no longer symmetric but can be symmetrized by a change of variables. The resulting system is equivalent to a preconditioned form of Chorin's equations. The preconditioned systems for incompressible flow were then generalized by Turkel for low-speed compressible flows. The two-parameter family of preconditioning matrices derived by Turkel are general enough to be effective in the transonic regime, though this was not apparently recognized at the time.

Viviand [179] devised preconditioning methods which deal with the entire range of Mach number, starting from the iso-energetic Euler equations. This system is hyperbolic and results from the assumption of homogeneous total specific enthalpy (stagnation enthalpy). In his analysis, Viviand considered a four-parameter family of preconditionings. However, his primary concern was to find preconditioned systems in full conservation form, or ones which have simple expressions for the characteristic speeds. In the equations he derived, the maximum characteristic speed equals the flow speed, but the range of characteristic speeds is not discussed and no numerical results are presented.

Recently van Leer, Lee and Roe have made significant progress in developing preconditioning methods for the Euler equations. Their work [169] first emerged in 1991 when a local preconditioning matrix was presented which dramatically reduces the spread among the characteristic speeds. It achieves what can be shown to be the optimal condition number for the characteristic speeds. Important work on preconditioning the Navier-Stokes equations has been carried out by Choi and Merkle [23], Lee and van Leer [85], Godfrey, Walters and van Leer [52] and Godfrey [51]. A useful review of preconditioning methods is given by Turkel [156].

In what follows, the connection between the different ways of time stepping and local preconditioning will be considered after which preconditioning of the one-dimensional Euler equations will be discussed as the basis of characteristic time stepping. This introduces some of the notions behind preconditioning in higher-dimensions. The two-dimensional local preconditioning method of van Leer, Lee and Roe [169] will then be described.

4.7.1 Choices for time stepping

The concept of local preconditioning may be placed in perspective by considering it as a way of choosing the timestep for a time marching method. Typically there are three distinct choices of timestep. Two have already been defined in section 2.6.4, namely global or fixed timesteps, and local timesteps. It will be recalled that with global time stepping, all cells within the computational domain are updated using the same value of Δt. This method of time stepping is normally used only for time-accurate computations. In contrast, with local time stepping, each cell has its own timestep which is scaled by the maximum eigenvalue that will ensure local stability. This produces a constant Courant number for all cells in the domain and accelerates convergence to the steady-state. Although time accuracy is lost, the steady solution is unaffected. The use of a local timestep is equivalent to preconditioning the residual by a different scalar value for each cell. It removes some stiffness due to spatial variations in the wave speed and

cell size.

The third approach to choosing a timestep is readily carried out in one-dimension and relies on diagonalizing the flux Jacobian. This transforms the governing equations from conserved variables to characteristic variables such that each equation in the transformed system describes the motion of a particular wave. On returning to the original variables, this is found to be equivalent to multiplying the time derivatives by a matrix. Hence this approach is called characteristic time stepping [169]. With multi-dimensional equations, the waves cannot be completely decoupled. Consequently, in higher dimensions the notion of characteristic time stepping is not as precise as it is in one-dimension. Characteristic time stepping removes stiffness due to variation in the characteristic speeds. Time-accuracy is further sacrificed but convergence to the steady-state, especially in subsonic and transonic solutions may be improved significantly. For maximum effect, characteristic time stepping may be used in tandem with local time stepping.

4.7.2 Characteristic time-stepping

The convergence of an iterative scheme may be accelerated essentially by multiplying the time derivative by a matrix, but in such a way that it does not alter the steady-state solution. The motivation for this becomes apparent on examination of the one-dimensional Euler equations. Instead of solving the usual linearized time-dependent problem

$$\mathbf{Q}_t + \mathbf{A}\mathbf{Q}_x = 0 \tag{4.111}$$

the preconditioned problem

$$\mathbf{Q}_t + \mathbf{P}\mathbf{A}\mathbf{Q}_x = 0 \tag{4.112}$$

may be considered, in which $\mathbf{P}$ is a preconditioning matrix. For computation on highly stretched mesh, this speeds up the arrival of transient disturbances or error waves at the external boundaries. Through appropriate choice of $\mathbf{P}$, the relative speeds of different types of wave can be altered such that the slowest ones are speeded up. For the above equation, this is done by taking $\mathbf{P}$ to be $|\mathbf{A}|^{-1}$ so that

$$\mathbf{Q}_t + |\mathbf{A}|^{-1}\mathbf{A}\mathbf{Q}_x = 0 \tag{4.113}$$

The diagonal factorization of $|\mathbf{A}|$ is given by $|\mathbf{A}| = \mathbf{R}|\Lambda|\mathbf{R}^{-1}$ where $\mathbf{R}$ is the matrix of right eigenvalues and $\mathbf{R}^{-1}$ is that of the left eigenvalues. The diagonal matrix $|\Lambda|$ contains the absolute value of the wave speeds and is given by

$$|\Lambda| = \text{diag}\,[|\lambda_1|, |\lambda_2|, |\lambda_3|] \tag{4.114}$$

$$= \text{diag}\,[|u-a|, |u|, |u+a|] \tag{4.115}$$

Applying this diagonalization, the system of equations to be solved becomes

$$\begin{aligned} \mathbf{Q}_t + (\mathbf{R}|\Lambda|\mathbf{R}^{-1})(\mathbf{R}\Lambda\mathbf{R}^{-1})\mathbf{Q}_x &= 0 \\ \mathbf{Q}_t + \left[\mathbf{R}\,\text{sgn}\,(\Lambda)\mathbf{R}^{-1}\right]\mathbf{Q}_x &= 0 \qquad (4.116) \\ \mathbf{R}^{-1}\mathbf{Q}_t + \left[\text{sgn}\,(\Lambda)\mathbf{R}^{-1}\right]\mathbf{Q}_x &= 0 \end{aligned}$$

Now $\mathbf{R}^{-1}d\mathbf{Q}$ defines the characteristic variables which are given by $\mathbf{C} = [c_1\ c_2\ c_3]^T$. The components of $\mathbf{C}$ represent the upstream-running acoustic or pressure wave, an entropy wave and a downstream-running acoustic wave, respectively. Thus, the system may be re-written as

$$\mathbf{C}_t + \operatorname{sgn}\Lambda\,\mathbf{C}_x = 0 \tag{4.117}$$

Because Λ is diagonal this equation is an ordinary differential equation for $\mathbf{C}$ along the characteristic curves defined by $dx/dt \equiv \operatorname{sgn}(\lambda_k),\ k = 1,3$. Furthermore, the components c_k are constant along the corresponding characteristic curves and can be regarded as travelling waves that propagate at the characteristic velocities of the system. Now since the entries of the diagonal matrix $\operatorname{sgn}\Lambda$ are the signs (± 1) of the wave speeds, it is evident that in the modified problem, the characteristic variables are propagated with speed ± 1. In addition, as the directions of wave propagation are conserved (i.e. waves in the original system which were left- or right-going retain their direction of propagation), the problem is still well-posed under the same boundary conditions. However, a disturbance of any kind starting in the middle of the domain will now reach all boundaries in equal time.

The definition of the characteristic condition number given earlier to quantify the stiffness of a system, may be written as

$$\sigma = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{4.118}$$

From this definition, the condition number of the Euler equations for the original problem is

$$\sigma = \frac{1 + |M|}{\min[|M|,\ 1 - |M|]} \tag{4.119}$$

and for the flow singularities which occur as $M \to 0, 1$, so $\sigma \to \infty$. Clearly these are the most difficult computational cases. For the preconditioning given by equation (4.117), $\sigma \equiv 1$ for all Mach numbers. However, this result holds only for the one-dimensional system.

The simplicity of developing the above preconditioning is attributable to the fact that the one-dimensional Euler equations are readily diagonalized. However, for multidimensional problems, devising a preconditioning becomes more complex because the equations cannot be simultaneously diagonalized except in the case of supersonic flow. In the work of van Leer, Lee and Roe [169], a two-dimensional preconditioning matrix is devised by first transforming the conserved variables to ones which facilitate a symmetric form for the linearized equations. Then a rotational transformation is carried out to align the original Cartesian coordinate system with the flow direction. The resulting equations become

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_s + \mathbf{B}\mathbf{U}_n = 0 \tag{4.120}$$

in which the subscripts represent partial derivatives, with s denoting the streamwise direction and n the direction normal to the streamlines. In addition,

$$d\mathbf{U} = \begin{bmatrix} dp/\rho a \\ du \\ dv \\ dp - a^2 d\rho \end{bmatrix} \tag{4.121}$$

and

$$\mathbf{A} = \begin{bmatrix} q & a & 0 & 0 \\ a & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & q \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.122}$$

Here q is the magnitude of the velocity in the streamwise direction which is determined as $q = u\cos\theta + v\sin\theta$. The variable θ is the angle through which the original Cartesian coordinate system is rotated to obtain the stream-aligned system.

Now first consider flow which is fully supersonic. In this case $|\mathbf{A}^{-1}| = \mathbf{A}^{-1}$, and multiplying the residual of equation (4.120) by $|\mathbf{A}^{-1}|$ gives

$$\mathbf{U}_t = -\left(\mathbf{I}\,\mathbf{U}_s + \mathbf{A}^{-1}\mathbf{B}\,\mathbf{U}_n\right) \tag{4.123}$$

Since the flow is supersonic, this equation may be simultaneously diagonalized using the transformation $d\mathbf{V} = \mathbf{T}^{-1}d\mathbf{U}$ to produce

$$\mathbf{V}_t = -\left(\mathbf{I}\,\mathbf{V}_s + \Lambda\mathbf{V}_s\right) \tag{4.124}$$

where Λ is the diagonal matrix of the eigenvectors of $\mathbf{A}^{-1}\mathbf{B}$ and $\mathbf{T}, \mathbf{T}^{-1}$ are the diagonalizing matrices. In this equation, the wave speeds in the streamwise direction are equalized. However, the disparity in wave speed between the acoustic waves in the direction normal to the streamlines is amplified so that the ratio of the acoustic wave speeds to the convective wave speed is given by $r_\lambda = M/\sqrt{M^2-1}$. In order to make the convective speeds equal, the acoustic waves must be scaled. This is done by multiplying the right-hand side of equation (4.123) by the matrix $\mathbf{X}$, given by

$$\mathbf{X} = \begin{bmatrix} \tau & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \tau \end{bmatrix} \tag{4.125}$$

in which $\tau = 1/r_\lambda$ is the scaling factor. Converting back to symmetry variables, the preconditioned system of equations in stream-aligned coordinates becomes

$$\mathbf{U}_t = -\mathbf{P}\left(\mathbf{A}\,\mathbf{U}_s + \mathbf{B}\,\mathbf{U}_n\right) \tag{4.126}$$

where

$$\mathbf{P} = \mathbf{T}\mathbf{X}\mathbf{T}^{-1}\mathbf{A}^{-1} = \begin{bmatrix} \frac{\tau}{\beta^2}M^2 & -\frac{\tau}{\beta^2}M & 0 & 0 \\ -\frac{\tau}{\beta^2}M & \frac{\tau}{\beta^2}+1 & 0 & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.127}$$

and

$$\begin{aligned} \beta &= \sqrt{M^2-1} \\ \tau &= \sqrt{1-M^{-2}} \end{aligned}, \qquad M \geq 1 \tag{4.128}$$

Now consider the case of subsonic flow. Here difficulties arise because the equations cannot be uncoupled by diagonalization as in the supersonic case. However, by assuming that the subsonic preconditioning matrix should have a similar structure to equation (4.127), it may be obtained through the requirement

that the convective waves are unchanged and that the acoustic waves travel at the flow velocity in the limit of zero Mach number. Thus the preconditioning matrix for the subsonic case in the stream-aligned coordinate system in symmetry variables is identical to equation (4.127), except that

$$\beta = \tau = \sqrt{1 - M^2}, \qquad M < 1 \tag{4.129}$$

To solve the system numerically, it is necessary to transform from the stream-aligned symmetry variables back to the original conserved variables in a Cartesian coordinate system. The transformation matrices required for this task are given in the work of Lee [86]. Finally, it should be noted that in order to apply this preconditioning effectively, with certain schemes the numerical viscosity must be modified so as to overcome stability restrictions that would otherwise impair the benefits of the preconditioning. For Roe's approximate Riemann solver [127], this amounts to modification of the numerical flux function. Details of this modification are also given by Lee [86].

Use of this preconditioning reduces the condition number of the characteristic speeds from $(1+|M|)/\min[|M|, 1-|M|]$ to $1/\sqrt{1-\min[M^2, M^{-2}]}$ which can be shown to be the lowest condition number attainable. Results from this preconditioning for flows about two-dimensional aerofoils are presented in [169] for the Mach number range $0.01 \geq M \geq 1.8$. In all cases, the preconditioning improves convergence significantly.

When moving from two- to three-dimensional preconditioning, further deterioration of the condition number results. The reason is that in three-dimensions a shear wave is possible that cannot be separated from the acoustic waves for the Euler equations. Consequently, for supersonic flow, the condition number reduces to $1/\sqrt{1-M^2}$ as $M \downarrow 1$. Nevertheless, a three-dimensional preconditioning matrix for the Euler equations in the stream-aligned symmetry variables has been derived [86] which is similar in form to its two-dimensional counterpart, equation (4.127).

The above preconditioning does not take into account the effect of the differences in mesh stretching in different coordinate directions. In viscous solutions, the need to resolve boundary layers necessitates the use of cells with high aspect ratio close to the boundary and in these regions a severe reduction in the rate of convergence to the steady-state can be expected. It is for this reason that preconditioners need to be specifically developed for viscous flows. Work has already begun to emerge on such preconditionings. The references cited earlier, namely [23, 85, 52, 51] deal with the subject.

The remaining difficulties for two- and three-dimensional preconditionings are the singularities in flow as $M \to 0$ and $M \to 1$. According to [169], these are no trivial matters and may require a considerable research effort. Recently Koren [78] has made progress with the $M \to 0$ singularity by improving the condition of the one- and two-dimensional upwind-discretized Euler equations for subsonic computations. This is done by adding an appropriately constructed regularizing matrix to the absolute-eigenvalue matrix. The approach is referred to as additive conditioning. In the section to follow, another recent approach based on the canonical form of the governing equations is briefly described which also addresses the $M \to 0$ singularity in two-dimensional inviscid flows. Later in chapter 6, an attempt is made to deal with the singularity at $M \to 1$ in a shock wave.

The method formulated there can be viewed as locally modifying the flux function of an upwind scheme in the vicinity of a shock wave and is termed *shock acceleration.*

4.8 Canonical Forms of the Euler Equations

A new approach for the discretization, solution and convergence acceleration of the Euler equations has been proposed by Ta'asan [149, 150] in which the governing equations are written in a canonical form. This separates the equations describing acoustic waves from those describing convection. In this form the two-dimensional steady flow equations are written in terms of the velocity components u and v, the temperature T, entropy s and total enthalpy H. In canonical variables the governing equations for compressible flow become

$$\begin{bmatrix} D_1 & D_2 & 0 & 0 \\ q\partial/\partial y & -q\partial/\partial x & -c^2 D_0/\gamma(\gamma-1) & D_0/q \\ 0 & 0 & -T\rho Q & 0 \\ 0 & 0 & 0 & \rho Q \end{bmatrix} \begin{bmatrix} u \\ v \\ s \\ H \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.130}$$

with the differential operators

$$\begin{aligned} D_1 &= \frac{\rho}{c^2}\left[(c^2-u^2)\frac{\partial}{\partial x} - uv\frac{\partial}{\partial y}\right] \\ D_2 &= \frac{\rho}{c^2}\left[(c^2-u^2)\frac{\partial}{\partial y} - uv\frac{\partial}{\partial x}\right] \\ D_0 &= v\frac{\partial}{\partial x} - u\frac{\partial}{\partial y} \\ Q &= u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} \end{aligned}$$

and

$$q = u^2 + v^2$$

Expressed in this form, the system is essentially uncoupled with the first two equations describing an elliptic subsystem if the flow is subsonic while the remaining equations describe a convective subsystem. Since the subsystems are of a very different nature, it is unlikely that the same numerical procedures will be optimal for both. The separation of the different subsystems allows the construction of an optimal solver for the full system. The canonical form therefore allows the elliptic subsystem to be discretized using a central discretization while the subsystem containing the physically based quantities, i.e. entropy and total enthalpy, may be upwind-biased. Multigrid acceleration using the FAS-FMG formulation is then used only on the elliptic subsystem where it is most effective.

Ta'asan [150] has presented preliminary numerical results for flow through 2-D nozzles at $M_\infty = 0.02, 0.2$ and about an ellipse at $M_\infty = 0.1$, $\alpha = 10°$. Unlike most other methods, the performance of the method does not degrade as the Mach number approaches zero.

4.9 Closing Discussion

This chapter has reviewed the principle convergence acceleration methods in use today for explicit time marching schemes with a view to placing in context the convergence acceleration techniques of the present study. The concepts that lie beneath the various methods have been identified with a view to affording insight to their efficacy, their limitations and where applicable their complementary nature. Enlargement of the stability region is one approach that is adopted by methods such as local time stepping, general multi-stage methods, residual smoothing, multigrid and Krylov methods. Another approach to convergence acceleration is to tackle the problem of stiffness in the equations as exemplified by preconditioning methods. One of the extreme cases of stiffness, the singularity occuring at $M \rightarrow 0$, may be tackled by adopting the canonical form of the governing equations so that the resulting equation sub-systems may each be treated optimally. An original approach to tackling the flow singularity as $M \rightarrow 1$ is formulated later in chapter 6, based on a procedure which is applied locally to the discrete shock structure.

Convergence acceleration methods may also be classed as locally convergent or globally convergent. This classification determines whether the ability of a numerical method lies in achieving convergence from some arbitrary initial guess within the basin of attraction or simply in converging quickly from a nearby state. Vector sequencing methods lie in the former category while only iterative procedures using full multigrid appear to be in the latter. Another way in which some convergence acceleration methods may be viewed is the frequency error range which they specifically target. For example, optimal smoothing multi-stage methods are devised to damp high-frequency error modes whereas multigrid may be seen as an approach based on the expulsion of low-frequency errors. An attempt has also been made to highlight some of the physical and numerical problems encountered in the development of convergence acceleration methods. Flow singularities have already been mentioned, the physically diffusive limit on timestep size is another. The latter occurs routinely in the application of numerical methods to high Reynolds number viscous flows. The proliferation of convergence acceleration methods indicates that there is no single approach that is optimal. Therefore it is likely that further gains in accelerating the convergence of numerical methods will continue to be found through the development of complementary techniques that can be used together for maximum effect.

5

Multigrid Acceleration for Viscous Flow

To accelerate the convergence of the present 3-D Euler upwind solver, the multigrid method was implemented. The multigrid coding modules were then extensively tested and validated for coarse and fine grid inviscid solutions about a wing in isolation. Once the correctness of these modules had been established and the multigrid options assessed, the code was extended to the Navier-Stokes equations by the inclusion of the diffusive terms based on the theory of chapter 2. Initially the viscous modules were validated on a 2-D laminar flat-plate test case for which the computed boundary layer profiles were found to be in excellent agreement with the analytic Blasius profiles. Next the turbulent flow capability of the solver was tested on high-Reynolds number flat-plate test cases. After that, the viscous modules of the code were tested together with multigrid on several realistic wing geometries using coarse and fine grids for both laminar and turbulent flows.

In the present chapter, specific details are given of the multigrid methodology. Following that, numerical results obtained from the new upwind multigrid Navier-Stokes solver are presented and compared to equivalent results from a state-of-the-art industrial CFD design code based on centred differences with scalar dissipation. The results presented encompass a formal grid convergence study for high Reynolds number turbulent flow over the ONERA M6 wing.

5.1 Multigrid Strategy

In this work a multigrid technique based on the Full Approximation Storage, Full Multigrid Scheme (FAS-FMG) of Brandt [9] is implemented, following the formulation proposed by Jameson [70]. The present formulation is based on an explicit optimal smoothing multi-stage scheme in which local timesteps are used. Successively coarser grids are formed by deleting every other mesh line on the next finest grid. The sequence of coarse grids so formed is then used to accelerate the rate of convergence by allowing the use of large time-steps on coarse grids which enables disturbances to be expelled more rapidly from the computational domain. For a more detailed discussion of multigrid fundamentals, refer to section 4.2.1. The principal elements of the present multigrid method are described below.

5.1.1 THE SMOOTHING SCHEME

The smoothing scheme is based on the theory of chapter 2. The convective derivatives are approximated using the upwind spatial operator given in section 2.4.3, while the diffusive derivatives are approximated using a centred discretization following section 2.5.1. The time integration is performed by a choice of optimally smoothed multi-stage schemes which are described in section 4.5. On the finest grid (h) in the multigrid cycle the solution is obtained for each interior cell as

$$\begin{aligned} \mathbf{Q}_h^{(0)} &\stackrel{\text{def}}{=} \mathbf{Q}_h^n \\ \mathbf{Q}_h^{(k)} &= \mathbf{Q}_h^{(0)} - \alpha_k \frac{\Delta t}{\mathcal{V}_h} \mathbf{R}_h[\mathbf{Q}_h^{(k-1)}] \qquad k = 1, m \\ \mathbf{Q}_h^{n+1} &\stackrel{\text{def}}{=} \mathbf{Q}_h^{(m)} \end{aligned} \tag{5.1}$$

cf. equation (2.113). The solution vector is given by the variable $\mathbf{Q}$, the coefficients α_k are the multi-stage coefficients, Δt is the local timestep, and $\mathcal{V}$ the cell volume. Each bracketed superscript refers to the integration stage at which its possessor is evaluated. Non-bracketed superscripts refer to the time level. Since the success of a multigrid scheme is critically dependent on the ability of the smoother to rapidly damp high-frequency modes, optimal smoothing multi-stage schemes were considered. For the present algorithm, the two most efficient multi-stage schemes appear to be the four-stage scheme of Lallemand and the five-stage hybrid scheme of Tai. (The former is a 4/4 scheme in which the dissipative flux is evaluated at each stage using the coefficients α_k given in Table 4.2, while the latter is a 5/3 scheme in which the dissipation is evaluated only at odd stages. Its coefficients are given in Table 4.3). The fine grid residual is evaluated as

$$\mathbf{R}_h\left(\mathbf{Q}_h^{(k-1)}\right) = \left[\mathbf{R}_\iota^{(k-1)} + \mathbf{R}_\upsilon^{(k-1)} - \mathbf{R}_{d2}^{(k-1)}\right]_h \tag{5.2}$$

in which $\mathbf{R}_\iota$ represents the inviscid component of the residual, $\mathbf{R}_\upsilon$ the viscous component and $\mathbf{R}_{d2}$ the contribution of the second-order numerical dissipation. For the 5/3 optimally smoothed scheme the dissipative component of the residual is recomputed only at odd-stages and weighted according to the residual function

$$\mathbf{R}_{d2}^{(k)} = \beta^{(k)} \mathbf{R}_{d2}\left[\mathbf{Q}^{(n+1,k)}\right] + \left(1 - \beta^{(k)}\right) \mathbf{R}_{d2}^{(k-1)} \tag{5.3}$$

The weights β_k are included in Table 4.3.

5.1.2 SOLUTION AND RESIDUAL RESTRICTION

In the FAS scheme (see section 4.2.1), both the solution and the residual are restricted from a given grid level (h) to the next coarser grid level (H) in the sequence. Separate operators are used for this purpose. For notation, the convention adopted is to use the subscripts h and H respectively to indicate successive fine and coarse grid levels. Once the fine grid solution has been updated using equation (5.1), the solution on the coarser grid is initialized by a volume-weighted restriction operator as

$$\mathbf{Q}_H^{(0)} = R_h^H \mathbf{Q}_h \tag{5.4}$$

The *solution* restriction operator is defined as

$$R_h^H \equiv \frac{\sum_h \mathcal{V}_h (\bullet)_h}{\mathcal{V}_H} \tag{5.5}$$

with the summation carried out over the eight h-grid cells that compose each coarser H-grid cell. From the definition of this operator, it is apparent that the solution restriction conserves mass, momentum and energy.

To ensure that the H-grid solution is consistent with that of the h-grid, the residuals $\mathbf{R}_h$ are restricted onto the coarser grid via a residual *forcing function* $\mathbf{F}_H$. It will be recalled from the earlier description of the FAS scheme that the forcing function, sometimes referred to as a *source term*, imparts nonlinearity to this type of algorithm. This forcing function is necessary to ensure that the solution on the H-grid is driven by the residuals calculated on the h-grid. It is defined as the difference between the aggregated residuals transferred from the h-grid and the residual computed on the H-grid and is given by

$$\mathbf{F}_H = \hat{R}_h^H \mathbf{R}_h - \mathbf{R}_H\left(\mathbf{Q}_H^{(0)}\right) \tag{5.6}$$

for which the *residual* restriction operator is defined as

$$\hat{R}_h^H \equiv \omega_R \sum_h (\bullet)_h \tag{5.7}$$

The difference between the restriction operators (5.5) and (5.7) should also be noted. The former produces a volume-weighted sum over the constituent finer grid cells which ensures that the solution transfer is conservative. In contrast the residual transfer operator effects a straight-forward summation which is *not* conservative. Finally, the restriction of residuals may be damped via the parameter ω_R. The actual value used is tuned to the specific problem. Typical values are in the range $0.6 \leq \omega_R \leq 1.0$. For transonic flows, damping was not found to be necessary and hence for the computations presented, full restriction was used (i.e. $\omega_R = 1.0$).

5.1.3 Coarse grid evolution

The multi-stage scheme defined by equation (5.1) is reformulated to produce the following coarse grid algorithm

$$\begin{array}{lcll} \mathbf{Q}_H^{(0)} & \stackrel{\text{def}}{=} & \mathbf{Q}_H^n = R_h^H \mathbf{Q}_h & \\ \mathbf{Q}_H^{(k)} & = & \mathbf{Q}_H^{(0)} - \alpha_k \frac{\Delta t_H}{\mathcal{V}_H}\left[\mathbf{R}_H\left(\mathbf{Q}_H^{(k-1)}\right) + \mathbf{F}_H\right] & k = 1, m \\ \mathbf{Q}_H^{n+1} & \stackrel{\text{def}}{=} & \mathbf{Q}_H^{(m)} & \end{array} \tag{5.8}$$

where the coarse grid residual is given by

$$\mathbf{R}_H\left(\mathbf{Q}_H^{(k-1)}\right) = \left[\mathbf{R}_i^{(k-1)} + \varphi_H \mathbf{R}_v^{(k-1)} - \mathbf{R}_{d1}^{(k-1)}\right]_H \tag{5.9}$$

It may be observed that the form of equation (5.8) differs from equation (5.1) only in the presence of the (coarse grid) forcing function $\mathbf{F}_H$, given above in equation (5.6). Thus, on the first stage of the coarse grid algorithm, the forcing term $\mathbf{F}_H$, simply replaces the coarse grid residual by the aggregated fine grid residuals. A key point to note is that the coarse grid algorithm ensures that the *converged* solution on the fine grid is not altered by coarse grid corrections so that the coarse grid correction is zero if $\mathbf{R}_h(\mathbf{Q}_h)$ is zero.

In order to reduce the computational expense of the matrix dissipation –one of the most costly elements of the solution– its evaluation on coarse grids is effected by a first-order spatial approximation, denoted $\mathbf{R}_{d1}$. This strategy is countenanced because it does not impair the second-order accuracy of the fine grid solution. Furthermore, it actually improves coarse grid damping since the high-frequency content of a first-order solution is less than that of a second-order one. To reduce expense further, the type of solution carried out on the coarser mesh is regulated via the switch φ_H. This may be set to allow laminar or even inviscid solutions on the coarser mesh rather than carry out full solutions on all mesh in a viscous solution cycle. Finally, with the 5/3 optimally smoothed scheme, the dissipative component of the residual is weighted in a similar manner to that used for the smoother. i.e.

$$\mathbf{R}_{d1}^{(k)} = \beta^{(k)}\mathbf{R}_{d1}\left[\mathbf{Q}^{(n+1,k)}\right] + \left(1-\beta^{(k)}\right)\mathbf{R}_{d1}^{(k-1)} \tag{5.10}$$

Again the parameters α_k and β_k appearing in equations (5.8) and (5.10) assume the same values as those used by the smoothing scheme.

5.1.4 Correction prolongation

Coarse grid corrections are transferred to the fine grid by the prolongation operation

$$\mathbf{Q}_h = \mathbf{Q}_h + P_H^h \Delta\mathbf{Q}_H \tag{5.11}$$

in which

$$\Delta\mathbf{Q}_H = \mathbf{Q}_H^{n+1} - \mathbf{Q}_H^n \tag{5.12}$$

The prolongation operator is based on trilinear interpolation, which may be defined as follows. A given grid cell denoted h_1 is surrounded by the cell centres of eight coarse grid cells denoted $H_1, H_2, \ldots, H_8$, numbered sequentially in terms of their distance away from the fine grid cell. Thus $\Delta\mathbf{Q}_{H_1}$ represents the coarse grid correction from the coarse grid cell H_1 which is closest to cell h_1, while $\Delta\mathbf{Q}_{H_8}$ corresponds to the coarse grid cell H_8 which is farthest away. Using this notation, trilinear interpolation of the coarse grid correction to the fine grid cell h_1 is effected as

$$\begin{aligned} P_H^{h_1}\Delta\mathbf{Q}_H \;=\; & [\bar{w}_1\Delta\mathbf{Q}_{H_1} + \bar{w}_2\left(\Delta\mathbf{Q}_{H_2} + \Delta\mathbf{Q}_{H_3} + \Delta\mathbf{Q}_{H_4}\right) + \\ & \bar{w}_3\left(\Delta\mathbf{Q}_{H_5} + \Delta\mathbf{Q}_{H_6} + \Delta\mathbf{Q}_{H_7}\right) + \bar{w}_4\Delta\mathbf{Q}_{H_8}] \end{aligned} \tag{5.13}$$

with the weightings $\bar{w}_1 = 27/64$, $\bar{w}_2 = 9/64$, $\bar{w}_3 = 3/64$, $\bar{w}_4 = 1/64$ such that the sum of all weights is unity. i.e.

$$\bar{w}_1 + 3(\bar{w}_2 + \bar{w}_3) + \bar{w}_4 = 1 \tag{5.14}$$

The coarse grid corrections are prolongated according to equation (5.13) for each interior cell. In the case of fine grid boundary cells, modification to the prolongation operator is necessary to prevent the coarse grid scheme from changing the fine grid boundary values so that the interior scheme converges to the fine grid problem.

5.1.5 Cycling strategy

Both V- (saw-tooth) and W-cycles are implemented in the present upwind solver as run-time options. In order to provide a well-conditioned starting solution for the fine mesh, Full Multigrid (FMG) is also used to combine successive grid refinement with FAS multigrid. In the present computations, saw-tooth cycles are adopted throughout as they offer better efficiency than W-cycles when the present scheme is applied to transonic flows. The maximum number of grid levels used in the present fine grid computations is four. With a four-level grid sequence, computation is typically commenced on the second grid level G^2, where 100 multigrid cycles are performed (between $G^2 \rightarrow G^1$). Then a further 100 multigrid cycles are performed at the third grid level (between levels $G^3 \rightarrow G^1$). Finally, several hundred iterations are performed at the fine grid level, cycling between all grids in the sequence until convergence to engineering precision is attained. Timesteps are performed only when moving down the cycle. No post-smoothing iterations are carried out between the interpolation steps on the way up the cycle. On the fine grid level G^4, three pre-smoothing iterations are performed prior to restriction while on all other grid levels, a single pre-smoothing iteration is carried out.

5.2 A Grid Convergence Study of Turbulent Flow Solutions

In recent years the open literature [172] has cast serious doubt on the accuracy of central difference algorithms employing scalar dissipation when applied to viscous flow simulations. In particular, a two-dimensional laminar flow grid convergence study by Allmaras [1] showed that the accuracy of various boundary layer parameters including skin friction, displacement thickness and momentum thickness, computed by a central difference scalar dissipation method is inferior to that of an upwind method utilizing matrix valued dissipation when compared against Blasius profiles. It was found that unlike the upwind solutions, those from the central difference method degraded severely with reduction in grid size. The poor performance of the central solutions was attributed to contamination of the boundary layer by excessive numerical dissipation. Furthermore, simple analysis [1] suggests that it is not possible to improve the poor accuracy of scalar based dissipation methods to the extent that they can compete with a properly formulated matrix valued dissipation method. Further work by Turkel and Vasta [158] has arrived at similar conclusions, based on the comparison of viscous solutions obtained from a scalar dissipation central difference scheme and a matrix dissipation central difference scheme. However, the Turkel and Vasta study did not include any results for the breakdown of drag, computed boundary layer profiles/parameters or momentum budgets.

The findings of the Allmaras study prompted the present grid dependency exercise. Its primary objective is to assess the accuracy of the present upwind solver in comparison with that of a state-of-the-art industrial CFD code for three-dimensional high Reynolds number turbulent flows. A secondary objective is to assess the extent of boundary layer contamination caused by numerical viscosity in realistic transonic computations. In what follows, numerical results

of the present 3-D multigrid Navier-Stokes upwind algorithm are compared to results from the BAe wing design code NS87, a central difference method that uses a scalar dissipation model [76, 97]. The test case involves high Reynolds number transonic flow over a wing for which three mesh discretizations are used. Turbulent flow solutions obtained from each solver on each mesh are presented in terms of computed pressure distributions, lift and drag coefficients, boundary layer velocity profiles, various boundary layer parameters and momentum equation budget plots. The findings of this part of the present study are to appear in the open literature [101].

5.2.1 Methodology and test case description

Both of the solvers use the Baldwin-Lomax turbulence model. The test case chosen is turbulent transonic flow with shock interaction about the ONERA M6 wing. Experimental pressure data for this case were obtained from wind-tunnel test [140]. However, although force measurements were also carried out, no values are given, nor have any appeared elsewhere in the literature for this fairly common test case. The wing is a symmetric section military/civil hybrid having a moderate aspect ratio ($\mathcal{AR} = 3.8$) with a leading edge sweep of 30°. The flow condition is $M_\infty = 0.84, \alpha = 3.06°, Re_{\bar{c}} = 11.72 \times 10^6$. The three grids used to generate the flowfield solutions are listed in Table 5.1. The density of each grid relative to that of the fine one is also given.

Mesh	*Size* $(NX \times NY \times NZ)$	*Relative mesh density*
Fine	$160 \times 56 \times 56$	1.0
Medium	$100 \times 36 \times 36$	0.26
Coarse	$72 \times 28 \times 28$	0.11

Table 5.1: Dimensions of ONERA M6 grids.

Planform views of the upper surface grids are shown in Figure 5.1. All grids share a C-H topology, necessitating an abrupt truncation of the wing tip. This produces a chiselled-tip shape which departs slightly from the smooth compound curvature of the true geometry. The solutions presented for each code were obtained from these common flowfield discretizations to facilitate unambiguous comparison between both solvers.

Each of the upwind solutions presented used the minmod limiter function and the entropy fix defined by equations (2.72–2.74) while the runs of the central scheme were performed using the recommended code defaults for all free-parameters including the dissipation coefficients. To ensure strict comparison all post-processing of solution data including the Sectional C_p's, boundary layer velocity profiles and other boundary layer parameters, was carried out using common software.

5.2.2 Coefficient of pressure

C_p isobars ($\Delta C_p = 0.025$) on the upper surface wing planform from the fine mesh solutions are given in Figure 5.2 while Figure 5.3 presents the corresponding

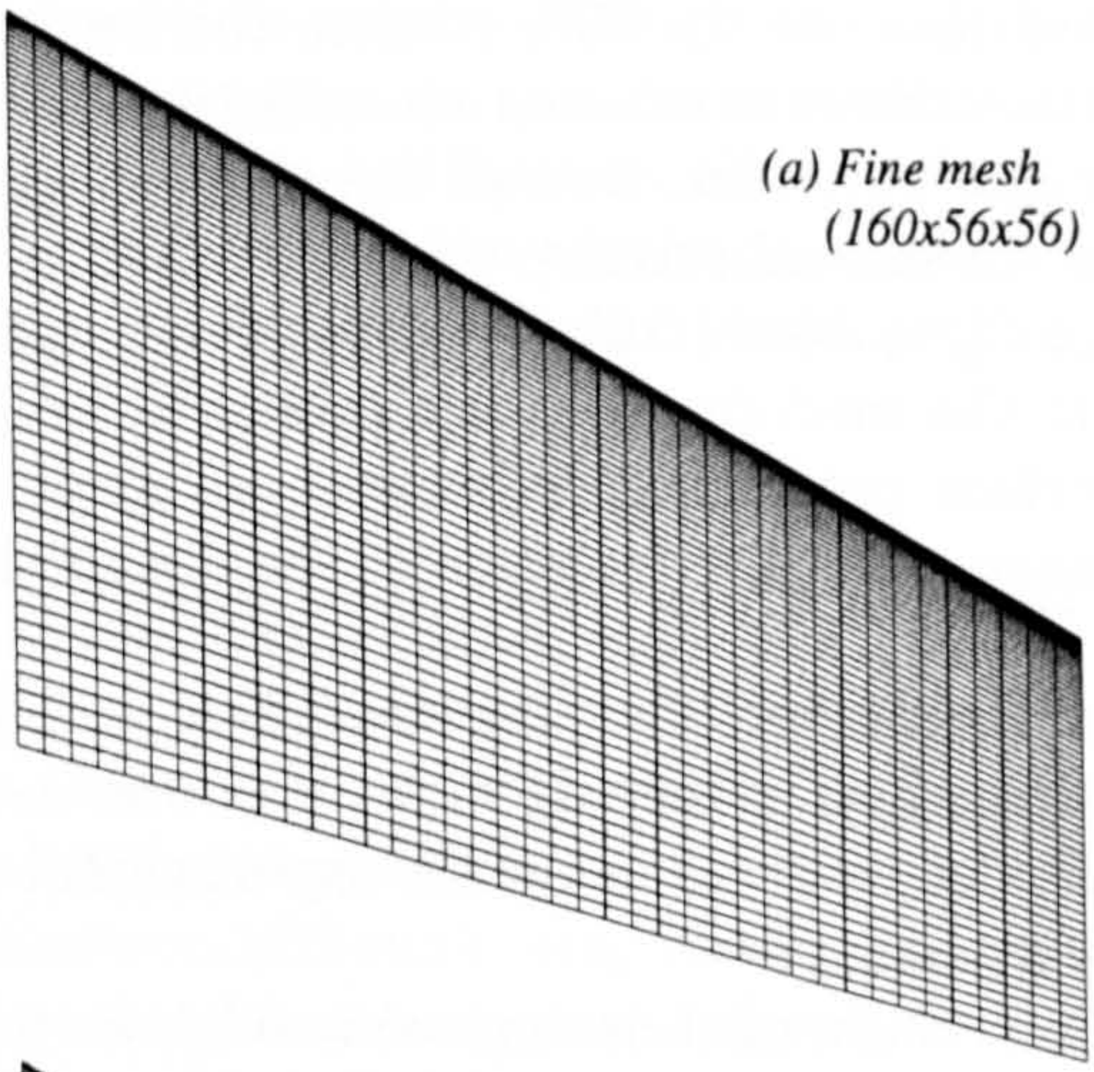
surface pressure distributions at the six span-wise stations (20%, 44%, 65%, 80%, 90% and 95%) used throughout the exercise. The suction peak from the upwind solution is higher than that of the central scheme and in closer agreement with the experimental data. At the 80% station, the front and rearward shock branches of the central difference solution are indistinguishable, having already merged prematurely into the stem. Overall the shock resolution of the upwind scheme is superior to the central scheme.

The wing planform C_p isobars ($\Delta C_p = 0.025$) of the upwind and central solutions obtained from the medium mesh are presented in Figure 5.4 while the corresponding C_p surface pressure distributions are given in Figure 5.5. The medium mesh has one-quarter the resolution of the fine mesh. The upwind solution maintains crisp shocks while those in the central solution degrade markedly with diminishing mesh resolution. The upwind suction peak remains better than that of the central scheme but, in keeping with the poorer mesh resolution in the vicinity of the leading edge, it drops below the experimental value, most notably at the 20% span-wise station.

The resolution of the coarse mesh is approximately one-tenth that of the fine mesh. Solutions from it by both solvers are given in terms of C_p isobars ($\Delta C_p = 0.025$) in Figure 5.6 and C_p surface pressure distributions in Figure 5.7. The low resolution of this mesh places high demands on both viscous solvers. The major casualty of the diminished mesh density is the suction peak. The suction peak and shock resolution of the upwind solution are still appreciably better than those of the central scheme. Indeed the captured shocks from the central scheme are poor on the coarse mesh.

Finally, in Figure 5.8 the wing planform upper surface C_p isobars ($\Delta C_p = 0.025$) from the central fine mesh solution are placed along side the isobars from the upwind medium mesh solution. The comparison demonstrates qualitatively that the upwind solver can produce sharper shocks from the medium mesh than those produced by the central solver from a mesh that is four times finer.

In all the computations discussed, the accuracy of computed flowfield behaviour near the tip (the last 5% or so of the semi-span), renders a poor comparison to the experimental data, even in the case of the fine mesh. As the resolution of the fine mesh is reasonably good by current standards, the discrepancy between the computations and experiment in the tip region is thought to be a consequence of the poor mesh representation of the true tip geometry. Indeed, tip vortex studies by Srinivasan *et al.* [145] demonstrate that the shape of the wing tip is extremely important in tip vortex formation as well as in the flow of the tip region. In addition, Kaynak and Flores [77] warn against the use of sharp-edged wedge shaped tips for modelling rounded tip geometries. Their observations are made in the light of detailed studies of transonic separated flow patterns over swept wings, based on careful comparisons between Navier-Stokes computations using sharp and rounded wing tips with high quality experimental data. This evidence together with that of the present study makes a strong case for adopting C-O mesh topologies in preference to C-H ones for wings with rounded tips.



(a) Fine mesh (160x56x56)

(b) Medium mesh (100x36x36)

(c) Coarse mesh (72x28x28)

FIGURE 5.1. ONERA M6 upper surface C-H grids.

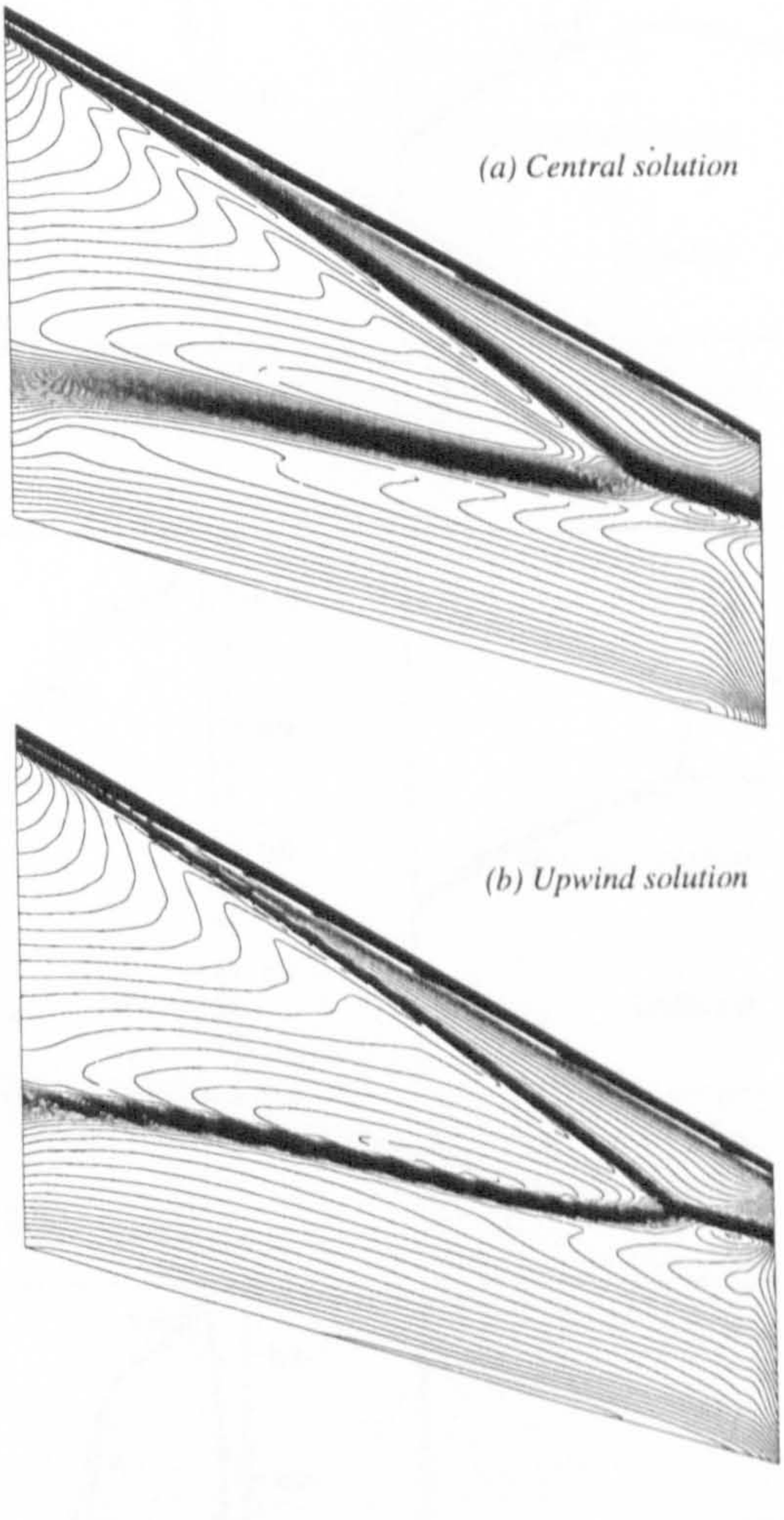


FIGURE 5.2. ONERA M6 wing upper-surface isobar contours for fine grid.

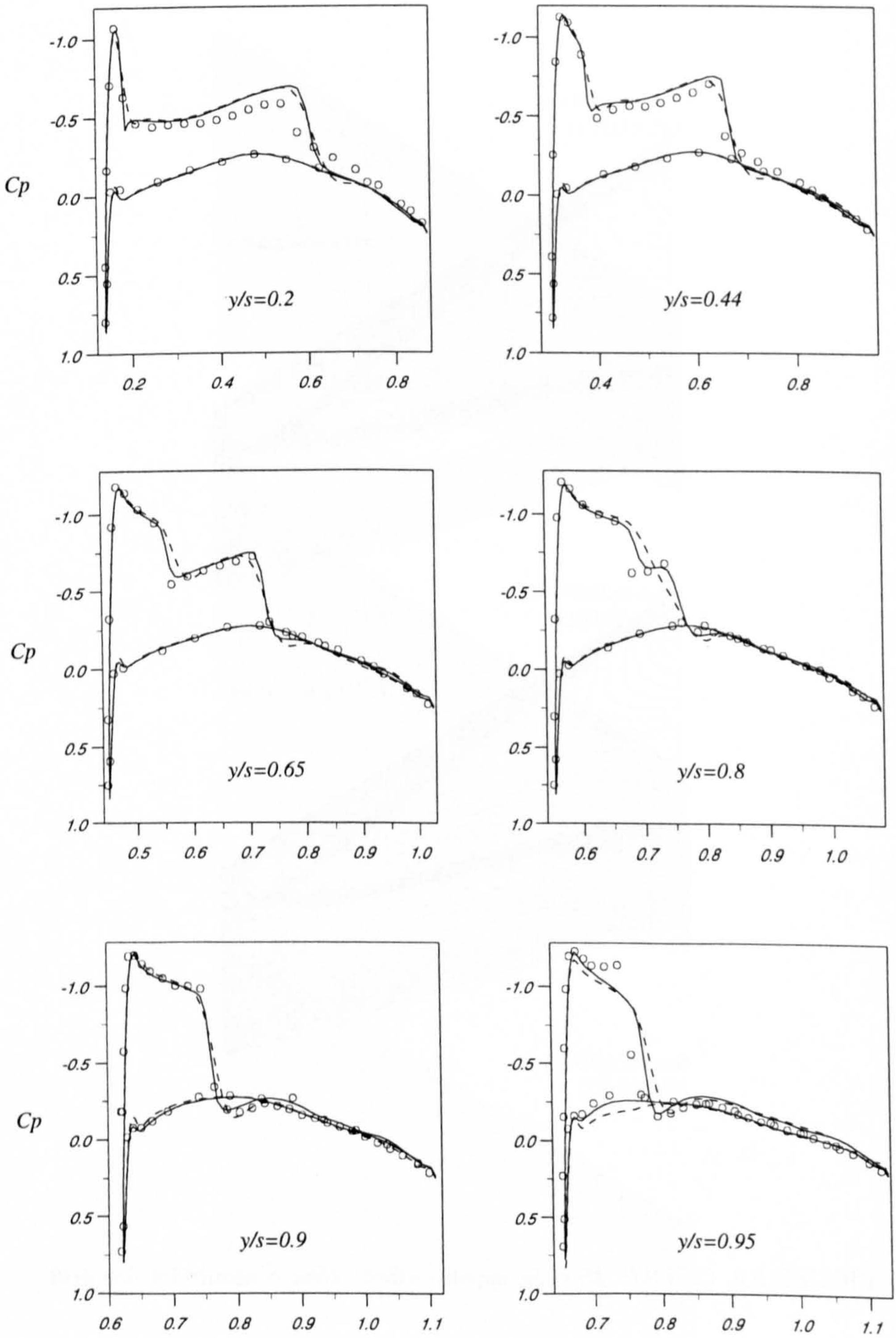


FIGURE 5.3. ONERA M6 wing sectional C_p profiles for fine grid.

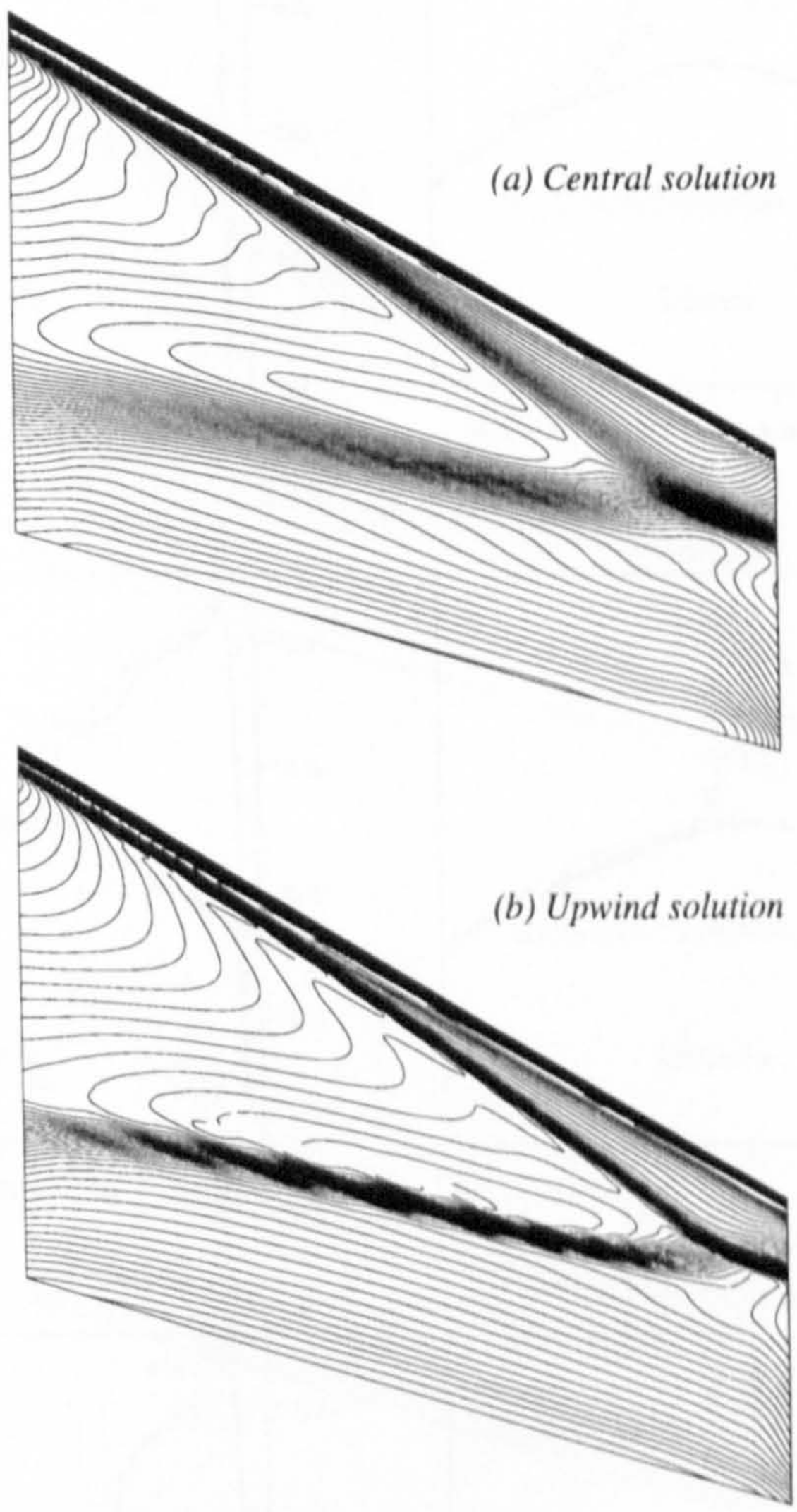


FIGURE 5.4. ONERA M6 wing upper-surface isobar contours for medium grid.

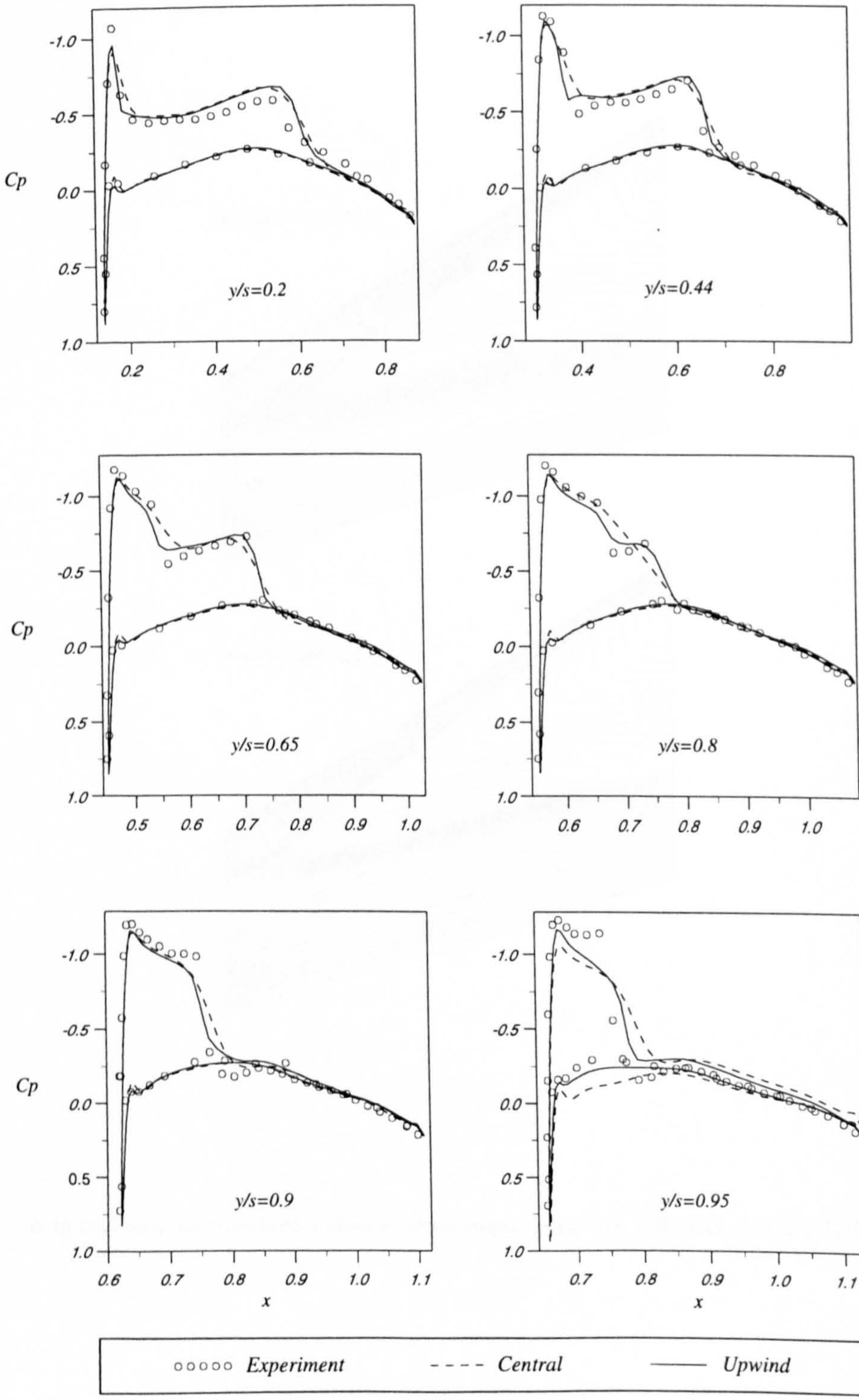


FIGURE 5.5. ONERA M6 wing sectional C_p profiles for medium grid.

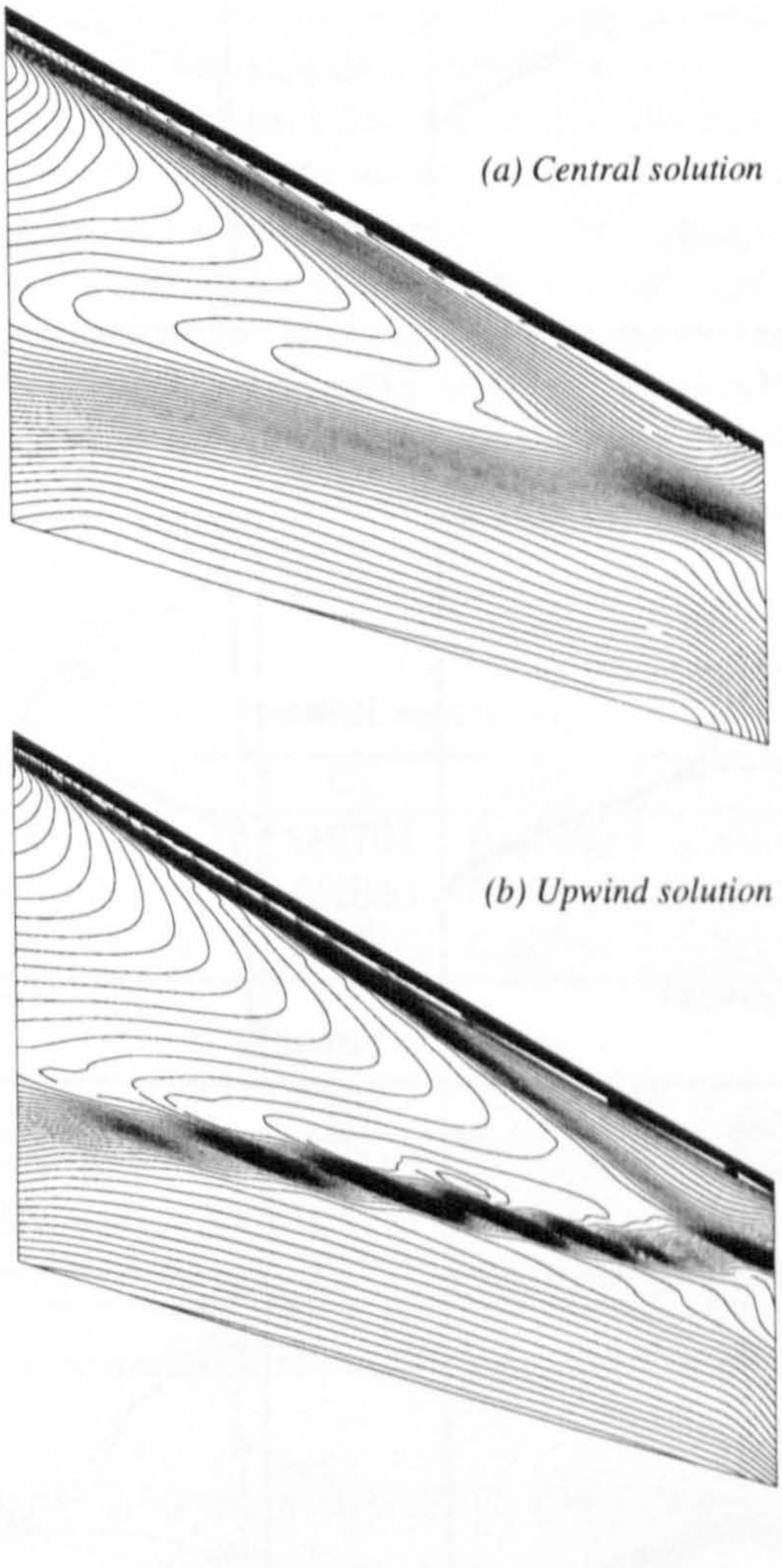


FIGURE 5.6. ONERA M6 wing upper-surface isobar contours for coarse grid.

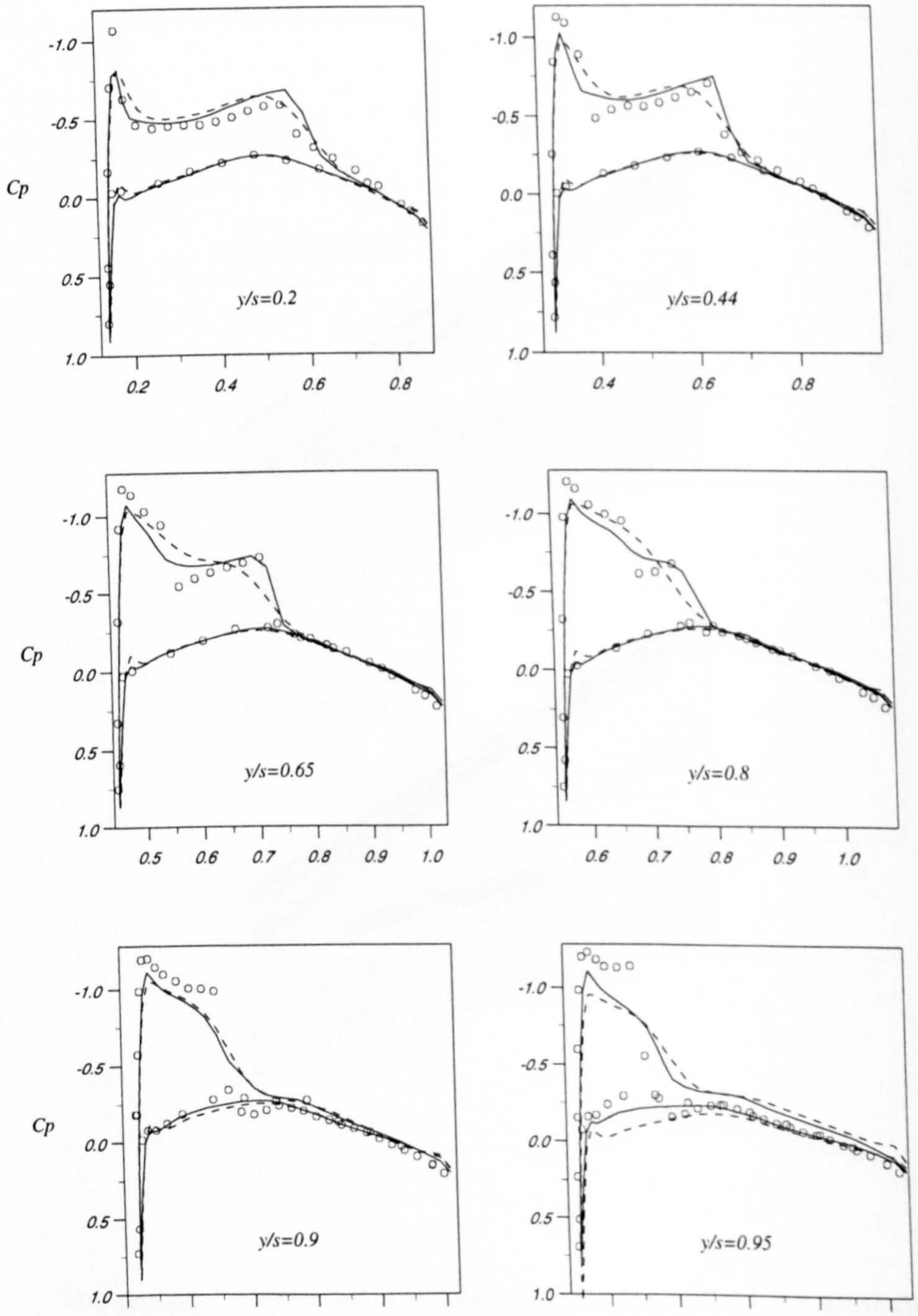


FIGURE 5.7. ONERA M6 wing sectional C_p profiles for coarse grid.

5.2.3 Lift and drag coefficients

The numerical values for lift coefficient C_L, total drag coefficient $C_{D_{total}}$, pressure drag coefficient C_{D_p} and skin friction drag coefficient C_{D_f}, for each of the central and upwind solutions are listed in Table 5.2. Since the accurate prediction of lift and drag is very important in design, a good estimate of these quantities for an infinitely fine mesh is very useful. The variation of each force coefficient component with mesh density (i.e. the reciprocal number of grid points) is plotted in Figures 5.9–5.12. In such plots it is desirable for the curves to have very shallow gradients. Were it possible a straight line parallel to the horizontal axis would in theory indicate a range of completely mesh independent solutions. Furthermore, a linear curve implies second-order accuracy for the range of mesh considered.

The plots of lift and pressure drag from the present upwind code are almost linear in contrast to those of the central solution. Also, the gradients in every upwind plot are of consistent sign unlike some of the central ones. Finally, the variation of total drag from the upwind solutions is about half that of the central solutions.

Central solutions

Mesh	C_L	C_{Dp}	C_{Df}	C_{Dtotal}
Fine: 160 × 56 × 56	0.2764	0.01400	0.00272	0.01671
Medium: 100 × 36 × 36	0.2852	0.01811	0.00353	0.02163
Coarse: 72 × 28 × 28	0.2805	0.02214	0.00178	0.02392

Upwind solutions

Mesh	C_L	C_{Dp}	C_{Df}	C_{Dtotal}
Fine: 160 × 56 × 56	0.3113	0.01596	0.00545	0.02141
Medium: 100 × 36 × 36	0.3164	0.01799	0.00528	0.02327
Coarse: 72 × 28 × 28	0.3257	0.02165	0.00360	0.02525

Table 5.2: Computed force coefficients for ONERA M6 wing.

5.2.4 Boundary layer velocity profiles

The purpose of this part of the exercise is to assess the sensitivity of computed boundary layer profiles to mesh density. Boundary layer profiles were computed on each mesh at three upper-surface positions situated in a line at the 65% station along the semi-span. These stations are indicated in Figure 5.13 where they are superimposed on the fine mesh upper-surface isobar solutions from each solver. This is done to afford a perspective of where the stations lie relative to the computed flowfield features. The first station denoted l_1 lies between the leading edge of the wing and the forward branch of the shock, the second station l_2, is roughly midway between the forward and rearward shock branches, while the final station l_3, is between the rearward branch and the trailing edge.

Figure 5.14 compares the normalized velocity profiles obtained from the central and upwind solutions on the sequence of mesh. On each mesh the central profiles differ noticeably whereas the upwind profiles are less distinguishable. The

profiles from each solver show a thickening of the boundary layer with decreasing mesh resolution. However, the central profiles thicken substantially more than the upwind ones. It is concluded that the upwind solver exhibits appreciably faster grid convergence.

5.2.5 Boundary layer parameters

The purpose of calculating boundary layer parameters is again to assess the grid sensitivity of the two flow solvers. As experimental data for these parameters are not available, it is possible only to determine relative errors. This is done for each solver in turn by using its respective fine mesh solution as a norm and then calculating the errors in skin friction C_{D_f}, displacement thickness δ^*, and momentum thickness θ, relative to that norm for the coarser mesh solutions. This was done at the three stations $l_1 - l_3$. The results are presented in the accompanying tables. Values of the computed boundary layer parameters and their relative errors in the case of the medium and coarse mesh solutions are listed in Tables 5.3 and 5.4 respectively, for the central scheme. Table 5.5 and 5.6 list the equivalent values computed by the upwind scheme.

Although some of the computed errors are inevitably due to the well-known resolution requirements for boundary layer equation solvers, the magnitude of these errors gives some indication of the demanding nature of the given flow conditions at the chosen stations. Indeed, the flow is strongly three-dimensional

Mesh	*Station*	C_f (×1000)	δ^* (×1000)	θ (×1000)
Fine	l_1	1.674	0.121	0.042
Fine	l_2	1.320	0.370	0.191
Fine	l_3	0.814	1.536	0.873
Medium	l_1	2.095	0.220	0.097
Medium	l_2	1.441	0.651	0.336
Medium	l_3	0.709	2.644	1.419
Coarse	l_1	0.893	0.355	0.151
Coarse	l_2	0.797	0.641	0.338
Coarse	l_3	0.406	3.078	1.723

Table 5.3: Boundary layer parameters computed by central scheme.

Mesh	*Station*	C_f *error (%)*	δ^* *error (%)*	θ *error (%)*
Medium	l_1	25.15	81.82	130.95
Medium	l_2	9.17	75.95	75.92
Medium	l_3	−12.90	72.14	62.54
Coarse	l_1	−46.65	193.39	259.52
Coarse	l_2	−39.62	73.24	76.96
Coarse	l_3	−50.12	100.39	97.37

Table 5.4: Relative Error in boundary layer parameters for central scheme.

Mesh	*Station*	C_f (×1000)	δ^* (×1000)	θ (×1000)
Fine	l_1	2.295	0.098	0.042
Fine	l_2	1.673	0.372	0.204
Fine	l_3	1.070	1.484	0.892
Medium	l_1	2.641	0.103	0.046
Medium	l_2	1.799	0.472	0.247
Medium	l_3	1.013	1.971	1.117
Coarse	l_1	1.277	0.081	0.036
Coarse	l_2	1.030	0.254	0.133
Coarse	l_3	0.693	1.281	0.774

Table 5.5: Boundary layer parameters computed by upwind scheme.

Mesh	*Station*	C_f *error (%)*	δ^* *error (%)*	θ *error (%)*
Medium	l_1	15.08	5.10	9.52
Medium	l_2	7.53	26.88	21.08
Medium	l_3	−5.33	32.82	25.22
Coarse	l_1	−44.36	−17.35	−14.29
Coarse	l_2	−38.43	−31.72	−34.80
Coarse	l_3	−35.23	−13.68	−13.23

Table 5.6: Relative Error in boundary layer parameters for upwind scheme.

and boundary layer development is influenced by the presence of multiple shocks. The relative errors of the upwind solutions on the medium and coarse mesh are clearly superior to the central ones in all the measured parameters, but particularly so in respect of momentum thickness and displacement thickness for which the central solution errors are enormous. On the medium mesh, the relative error in displacement thickness from the central solver varies between approximately 2–16 times that of the upwind solver while in terms of momentum thickness on the coarse mesh its relative error varies between 2–18 times that of the upwind solver. Though it is not possible to compute absolute errors for this test case, the relative errors suggest strongly that the upwind boundary layer parameters are likely to be far more accurate as grid size is decreased than would be the case with the central solver. In any event, the computed boundary layer thickness of the central solutions increases exponentially with decreasing grid size.

5.2.6 Momentum equation budgets

The central and upwind solvers use the same turbulence model and a similar discretization of the diffusive terms. In terms of spatial discretization, the principal difference between the two methods lies in their dissipation models. With this in mind and in the light of the numerical results presented in the last two sections, it would seem likely that the substantial errors in the central boundary layer parameters must be due to contamination of the boundary layer by excessive numerical dissipation. In order to verify this, budget plots for the x-momentum equation on the fine mesh at the stations $l_1 - l_3$ were obtained for both solvers. Each budget plot is produced from the definition of the solution residual for the x-momentum equation, broken down into the individual inviscid, viscous, and

dissipative components.

For an ideal momentum budget in a viscous flow solution, one is looking for as little numerical dissipation within the boundary layer as possible. The dominant momentum component should be the inviscid one. Figure 5.15 shows the budget plots for both solvers at the given stations. It can be seen that the central solution has levels of dissipation that vary between one and two orders of magnitude greater that the upwind solution.

5.3 Computational Issues

5.3.1 Convergence histories

The convergence histories of each fine grid solution is given in terms of the log of the l_2 norm ($\log ||d\rho/dt||_2$), the number of supersonic points, the lift coefficient C_L, and the drag coefficient C_D, each plotted against Work Units (WU). One WU is defined to be the computational effort required to evaluate the residual $\mathbf{R}_h(\mathbf{Q}_h)$ in equation (5.1) for one complete multi-stage cycle on the finest grid. This is a convenient measure of work as it facilitates direct comparison with the basic single-grid scheme so that relative efficiency may be readily assessed. (By definition, one multi-stage iteration of a single-grid scheme is effectively equivalent to 1WU, coding issues aside).

Each solution presented used local time stepping. The first set of plots compares the convergence of the present multigrid upwind solution using the 5/3 optimal smoothing multi-stage scheme (Tai coefficients) to a single-grid solution. The latter was obtained from the multigrid code run in single-grid mode using a standard two-stage Runge-Kutta scheme. The first two plots of Figure 5.16 show the l_2 norm and the number of supersonic points, while Figure 5.17 gives the convergence histories of the force coefficients. The multi-grid solution achieves a drop in the l_2 norm of about 3 orders of magnitude in 400WU while the single-grid solution achieves a comparable drop in 12000WU, a factor of 30 times greater. Since a single-grid WU is roughly equivalent to 1.5WU of the 5/3 multi-stage multigrid scheme, this indicates that the multigrid algorithm is approximately 20 times more efficient than the single-grid algorithm for this test case. The number of supersonic points and the magnitude of the force coefficients from both the single-grid and the multigrid solution are within very close agreement.

The second set of convergence plots follows the format of the preceding set. Here, the aim is to compare the convergence of the upwind scheme using both a four-stage optimal smoothing scheme (Lallemand 4/4 coefficients) and the 5/3 optimal smoothing scheme (Tai coefficients) to that of the scalar dissipation central scheme. The numerical control parameters used for the central scheme solution were the code defaults. These include a W-cycle and the use of both enthalpy damping and Implicit Residual Smoothing (IRS) which permits a Courant number of 6. For the upwind scheme, FMG was used as it improves convergence at little extra expense. Figure 5.18 shows the convergence in terms of the log of the l_2 norm and the number of supersonic points. The 5/3 upwind scheme exhibits slightly better performance than the 4/4 scheme with the number of supersonic points asymptoting quicker, though the 4/4 scheme reaches a lower level

of convergence before bottoming. The central scheme exhibits superior efficiency to the upwind solutions with a drop of 4 orders of magnitude in the l_2 norm in 200WU. The 5/3 upwind solution converged with a 3 order drop after about 400WU (i.e. two coarse grid solutions of 100 cycles each, followed by 320 fine mesh cycles). The superior convergence of the central scheme is to be expected since the use of IRS permitted it to run at a Courant number 2.5 times higher than the 5/3 upwind scheme. In addition, the central algorithm is far more dissipative than the upwind scheme and therefore it can damp high-frequency errors better. Convergence in terms of the force coefficients is given in Figure 5.19.

5.3.2 Coding efficiency

Two parameters that are commonly used to assess an algorithm's coding efficiency on vector platforms are the number of floating point operations that it can sustain per second (i.e. the extent to which it is vectorized) and, its computational intensity (the ratio of flops/memory references). Generally, the higher these figures the better the efficiency. However, they both depend on the particular algorithm used as well as the platform on which the code is run. Both the central difference code and the present upwind code have been optimized for vector machines and so *Cray Perftraces* were obtained for both on a single processor of a *Cray YMP-4E* (6 nanosecond clock). The central code returned a performance of 111 Mflops while the upwind code produced 134 Mflops. In terms of a computational intensity, the central code demonstrated a figure of 0.76 while the upwind code showed an equivalent figure of 1.39. Extracts from the *Preftraces* produced for both codes showing timings of the individual routines are given in Figure 5.20. The high performance of the upwind code indicates that little scope exists for increasing its efficiency through coding improvement.

5.3.3 Computational expense

In the present viscous fine mesh computation which involves over 50 000 cells, the new upwind code had a run time that was approximately 2.2 times that of the central scheme, NS87. This represents a run time for the upwind code of 4.4 hours on a single processor of a *Cray YMP-4E* computer. In terms of memory usage, the upwind code used approximately 5% more in-core memory than the central code. The difference between run-times is not surprising since any algorithm that employs a properly formulated matrix dissipation must by nature carry out considerably more arithmetic operations than one based on scalar dissipation. Thus, on the basis of operation count alone, it is clear that upwind schemes are more computationally expensive than scalar dissipation methods. In addition, the high efficiency of the industrial design code used in the present comparison is the culmination of several years of development of well-established algorithm technology, whereas the present upwind solver has enjoyed considerably less investment.

However, taking into account the present finding that the upwind scheme can produce a viscous solution of equivalent accuracy on a grid that is at least one-quarter the size of that used by the central scheme, the relative cost of upwind solutions improves significantly. In fact, if grids are carefully selected, an upwind solution can be obtained that is both more accurate and cheaper than that

possible from the central scheme. Furthermore, aside from improved accuracy, sight cannot be lost of the excellent robustness that upwind schemes offer.

5.4 Closing Discussion

The convergence acceleration methods developed for the upwind solver of chapter 2 have been described. They comprise a FAS-FMG multigrid procedure which is used in conjunction with optimal multi-stage time stepping with local timesteps. A grid convergence study for high Reynolds number turbulent flow over a wing has also been undertaken. In this study, the solutions from the present upwind solver are compared to ones obtained from an established industrial design code based on a central difference scalar dissipation algorithm. The findings are as follows:

1. For the upwind scheme, the variation of lift and drag with mesh refinement is linear across the range of mesh considered, indicating full second-order accuracy. The central scheme does not possess such consistency;

2. The upwind scheme demonstrates faster grid convergence than the central scheme in computing viscous boundary layers. In contrast, the central scheme shows a substantial thickening of the boundary layer with decreasing mesh size;

3. Comparison of x-momentum budget plots reveal that the levels of dissipation in the central scheme are between one and two orders of magnitude greater than the upwind scheme. This indicates that the central boundary layer profiles are severely contaminated by excessive numerical dissipation. It also corroborates and extends the results of other studies [1, 158]; and,

4. Accurate results for high Reynolds number turbulent flow solutions may be obtained from the upwind scheme on grids of one-quarter the density of the central scheme.

Convergence acceleration makes the upwind algorithm 20 times more efficient than the baseline algorithm for transonic viscous flow computations. This figure is based on total CPU usage for computations carried out using the same grid. Similar comparison between the central scheme and the most efficient upwind scheme indicates that the upwind scheme is approximately twice the expense of the central scheme on the same grid. However, the upwind scheme can produce the same level of accuracy as a fine grid central solution, using a grid of substantially reduced density. In this context, the present upwind scheme is both cheaper and more accurate than the central scheme.

(a) Central solution for fine mesh (160x56x56)

(b) Upwind solution for medium mesh (100x36x36)

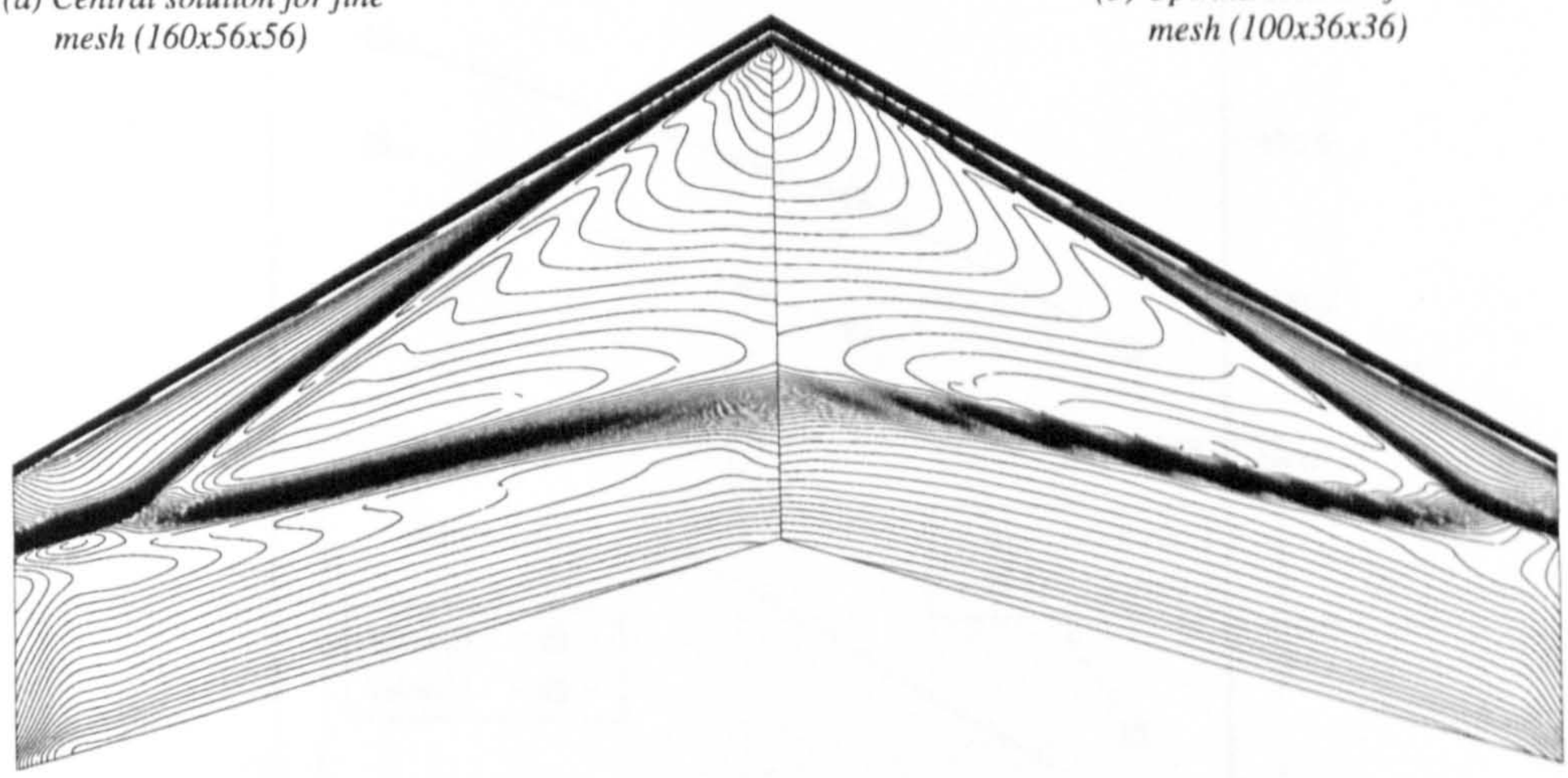


FIGURE 5.8. Comparison of ONERA M6 wing upper-surface isobar contours. Central solution on fine grid, upwind solution on medium grid.

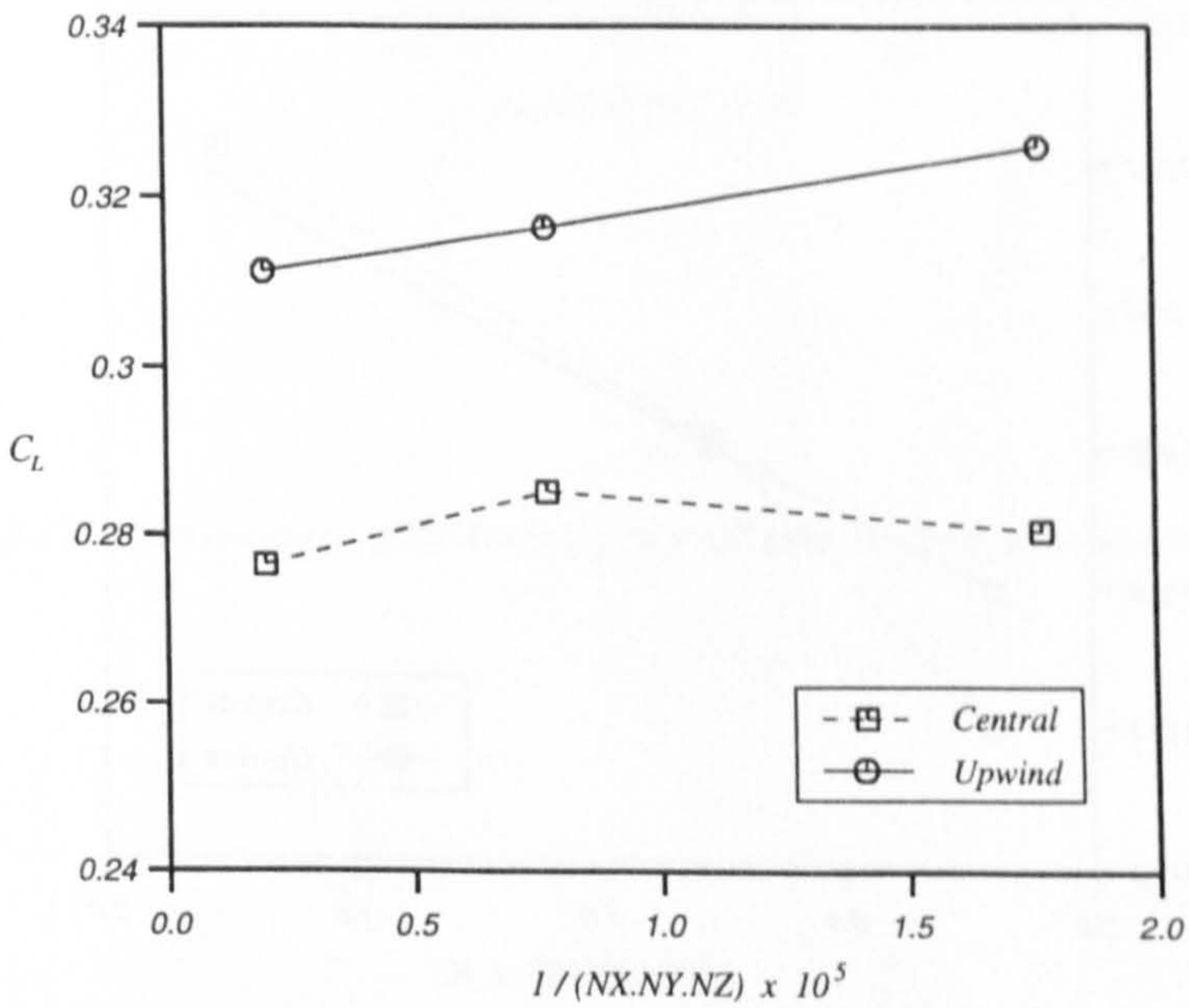


FIGURE 5.9. Comparative grid convergence of lift coefficient for ONERA M6 wing.

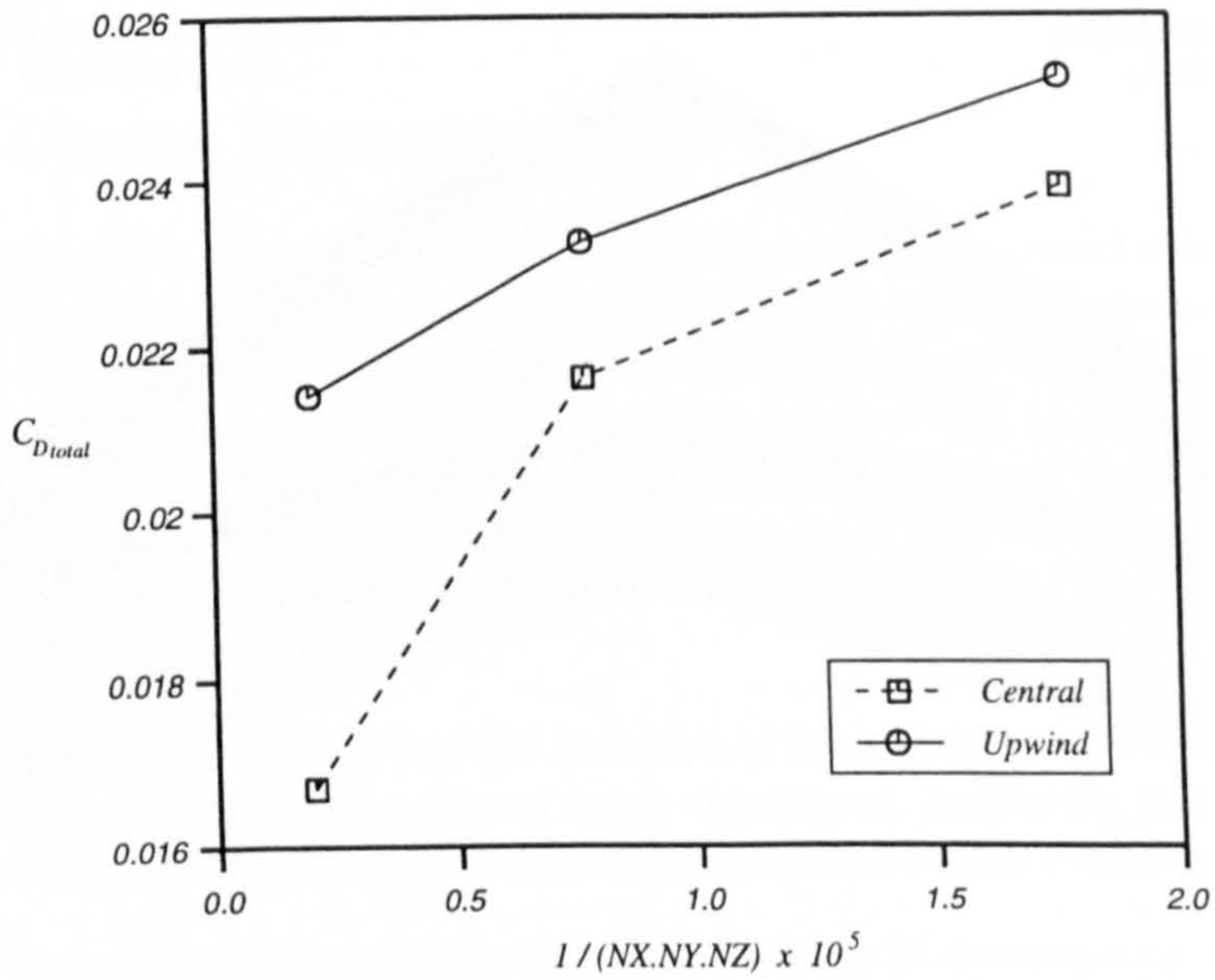


FIGURE 5.10. Comparative grid convergence of total drag coefficient for ONERA M6 wing.

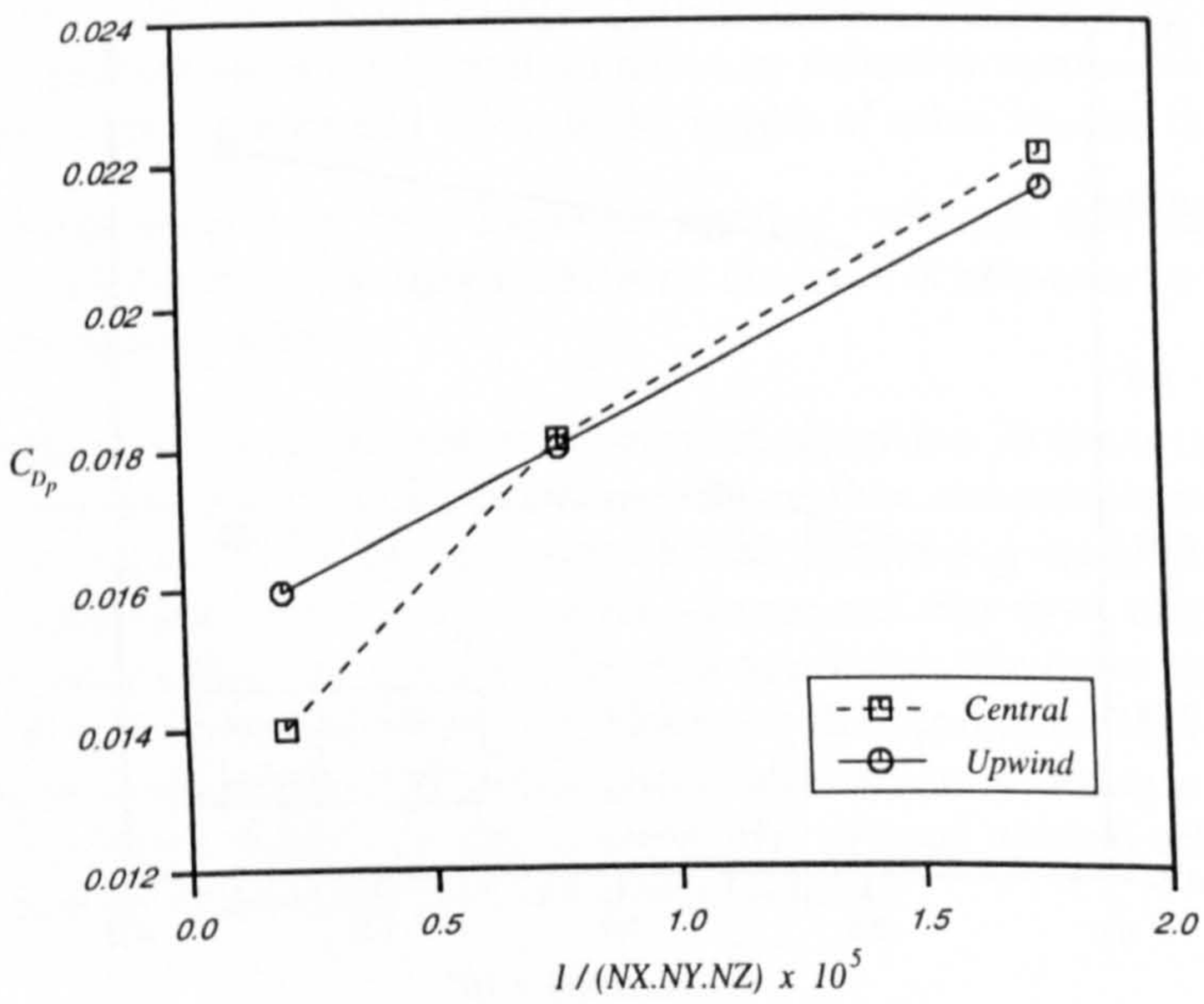


FIGURE 5.11. Comparative grid convergence of pressure drag coefficient for ONERA M6 wing

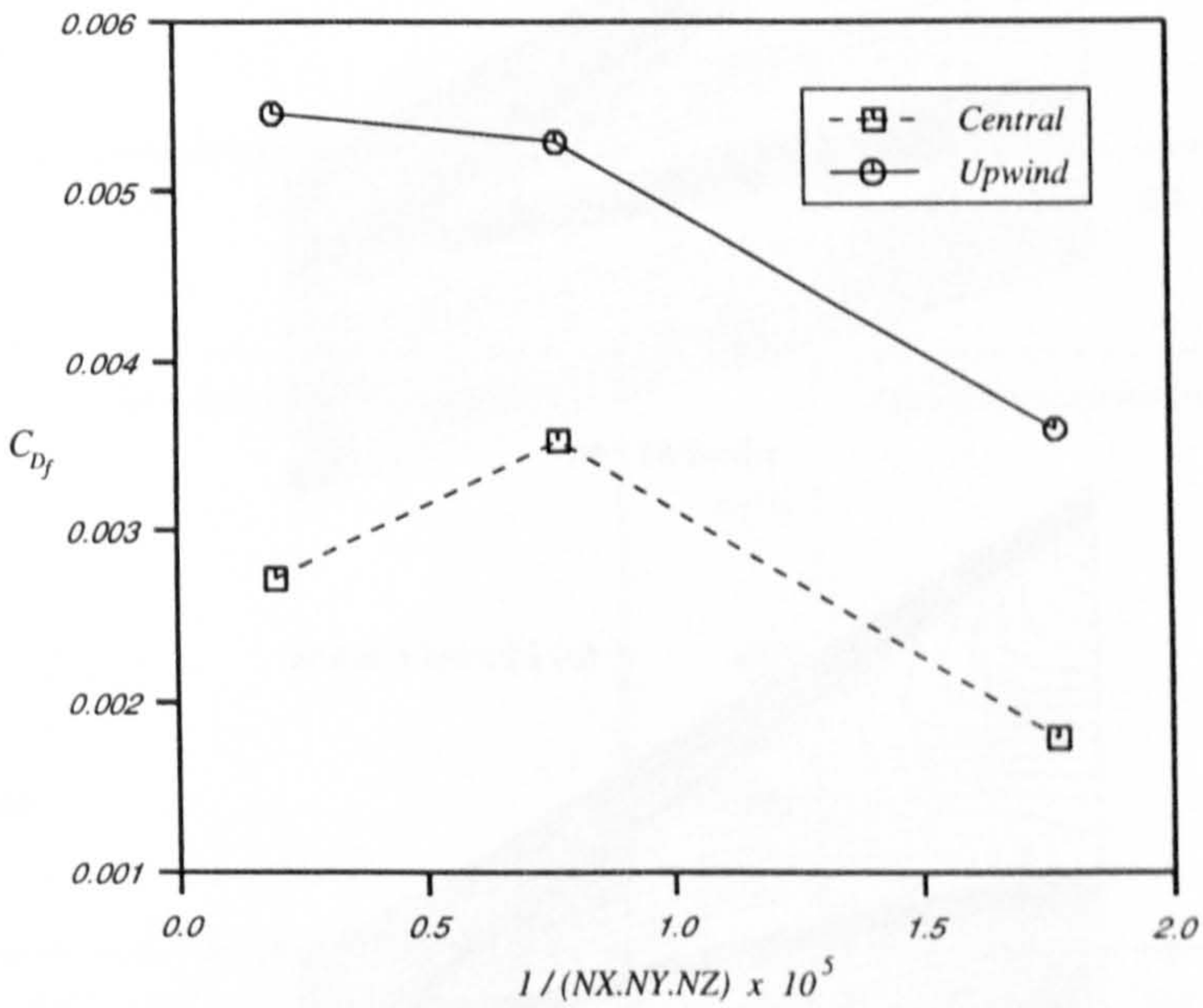


FIGURE 5.12. Comparative grid convergence of skin-friction coefficient for ONERA M6 wing

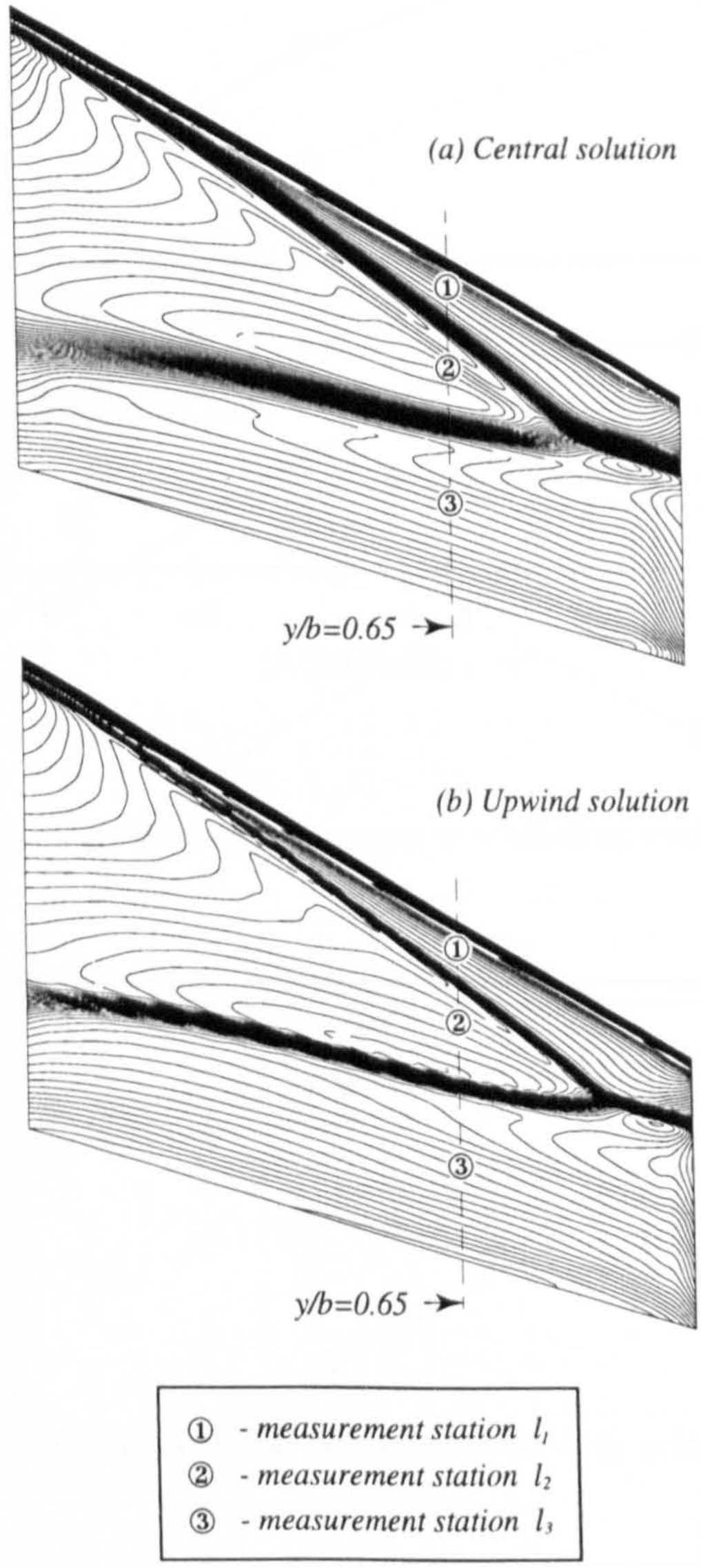


FIGURE 5.13. Location of boundary layer measurement stations on ONERA M6 upper-surface planform relative to computed isobars.

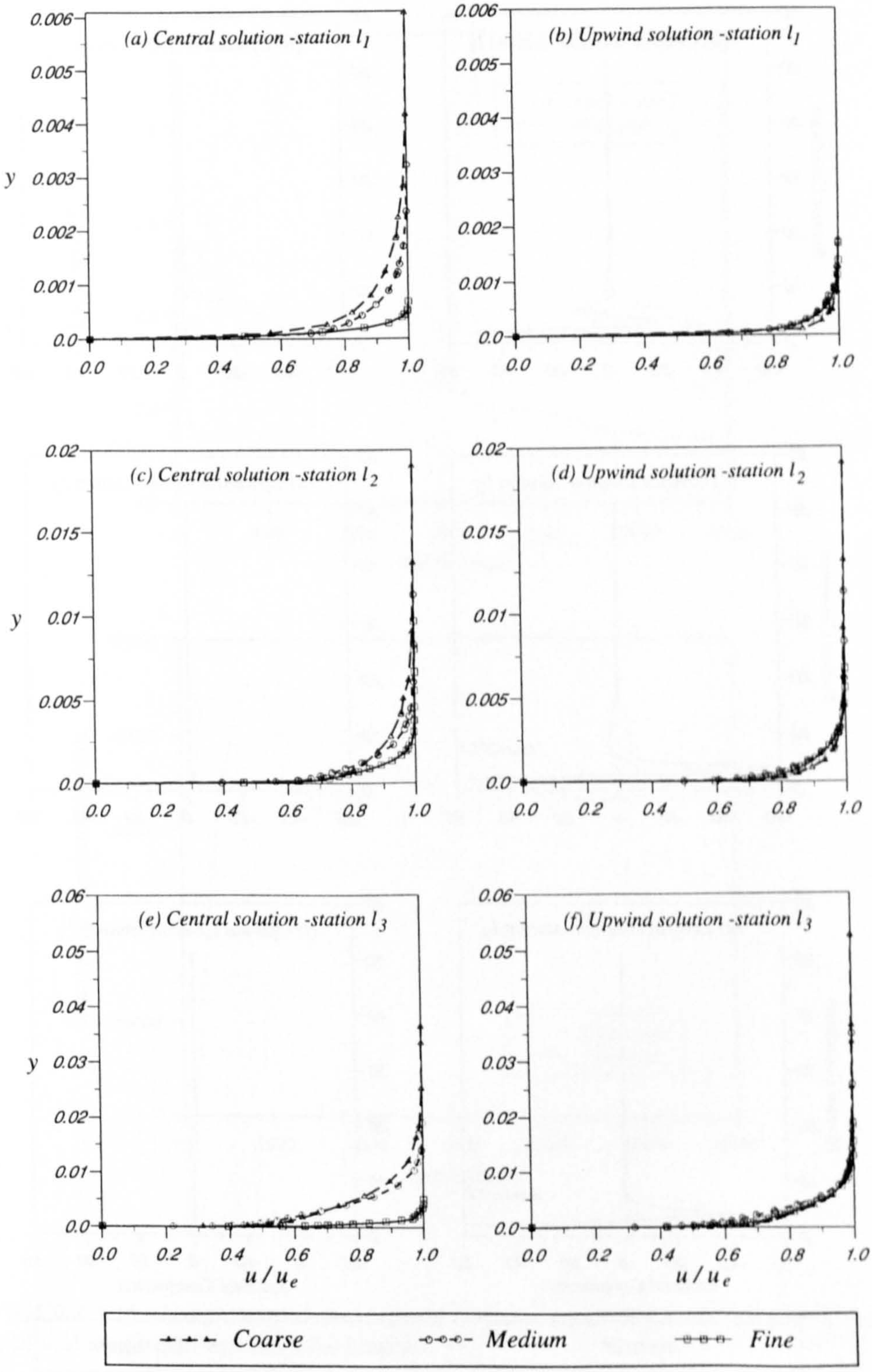


FIGURE 5.14. Comparison of velocity profiles computed by central and upwind solvers for ONERA M6 wing.

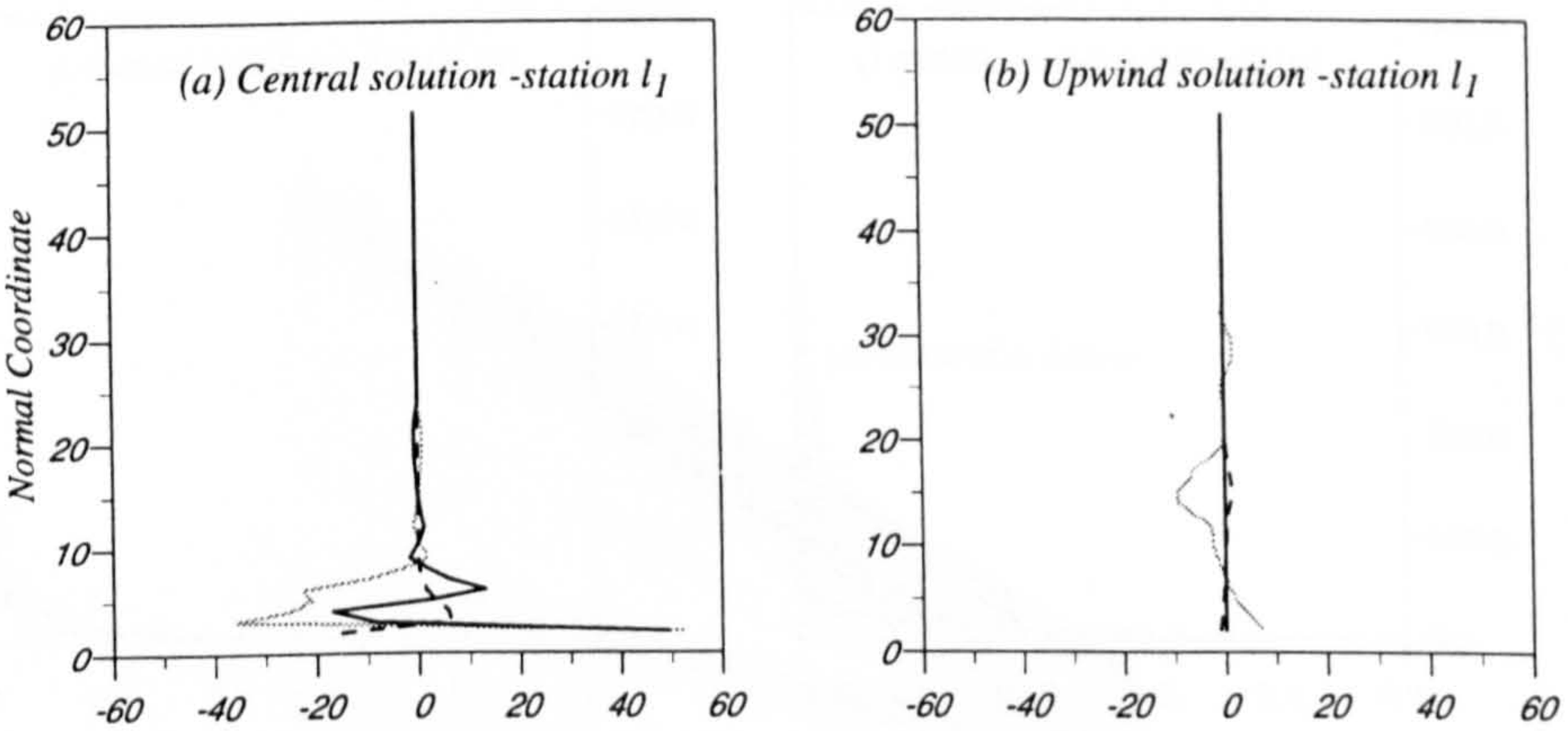
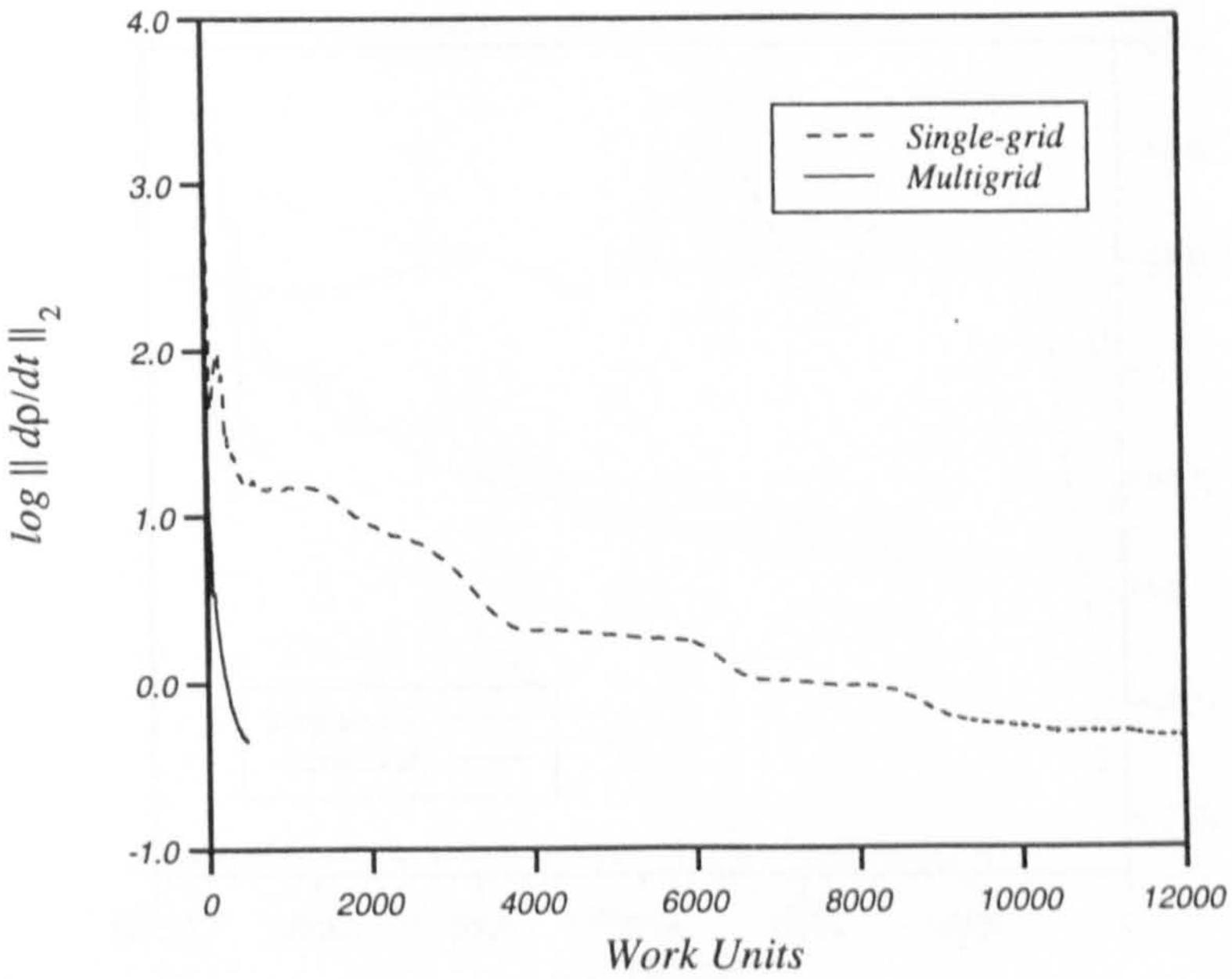


FIGURE 5.15. Comparison of x-momentum budget plots obtained from central and upwind solutions of ONERA M6 wing.



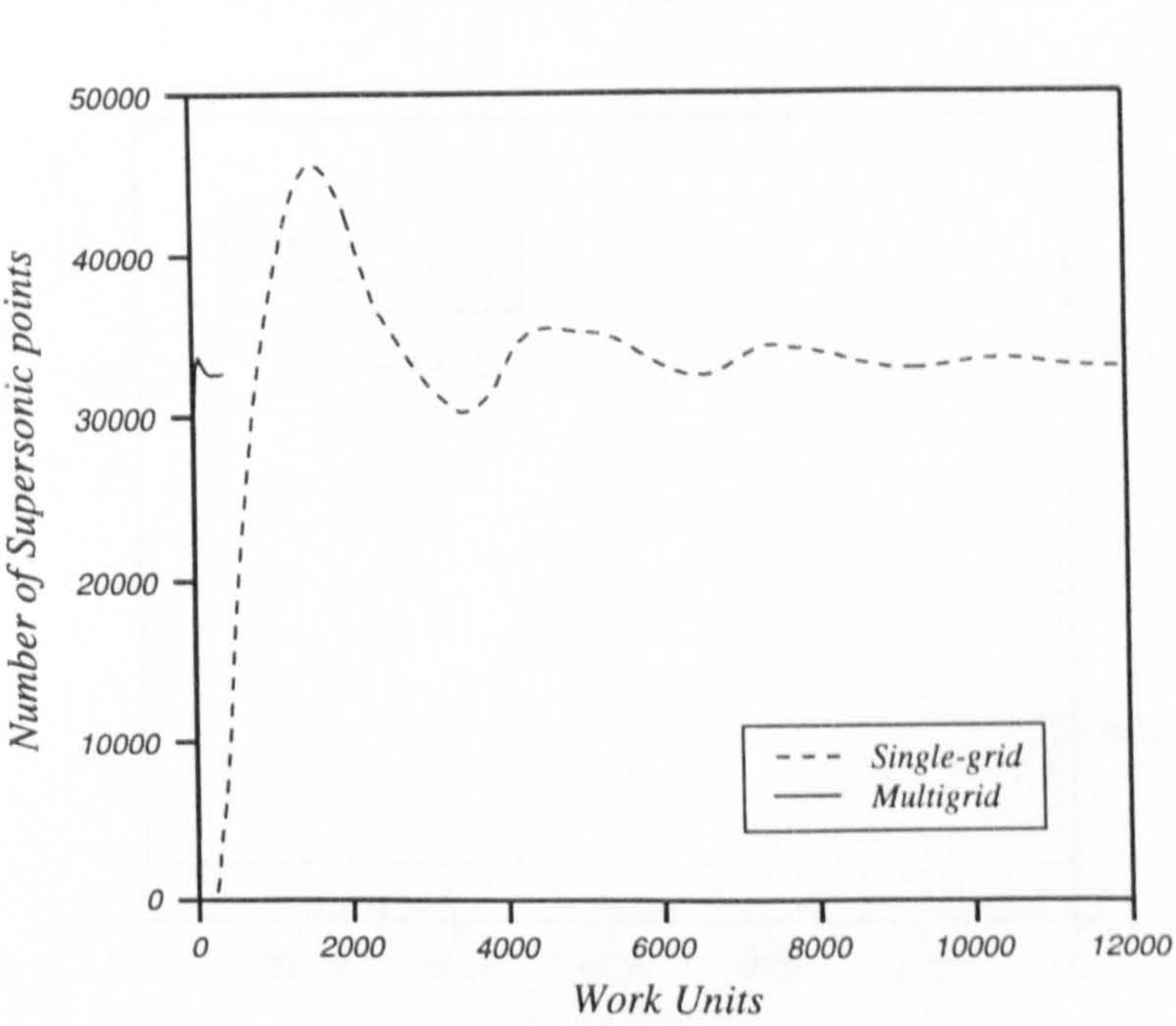


FIGURE 5.16. Comparison of convergence histories for ONERA M6 wing from multigrid and single-grid upwind solutions.

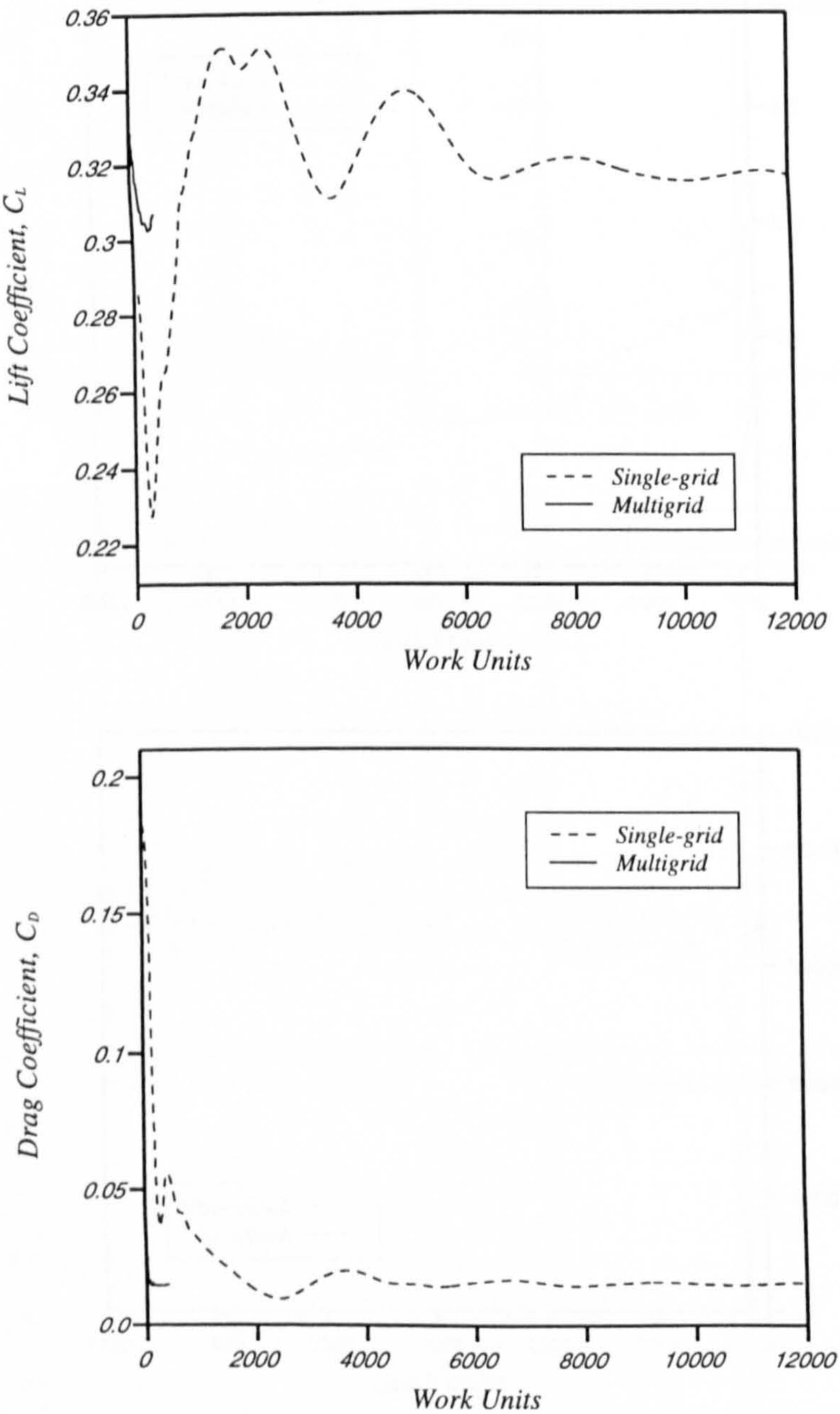


FIGURE 5.17. Comparison of force coefficient convergence for ONERA M6 wing from multigrid and single-grid upwind solutions.

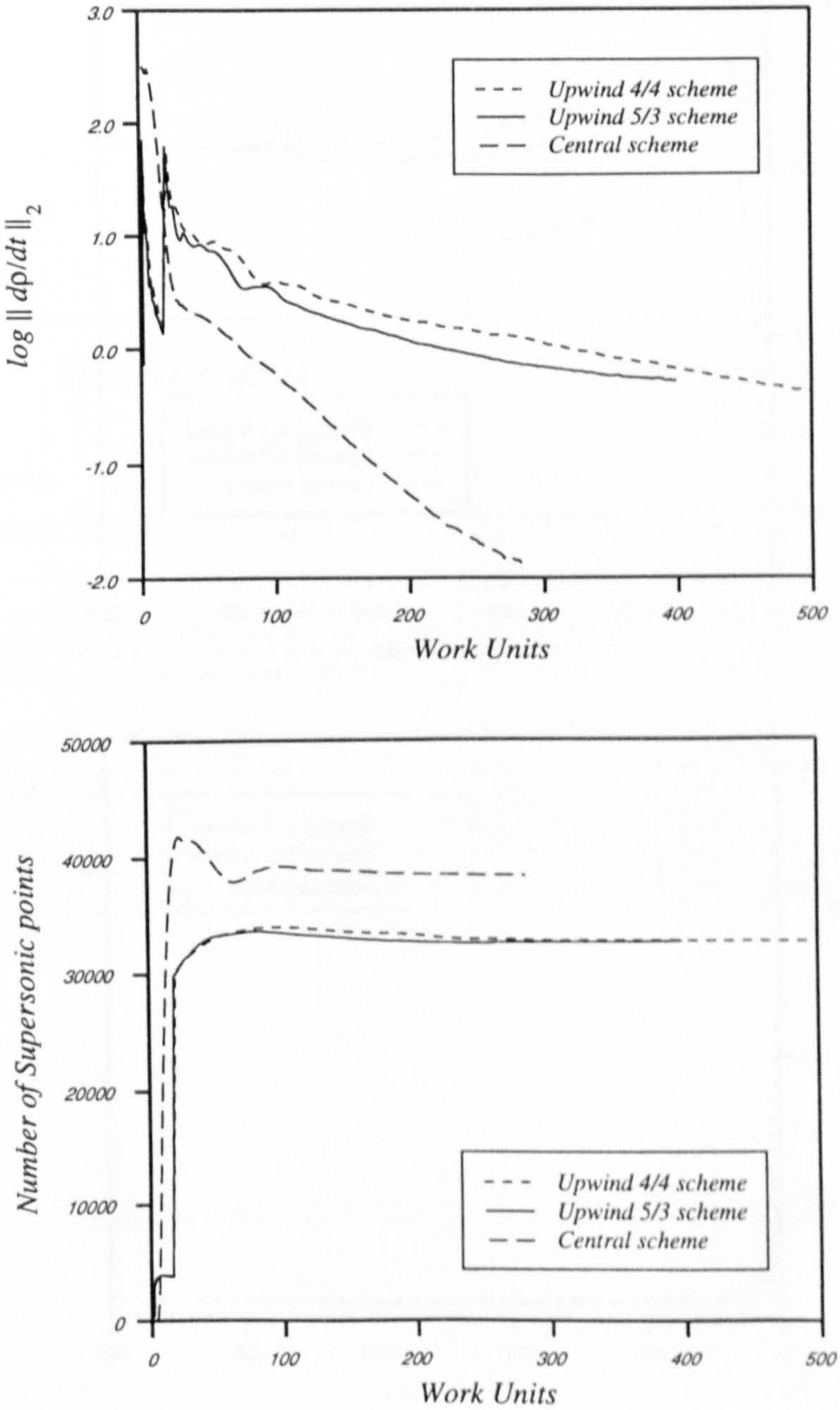


FIGURE 5.18. Comparison of convergence histories for ONERA M6 wing from central solution and two upwind solutions.

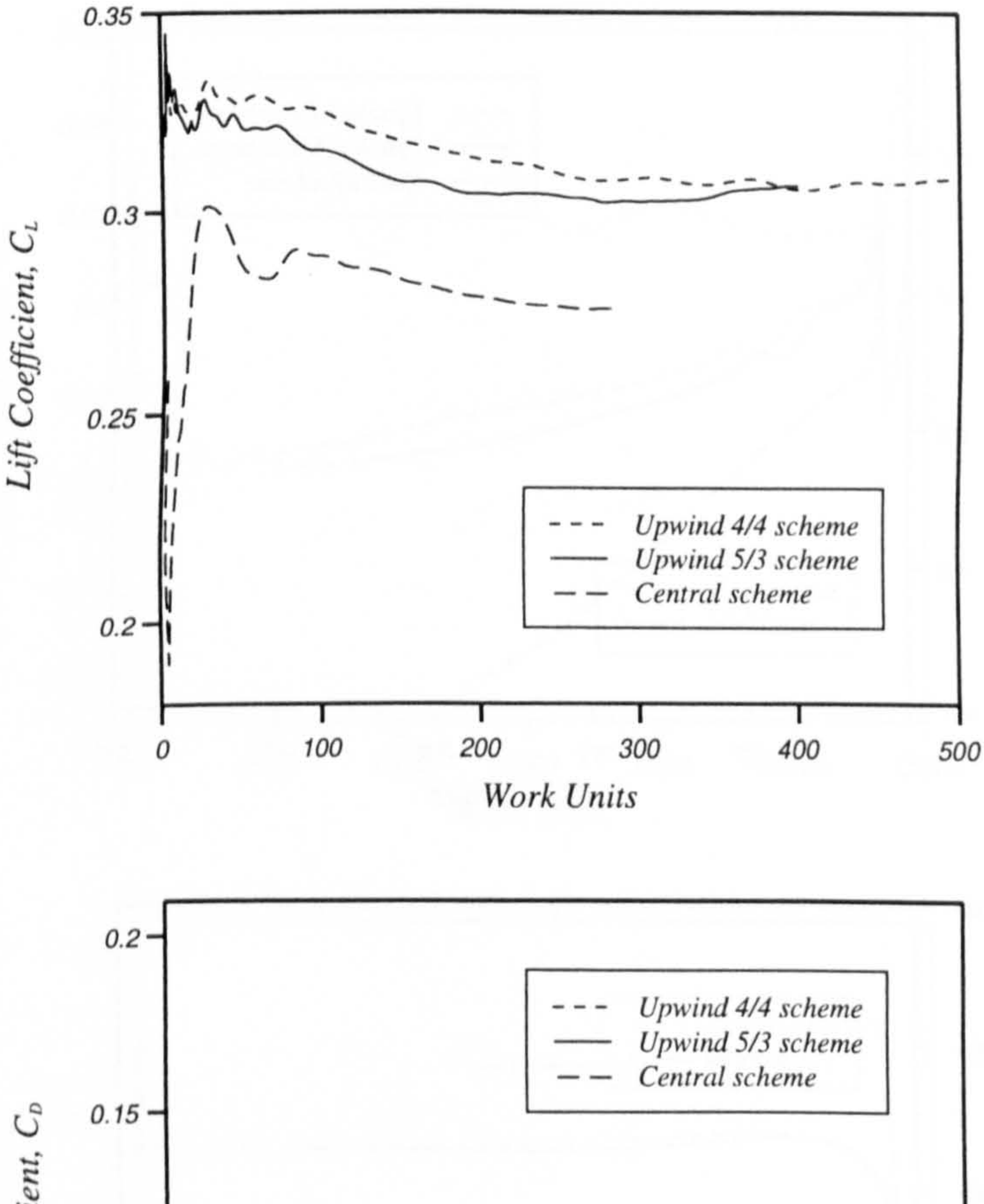


FIGURE 5.19. Comparison of force coefficient convergence for ONERA M6 wing from central solution and two upwind solutions.

CENTRAL SCHEME

```
                    Perfview Observations Report

Observations:

This program achieved 110.8 million floating-point
operations per second during its execution.

This program appears to be an efficient vectorized code.

This program showed a computational intensity of 0.76.
(Computational intensity is the ratio of flops to memory
references.) This value is considered to be moderate,
indicating an algorithm with mid-range efficiency.


User Program Was Run for HPM Groups:
       0: on 03/17/95, at 15:18:32
Machine Serial Number 1909, a CRAY Y-MP,
  with a Clock Speed of 6000 Picoseconds
Program was running in Y mode.
AVL was enabled during this program's execution.
High-water mark for the stack was 21523 (octal words)
     or 9043 (decimal words).
Original Executable Filename = /[redacted]/ns87
  Timestamp for this File: Fri Mar 17 14:42:37 1995
```

UPWIND SCHEME

```
                    Perfview Observations Report

Observations:

This program achieved 133.6 million floating-point
operations per second during its execution.

This program appears to be an efficient vectorized code.

This program showed a computational intensity of 1.39.
(Computational intensity is the ratio of flops to memory
references.) This value is considered to be very good,
indicating an efficient algorithm.


User Program Was Run for HPM Groups:
       0: on 03/17/95, at 18:08:28
Machine Serial Number 1909, a CRAY Y-MP,
  with a Clock Speed of 6000 Picoseconds
Program was running in Y mode.
AVL was enabled during this program's execution.
High-water mark for the stack was 22606 (octal words)
     or 9606 (decimal words).
Original Executable Filename = /[redacted]/nsr3d
  Timestamp for this File: Thu Mar 16 14:28:46 1995
```

FIGURE 5.20. Extracts from Cray Perftraces obtained for central and upwind solvers.

6

The Shock Acceleration Method

Many of the difficulties encountered with systems of conservation laws are common to their scalar counterparts. Thus, an appropriately chosen scalar equation may serve as a simple model that affords a useful starting point for obtaining a good understanding of fundamental behaviour before proceeding to consider the system of primary interest. This chapter begins by investigating a scalar equation that may be considered an analogue for the inhomogeneous Euler equations for flow through a duct or nozzle. This equation is examined at the analytic level and at the discrete level using the Roe and the Engquist-Osher scheme. Then a new approach to convergence acceleration called the *shock acceleration method* is proposed for these schemes. It is founded on a localized treatment of the transition points in a shock wave and works by damping the high-frequency error modes associated with moving shock waves. Several acceleration models are formulated for the scalar case and numerical results are presented which demonstrate the efficacy of this new approach.

After consideration of the scalar case, the Euler equations for flow through a duct are examined at both the analytic and discrete levels. The baseline numerical algorithm considered here is the Roe scheme which is extended to inhomogeneous problems via source-term splitting. The latter technique ensures that the source terms are discretized consistently with the convection term. A particular shock acceleration model derived for the scalar case is then extended to produce a formulation for the system of equations. The feasibility of the new accelerated scheme is demonstrated by the numerical results presented for the quasi one-dimensional Euler equations for duct flow.

6.1 Inhomogeneous Scalar Conservation Laws

6.1.1 A MODEL EQUATION WITH SLOW CONVERGENCE

With a view to obtaining an insight into the slow convergence phenomena of high resolution methods and, to devising methods of accelerating their convergence, a model equation which exhibits slow convergence was investigated. It is a time-dependent inhomogeneous, nonlinear partial differential equation of the form

$$u_t + f_x = q(u, x) \tag{6.1}$$

which is satisfied by $u \equiv u(x,t)$ and, in which f defines the flux function and q a nonlinear source term. The subscripts t and x denote partial derivatives with respect to time and space. The precise form of the equation considered here was

devised by Roe and van Leer [134] and is given by

$$u_t + \left(u + \frac{1}{u}\right)_x = \frac{kx}{u} \tag{6.2}$$

in the interval $x : [-1, 1]$ for $t \geq 0$, subject to the boundary conditions $u_{b-} \equiv u(-1, t)$, $u_{b+} \equiv u(1, t)$. It may be considered a scalar analogue to transonic flow through an isothermal nozzle of area distribution $A = A_o exp(kx^2/2)$. The non-dimensional solution variable u is identified with Mach number while the parameter k is constant. For an exposition on non-existence and non-uniqueness for a class of conservation laws which includes the present model equation, the interested reader is referred to the paper by Roe and van Leer [134].

The analytic solution of the model equation is discussed in the next section. Following that, the numerical solutions of two upwind methods are described, namely that of Roe [126] and, that of Enquist-Osher [42].

6.1.2 The analytic solution

To understand the model equation, it will be fruitful to begin with its homogeneous counterpart. In the steady-state, the homogeneous form of equation (6.2) is

$$\frac{d}{dx}\left(u + \frac{1}{u}\right) = 0 \tag{6.3}$$

Integrating with respect to x

$$\int \frac{d}{dx}\left(u + \frac{1}{u}\right) dx = 0 \tag{6.4}$$

produces

$$u + \frac{1}{u} = b \tag{6.5}$$

where b is the constant of integration. This equation may be written in a more familiar quadratic form as

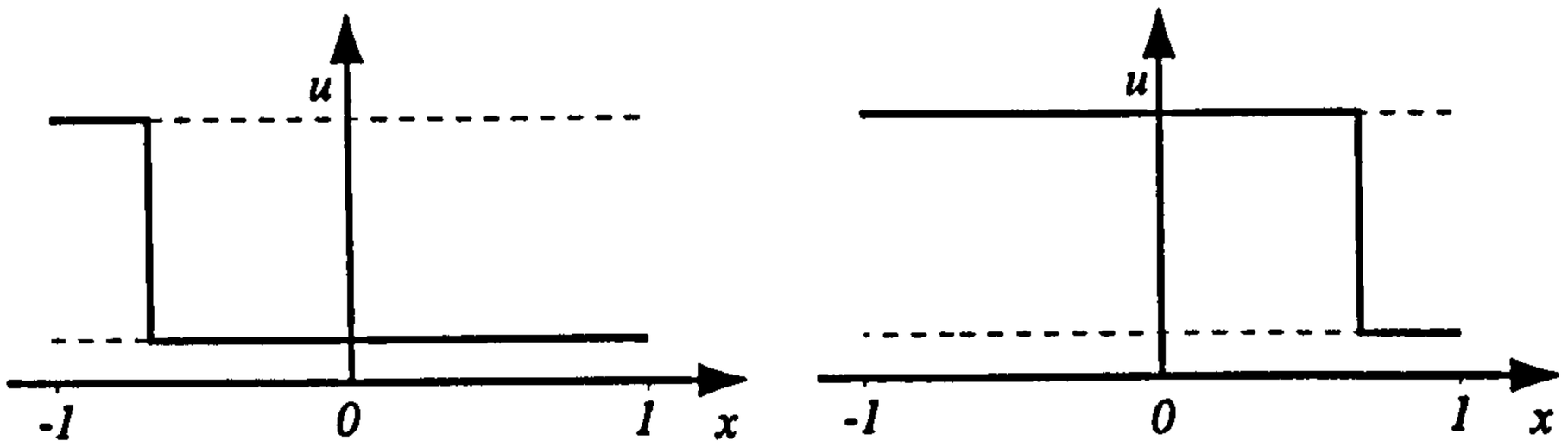
$$u^2 - bu + 1 = 0 \tag{6.6}$$

Denoting the branches of its roots as $u_-(x)$ and $u_+(x)$, it is found that

$$u_-(x) + u_+(x) = b \tag{6.7}$$

$$u_-(x)u_+(x) = 1 \tag{6.8}$$

The branches of the solutions to equation (6.3) are indicated in Figure 6.1 by dashed lines with two possible solutions superimposed. In part (a) of the figure, the solid line denotes an unstable root while in part (b), the solid line denotes the stable root. In the latter, the value at the head of the discontinuity is u_{s-} while that at the tail is u_{s+}. The solution indicates constant states existing either side of the discontinuity. Indeed this is evident from equation (6.3) in which the derivative of the flux function is zero. The discontinuity is readily identified as a shock wave. One may also observe from equation (6.8) that the value at the right-hand boundary is the reciprocal value of the left-hand boundary.



(a) Unstable root *(b) Stable root*

FIGURE 6.1. Analytic solutions of the equation $[u + 1/u]_x = 0$.

From Rankine-Hugoniot shock relations, the speed of a shock wave $a(u)$ may be expressed in terms of the jump condition

$$a(u) = \frac{\Delta f}{\Delta u}$$

For the model equation, the speed of the shock wave is

$$\begin{aligned} a(u) &= \left[\left(u_{s+} + \frac{1}{u_{s+}}\right) - \left(u_{s-} + \frac{1}{u_{s-}}\right)\right] / (u_{s+} - u_{s-}) \\ &= 1 - \frac{1}{u_{s-} u_{s+}} \end{aligned} \tag{6.9}$$

In the steady-state the wave speed is zero. Thus

$$u_{s-} u_{s+} = 1 \qquad \text{(condition I)} \tag{6.10}$$

The steady shock relationship (condition I) is a particular case of equation (6.8). In the inhomogeneous case, condition I is sufficient for a stable shock to exist. However, in the homogeneous case, although condition I is a necessary condition, a further condition is required to ensure conservation. This second condition determines the shock position and is given by

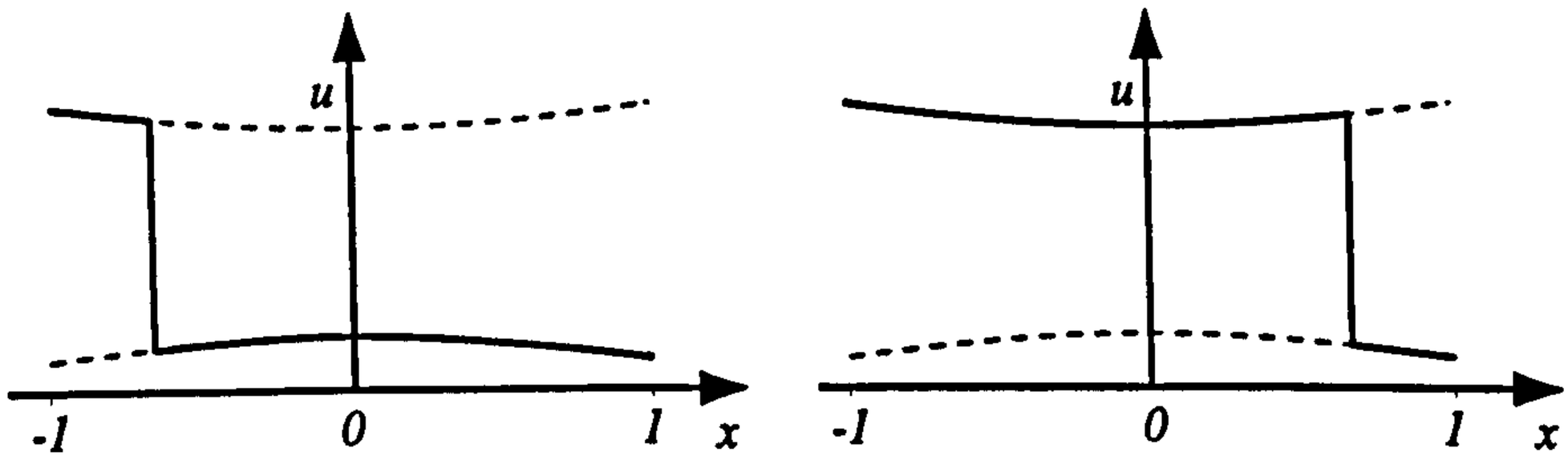
$$(1 + x_s)u_- + (1 - x_s)u_+ = \int_{-1}^{+1} u(x)dx \quad \text{(condition II)} \tag{6.11}$$

As conservation is not strictly valid when source terms are present, condition II is *not* applicable to the inhomogeneous case. Furthermore, unstable solutions are not precluded when these conditions hold, be they homogeneous or otherwise. Where they do hold, it is clear that

$$\begin{aligned} u(x) &> 1 \qquad \text{(supercritical case)} \\ u(x) &= 1 \qquad \text{(critical case)} \\ u(x) &< 1 \qquad \text{(subcritical case)} \end{aligned} \tag{6.12}$$

By analogy to real flow, the supercritical case is equivalent to supersonic flow, the subcritical one equivalent to subsonic flow. When $u = 1$, there is a sonic point in the solution.

Returning to the inhomogeneous case, it is obvious that like its homogeneous counterpart, the inhomogeneous case may admit two possible steady solutions. These are illustrated in Figure 6.2. Now, considering equation (6.2) at steady-



(a) Unstable root *(b) Stable root*

FIGURE 6.2. Analytic solutions of the equation $[u + 1/u]_x = kx/u$.

state and differentiating its left-hand side with respect to x

$$\left(1 - \frac{1}{u^2}\right)\frac{du}{dx} = \frac{kx}{u} \tag{6.13}$$

multiplying through by u, separating variables and integrating with respect to x as

$$\int \frac{du}{dx}\left(u - \frac{1}{u}\right)dx - \int kx dx = 0 \tag{6.14}$$

has the solution

$$\frac{1}{2}u^2 - \ln u - \frac{1}{2}kx^2 = c \tag{6.15}$$

where c is the constant of integration. Using this result, the value at any arbitrary position along a branch of the solution may be determined in terms of its respective boundary value. If a candidate shock point position $x = x_c$ is chosen and substituted into the above equation for each solution branch, the equations

$$\frac{1}{2}(u_{c-}^2 - u_{b-}^2) - \ln\left(\frac{u_{c-}}{u_{b-}}\right) = \frac{1}{2}k(x_{c-}^2 - x_{b-}^2) \tag{6.16}$$

$$\frac{1}{2}(u_{c+}^2 - u_{b+}^2) - \ln\left(\frac{u_{c+}}{u_{b+}}\right) = \frac{1}{2}k(x_{c-}^2 - x_{b+}^2) \tag{6.17}$$

emerge. To ascertain whether a candidate position is indeed that of a permissible steady shock, one need only substitute the trial solutions $u_{c\pm}$ obtained from equations (6.16) and (6.17) into equation (6.10). If the latter does not hold for

the candidate value $x_c : [-1, 1]$, a weak solution cannot exist at that point for the given boundary data. However, if it does hold, a valid weak solution occurs at $x_s = x_c$ such that

$$f_{s-} = f_{s+} \tag{6.18}$$

i.e. the flux either side of the shock is in a state of balance. On finding a weak solution, there remains the question of determining whether it is stable or unstable. To do this, a spatial perturbation δx_s is applied. Then the perturbed solution $u'_{\pm}$ corresponding to the position $x' = (x_s + \delta x_s)$ may be calculated from equations (6.16) and (6.17) using x' in place of x_c. The *perturbed* wave speed S' is calculated as

$$S' = 1 - \frac{1}{u'_+ u'_-} \tag{6.19}$$

The shock wave is stable *iff* $S' < 0$.

6.1.3 The numerical solution –the Roe scheme

The upwind method proposed by Roe in [126] for the homogeneous scalar conservation law

$$u_t + f(u)_x = 0 \tag{6.20}$$

is now considered in the context of the model problem. Equation (6.2) may be written in the form

$$u_t + a(u)u_x = q(u, x), \qquad a(u) = \frac{df}{du} \tag{6.21}$$

The solution domain is discretized into finite difference cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ with centroids x_i. A piecewise constant mesh function W takes on cell centred values as $W_i = W(x_i)$. The following notation for differences is used:

$$\Delta W_{i+\frac{1}{2}} = \Delta_- W_{i+1} = \Delta_+ W_i = W_{i+1} - W_i \tag{6.22}$$

and the grid ratio μ, is defined by

$$\mu \equiv \Delta t / \Delta x \tag{6.23}$$

The discrete characteristic wave speed $a_{i+\frac{1}{2}} = a(u_i, u_{i+1})$ is given by

$$a_{i+\frac{1}{2}} = \frac{\Delta f_{i+\frac{1}{2}}}{\Delta u_{i+\frac{1}{2}}} = 1 - \frac{1}{u_i u_{i+1}} \tag{6.24}$$

while the local Courant number ν, is

$$\nu_{i+\frac{1}{2}} = \mu a_{i+\frac{1}{2}} \tag{6.25}$$

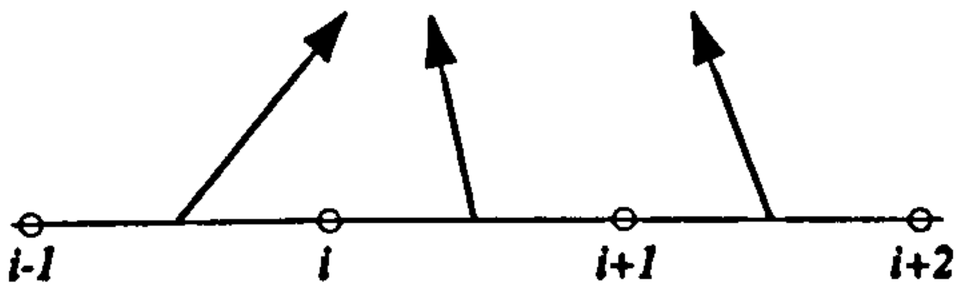
For each interval $[i, i+1]$, a residual $\phi_{i+\frac{1}{2}}$ is computed as

$$\phi_{i+\frac{1}{2}} = \frac{f_i - f_{i+1}}{\Delta x} + \bar{q}_{i+\frac{1}{2}} \tag{6.26}$$

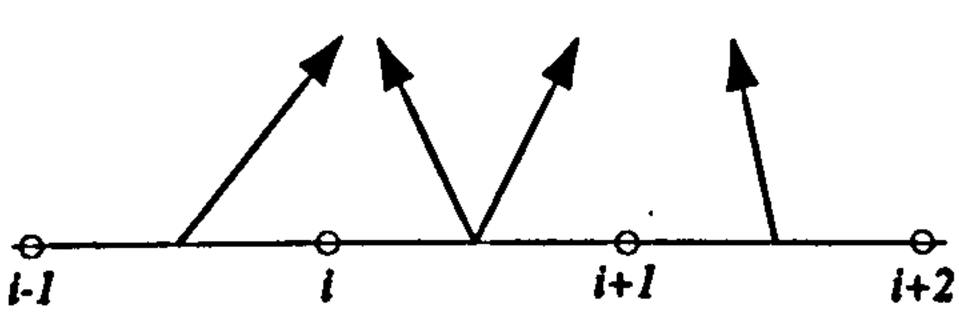
where $\bar{q}_{i+\frac{1}{2}}$ is a discrete approximation to the source term, evaluated at the cell interface. The approximation chosen here is the mean average given by

$$\bar{q}_{i+\frac{1}{2}} = k(x_i + x_{i+1})\left(\frac{1}{u_i} + \frac{1}{u_{i+1}}\right) \tag{6.27}$$

At any given iteration, non-shock cells, including ones through which the flow may be expanding, receive their upwind information via a single interface component $\phi_{i\pm\frac{1}{2}}$. In contrast, at a shock the situation is different with the flow of information being as that illustrated by Figure 6.3(a). In the diagram, space is



(a) The Roe Method (shock located at cell i)



(b) The Engquist-Osher Method (shock located between cells i and i+1)

FIGURE 6.3. Flow of information at a shock wave for the model equation using using two different upwind methods.

shown to vary in the horizontal direction while a notional time axis is implied in the vertical direction. The arrows shown indicate the direction in the x-t plane in which the residual components $\phi_{i\pm\frac{1}{2}}$ are acting and hence the cells to which they contribute. It is the single cell in which the shock is located, cell i in the diagram, that receives information from both interfaces either side of it. If the scheme converges to a steady-state, it does so when

$$\phi_{i+\frac{1}{2}} = 0 \qquad \text{(for a non-shock point)} \tag{6.28}$$

$$\phi_{i+\frac{1}{2}} + \phi_{i-\frac{1}{2}} = 0 \qquad \text{(for a shock point, } s = i\text{)} \tag{6.29}$$

Adopting a backward Euler update to approximate the temporal derivative with successive time levels denoted by superscripts $(n, n+1)$, the first-order scheme becomes

$$u_l^{n+1} = u_l^n + \Delta t_l \phi_{i+\frac{1}{2}}^n \tag{6.30}$$

in which the upwind counter l determines the interface to which the residual $\phi^n_{i+\frac{1}{2}}$ is to be sent, according to the following logic

$$\begin{aligned} l &= i+1, && \text{for } (a_{i+\frac{1}{2}} \geq 0) \\ l &= i, && \text{for } (a_{i+\frac{1}{2}} < 0) \end{aligned} \tag{6.31}$$

The update is formally first-order accurate in space and time. As described, both source and convection terms are upwind-biased. This difference scheme may be cast in an alternative form as

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right) + \frac{\Delta t}{2}\left(\bar{q}_{i-\frac{1}{2}} + \bar{q}_{i+\frac{1}{2}}\right) \tag{6.32}$$

in which the numerical flux function is given by

$$f^{Roe}_{i+\frac{1}{2}}(u_i, u_{i+1}) = \frac{1}{2}\left[f(u_i) + f(u_{i+1})\right] - \frac{1}{2}\left[|a|_{i+\frac{1}{2}}\,\Delta u_{i+\frac{1}{2}} - |\bar{q}|_{i+\frac{1}{2}}\,\Delta x\right] \tag{6.33}$$

The method is stable and free from oscillations for

$$|\nu| \leq 1 \tag{6.34}$$

Although a first-order time stepping scheme, equations (6.30) and (6.33) become second-order accurate in the steady-state: each flux difference is centred on the same interval as the source term that must balance it [168]. As described, the Roe scheme is not entropy-satisfying in the sense that where a sonic expansion is present, a mathematically correct solution will result that may be physically implausible (viz. an expansion shock). However, such adverse behaviour is readily corrected by simple modification of the scheme as described in the literature. For example see [168]. In the model problem of the present study, boundary conditions and initial data were chosen that exclude non-entropy-satisfying solutions as they are not central to present purposes. Finally, it is noted that the convergence of the Roe scheme has been proven by Sweby and Baines [148] for a generalized nonlinear scalar wave equation.

6.1.4 The numerical solution –the Engquist-Osher scheme

In the upwind method of Engquist-Osher [42], each possible state of fluid is considered as a point in phase space in which all changes are brought about by the action of the characteristics. Taking u_L and u_R to be fixed points in phase space, there is a unique path Γ linking them. A formal expression for the Enquist-Osher interface flux is

$$h^{EO}_{i+\frac{1}{2}}(u_L, u_R) = \frac{1}{2}(f_L + f_R) - \frac{1}{2}\int_{u_L}^{u_R} |S(u)|\,du \tag{6.35}$$

in which the integral follows the path Γ. For a homogeneous problem, $S(u)$ is the wave speed at each point along the integration path. However for the inhomogeneous problem in which both the source and convection terms are upwinded, the function includes the source term as

$$S(u) = \frac{\partial}{\partial u}(q - f) \tag{6.36}$$

In order to evaluate the integral of equation (6.35), one must first determine the local nature of the flow. Figure 6.4 illustrates the four generic types of flow that may be encountered, namely right moving flow, left moving flow, flow expansion (an expansion wave) and, flow compression (a shock wave). The arrows in the figure indicate the direction of the flow in the x-t plane. An arrow pointing towards the left indicates a negative wave velocity while one pointing towards the right indicates a positive wave velocity. Evaluation of the integral above is carried out with reference to these types of flow. For convenience, the numerical counters $(i, i+1)$ may be used interchangeably with the indices (L, R). The case of flow in the absence of a sonic region is considered first. Physically, this situation corresponds to flow moving to the left or to the right without any change in the sign of the characteristics. Equation (6.1) is re-written as

$$\frac{\partial u}{\partial t} = q(u,x) - \frac{\partial f}{\partial x} \tag{6.37}$$

Integrating with respect to x in the interval $[x_i, x_{i+1}]$ one has

$$\int_{x_i}^{x_{i+1}} \frac{\partial u}{\partial t} dx = \frac{\partial u}{\partial t}\Delta x = \int_{x_i}^{x_{i+1}} \left(q - \frac{\partial f}{\partial x}\right) dx \tag{6.38}$$

in which $\Delta x \equiv x_{i+1} - x_i$. For brevity the wave speed will be denoted as a. Using the identity

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} = a\frac{\partial u}{\partial x} \tag{6.39}$$

together with the definition

$$\frac{\partial u}{\partial t} \equiv \Delta u^-(u_i) + \Delta u^+(u_{i+1}) = \Delta u^- + \Delta u^+ \tag{6.40}$$

produces

$$\Delta u^- + \Delta u^+ = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} a\left(\frac{q}{a} - \frac{\partial u}{\partial x}\right) dx \tag{6.41}$$

The terms Δu^- and Δu^+ represent the upwind contributions to the left and right end points of the integration path. The wave speed may be decomposed as

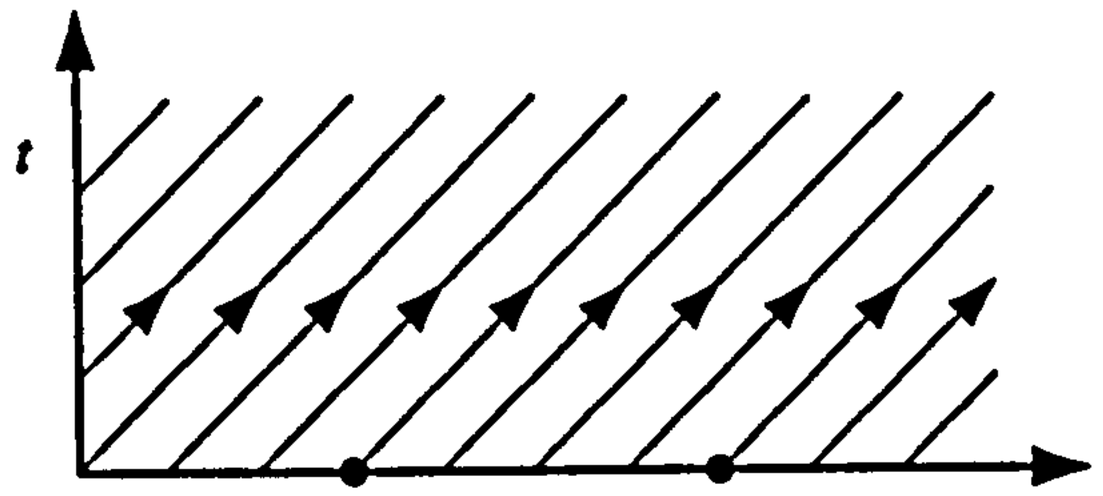
$$a \equiv a^+ + a^- \tag{6.42}$$

The component a^+ represents waves with positive velocity while a^- represents those that are of negative velocity. The absolute wave speed is

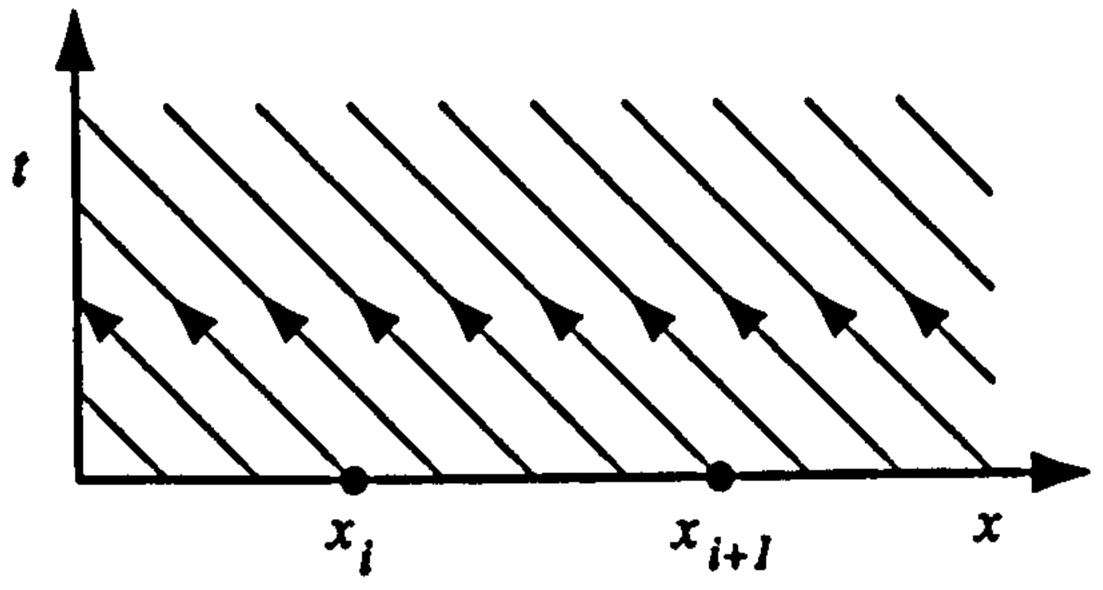
$$|a| \equiv a^+ - a^- \tag{6.43}$$

With the aid of these definitions, equation (6.41) may be expressed as

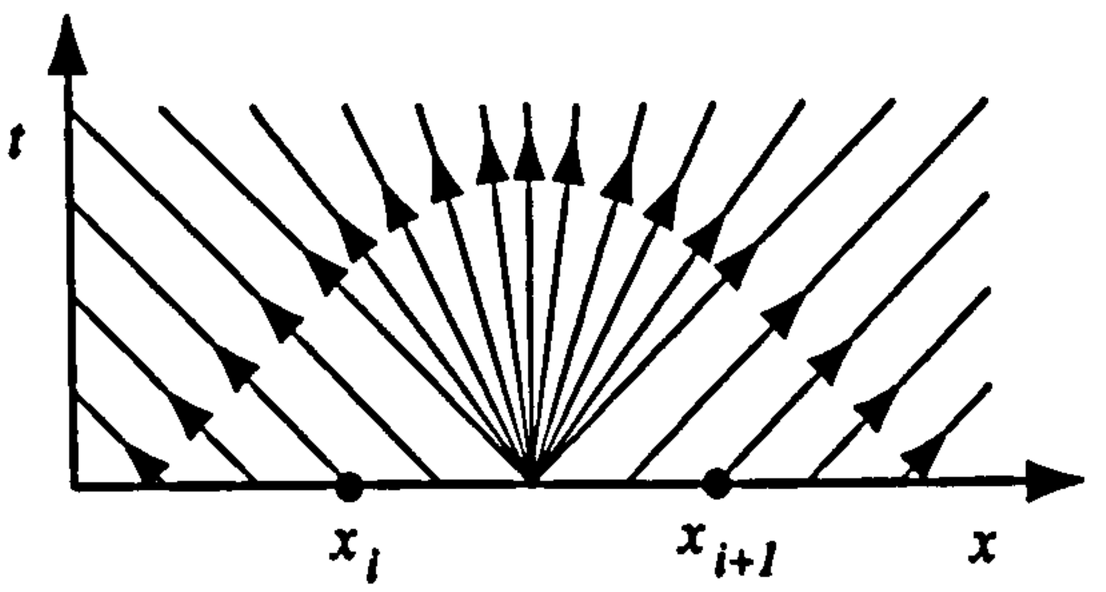
$$\begin{aligned} \Delta u^- + \Delta u^+ &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+1}} \frac{q}{a}(a^+ + a^-)\, dx - \int_{x_i}^{x_{i+1}} \frac{\partial u}{\partial x}(a^+ + a^-) dx\right] \\ &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+1}} \frac{q}{a}(a^+ + a^-)\, dx - \int_{u_i}^{u_{i+1}} (a^+ + a^-) du\right] \end{aligned} \tag{6.44}$$



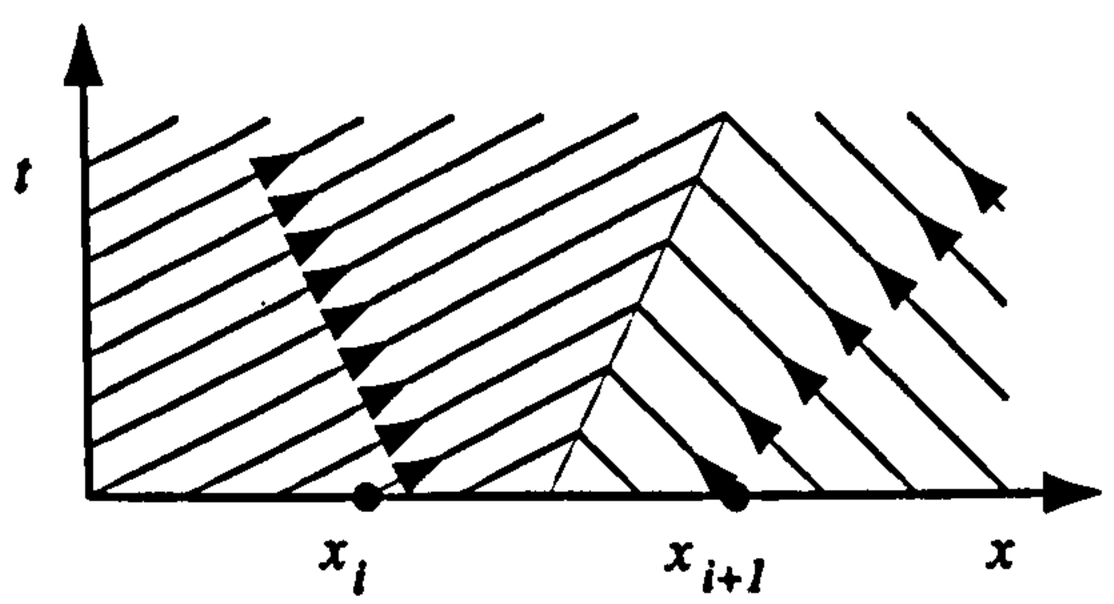
Case (i): Right moving flow



Case (ii): Left moving flow



Case (iii): Flow expansion



Case (iv): Flow compression (shockwave)

FIGURE 6.4. Generic types of flow.

Now consider the special case of a sonic region occurring within the interval $[x_i, x_{i+1}]$. (Conditions at the sonic point will be denoted by the subscript s). Physically, this situation arises as either an expansion of flow through a sonic region or as a compression that may result in a shock wave. To account for the change in sign of the characteristics that occur at such singularities, the path of integration Γ is divided into two segments corresponding to the interval $x : [i, s)$ to the left of the sonic point s and $x : (s, i+1]$ to the right of it. Accordingly, equation (6.44) becomes

$$\Delta u^- + \Delta u^+ = \frac{1}{\Delta x}\left[\int_{x_i}^{x_{\inf s}} \frac{q}{a}\left(a^+ + a^-\right) dx + \int_{x_{\sup s}}^{x_{i+1}} \frac{q}{a}\left(a^+ + a^-\right) dx \right.$$
$$\left. - \int_{u_i}^{u_{\inf s}} \left(a^+ + a^-\right) du - \int_{u_{\sup s}}^{u_{i+1}} \left(a^+ + a^-\right) du\right] \qquad (6.45)$$

Henceforth, whenever the theoretical distinction between $\inf s, s, \sup s$ is not relevant to implementation detail, the notation s will be adopted preferentially. From the last equation, it is evident that the source term must be decomposed into constituent components corresponding to the local nature of flow conditions in the interval. Equations (6.44) and (6.45) are the general expressions that are evaluated below to produce the algebraic flux formulae needed to update the numerical solution of the model problem. Now consider each type of flow in turn.

Case (i): Right moving flow $\quad u_i, u_{i+1} > 1$

Since all flow in the interval is right moving, the characteristic wave speed $a > 0$, and the identities

$$\frac{a^+}{a} = 1, \qquad \frac{a^-}{a} = 0 \qquad (6.46)$$

are applicable. Substituting them into equation (6.44) produces

$$\Delta u^- = 0 \qquad (6.47)$$

$$\Delta u^+ = \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+1}} q dx - \int_{u_i}^{u_{i+1}} a^+ du\right]$$
$$= \frac{1}{\Delta x}\left\{\bar{q}_{i+\frac{1}{2}} - \left[u + \frac{1}{u}\right]_i^{i+1}\right\}$$
$$= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{i+\frac{1}{2}}\right] \qquad (6.48)$$

The term $\bar{q}_{i+\frac{1}{2}}$ represents the discretized contribution of the source term over the given interval. Intentionally, the numerical approximation used for its evaluation was chosen to be identical to that used in section 6.1.3 for the Roe scheme (i.e. equation (6.27)).

Case (ii): Left moving flow $\quad u_i, u_{i+1} < 1$

With all flow in the interval being left moving, the wave speed is $a < 0$, and the identities

$$\frac{a^+}{a} = 0, \qquad \frac{a^-}{a} = 1 \qquad (6.49)$$

hold. Substitution into equation (6.44) gives

$$\begin{aligned}
\Delta u^- &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+1}} q dx - \int_{u_i}^{u_{i+1}} a^- du\right] \\
&= \frac{1}{\Delta x}\left\{\bar{q}_{i+\frac{1}{2}} - \left[u + \frac{1}{u}\right]_i^{i+1}\right\} \\
&= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{i+\frac{1}{2}}\right] \qquad (6.50) \\
\Delta u^+ &= 0 \qquad (6.51)
\end{aligned}$$

Case (iii): Flow expansion $u_i \leq 1, u_{i+1} \geq 1$
The wave speed is of mixed sign such that in the interval to the left of the sonic point, $a : [i, s) < 0$ and

$$\frac{a^+}{a} = 0, \quad \frac{a^-}{a} = 1, \quad \forall x : [i, s) \qquad (6.52)$$

Substituting these identities into equation (6.45) produces

$$\begin{aligned}
\Delta u^- &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_s} q dx - \int_{u_i}^{u_s} a^- du\right] \\
&= \frac{1}{\Delta x}\left\{\bar{q}_{s_L} - \left[u + \frac{1}{u}\right]_{u_i}^{1}\right\} \\
&= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - 2 + \bar{q}_{s_L}\right] \qquad (6.53)
\end{aligned}$$

The term $\bar{q}_{s_L}$ represents the discretized contribution of the source term along the interval to the left of the sonic point. For flow in the interval to the right of the sonic point in which $a : (s, i+1] > 0$, the identities become

$$\frac{a^+}{a} = 1, \quad \frac{a^-}{a} = 0, \quad \forall x : (s, i+1] \qquad (6.54)$$

Substitution into equation (6.45) renders

$$\begin{aligned}
\Delta u^+ &= \frac{1}{\Delta x}\left[\int_{x_s}^{x_{i+1}} q dx - \int_{u_s}^{u_{i+1}} a^+ du\right] \\
&= \frac{1}{\Delta x}\left\{\bar{q}_{s_R} - \left[u + \frac{1}{u}\right]_1^{i+1}\right\} \\
&= \frac{1}{\Delta x}\left[2 - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{s_R}\right] \qquad (6.55)
\end{aligned}$$

The term $\bar{q}_{s_R}$ represents the discretized contribution of the source term along the interval to the right of the sonic point.

Case (iv): Flow compression (shock wave) $u_i \geq 1, u_{i+1} \leq 1$

The wave speed has mixed sign being opposite to that of a flow expansion. In the interval to the left of the sonic point, $a : [i, s) > 0$ and

$$\frac{a^+}{a} = 1, \quad \frac{a^-}{a} = 0, \quad \forall x : [i, s) \tag{6.56}$$

Substitution into equation (6.45) produces

$$\begin{aligned} \Delta u^+ &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_s} q dx - \int_{u_i}^{u_s} a^+ du\right] \\ &= \frac{1}{\Delta x}\left\{\bar{q}_{s_L} - \left[u + \frac{1}{u}\right]_{u_i}^{1}\right\} \\ &= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - 2 + \bar{q}_{s_L}\right] \end{aligned} \tag{6.57}$$

For flow to the right of the sonic point, $a : (s, i+1] < 0$ and

$$\frac{a^+}{a} = 0, \quad \frac{a^-}{a} = 1, \quad \forall x : (s, i+1] \tag{6.58}$$

Substitution into equation (6.45) gives

$$\begin{aligned} \Delta u^- &= \frac{1}{\Delta x}\left[\int_{x_s}^{x_{i+1}} q dx - \int_{u_s}^{u_{i+1}} a^- du\right] \\ &= \frac{1}{\Delta x}\left\{\bar{q}_{s_R} - \left[u + \frac{1}{u}\right]_{1}^{u_{i+1}}\right\} \\ &= \frac{1}{\Delta x}\left[2 - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{s_R}\right] \end{aligned} \tag{6.59}$$

In order to discretize the component source terms $\bar{q}_{s_L}$ and $\bar{q}_{s_R}$ needed for cases (iii) and (iv) consistently with the overall source contribution, given by equation (6.27), the relationship

$$\bar{q}_{i+\frac{1}{2}} = \bar{q}_{s_L} + \bar{q}_{s_R} \tag{6.60}$$

must hold. The source components $\bar{q}_{s_L}$ and $\bar{q}_{s_R}$ are obtained by splitting equation (6.27) according to the above relation such $\bar{q}_{s_L}$ is defined over the interval $[i, s)$ and $\bar{q}_{s_R}$ is similarly defined over $(s, i+1]$. The resulting expressions are then integrated and after algebraic manipulation, the source term components for sonic regions are found to be

$$\bar{q}_{s_L} = k\left(\frac{x_i^2 - x_s^2}{x_i - x_{i+1}}\right)\left(1 + \frac{1}{u_i}\right) \tag{6.61}$$

$$\bar{q}_{s_R} = k\left(\frac{x_s^2 - x_{i+1}^2}{x_i - x_{i+1}}\right)\left(1 + \frac{1}{u_{i+1}}\right) \tag{6.62}$$

The position of the sonic point x_s needed in equations (6.61) and (6.62) is found by equating the individual source components given by equations (6.61–6.62) with the overall source contribution, equation (6.27). After further manipulation it may be expressed as

$$x_s^2 = \left[x_i^2\left(\frac{1}{u_{i+1}} - 1\right) + x_{i+1}^2\left(1 - \frac{1}{u_i}\right)\right] / \left(\frac{1}{u_{i+1}} - \frac{1}{u_i}\right) \tag{6.63}$$

Finally, the update procedure for the Engquist-Osher scheme may be implemented in a computer program as follows: compute Δu_i^- and Δu_i^+ for each cell in the domain according to the local nature of the flow (i.e. right moving, left moving, flow expansion or, flow compression), summing the contributions that make up the residual R_i, for each cell in a single array such that

$$R_i = \Delta u_i^- + \Delta u_i^+$$

Then apply the update

$$u_i^{n+1} = u_i^n + \Delta t R_i \tag{6.64}$$

across the domain. For convenience, the expressions for the terms Δu_i^- and Δu_{i+1}^+ are summarized here:

$u_i, u_{i+1} > 1$:

$$\begin{aligned} \Delta u^- &= 0 \\ \Delta u^+ &= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{i+\frac{1}{2}}\right] \end{aligned}$$

$u_i, u_{i+1} < 1$:

$$\begin{aligned} \Delta u^- &= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{i+\frac{1}{2}}\right] \\ \Delta u^+ &= 0 \end{aligned}$$

$u_i \leq 1, u_{i+1} \geq 1$:

$$\begin{aligned} \Delta u^- &= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - 2 + \bar{q}_{s_L}\right] \\ \Delta u^+ &= \frac{1}{\Delta x}\left[2 - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{s_R}\right] \end{aligned}$$

$u_i \geq 1, u_{i+1} \leq 1$:

$$\begin{aligned} \Delta u^+ &= \frac{1}{\Delta x}\left[\left(u_i + \frac{1}{u_i}\right) - 2 + \bar{q}_{s_L}\right] \\ \Delta u^- &= \frac{1}{\Delta x}\left[2 - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{s_R}\right] \end{aligned}$$

Note that in regions of flow advancing to the left, each cell i receives a single upwind update from Δu_i^-, while in flow advancing to the right, each cell i receives its upwind update from Δu_i^+. For a shock wave the flow of information at a given point in time is illustrated in Figure 6.3(b). Here, distance is shown to vary horizontally, while the time axis (not shown), is vertical. The arrows indicate the direction in the x-t plane in which the residual components $\Delta u^{\pm}$ act and hence the cells to which they contribute. Those pointing to the left represent the propagation of Δu^- information while those pointing to the right represent that of Δu^+ information. As shown in the diagram, the shock is located between the centres of cells i and $i+1$, but not necessarily at the interface between them. The method produces two transition points in the shock, cell i which contains

supercritical flow and, cell $i+1$ containing subcritical flow. Thus the *pair* of cells straddling the shock, receives updates from Δu^- and Δu^+. (Similarly with expansions through sonic regions, the pair of cells that straddle the $M = 1$ singularity, receive updates from Δu^- and Δu^+). Thus the residuals for the shock transition cells are composed as

$$R_i = \Delta u_i^- + \Delta u_i^+ \tag{6.65}$$

and

$$R_{i+1} = \Delta u_{i+1}^- + \Delta u_{i+1}^+ \tag{6.66}$$

In contrast, the Roe scheme produces a single transition point representing shock location in the discrete sense, that may contain subcritical, supercritical or even critical flow, cf. Figure 6.3(a). Hence one of the principal differences between the two schemes is apparent, namely, the Engquist-Osher scheme captures a shock with two-point resolution while the Roe scheme renders shocks with single-point resolution.

As described above, the Engquist-Osher scheme is first-order accurate (in space and time) with both the source and convection terms being upwinded. It is stable for

$$|\nu| \leq 1$$

Unlike the Roe scheme, in the steady-state, it does not achieve second-order accuracy. A variety of properties have been proven in [42] for solutions produced by the Engquist-Osher scheme. These include sharp monotone shock profiles, existence and uniqueness of solution. The scheme is also monotone for arbitrary flux functions and therefore its approximate solution will always converge to the physically correct solution. In contrast, the Roe scheme is not naturally entropy-satisfying. As the Engquist-Osher scheme satisfies the Entropy condition, it is sometimes referred to as an *E scheme.* In view of the extra logic necessary to determine the correct form of the upwind terms Δu^- and Δu^+, the scheme is more computationally intensive than the Roe scheme.

6.1.5 Shock Acceleration for Roe's scheme

Chapter 4 considered the convergence of explicit time marching methods in hyperbolic problems and discussed the phenomena through which persistent high-frequency error components retard convergence. In a general sense, the notion behind the convergence acceleration techniques developed in the present work is to dispel the most persistent high-frequency errors. Furthermore, in addition to eliminating persistent error modes, the method should also reduce the spectrum of error frequencies so that residuals can decay at less disparate rates. Hence convergence is accelerated. As discussed already, the dominating errors are primarily associated with singularities in the flow that exist at sonic expansions and at shock waves, viz. where $M \rightarrow 1$. In terms of undesirable convergence pathology, generally behavior at the shock wave is the more severe. The convergence acceleration methods proposed in this study confine attention solely to the shock region, using a mechanism that is equivalent to accelerating the motion of the shock to its terminal location and state in steady flow. Hence the term that we have coined for this is *Shock Acceleration.* Reducing

the high-frequency errors that are associated with slow shock motion is treated as a local task independent of other regions within the flow that are updated by the standard non-accelerated method. This necessitates the development of custom update techniques for shock regions that must be consistent with the unmodified scheme used in the other regions of the flow. Consequently, the acceleration techniques proposed here are derived for use with specific numerical algorithms. The particular schemes considered for the scalar model problem were introduced in the previous chapter. They are the upwind methods of Roe [126] and Engquist-Osher [42]. However, it is stressed that the methods developed may be generalized to wave decomposition methods not considered presently.

Algebraic Single-point Model

In Section 6.1.3, the explicit first-order update procedure for Roe's scheme is given by equation (6.30). In the course of time marching the solution to a hyperbolic conservation law, the residual for the ith cell is

$$\begin{aligned} R_i &= \phi_{i-\frac{1}{2}}, && \text{for a non-shock point, } (a_{i-\frac{1}{2}} \geq 0) \\ R_i &= \phi_{i+\frac{1}{2}}, && \text{for a non-shock point, } (a_{i+\frac{1}{2}} < 0) \\ R_i &= \phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}}, && \text{for a shock point, } (a_{i-\frac{1}{2}} \geq 0 > a_{i+\frac{1}{2}}) \end{aligned} \tag{6.67}$$

In what follows the subscript s will be taken to represent the counter of the cell in which the shock is located. Such a cell will be referred to as a *shock point* or *shock cell* which can exist only in the shock transition zone. The transition zone is that region of flow that separates conditions at the head of the shock from conditions at its foot. The shock cell may be defined as being the cell across which the cell interface wave speed $a(u)$, changes sign. However, this should not be confused with a sonic expansion, where there is also a change of sign in wave speed. The cell in which a shock is resident is defined uniquely from the relationship

$$a(u_{i-\frac{1}{2}}) \geq 0 \geq a(u_{i+\frac{1}{2}}) \tag{6.68}$$

Where it holds, the shock cell is given by $s = i$. Furthermore, passing through a shock wave normal to its front, the wave speed may change sign only once, ensuring that the above condition is met in a single cell, even in instances when the numerical scheme captures the shock wave across more than one cell. As the solution approaches the asymptotic limit of a steady solution, all residuals tend to zero, though they do not necessarily do so smoothly or at the same rate. A steady solution is reached when

$$\phi_{i+\frac{1}{2}} = 0 \qquad \text{(for a non-shock point)} \tag{6.69}$$

$$\phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}} = 0 \qquad \text{(for a shock point, } s = i\text{)} \tag{6.70}$$

The convergence acceleration method proposed here is based on satisfaction of the latter equation for the shock cell residual *prior* to its natural attainment in time, i.e. by enforcing the residual in the shock cell to be zero. As such, equation (6.70) represents the acceleration model. Implementing this procedure has the effect of increasing the wave speed of the shock artificially so that the high-frequency error component associated with its naturally slow convection

speed is effectively reduced. In turn, the errors associated directly with the shock motion may be damped at rates that are less disparate compared to the error modes from other parts of the flow. As the technique modifies only the behavior of the shock cell and, as it does so via algebraic manipulation of equation (6.70), it is termed a *one-point algebraic shock acceleration model.*

To understand how this acceleration model is implemented in the scalar problem, recall the model equation

$$u_t + \left(u + \frac{1}{u}\right)_x = \frac{kx}{u} \quad .$$

Now, the quantity $\phi_{i+\frac{1}{2}}$, given by equation (6.26), which represents the component of the shock cell residual from the right-hand interface, can be re-written as

$$\phi_{i+\frac{1}{2}} = -\frac{1}{\Delta x} a_{i+\frac{1}{2}} \Delta u_{i+\frac{1}{2}} + \bar{q}_{i+\frac{1}{2}} \tag{6.71}$$

in which the standard difference notation

$$\Delta u_{i+\frac{1}{2}} = u_{i+1} - u_i \tag{6.72}$$

applies. The interface wave speed $a_{i+\frac{1}{2}}$ and the source term $\bar{q}_{i+\frac{1}{2}}$ are given by equations (6.24) and (6.27) respectively. Substitution of these expressions into equation (6.71), produces

$$\phi_{i+\frac{1}{2}} = -\frac{1}{\Delta x}\left(1 - \frac{1}{u_i u_{i+1}}\right)(u_{i+1} - u_i) + k(x_i + x_{i+1})\left(\frac{1}{u_i} + \frac{1}{u_{i+1}}\right) \tag{6.73}$$

The equivalent expression for the left hand interface is

$$\phi_{i-\frac{1}{2}} = -\frac{1}{\Delta x}\left(1 - \frac{1}{u_{i-1} u_i}\right)(u_i - u_{i-1}) + k(x_{i-1} + x_i)\left(\frac{1}{u_{i-1}} + \frac{1}{u_i}\right) \tag{6.74}$$

In accordance with the acceleration model, equation (6.70), the latter two expressions are summed, the result is then set to zero and re-arranged in terms of u_s. The accelerated solution update for the shock cell is then found to be

$$u_s = u_i = -k\left(\bar{x}_{i-\frac{1}{2}} + \bar{x}_{i+\frac{1}{2}}\right) / (\Omega + \Psi) \tag{6.75}$$

in which

$$\Omega = \left(u_{i-1} + \frac{1}{u_{i-1}} - u_{i+1} - \frac{1}{u_{i+1}}\right) / \Delta x$$

$$\Psi = k\left(\frac{\bar{x}_{i-\frac{1}{2}}}{u_{i-1}} + \frac{\bar{x}_{i+\frac{1}{2}}}{u_{i+1}}\right)$$

and,

$$\begin{aligned} \bar{x}_{s-\frac{1}{2}} &= x_{i-1} + x_i \\ \bar{x}_{s+\frac{1}{2}} &= x_i + x_{i+1} \end{aligned}$$

Evaluation of equation (6.75) renders an accelerated update for the shock cell, which is consistent with the underlying flux function of the standard Roe scheme

used elsewhere to update the non-shock points. The new accelerated shock update depends solely on conditions in one cell on each side of the shock point. As a test for consistency, the accelerated update can be applied after the standard scheme has marched the solution to convergence, and it should not be found to change the solution at all. In fact, this test was carried out for all the acceleration methods proposed here and all passed it. However, it must be stressed that for the one-point acceleration model to work, the shock cannot be permitted to transfer into a neighbouring cell by the action of an accelerated update. If a shock is allowed to pass into a new cell on an accelerated update from the single-point model, a divergent solution will result.

The success of the method depends to a large measure upon implementation detail. Non-optimal implementations were found to reduce the rate of acceleration. For the present problem, it was found that the best results were obtained by updating non-shock points in the field before updating the shock via the acceleration procedure. It is believed that this is because the acceleration procedure benefits more from using the latest data of neighbouring points $u_{s\pm1}^{n+1}$ than from the older data $u_{s\pm1}^{n}$. In addition, it is crucial that care be taken in deciding when to apply the procedure. When applied from the onset of computation ($t = 0$), a spurious solution may result or the solution may diverge. Once the acceleration procedure has been safely initiated, to maintain robustness each accelerated update must be monitored for its reasonableness and when appropriate, discarded in favour of the standard (non-accelerated) solution update. Each of these aspects is considered further below.

Admissibility of Accelerated Update

At the present time, no theory exists for predicting the basin of attraction for a time accurate numerical solution to a nonlinear Initial Boundary Value Problem (IBVP) *a priori*. Nonlinear dynamical systems is an active area of research which has only recently been applied to the study of model problems of interest to CFD [190, 189, 81, 59, 191]. However, it offers potential insight into understanding the mechanisms that couple the solution of a PDE to its numerical discretization and, the factors which determine the basin of attraction of a discretized solution. The most obvious factor effecting the ability of a numerical method to attain the solution of its PDE for a given problem is the initial conditions assumed for the computation. However, the mechanisms by which a discrete approximation can attain the basin of attraction of the governing PDE, whether from the onset of computation or after a certain advancement in time, are not adequately understood. Thus, at present there is little theoretical guidance for the fundamental development of convergence acceleration methods, particularly in the case of hyperbolic time marching algorithms. This background should place in context the need for admissibility criteria when the shock acceleration method is used. Such criteria serve to screen accelerated updates in instances when the given acceleration model is inappropriate for the local flow conditions. With the single-point acceleration model, the latter situation arises when a shock is on the verge of moving into a new cell.

When the one-point algebraic shock acceleration model was applied naïvely from the commencement of the solution process ($t = 0$), the solution diverged within a few iterations. However, accelerated convergence to the correct solution

was achieved when applied at later times during the computation. Commencing acceleration too late, (i.e. after the solution has evolved for a large number of iterations), reduces the cost savings that can be achieved. Clearly, for each problem, there is an optimal time at which convergence acceleration should be commenced in order to achieve maximum savings in computational expense yet without suffering an unnecessary risk of divergence or indeed, convergence to a spurious solution that is outside the basin of attraction of the PDE.

For the one-point shock acceleration model, the problem of deciding the earliest time at which to commence acceleration is determined heuristically through experiment. The remaining problem of deciding if and when to abstain from an accelerated update once acceleration has commenced, is overcome by applying two criteria. The first is derived from the characteristic relationship at the shock, equation (6.68). Substituting the wave speed from the model equation into that relationship produces

$$1 - \frac{1}{u_{s-1}u_s} \geq 0 \geq 1 - \frac{1}{u_s u_{s+1}}$$

After rearrangement, the useful expression

$$\frac{1}{u_{s-1}} \leq u_s \geq \frac{1}{u_{s+1}} \tag{6.76}$$

emerges, which is termed the *entropy criterion.* It describes an admissible range of values within which an accelerated update of the solution in the shock cell should lie, if it is to be accepted. When an accelerated update lies outside this range, it suggests one of the following possibilities:

1. the accelerated update is non-monotonic and too severe a change to the solution. That is to say that the solution data in the cells neighbouring the shock cell have not evolved sufficiently to support the accelerated update of the shock cell and consequently the solution cannot be maintained within its basin of attraction or,

2. at the time of the update, the data in the shock cell and its neighbours are indicative of the shock being near the time of migrating into a new cell.

In either case, the accelerated update is unacceptable. To apply the entropy criterion, the following procedure is adopted. After the shock cell has been identified, a trial update value is determined from the shock acceleration formula, equation (6.75). Provided the shock lies within the bounds implied by equation (6.76), it is used. Otherwise it is rejected and either one may revert back to the standard method to affect the update or, if shock transfer is taking place, another acceleration model may be used. At this point, the second test known as a *shock transfer test* is necessary to determine whether the unusable update is indicative of shock migration into another cell. To carry out the test, the rejected update is retained to compute a hypothetical wave speed for the downstream interface to the shock cell: a wave speed of different sign to that calculated for the same interface at the previous iteration indicates that the time has come for the shock to migrate into an adjacent cell. In this case, one may proceed by choosing one of three procedures.

The first and most conservative is to abstain from any acceleration and instead update the shock cell using the standard scheme while shock transfer is taking place. Once the transfer is complete, the acceleration procedure may be resumed to commence acceleration of the shock in its new cell. The second option is to introduce a mechanism that will limit the extent of the accelerated update. This was achieved via equation (6.76) which may be re-written as

$$u^{\min} \leq u_s \leq u^{\max} \tag{6.77}$$

in which

$$u^{\min} = \frac{1}{u_{s-1}}$$

and

$$u^{\max} = \frac{1}{u_{s+1}}$$

If the acceleration update lies outside the bounds $u^{\min}, u^{\max}$, it is set to the bound to which it is closest. This enables acceleration to be maintained such that it is monotonically bounded. The third and final option is to cease using the single-point acceleration model during shock cell transfer and switch to a more sophisticated shock acceleration model while shock transfer is taking place.

A family of three models are proposed to maintain acceleration during shock transfer. Due to the nature of the transfer process, the new shock acceleration models operate on two cells –the shock cell and, its downstream neighbour (viz. the shock cell prior to transfer and, the new shock cell after transfer). Hence, they are termed *two-point algebraic models*. These new models are discussed in the next section.

Two-point Algebraic Models for Shock Transfer

Like their one-point counterpart, each of the two-point family of models that accelerate shock motion while the shock is moving from one cell to another, are based on the principles of wave propagation and, the enforcement of conditions deduced from the steady-state behaviour of the underlying numerical algorithm. This approach ensures that the hyperbolicity constraints on the governing equations are respected. Also, it ensures consistency between the baseline method used outside shock regions and the acceleration model. It is important to note that for scalar equations using Roe's method, two-point models of the type described below, should *not* be used unless shock transfer is taking place. In other words, they must be used in conjunction with one-point algebraic models, not in isolation. Three two-point shock transfer models are now described, but it should be noted that other transfer models of increasing sophistication can be devised that may have greater merit in problems involving extremely strong shock waves such as reactive flows.

Shock Transfer Model I

To formulate the acceleration procedure, the physical model of shock cell transfer is prescribed in terms of the cell interface residual components $\phi_{i\pm\frac{1}{2}}$. The model is described conceptually in Figure 6.5(a) which shows the flow of information at two successive time levels (t^n, t^{n+1}). The indexed circles represent cell locations.

Shock cells are represented by filled circles. The arrows indicate the direction of the interface wave speed and as such, the cells to which each residual component contributes. Prior to attaining steady-state values, the residuals at the shock cell and at its downstream neighbour are composed as

$$R_i^n = \phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}} \qquad (s = i) \tag{6.78}$$

$$R_{i+1}^n = \phi_{i+\frac{3}{2}} \tag{6.79}$$

which can be inferred from the figure. Again, superscripts denote time level. Assuming that the shock moves into the right adjacent cell on the next flowfield update, the residuals within the same two cells become

$$R_i^{n+1} = \phi_{i-\frac{1}{2}} \tag{6.80}$$

$$R_{i+1}^{n+1} = \phi_{i+\frac{1}{2}} + \phi_{i+\frac{3}{2}} \qquad (s = i+1) \tag{6.81}$$

Now the present acceleration model enforces the condition that the residuals in the cells $(i, i+1)$ must be zero. Thus

$$\phi_{i-\frac{1}{2}} = 0 \tag{6.82}$$

$$\phi_{i+\frac{1}{2}} + \phi_{i+\frac{3}{2}} = 0 \qquad (s = i+1) \tag{6.83}$$

Ordinarily this would be the case in the steady-state were the solution to be marched to convergence by the standard scheme *provided* the shock did not shift position again into another cell. The two equations above constitute the accelerated shock transfer model now proposed. Evaluation of the first acceleration equation for the model problem is readily achieved by setting the equation for the residual at the $i - \frac{1}{2}$ interface, equation (6.82), to zero, i.e.

$$-\frac{1}{\Delta x}\left(1 - \frac{1}{u_{i-1}u_i}\right)(u_i - u_{i-1}) + k(x_{i-1} + x_i)\left(\frac{1}{u_{i-1}} + \frac{1}{u_i}\right) = 0$$

After rearrangement, its quadratic form becomes apparent:

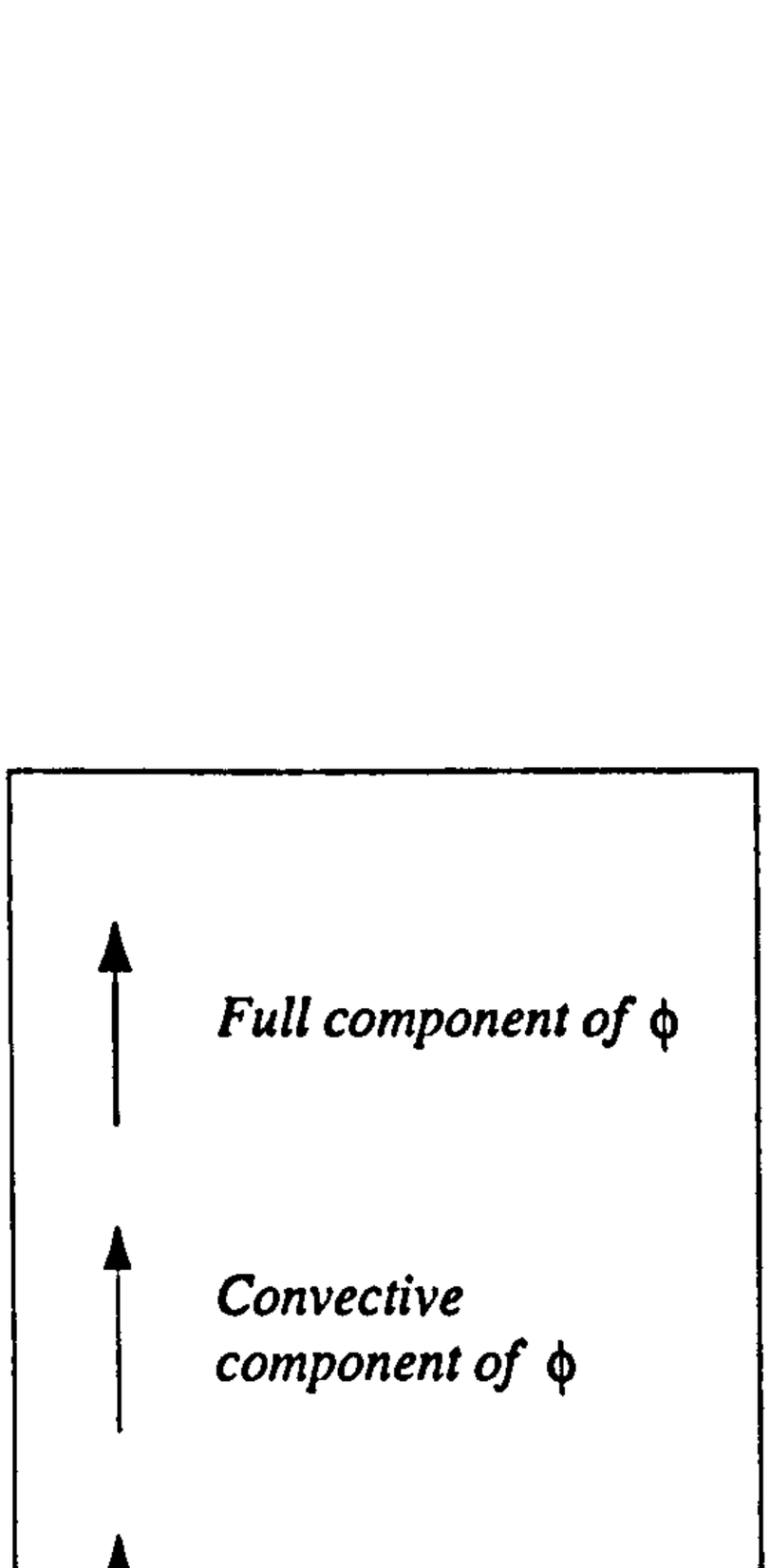
$$u_i^2 - \left(u_{i-1} + \frac{1}{u_{i-1}} - \frac{k\bar{x}_{i-\frac{1}{2}}\Delta x}{u_{i-1}}\right)u_i + \left(1 + k\bar{x}_{i-\frac{1}{2}}\Delta x\right) = 0 \tag{6.84}$$

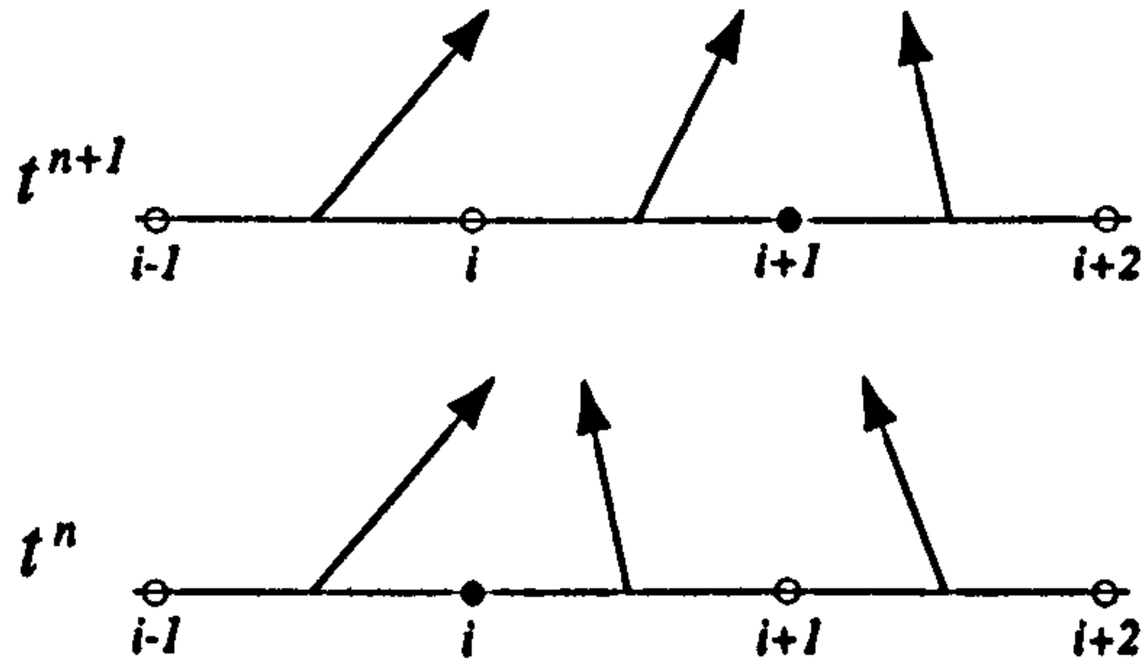
To avoid confusion in notation when shock transfer is taking place, the index s denotes the new shock cell after transfer has taken place. Solving the latter equation for u_i gives the update for the former shock cell as $u_{s-1}^{n+1} = u_i$. Now consider the second acceleration equation which is none other than the one-point acceleration model, equation (6.70), shifted spatially by one index. As such the solution in the new shock cell is found as $u_s^{n+1} = u_{i+1}$. Indeed for the present model problem, the acceleration update is obtained as

$$u_s = u_i = -k\left(\bar{x}_{i+\frac{1}{2}} + \bar{x}_{i+\frac{3}{2}}\right)/(\Omega + \Psi) \tag{6.85}$$

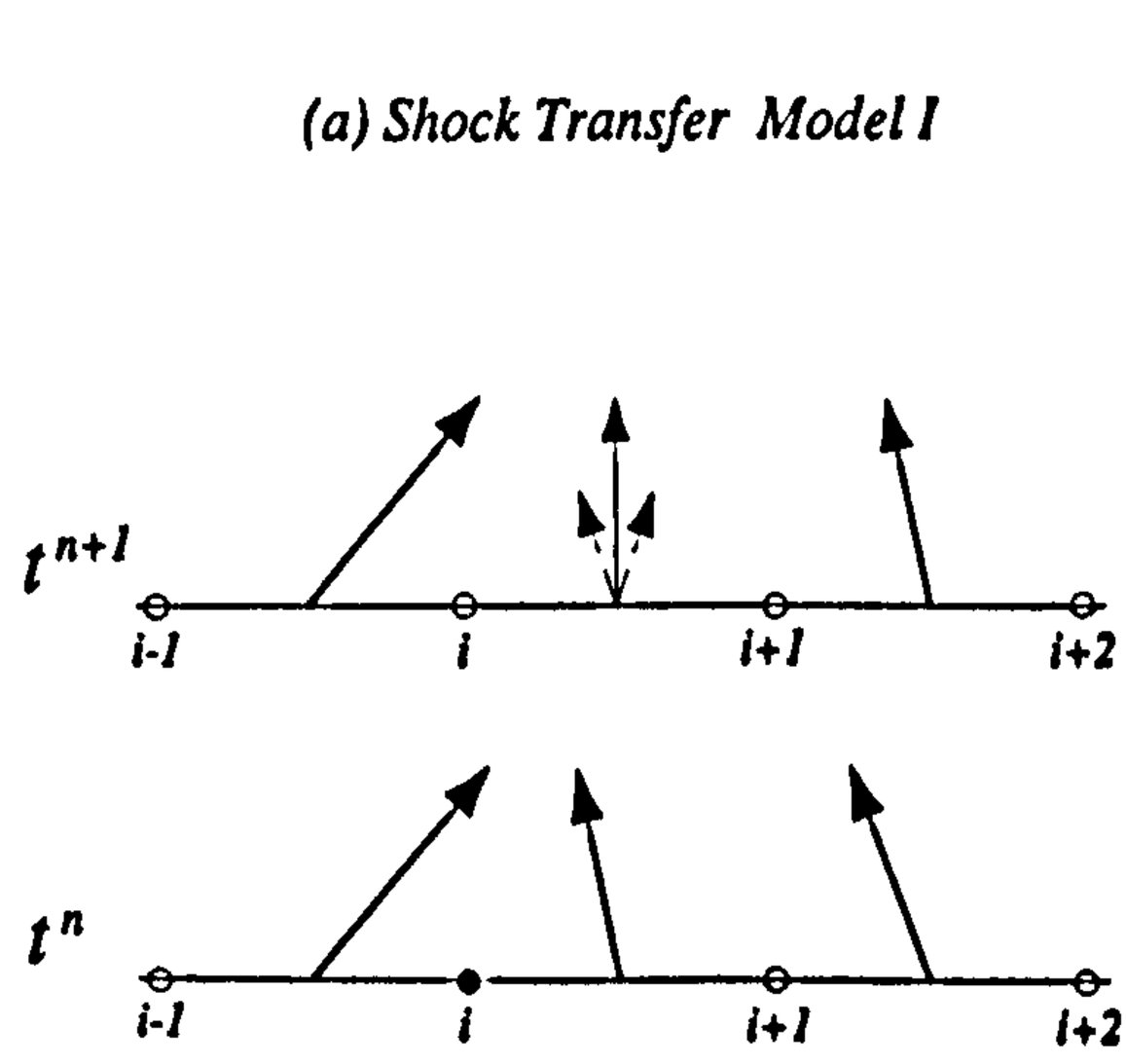
where

$$\Omega = \left(u_i + \frac{1}{u_i} - u_{i+2} - \frac{1}{u_{i+2}}\right)/\Delta x$$





(a) Shock Transfer Model I



(b) Shock Transfer Model II



(c) Shock Transfer Model III

FIGURE 6.5. Two-point shock transfer models.

and

$$\Psi = k\left(\frac{\bar{x}_{i+\frac{1}{2}}}{u_i} + \frac{\bar{x}_{i+\frac{3}{2}}}{u_{i+2}}\right)$$

The accelerated shock transfer equations (6.84) and (6.85) for the two-point model are readily solved. Since the former does not rely on data from the new shock cell $(i+1)$ it may be solved first, independently of the latter equation as a normal quadratic. Selection of the appropriate root is trivial since both roots are real and of opposite sign. The positive root is selected. The latter equation is evaluated as written using the value $u_{s-1}^{n+1} = u_i$ found from the quadratic. A more general approach is to solve both the acceleration equations simultaneously using the Newton iterative procedure with evaluation of first derivatives obtained analytically for the flux function of the model problem. However, care must be taken to use sensible initial guesses to initiate the iteration process. In practice this is accomplished by setting $u_{s-1}^{guess} = u_i^n, u_s^{guess} = u_{i+1}^n$. This is necessary to ensure that the iteration process converges to the correct root of the quadratic. Both approaches have been implemented successfully and found to produce identical results in single precision arithmetic. However, the second approach is preferred as it is more general in that it can be used for other equations of more complex form. Furthermore, in cases where the first derivatives needed by the Newton procedure are not available analytically, they may be approximated by carefully selected finite difference quotients.

Shock Transfer Model II

The second in the family of shock transfer models is illustrated in Figure 6.5(b). Here however, the acceleration model requires the component of the shock residual $\phi_{i+\frac{1}{2}}$, to be split into its convective and source elements. The convective component of ϕ is denoted by a thin arrow while the source component is denoted by a dashed arrow. Full components of ϕ are denoted by the thicker arrows. Transfer model II produces a more conservative update than the first model by assuming that the solution at time level t^{n+1}progresses the shock as far as the interface, but not beyond into a new cell. The notion of the shock being resident within a cell and hence the term shock cell, is therefore lost as the shock position is between cells on the interface. Prior to update, the residual at the shock and at its downstream neighbour are, as before

$$\begin{aligned} R_i^n &= \phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}} \qquad (s=i) \\ R_{i+1}^n &= \phi_{i+\frac{3}{2}} \end{aligned}$$

Since the wave speed $a_{i+\frac{1}{2}}$ is forced to be zero, the convective component vanishes and equation (6.71) reduces to

$$\phi_{i+\frac{1}{2}} = \bar{q}_{i+\frac{1}{2}} \tag{6.86}$$

Thus, a source component in isolation will constitute the flux at the $i+\frac{1}{2}$ interface. With the current model, this is split equally between the two cells straddling the interface. The update described so far takes the form

$$R_i^{n+1} = \phi_{i-\frac{1}{2}} + \frac{1}{2}\bar{q}_{i+\frac{1}{2}}$$

$$R_{i+1}^{n+1} = \phi_{i+\frac{3}{2}} + \frac{1}{2}\bar{q}_{i+\frac{1}{2}}$$

Next, these residuals are forced to be zero and the acceleration model is given by the two equations

$$\phi_{i-\frac{1}{2}} + \frac{1}{2}\bar{q}_{i+\frac{1}{2}} = 0 \tag{6.87}$$

$$\phi_{i+\frac{3}{2}} + \frac{1}{2}\bar{q}_{i+\frac{1}{2}} = 0 \tag{6.88}$$

Evaluating the acceleration equations in terms of the model problem produces the two quadratics

$$u_i^2 - \left[u_{i-1} + \frac{1}{u_{i-1}} + k\Delta x\left(\frac{\bar{x}_{i-\frac{1}{2}}}{u_{i-1}} + \frac{\bar{x}_{i+\frac{1}{2}}}{2u_{i+1}}\right)\right]u_i + \left[1 - k\Delta x\left(\bar{x}_{i-\frac{1}{2}} + \frac{\bar{x}_{i+\frac{1}{2}}}{2}\right)\right] = 0 \tag{6.89}$$

and

$$u_{i+1}^2 - \left[u_{i+2} + \frac{1}{u_{i+2}} - k\Delta x\left(\frac{\bar{x}_{i+\frac{1}{2}}}{2u_i} + \frac{\bar{x}_{i+\frac{3}{2}}}{u_{i+2}}\right)\right]u_{i+1} + \left[1 + k\Delta x\left(\frac{\bar{x}_{i+\frac{1}{2}}}{2} + \bar{x}_{i+\frac{3}{2}}\right)\right] = 0 \tag{6.90}$$

These are the final expressions to be solved for the cells (u_i, u_{i+1}), straddling the interface shock. As these equations are inter-dependent, they should be solved iteratively by a procedure such as the Newton method, taking care to initiate the iterative process with a sensible guess, such as the solution at the last update.

Shock Transfer Model III

This model is shown in Figure 6.5(c) using the familiar notation of the previous two figures. Again, $\phi_{i+\frac{1}{2}}$ is split into convective and source components. As before, prior to update, the residual at the shock and that at its downstream neighbour are made up as

$$\begin{aligned} R_i^n &= \phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}} \qquad (s = i) \\ R_{i+1}^n &= \phi_{i+\frac{3}{2}} \end{aligned}$$

At update to time level t^{n+1}, the convective component of the residual at the $(i+1)$ interface is distributed to the right hand cell, while the source component continues acting on the left hand cell. The reasoning behind this is that when the wave velocity is small, it may not be justifiable to send the source component to the downstream cell since it will take some time before the full effect of the wave will reach there. Originally such reasoning was advanced by Toro [153] in proposing a technique for solving inhomogeneous systems arising from combustion problems, based on Roe's method[1].

---

[1]The technique Toro proposed is a modification to the standard approach of upwinding both convective and source terms in their entirety. The idea is that for sufficiently small wave speeds, only a proportion of the source contribution should be upwinded. The proportion to be upwinded is determined as a function of the local Courant number.

Thus, using this model for shock transfer, the residuals are given by

$$R_i^{n+1} = \phi_{i-\frac{1}{2}} + \bar{q}_{i+\frac{1}{2}} \tag{6.91}$$

$$R_{i+1}^{n+1} = a_{i+\frac{1}{2}} \Delta u_{i+\frac{1}{2}} + \phi_{i+\frac{3}{2}} \qquad (s = i+1) \tag{6.92}$$

and setting them to zero so as to drive the acceleration process produces

$$\phi_{i-\frac{1}{2}} + \bar{q}_{i+\frac{1}{2}} = 0 \tag{6.93}$$

$$a_{i+\frac{1}{2}} \Delta u_{i+\frac{1}{2}} + \phi_{i+\frac{3}{2}} = 0 \qquad (s = i+1) \tag{6.94}$$

Evaluation of these two equations for the model equation renders algebraic formulae of familiar form

$$u_i^2 - \left[u_{i-1} + \frac{1}{u_{i-1}} + k\Delta x \left(\frac{\bar{x}_{i-\frac{1}{2}}}{u_{i-1}} + \frac{\bar{x}_{i+\frac{1}{2}}}{u_{i+1}} \right) \right] u_i +$$

$$\left[1 - k\Delta x \left(\bar{x}_{i-\frac{1}{2}} + \bar{x}_{i+\frac{1}{2}} \right) \right] = 0 \tag{6.95}$$

and

$$u_s = u_{i+1} = -k\bar{x}_{i+\frac{3}{2}} \Delta x \Big/ \left(u_i + \frac{1}{u_i} - u_{i+2} - \frac{1}{u_{i+2}} + \frac{k\bar{x}_{i+\frac{3}{2}} \Delta x}{u_{i+2}} \right) \tag{6.96}$$

These are the final acceleration equations. As they are inter-dependent, they may be solved iteratively using the Newton method.

6.1.6 Performance of the accelerated Roe Scheme

Before discussing the performance of the new acceleration methods, a brief description of the methodology used in their development is required. Prior to embarking upon formulation of the acceleration models, a computer code was written to obtain analytic solutions, based on the theory of section 6.1.2. From the analytic solution it was possible to determine the range of boundary conditions that would produce both stable and unstable solutions. The latter could then be avoid when the numerical method was later used. Solutions from the baseline numerical method (i.e. the standard algorithm without acceleration) were then compared against the exact solutions and assessed for a range of different conditions. In the cases tested, the numerical solutions from the baseline scheme were found to be in excellent agreement with the exact solutions, rendering steady shocks with a single transition point. A large number of tests were conducted during the development and refinement of the proposed shock acceleration models. Once an acceleration procedure had been proven to work, it was validated against the analytic solution and numerical solutions from both the baseline numerical scheme and the Engquist-Osher scheme.

Throughout the study, the following criteria were used to validate and assess the newly developed acceleration procedures:

1. Accuracy of the solution –the extent to which an accelerated solution can replicate the converged solution of the baseline scheme;

2. Robustness –the ability to produce accurate solutions for a wide range of boundary conditions reliably without producing divergence or spurious behaviour that is inconsistent with the baseline method; and,

3. Efficiency –measured in terms of computational expense, taking into account both the increased rate of convergence as well as the increased operation count of the acceleration procedure.

In terms of the first criterion, each of the acceleration procedures for the Roe scheme were found to replicate the solution accuracy of the baseline Roe scheme to within the sixth decimal place in single precision arithmetic. An exception to this arose with certain solutions from the non-accelerated scheme in which a limit cycle developed between the shock transition cell and its downstream neighbour. This happened for the model equation over a very small range of boundary condition and the phenomenon has already been reported in the literature by Roe and van Leer [134]. On the issue of robustness, each of the models required only one free parameter, namely when to commence acceleration. When to desist from acceleration once it had been commenced was handled through enforcement of the admissibility criteria. These criteria also served to preclude divergent/spurious solutions, significantly improving the overall robustness of the methods developed.

Many computations were carried out using a wide range of boundary conditions and different sized mesh to properly assess the new acceleration procedures. From these, a small number of representative results have been chosen. Since the solutions obtained using the new acceleration method do not differ from solutions obtained from the baseline method to the extent that they can be judged by eye, they are not presented here. Consequently, only convergence data is presented. Of the results presented, each corresponds to a mesh containing 11 cells. Furthermore, all the results presented, both accelerated and non-accelerated, use local time stepping with a constant Courant number $\nu = 0.8$. The initial conditions are constant data from the left boundary to the first point on the positive side of the x-axis and, some other constant data from the second cell to the right boundary. This initialization produces a discontinuity or shock between the first and second cells on the positive axis.

Figure 6.6 compares the convergence history of the baseline Roe scheme to that of the scheme using the two-point acceleration method with shock transfer model II. The boundary conditions for this case are given by $u(-1,t) = 2.0$, and $u(1,t) = 0.42$. The convergence histories are plotted in terms of log $\|du/dt\|_2$ against iteration count. The baseline scheme shows a peculiar convergence pathology levelling out shortly after 500 iterations before reaching machine zero. The accelerated solution is commenced at iteration 12. Its introduction produces an immediate jump in the l_2 norm followed by a rapid and sustained drop to near machine zero. Convergence is attained in 49 iterations. This represents an order of magnitude improvement in terms of the rate of convergence. In terms of the operation count per iteration, the accelerated algorithm is less than 10% more expensive than the standard algorithm using the present sized grid on a computer with scalar architecture. Taking into account both the increased computational expense of the acceleration procedure and the improvement in convergence rate, the solution using shock acceleration effects a saving in computational expense of more than 90%.

The second test case to be considered differs from the first one only in the boundary condition used. Here the boundary conditions are $u(-1,t) = 2.0$, and $u(1,t) = 0.45$. The given initialization and boundary conditions imply that the shock must transfer cells on three occasions before the steady solution can be attained. The convergence histories for this test case are presented in Figure 6.7. The baseline scheme demonstrates fairly smooth convergence, reaching machine zero in just over 400 iterations. Again the accelerated solution uses the two-point procedure based on shock transfer model II which is commenced at iteration 12. For the present boundary conditions, the acceleration kicks-in smoothly, affecting a rapid and sustained drop until the computation is halted with the residual near machine zero. Convergence is achieved in a total of 28 iterations. Off-setting the increased rate of convergence against increased operation count, the accelerated solution still works out at over 90% cheaper than the baseline method.

The final set of results to be presented compares the performance of the single-point acceleration scheme with the two-point acceleration scheme based on shock transfer models I and II. The test case used here is the same test as the previous one. The convergence histories are presented in Figure 6.8. As expected, when used in isolation the single-point algebraic model was less efficacious than any of the two-point algebraic models. With the single-point model, acceleration could not be commenced before iteration 21, and could not benefit from accelerated updates when the shock moved from one cell to the next. In spite of this the model produced a converged solution in 33. Shock transfer model I produced a large jump in the l_2 norm at the commencement of acceleration, but converged within the same number of iterations as transfer model II. It will be recalled that transfer model II produces a more conservative update than model I. This is borne out by its convergence trace which is well behaved at the commencement of acceleration and, by its more shallow drop. Acceleration for both models I and II was started at iteration 12 and convergence was attained in 28 iterations.

It should be noted that when the shock transfer model III was used for this test case, it was unsuccessful in that the Newton method did not converge to a plausible root. In such instances the scheme reverts to the single-point acceleration model with the baseline method used for shock transfer. This prevents divergence, but with a loss in efficiency when compared to the performance of the other transfer models. Nevertheless, it should be noted that shock transfer model III did work well on other test cases, though not as consistently well as models I and II. For equations with stiffer source terms, the results of the above comparison of shock transfer models could well be different. However, on the basis of the results obtained, both transfer models I and II performed equally well, though transfer model II is the safer one to use.

The admissibility criteria worked flawlessly in all test cases preventing divergence once acceleration had been safely commenced. However, it proved insufficient to prevent divergence if acceleration was commenced from the onset of a computation or from a very early point in time. This is in contrast to the accelerated Engquist-Osher scheme, in which acceleration can be attempted on the first iteration with any unusable update being screened successfully through more robust admissibility criteria.

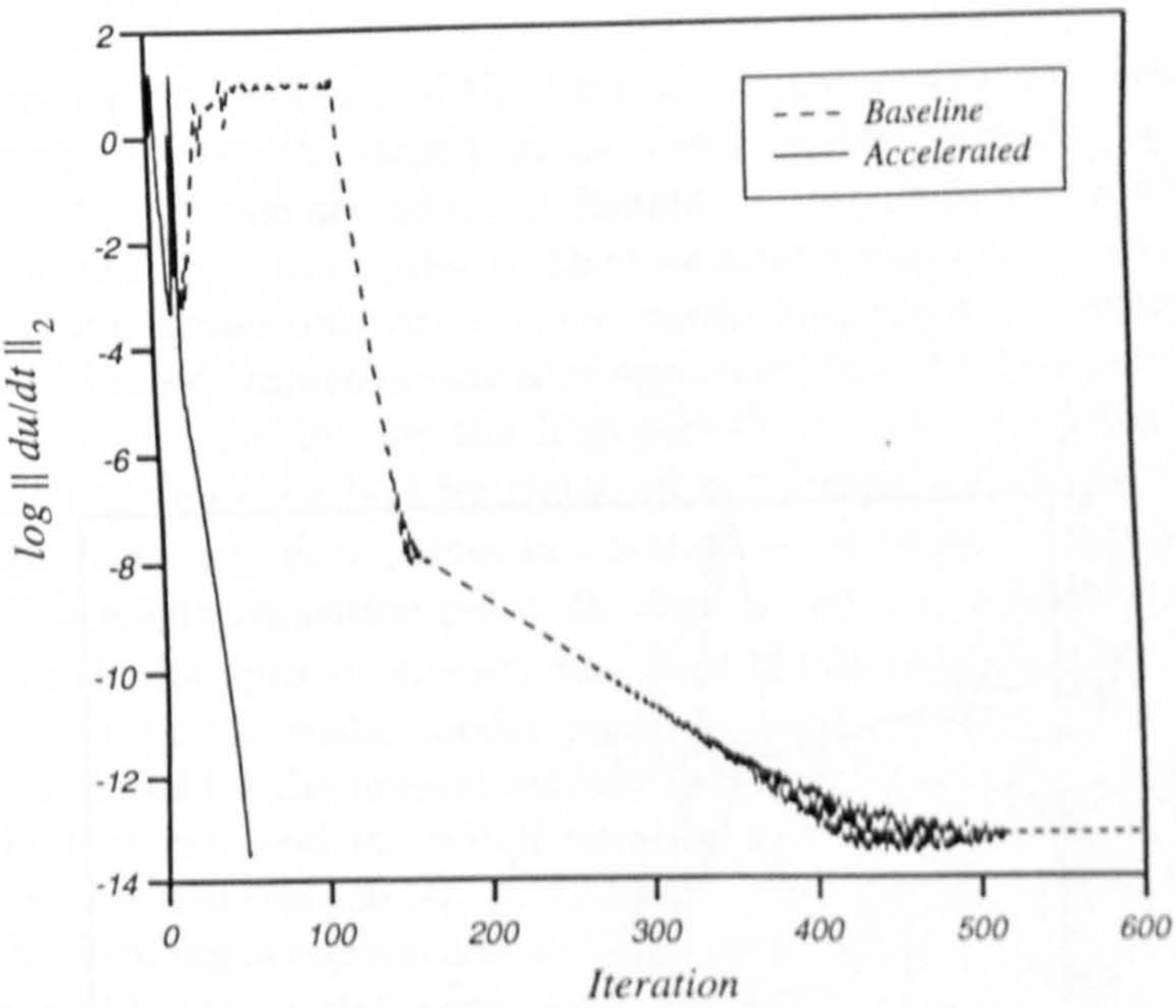


FIGURE 6.6. Comparison of convergence histories of Roe's scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.42$.

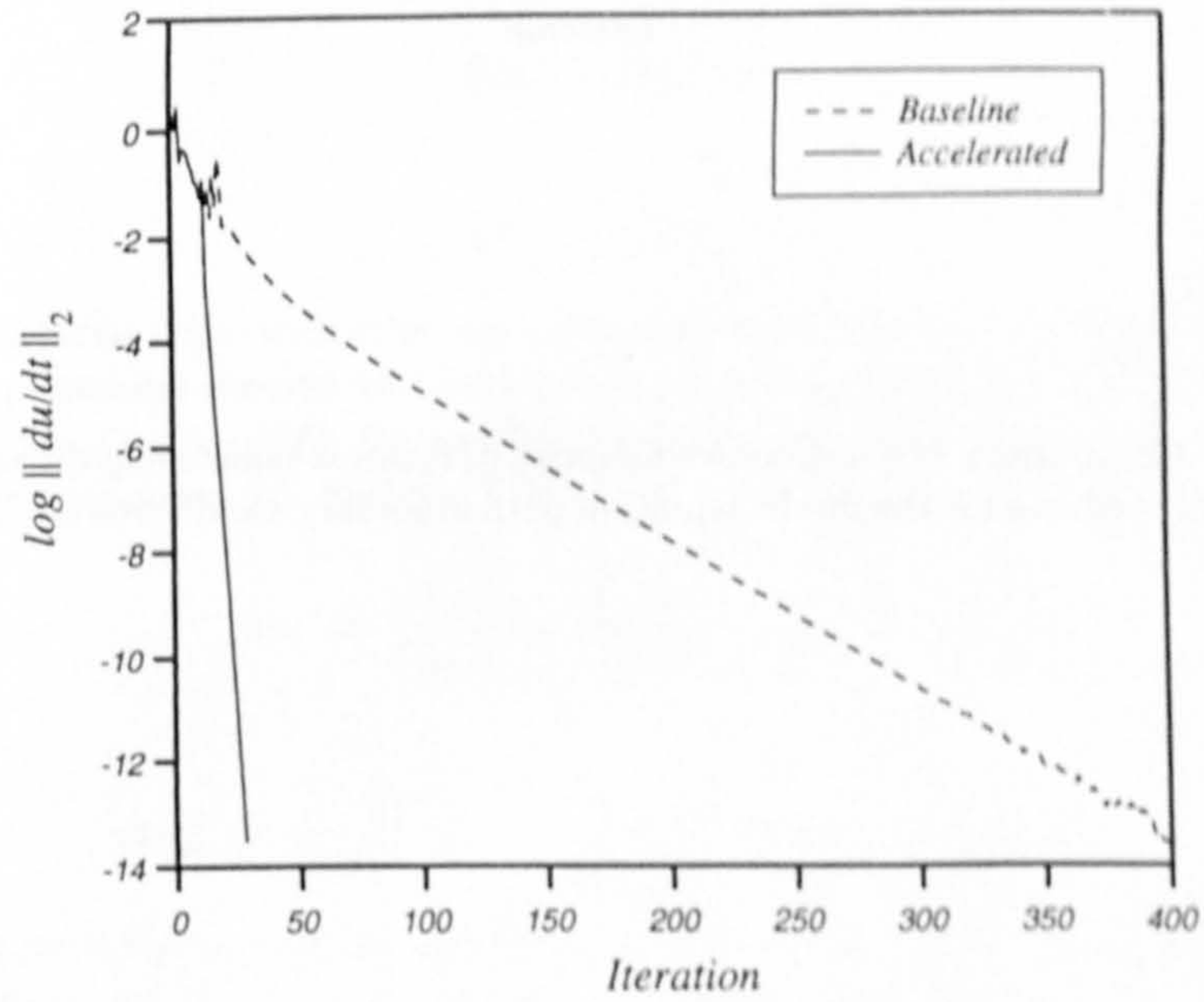


FIGURE 6.7. Comparison of convergence histories of Roe's scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$.

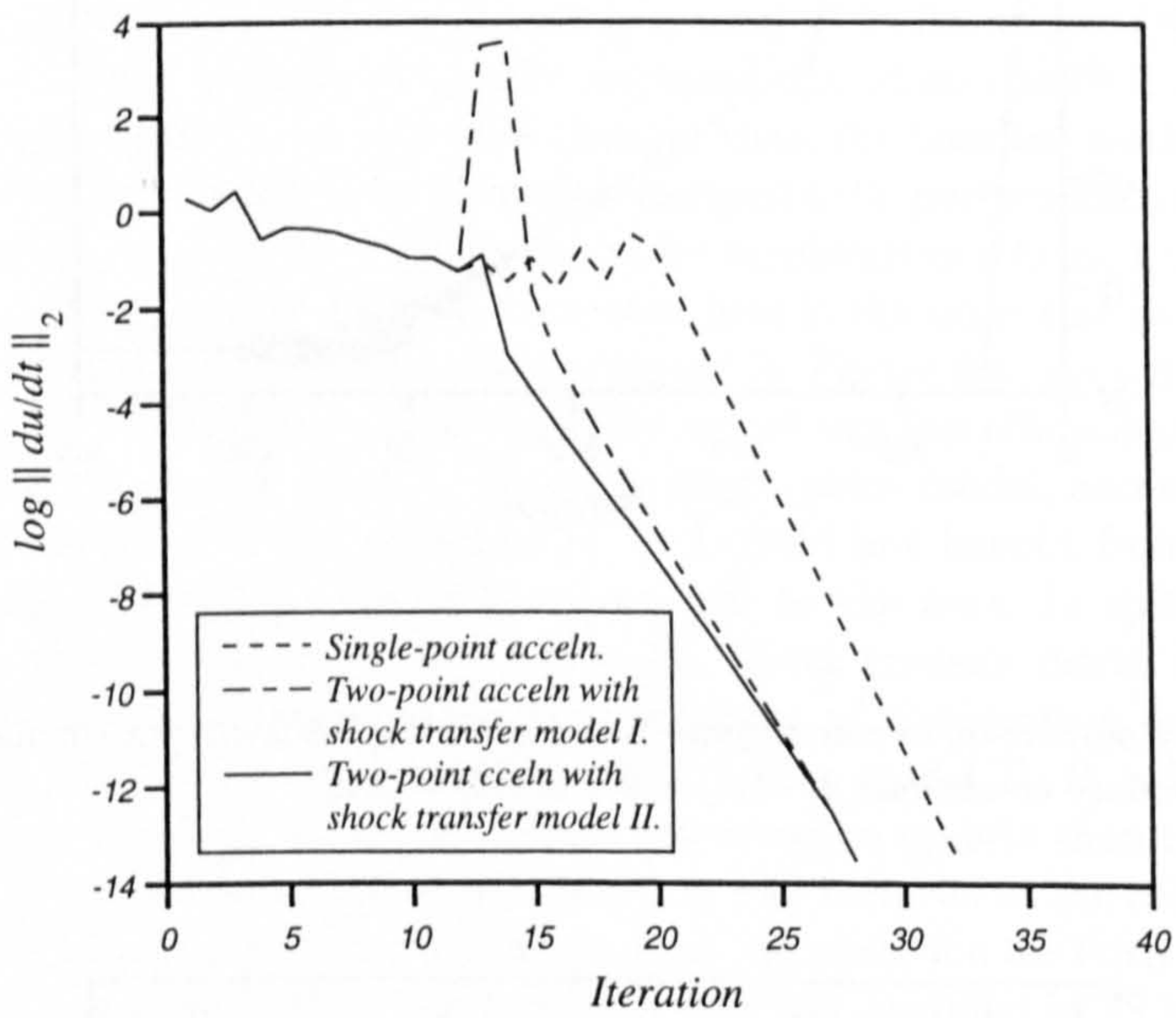


FIGURE 6.8. Comparison of convergence histories of Roe's scheme using different shock acceleration procedures for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$.

6.1.7 Shock Acceleration for the Engquist-Osher scheme

To accelerate the convergence of the Engquist-Osher scheme by localized treatment of the shock wave, the same principles espoused for accelerating the convergence of the Roe scheme are adopted. Namely, residuals in the shock transition zone are enforced to be zero prior to their natural attainment in time. Ordinarily this situation would only arise in the asymptotic limit of a steady solution. Devising and indeed implementing a suitable method of achieving convergence acceleration in this fashion for the Engquist-Osher method differs from shock acceleration for Roe's method by virtue of the former's disposition to capture shocks with two transition points in contrast to the latter's nature to render shocks with a single transition point. In view of this, the acceleration technique proposed for the Engquist-Osher scheme, must of necessity be a two-point model. As it turns out for the scalar model equation, implementation of a convergence acceleration model for the present method is easier than it is for the Roe method in that there is no need to switch between the single-cell shock model to a two-point shock transfer model. In addition, only one admissibility criterion is required for deciding acceptance of an accelerated update, as opposed to the two criteria needed for two-point acceleration models for Roe's scheme.

Recalling equations (6.65) and (6.66) for the two cell residuals in the shock transition zone

$$R_i = \Delta u_i^- + \Delta u_i^+$$

and

$$R_{i+1} = \Delta u_{i+1}^- + \Delta u_{i+1}^+$$

the acceleration model enforces each to be zero as

$$\Delta u_i^- + \Delta u_i^+ = 0 \tag{6.97}$$

and

$$\Delta u_{i+1}^- + \Delta u_{i+1}^+ = 0 \tag{6.98}$$

These equations constitute the acceleration model for the Engquist -Osher scheme.

The expressions needed to evaluate the acceleration update for the first transition cell (cell i), are given by equations (6.59) and (6.48), the latter with indices shifted by one count to the left. Thus

$$\Delta u_i^- = \frac{1}{\Delta x}\left[2 - \left(u_{i+1} + \frac{1}{u_{i+1}}\right) + \bar{q}_{s_R}\right] \tag{6.99}$$

and

$$\Delta u_i^+ = \frac{1}{\Delta x}\left[\left(u_{i-1} + \frac{1}{u_{i-1}}\right) - \left(u_i + \frac{1}{u_i}\right) + \bar{q}_{i-\frac{1}{2}}\right] \tag{6.100}$$

Together with the equations already provided for the source contributions, equations (6.62) and (6.27), the latter shifted one index count to the left, the two expressions above may be expressed as a function

$$F_i(u, \bar{q}) = 0 \tag{6.101}$$

This is the first of three inter-dependent functions needed to formulate the accelerated update for the Engquist-Osher method. Moving on to the second transition cell $(i+1)$, equations (6.50) and (6.57) make up the necessary contributions as

$$\Delta u^{-}_{i+1} = \frac{1}{\Delta x}\left[\left(u_{i+1} + \frac{1}{u_{i+1}}\right) - \left(u_{i+2} + \frac{1}{u_{i+2}}\right) + \bar{q}_{i+\frac{3}{2}}\right] \tag{6.102}$$

and

$$\Delta u^{+}_{i+1} = \frac{1}{\Delta x}\left[\left(u_{i} + \frac{1}{u_{i}}\right) - 2 + \bar{q}_{s_L}\right] \tag{6.103}$$

with adjustment of cell indices as appropriate. Again, from the expressions already provided for the source contributions, equation (6.27) with indices suitably adjusted and, equation (6.61), the two equations immediately above may be expressed as a second function

$$G_{i+1}(u, \bar{q}) = 0 \tag{6.104}$$

Now, the terms $\bar{q}_{s_L}$ and $\bar{q}_{s_R}$ appearing in the functions F_i and G_{i+1}, depend on the grid spacing, the solution variable and the location of the sonic condition x_s. In fact, it is the location of the sonic condition that provides the necessary third function needed for closure in solving for the accelerated updates in the two shock transition cells $(i, i+1)$. Recalling the equation for the sonic condition,

$$x_s^2 = \left[x_i^2\left(\frac{1}{u_{i+1}} - 1\right) + x_{i+1}^2\left(1 - \frac{1}{u_i}\right)\right] / \left(\frac{1}{u_{i+1}} - \frac{1}{u_i}\right)$$

it is readily cast in the functional form

$$H_s(u, x) = 0 \tag{6.105}$$

This is the final function required. Knowing the inter-dependent functions F_i, G_{i+1} and H_s, it is possible to solve for the two accelerated states u_i and u_{i+1} in the shock transition zone. To this end, an iterative solution procedure such as the Newton method is used.

Admissibility of Accelerated Update

Earlier comments concerning whether at a given point in time in marching a solution to an IVBP, one has attained the basin of attraction of the governing PDE and, whether it is safe to introduce the acceleration process or indeed maintain it are applicable to all numerical methods. Consequently, the question of admissibility of an accelerated update also arises with the Engquist-Osher scheme. The matters of deciding when to commence acceleration and whether to desist from acceleration once initiated, were both successfully overcome by introducing two admissibility criteria. The first admissibility criterion is slightly different to the single transition point entropy criterion used in accelerating the Roe scheme since the Engquist-Osher scheme captures shocks with two transition points. Indeed, the way in which the Engquist-Osher scheme represents discrete shock structure allows for the development of a more rigorous entropy criterion than that developed for Roe's scheme. The second criterion developed sets a bound on the shock location.

The admissibility criteria are carried out as such. First, the pair of cells in the transition zone are identified. Generally one of these two cells contains supercritical flow, the other contains flow in a subcritical state and, somewhere between them the flow is notionally sonic. Denote the supercritical transition point, as $u^{\uparrow}$, its subcritical neighbour as $u^{\downarrow}$, and the value of the notional sonic condition as u_s. Then

$$u_i > u_s > u_{i+1} \tag{6.106}$$

For the model equation, the solution at the sonic point has a value of unity. Thus, under the above condition it is obvious that

$$\begin{aligned} u^{\uparrow} &= u_i \\ u^{s} &= 1 \\ u^{\downarrow} &= u_{i+1} \end{aligned} \tag{6.107}$$

Next, define the cell at the head of the shock as having a value u^h, while that at the foot of the shock has a value u^f. Then a suitable criterion for assessing the admissibility of an accelerated update is

$$u^h > u^{\uparrow} > u^s > u^{\downarrow} > u^f \tag{6.108}$$

In practice, this is best implemented as

$$u_{i-1} - \epsilon > u_i > 1 > u_{i+1} > u_{i+2} - \epsilon \tag{6.109}$$

with the familiar use of indices. The coefficient ϵ, that has been introduced is a small number used as a safety factor to ensure that any non-monotonic acceleration update does not beat the test, as such an occurrence could cause potential divergence or a spurious solution.

In addition, a second test was devised, originally as a final safety net to guard against the possibility of the Newton procedure converging to a spurious root. The test places a simple bound on the range of values that the shock location x_s may assume. The permissible range is given by

$$x_i \leq x_s \leq x_{i+1}$$

in which (x_i, x_{i+1}) are the locations of the shock transition points. If the value of x_s returned from the Newton procedure is outside this range, the accelerated update for (u_i, u_{i+1}) is rejected. In any event, provided suitable guesses are chosen to initiate the Newton iterative procedure in solving for the transition point updates, problems are not encountered for the range of conditions tested.

Unlike convergence acceleration for the Roe scheme, no special tests were implemented for the shock accelerated Engquist-Osher method when the shock was migrating from one cell to another. If an update was encountered that was deemed by the above criteria to be inadmissible, the procedure was simply to reject it in favour of the standard non-accelerated update.

6.1.8 Performance of the accelerated Engquist-Osher scheme

The Standard Engquist-Osher scheme was validated for correctness against the analytic solution before commencing work on convergence acceleration. Once

shock acceleration procedures had been successfully devised, the accelerated Engquist-Osher scheme was validated in comparison with numerical results from both the baseline scheme as well as the Roe scheme using several different boundary conditions and grids. As with the Roe scheme, the accuracy of the accelerated solutions was found to be commensurate with the baseline scheme. Unlike solutions from the accelerated Roe scheme, those from the Engquist-Osher scheme do not require specification of when acceleration is to be commenced. This is handled automatically inside the code through the admissibility criteria which reliably screens unacceptable updates even from the onset of computation. The combined features of the two-point shock acceleration model and the more rigorous entropy criterion would appear to account for the shock accelerated Engquist-Osher scheme's increased robustness over the similarly accelerated Roe scheme.

Of the test cases carried out, two are presented here to demonstrate the efficacy of the new acceleration method. The results shown are for the same cases as used in section 6.1.6 for the Roe scheme. Again, since the accelerated solutions cannot be distinguished within plotting resolution from the solutions of the baseline scheme, they are not presented. The first comparison between an accelerated solution and the baseline method is given in Figure 6.9. The initial conditions are constant data from the left boundary to the first point on the positive side of the x-axis and, other constant data from the second cell to the right boundary. The mesh size is 11 points while the boundary conditions are $u(-1,t) = 2.0$, and $u(1,t) = 0.42$. The convergence histories are of $\log \|du/dt\|_2$ plotted against iteration count. The standard scheme without acceleration exhibits a similar trace to the standard Roe scheme for the same problem (cf. Figure 6.6). In the latter stages of convergence the solution develops some oscillation which settles down after 500 iterations to a high-frequency, low amplitude limit cycle. Convergence of the accelerated scheme also terminates in a limit cycle, after sustaining a slightly lower drop in the l_2 norm after 80 iterations. In terms of the operation count, the accelerated algorithm is approximately 15% more expensive per iteration than the standard algorithm for the model equation. Thus the savings in computational expense through convergence acceleration is in excess of 80%.

The second test case uses the boundary conditions $u(-1,t) = 2.0$, and $u(1,t) = 0.45$. Here the initialization that is adopted imposes a shock three cells upstream from its terminal location in steady flow. The relevant convergence histories are presented in Figure 6.10. The trace of the standard scheme bottoms after 327 iterations in comparison with the accelerated solution which takes 63 iterations. When the increased computational expense of evaluating accelerated updates is taken into account, the accelerated scheme demonstrates a saving of more than 75% in computational expense.

6.2 Inhomogeneous Systems of Conservation Laws

The flow of a fluid through a variable area duct or nozzle may be approximated by the quasi one-dimensional Euler equations. These equations constitute an inhomogeneous, coupled system of nonlinear PDEs that admit elliptic, hyperbolic and mixed flow solutions. They have become popular in constructing test cases for multi-dimensional numerical shock capturing algorithms because they can

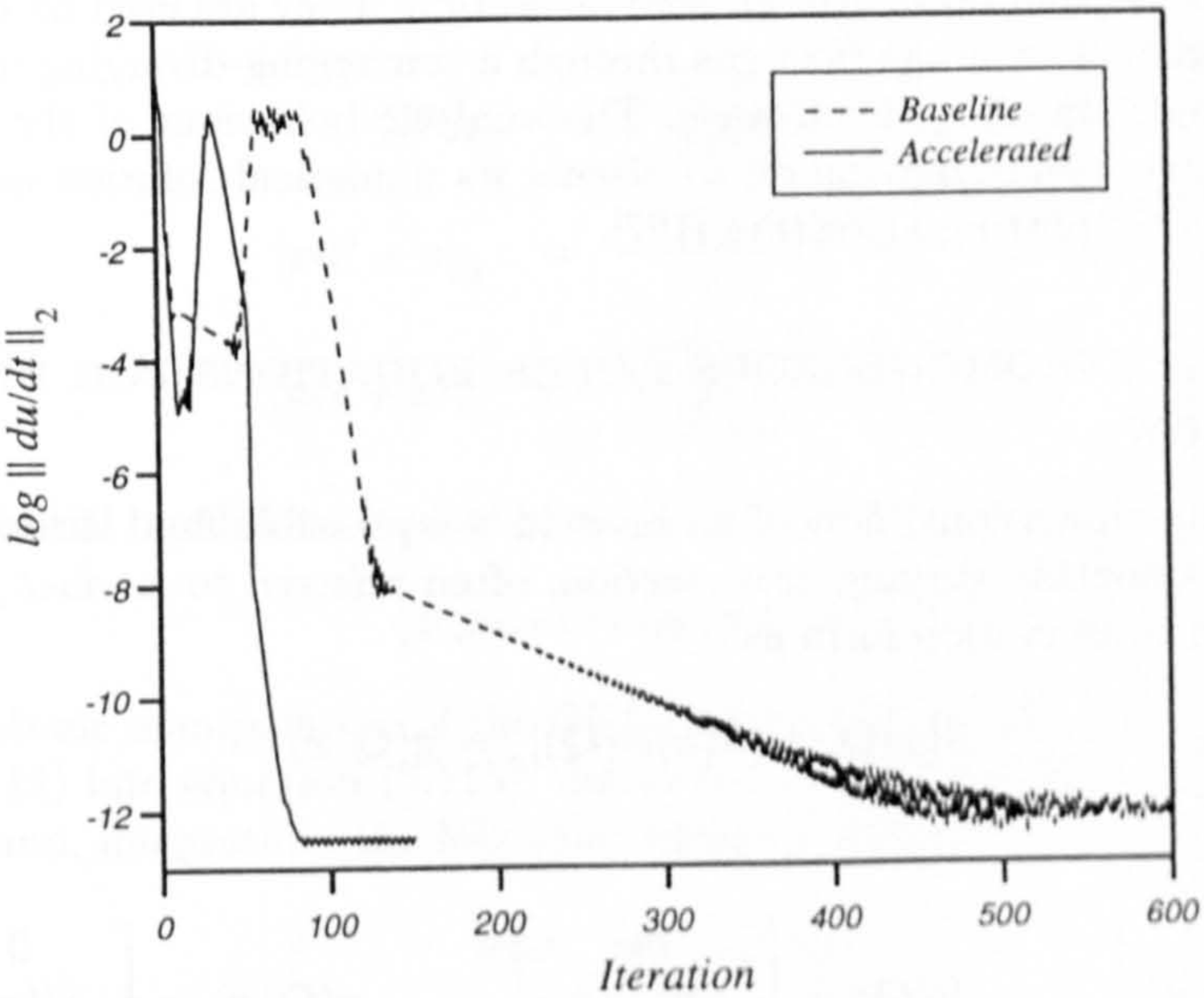


FIGURE 6.9. Comparison of convergence histories of the Engquist-Osher scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.42$.

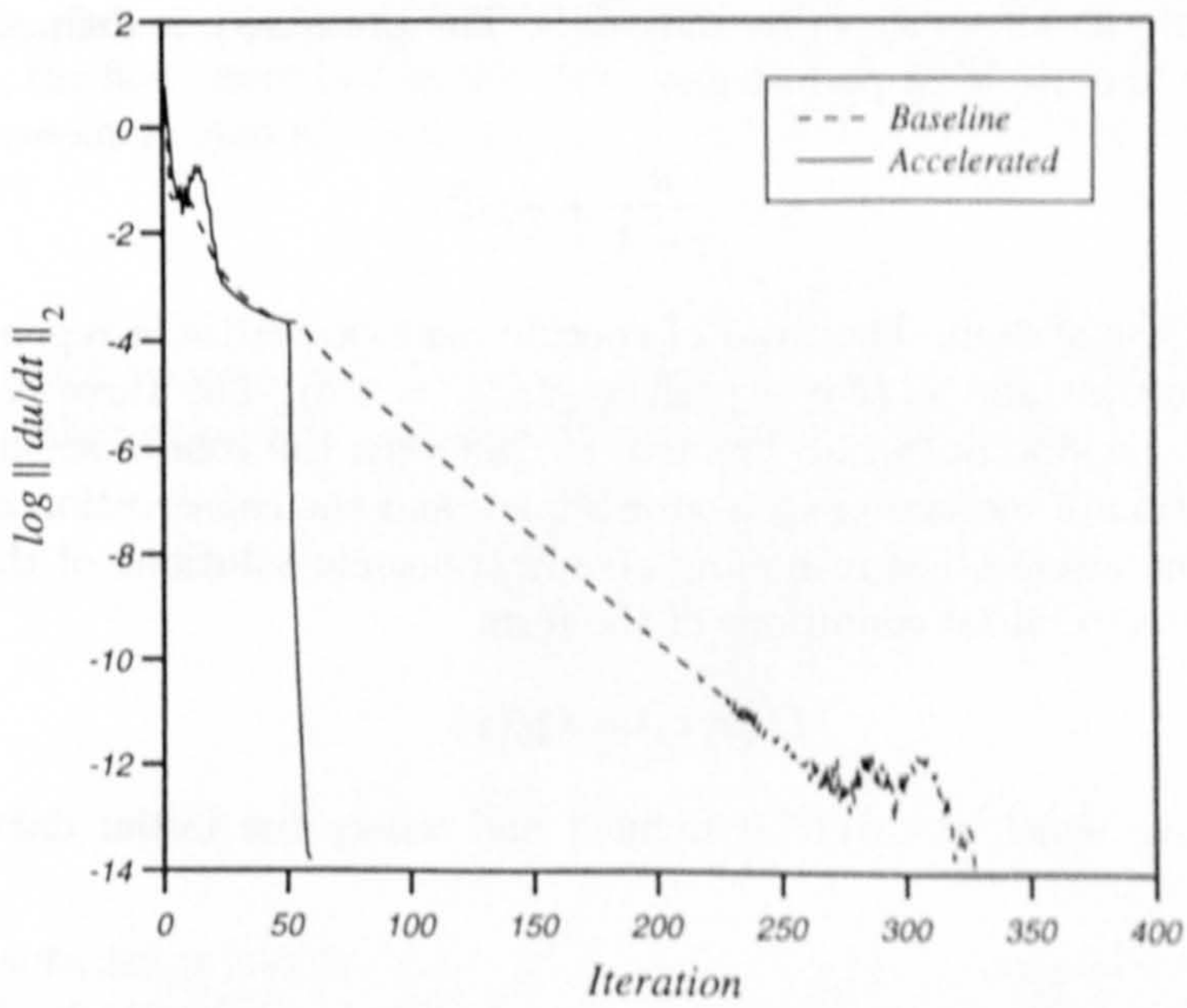


FIGURE 6.10. Comparison of convergence histories of the Engquist-Osher scheme for the model equation with boundary conditions $u(-1,t) = 2.0$ $u(1,t) = 0.45$.

render explicit steady shock wave solutions with smooth structure that can resemble the pressure profiles over an aerofoil section. They are used here to model steady transonic flow of a perfect gas through a converging-diverging nozzle containing a single standing shock wave. The analytic behaviour of the system is considered first, before moving on to discuss its numerical solution using Roe's Flux Difference Splitting algorithm [127].

6.2.1 The inhomogeneous Euler equations for duct flow

The quasi one-dimensional flow of an inviscid compressible fluid through a duct or nozzle of smoothly varying cross-section, often referred to as *duct flow*, may be written in conservation form as

$$S(x)\mathbf{Q}_t + [S(x)\mathbf{F}(\mathbf{Q})]_x = \mathbf{g}(\mathbf{Q}, x) \tag{6.110}$$

in which

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \qquad \mathbf{F}(\mathbf{Q}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix}, \qquad \mathbf{g}(\mathbf{Q}, x) = \begin{bmatrix} 0 \\ S'(x)p \\ 0 \end{bmatrix} \tag{6.111}$$

Here the subscripts t and x represent differentials with respect to time and to distance along the duct. $S(x)$ is the area of the cross-section of the duct at x, ρ the fluid density, u the velocity and, e the total energy. The source term $\mathbf{g}$ arises from the area variation of the duct geometry, though more generally, it could be constructed to represent other effects such as chemical reaction, mass or energy release or, interactions with other materials. The pressure p is defined through the equation of state for a perfect gas

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \tag{6.112}$$

which closes the system. The ratio of specific heat capacities is represented by the adiabatic constant γ. (For a perfect gas, $\gamma = 1.4$). The three component equations of the system (6.110) express, respectively, the conservation of mass, the conservation of momentum (Newton's law), and the conservation of energy.

The present interest lies in finding steady transonic solutions of the system (6.110) subject to initial conditions of the form

$$\mathbf{Q}(x, t_0) = \mathbf{Q}_0(x) \tag{6.113}$$

for all x along which the duct is defined and where the initial data $\mathbf{Q}_0$ are prescribed.

6.2.2 The analytic solution

First consider steady-state solutions in smooth regions of flow. The analysis described below follows that given by Glimm, Marshall and Plohr [50]. In the steady-state $\mathbf{Q}$ is independent of time, so that $\mathbf{Q}_t = 0$ and $\mathbf{Q}(x, t) = \mathbf{Q}_0(x)$ for

all x and t. Thus, for $\mathbf{Q}$ to be the solution of a steady flow, the system (6.110) reduces to the system of ordinary differential equations

$$[\rho u]_x = -\frac{S'(x)}{S(x)}\rho u \tag{6.114}$$

$$[\rho u^2 + p]_x = -\frac{S'(x)}{S(x)}\rho u^2 \tag{6.115}$$

$$[u(e+p)]_x = -\frac{S'(x)}{S(x)}[u(e+p)] \tag{6.116}$$

with the initial conditions that at $x = x_0$,

$$(\rho, \rho u, e) = (\rho_0, \rho u_0, e_0)$$

After algebraic manipulation of the ODE system; substitution of the equation of state (6.112) into equation (6.115); use of the relation $p = \Sigma\rho^\gamma$, where Σ is the entropy; and, integration, the following relations emerge:

$$\rho u S = \rho_0 u_0 S_0 \tag{6.117}$$

$$\frac{p}{\rho^\gamma} = \frac{p_0}{\rho_0^\gamma} \tag{6.118}$$

$$\frac{1}{2}u^2 + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = \frac{1}{2}u_0^2 + \frac{\gamma}{\gamma-1}\frac{p_0}{\rho_0} \tag{6.119}$$

where $S = S(x)$, $S(x_0) = S_0$. The latter three equations state respectively that the mass flux, the entropy and the total energy are constant in smooth, steady duct flow. Conditions denoted by the subscript 0 may be taken as reference conditions, such as the fluid state at the duct inlet, the throat or the exit. Furthermore, the flow described by this system must be isentropic since shock waves are not present in smooth regions. Introducing the speed of sound a as

$$a^2 = \frac{\gamma p}{\rho} \tag{6.120}$$

and the Mach number $M = u/a$, the system of equations immediately above may be further simplified and expressed in terms of M, p and a as

$$\frac{MpS}{a} = \frac{M_0 p_0 S_0}{a_0} \tag{6.121}$$

$$\frac{p^{\gamma-1}}{a^{2\gamma}} = \frac{p_0^{\gamma-1}}{a_0^{2\gamma}} \tag{6.122}$$

$$\left(1 + \frac{\gamma-1}{2}M^2\right)a^2 = \left(1 + \frac{\gamma-1}{2}M_0^2\right)a_0^2 \tag{6.123}$$

Finally, eliminating p and a from equations (6.121) and (6.122), then substituting the result into equation (6.123) produces a single equation for M, which may be written in the form

$$F(M) = \frac{S_0}{S}F(M_0) \tag{6.124}$$

for which the function F is defined as

$$F(M) = \left(\frac{(\gamma - 1)/2}{1 + ((\gamma - 1)/2)M^2} \right)^{(\gamma+1)/2(\gamma-1)} \tag{6.125}$$

Solution of the steady-state equations for smooth flow is now reduced to the following procedure:

1. Solve equation (6.124) for M using an iterative technique with appropriately chosen reference conditions M_0. (For critical flow in a converging-diverging nozzle, $M_0 = 1$ at the throat);
2. Use equation (6.123) to find a;
3. From equation (6.122) determine the pressure distribution p; and,
4. Finally obtain the density ρ from equation (6.120).

Now consider the properties of the function $F(M)$:

$$F(0) = 0$$

$$F(M) \to 0 \quad \text{as} \quad M \to \infty$$

$$F_{\max}(M) = F(1) = 1$$

and the implications that they have on the solution. Figure 6.11 illustrates the graph of $F(M)$ plotted against M. Note that if $S < S_0 F(M_0)$, a solution cannot

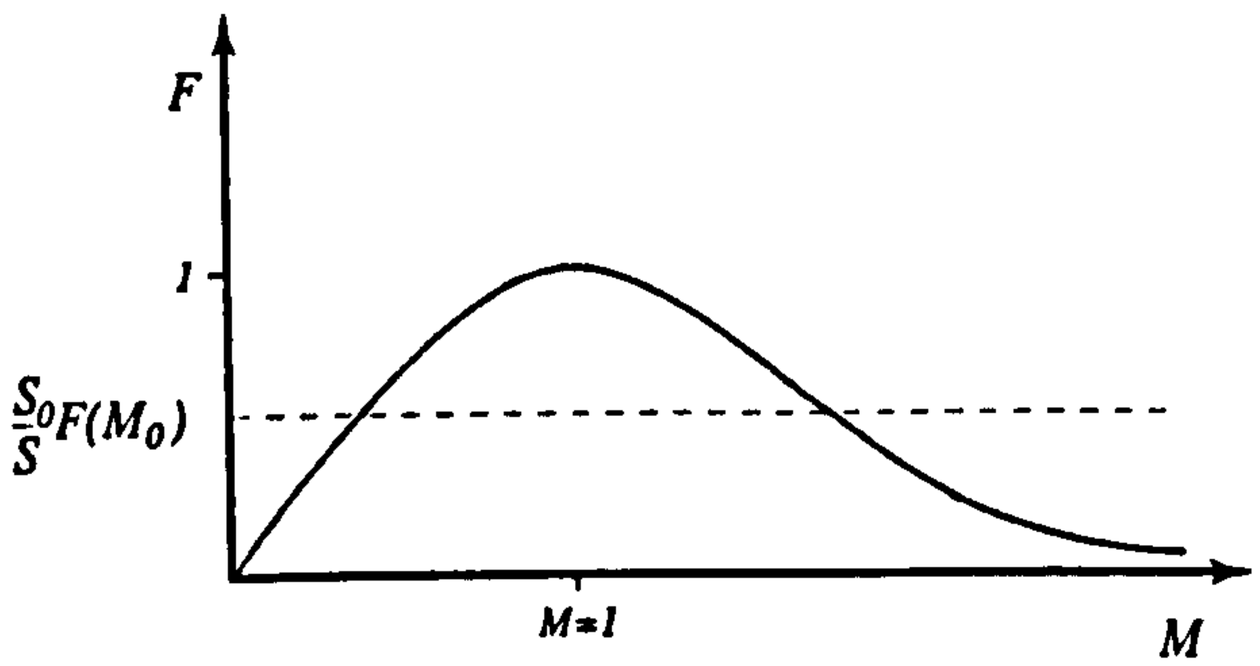
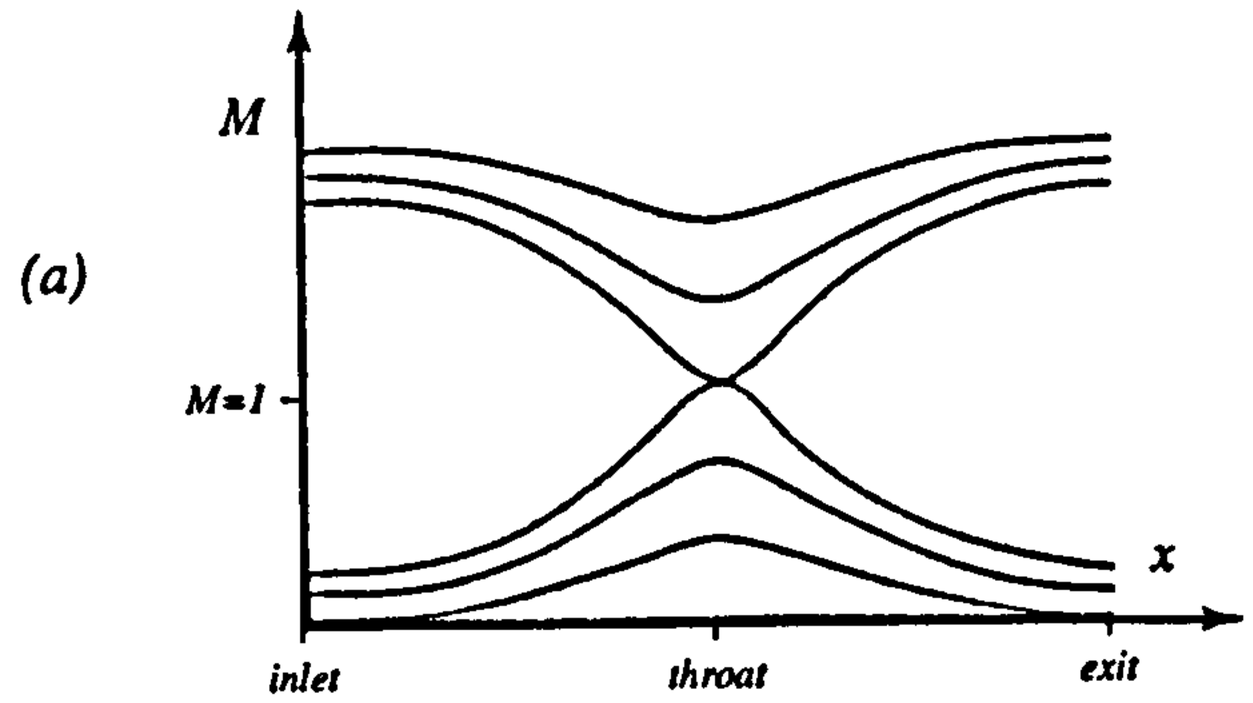


FIGURE 6.11. Graph of the function defined by equation (6.124).

exist since the right-hand side of equation (6.124) would have to exceed the maximum value of F. If $S > S_0 F(M_0)$, equation (6.124) has two roots, one subcritical, the other supercritical, corresponding to subsonic and supersonic flow respectively. The function is singular at its maximum $F(1)$ where the flow is sonic. Thus for critical flow through a converging-diverging nozzle (i.e. $M_{throat} = 1$), there are two solutions to equation (6.124), one corresponding to a subsonic Mach number at the inlet to the nozzle, the other to a supersonic inlet Mach number. Figure 6.12(a) shows the graph of Mach number against distance along the nozzle



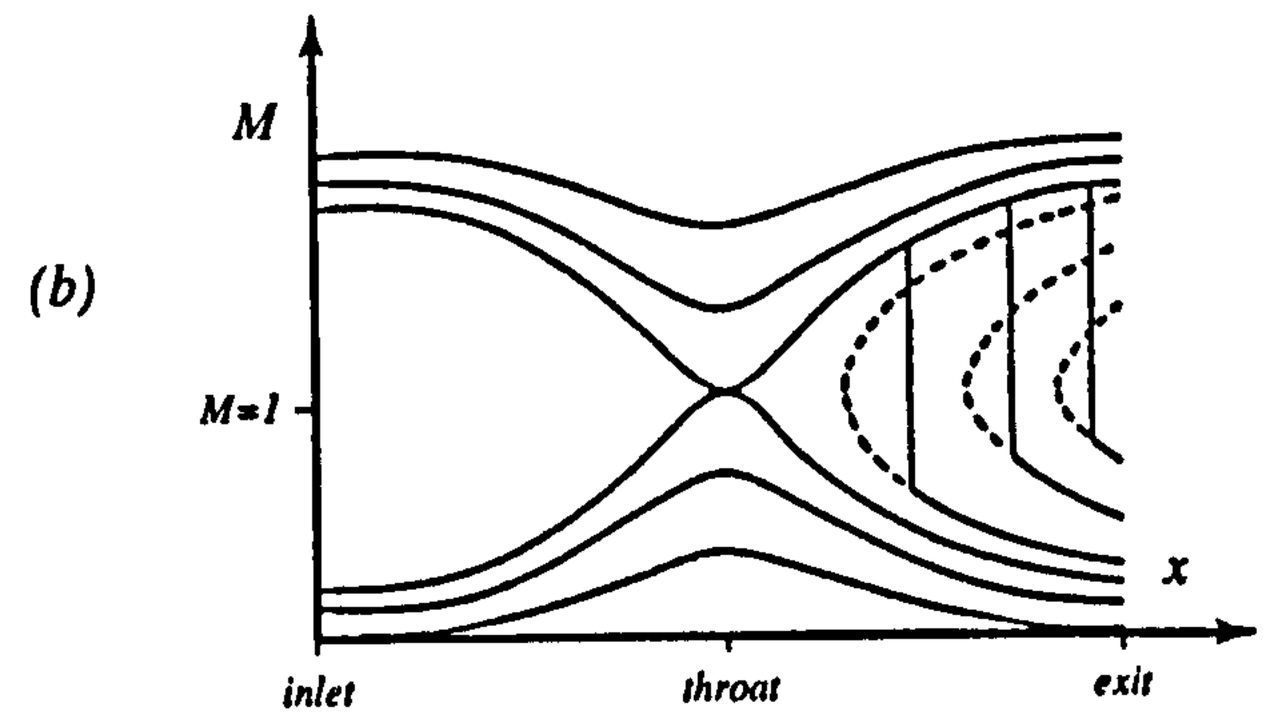


FIGURE 6.12. Various graphs of Mach number against distance along a nozzle. (a) Shockless steady flows and (b) steady flows with a standing shock.

for a range of inlet Mach numbers. The lower family of curves correspond to subsonic flow throughout the nozzle, with a rise in the speed of flow at the throat as predicted by Bernoulli's theorem. The upper family of curves represent supersonic flow solutions throughout the nozzle with decreasing speed of flow as the throat is approached. The critical curves, one subsonic, the other supersonic, meet in the throat at the singular point $M = 1$. It is important to note that for the *subsonic* inlet Mach number that admits critical flow at the throat, there are two solutions, one for which the flow is subsonic throughout, the other for which the flow is subsonic up to the throat, but supersonic from the throat to the outlet. In the latter case the solution branches at the sonic point. Indeed, the critical curves have zero slope at the throat, a fact that is often overlooked in texts that do not discuss the analytic behaviour thoroughly and which illustrate the subsonic flow transitioning into supersonic flow without any hint of inflection. Obviously, the severity of the point of inflection in the Mach number or pressure distribution that occurs at the nozzle throat when the flow transitions from subsonic to supersonic is a function of the rate of area change in a given nozzle $S'(x)$, and varies from one geometry to another. As an aside, it should be noted that the non-uniqueness resulting from the branching of the solution when the flow passes through a Mach number of unity is precisely the cause of difficulty

from which non-entropy-satisfying numerical methods fall foul, resulting in the possible admission of an expansion shock, glitch, two-cell plateau or something in between. In addition, solutions containing these oddities are often slow to converge [168].

For a steady shock wave to develop in the nozzle, the outlet or exit Mach number may *not* be one of those shown in Figure 6.12(a). Some of the exit Mach numbers that do permit a shock wave are shown in part (b) of the figure which includes the same solution branches for smooth flow given in part (a). Solutions containing shock waves are composed of four flow regimes: a subsonic critical flow from inlet to the throat; a supersonic critical flow from the throat to the head of shock; a discontinuity in the divergent portion of the nozzle corresponding to the shock position; and, a subsonic region joining the foot of the shock wave to the nozzle exit. Shock solutions may be parametrized by shock strength, position, exit Mach number or exit pressure (back pressure). In a physical experiment, it is the back pressure that is used as the control agent for shock location. When an appropriate pressure is specified at the exit to the duct, a standing shock wave develops in the diverging section of the duct. In the asymptotic limit of a steady solution, across the shock the Rankine-Hugoniot jump condition

$$\rho_1 u_1 = \rho_2 u_2 \tag{6.126}$$

$$\rho_1 u_1^2 + p_1 = \rho_2 u_2^2 + p_2 \tag{6.127}$$

$$\frac{1}{2}\rho_1 u_1^2 + h_1 = \frac{1}{2}\rho_2 u_2^2 + h_2 \tag{6.128}$$

and the entropy condition

$$u_1 - a_1 > 0 > u_2 - a_2 \tag{6.129}$$

are satisfied. The equation immediately above assumes left to right flow. The numbered subscripts denote limiting values as the shock is approached from the left (subscript 1) and as it is approached from the right (subscript 2). i.e. conditions at the head and at the foot of the shock, respectively. h is the specific enthalpy which is given by

$$h = (e + p)/\rho \tag{6.130}$$

For a standing shock, the Rankine-Hugoniot relations above state that the mass flux does not change across the shock and, that the momentum and energy fluxes change only according to the drop in pressure across it, as determined by Newton's law. Thus, equations (6.126–6.128) determine the state at the foot of the shock in terms of (known) conditions at its head. The flow downstream of the shock may be determined by using the state at the foot of the shock found from the Rankine-Hugoniot relations, as the reference conditions for solution of equations (6.124), (6.123), (6.122) and (6.120), as described in the numbered procedure listed above for regions of smooth flow.

For fluids that satisfy the given equation of state, the geometric entropy condition of equation (6.129) is a statement to the fact that the entropy increases as a fluid particle crosses a standing shock. It is equivalent to the requirement that the flow ahead of the standing shock be supersonic. This precludes solutions that one could envisage such as supersonic critical flow from the inlet to the throat, subsonic critical flow from the throat to a shock and, supersonic flow again from

the shock to the exit. Other solutions are possible such as supersonic inflow with a shock at the inlet, subsonic critical flow from the foot of the shock to the throat and, supersonic critical flow downstream of the throat, as well as solutions containing multiple shock waves as discussed by Embid, Goodman and Majda [39]. These type of solutions are not considered in the present study which is confined to transonic flows that are representative of typical aeronautical applications.

6.2.3 The numerical solution –the Roe scheme

The numerical solution for the one-dimensional homogeneous Euler equations using the Roe scheme has already been described in section 2.4.2. Here it is extended to the inhomogeneous Euler equations for duct flow. Various techniques have been developed for dealing with source terms in hyperbolic problems. Some discussion of these techniques is given in [153] and [129]. The approach adopted here was first advanced by Roe [129, 131] and is termed *source-term splitting.* Using this approach, the source term is upwinded together with the convective term, resulting in a consistent spatial discretization. Lafon and Yee [81] have demonstrated for a model reaction-convection problem that upwinding both source and convective terms produces better nonlinear behaviour than upwinding only the convective term either with point-wise evaluation of the source term or mean averaging. Source splitting also offers the advantage that it imparts second-order accuracy to first-order upwind solvers when the solution becomes steady [164, 129, 168].

In section 6.2.1 the inhomogeneous Euler equations for duct flow were presented. They will be recalled as

$$S(x)\mathbf{Q}_t + [S(x)\mathbf{F}(\mathbf{Q})]_x = \mathbf{g}(\mathbf{Q}, x)$$

in which

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \qquad \mathbf{F}(\mathbf{Q}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix}, \qquad \mathbf{g}(\mathbf{Q}, x) = \begin{bmatrix} 0 \\ S'(x)p \\ 0 \end{bmatrix}$$

Since $S(x)\mathbf{F}(\mathbf{Q}) = \mathbf{F}[S(x)\mathbf{Q}]$ and $S(x)\mathbf{Q}_t = [S(x)\mathbf{Q}]_t$, the first equation may be re-cast as

$$\bar{\mathbf{Q}}_t + \bar{\mathbf{F}}(\mathbf{Q})_x = \bar{\mathbf{g}}(\mathbf{Q}, x) \tag{6.131}$$

This gives rise to the new set of conserved variables $\bar{\rho}$, $\bar{m}$ and $\bar{e}$, where $\bar{\rho} = S(x)\rho$, $\bar{m} = \bar{\rho}u$ and $\bar{e} = S(x)e$. It also introduces a new pseudo-pressure variable $\bar{p} = S(x)p$. Quantities having the dimension of velocity such as the speed of sound a and enthalpy H, remain unchanged. In particular, the flux Jacobian of the new system is unaltered, i.e. $\bar{\mathbf{A}} = \partial\bar{\mathbf{F}}(\mathbf{Q})/\partial\mathbf{Q} = \partial\mathbf{F}(\mathbf{Q})/\partial\mathbf{Q}$. In the new variables, the Euler equations for duct flow become

$$\bar{\mathbf{Q}} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}u \\ \bar{e} \end{bmatrix}, \qquad \bar{\mathbf{F}}(\mathbf{Q}) = \begin{bmatrix} \bar{\rho}u \\ \bar{\rho}u^2 + \bar{p} \\ u(\bar{e}+\bar{p}) \end{bmatrix}, \qquad \bar{\mathbf{g}}(\mathbf{Q}, x) = \begin{bmatrix} 0 \\ \frac{S'(x)}{S(x)}\bar{p} \\ 0 \end{bmatrix} \tag{6.132}$$

They are closed by the equation of state for an ideal gas. In the new variables it is given by

$$\bar{p} = (\gamma - 1)\left[\bar{e} - \bar{\rho}u^2/2\right] \tag{6.133}$$

Using the forward-Euler time update together with Roe's flux function, a formally first-order accurate discrete approximation for equation (6.131) is

$$\bar{\mathbf{Q}}_i^{n+1} = \bar{\mathbf{Q}}_i^n - \frac{\Delta t}{\Delta x}\left[\bar{\mathbf{F}}_{i+\frac{1}{2}}^n - \bar{\mathbf{F}}_{i-\frac{1}{2}}^n - \frac{\Delta x}{2}\left(\tilde{\bar{\mathbf{g}}}_{i-\frac{1}{2}}^n + \tilde{\bar{\mathbf{g}}}_{i+\frac{1}{2}}^n\right)\right]^n \tag{6.134}$$

for which

$$\bar{\mathbf{F}}_{i+\frac{1}{2}}(\bar{\mathbf{Q}}_L, \bar{\mathbf{Q}}_R) = \frac{1}{2}\left[\bar{\mathbf{F}}_L + \bar{\mathbf{F}}_R - \sum_{k=1}^{3} \mathbf{r}_k \left[|\lambda_k|\alpha - |z_k|\Delta x\right]\right] \tag{6.135}$$

The notation for the first term of the summation appearing in the above equation has already been introduced in section 2.4.2. It will be recalled that λ_k is the kth characteristic speed, $\mathbf{r}_k$ is the corresponding right eigenvector and, α the projection of the jump in the state vector onto the left eigenvectors of the flux Jacobian $\tilde{\mathbf{A}}$, equations (2.45–2.47) refer. All these interface quantities are evaluated using Roe-averages denoted by the tilde, see equations (2.40–2.42). The last term of the summation in the above equation represents the decomposition

$$\tilde{\bar{\mathbf{g}}}(\mathbf{Q}^n) = \sum_{k=1}^{3} |z_k| \mathbf{r}_k \tag{6.136}$$

and is analogous to Roe's method for homogeneous problems where the flux difference $\Delta\mathbf{F}$ is projected onto the eigenvectors of $\tilde{\mathbf{A}}$. For the convective term, each projection represents the contribution of a wave system to $\Delta\mathbf{F}$; for the source term each projection represents a wave system contribution to $\mathbf{g}(\mathbf{Q}^n)$. This procedure leads to the correct upwind treatment of source terms as shown by Roe [131] for linear conservation laws. With its use the source-term vector is decomposed into a linear combination of the right eigenvectors of $\tilde{\mathbf{A}}$. Expanding the above equation produces the algebraic system

$$\begin{aligned}
\tilde{\bar{g}}_1 &= z_1 + z_2 + z_3 \\
\tilde{\bar{g}}_2 &= z_1(\tilde{u} - \tilde{a}) + z_2\tilde{u} + z_3(\tilde{u} - \tilde{a}) \\
\tilde{\bar{g}}_3 &= z_1(\tilde{H} - \tilde{u}\tilde{a}) + \frac{1}{2}z_2\tilde{u}^2 + z_3(\tilde{H} + \tilde{u}\tilde{a})
\end{aligned} \tag{6.137}$$

where z_k $(k = 1, 3)$ are the components of $\mathbf{Z}$ associated with the source contribution from the kth wave. These equations are now solved for z_k. After simplification using the identity

$$H = \frac{\tilde{a}^2}{\gamma - 1} + \frac{1}{2}\tilde{u}^2$$

the following expressions emerge

$$\begin{aligned}
z_1 &= \frac{\gamma - 1}{2\tilde{a}^2}\left[-\left(\frac{1}{\gamma - 1} + \frac{\tilde{u}}{2}\right)\tilde{u}\tilde{\bar{g}}_1 + \left(\frac{\tilde{a}}{\gamma - 1} + \tilde{u}\right)\tilde{\bar{g}}_2 - \tilde{\bar{g}}_3\right] \\
z_2 &= \frac{\gamma - 1}{2\tilde{a}^2}\left[\left(\tilde{u}^2 - \frac{2\tilde{a}^2}{\gamma - 1}\right)\tilde{\bar{g}}_1 - 2\tilde{u}\tilde{\bar{g}}_2 + 2\tilde{\bar{g}}_3\right] \\
z_3 &= \frac{\gamma - 1}{2\tilde{a}^2}\left[\left(\frac{\tilde{a}}{\gamma - 1} - \frac{1}{2}\tilde{u}\right)\tilde{u}\tilde{\bar{g}}_1 + \left(\tilde{u} - \frac{\tilde{a}}{\gamma - 1}\right)\tilde{\bar{g}}_2 - \tilde{\bar{g}}_3\right]
\end{aligned} \tag{6.138}$$

As written, these equations are valid for any source-term vector g and the method may be applied to a wide range of problems in gas dynamics involving source terms. For the present problem of flow through a variable-area duct, it will be recalled from equation (6.111) that the source components $\bar{g}_1$ and $\bar{g}_3$ are zero. Thus only the second component must be treated. Now using the relation

$$a^2 = \frac{\gamma p}{\bar{\rho}}$$

the second source component becomes

$$\bar{g}_2 = S' \frac{\bar{\rho}\tilde{a}^2}{\gamma} \tag{6.139}$$

At the discrete level it may be approximated as

$$\tilde{\bar{g}}_2 = \frac{S_i - S_{i-1}}{\Delta x} \frac{\tilde{\bar{\rho}}\tilde{a}^2}{\gamma} \tag{6.140}$$

Substituting equation (6.140) into equation (6.138), and following the algebra through, the components z_k simplify to

$$\begin{aligned} z_1 &= \frac{\tilde{\bar{\rho}}\Delta S}{2\gamma\tilde{S}} [(\gamma-1)\tilde{u} + \tilde{a}] \\ z_2 &= -\frac{(\gamma-1)\tilde{\bar{\rho}}\tilde{u}\Delta S}{2\gamma\tilde{S}} \\ z_3 &= \frac{\tilde{\bar{\rho}}\Delta S}{2\gamma\tilde{S}} [(\gamma-1)\tilde{u} - \tilde{a}] \end{aligned} \tag{6.141}$$

To ensure consistency with the new variables, the cell interface area $\tilde{S}$ is also Roe averaged. In addition, the wave strengths α may be written in the new variables as

$$\begin{aligned} \alpha_1 &= \left(\Delta\tilde{\bar{p}} - \tilde{\bar{\rho}}\tilde{a}\Delta\tilde{u}\right)/2\tilde{a}^2 \\ \alpha_2 &= \Delta\tilde{\bar{\rho}} - \Delta\tilde{\bar{p}}/\tilde{a}^2 \\ \alpha_3 &= \left(\Delta\tilde{\bar{p}} + \tilde{\bar{\rho}}\tilde{a}\Delta\tilde{u}\right)/2\tilde{a}^2 \end{aligned} \tag{6.142}$$

The remaining question to be dealt with is entropy satisfaction. Section 2.4.5 discussed the need for a mechanism to ensure that the solutions of numerical methods which do not naturally satisfy entropy, do in fact converge to the physically relevant solution in the case of flow containing a sonic expansion. In that section two entropy fixes are described. One of them has been extended by van Leer, Lee and Powell [168] to deal with source terms. This is carried out so that the entropy corrections to the source terms are consistent with those of the convective fluxes. Recall that the Harten/van Leer entropy fix redefines the absolute characteristic speeds $|\lambda_k|$ for the acoustic waves as

$$|\lambda_k^*| = \begin{cases} |\lambda_k|, & |\lambda_k| \geq \frac{1}{2}\delta\lambda_k \\ \frac{\lambda_k^2}{\delta\lambda_k} + \frac{1}{4}\delta\lambda_k, & |\lambda_k| < \frac{1}{2}\delta\lambda_k \end{cases}, \qquad k = 1 \text{ and } k = 3 \tag{6.143}$$

where

$$\begin{aligned} \Delta\lambda_k &= \lambda_{k,R} - \lambda_{k,L} \\ \delta\lambda_k &= \max(4\Delta\lambda_k, 0) \end{aligned}, \qquad k = 1 \text{ and } k = 3 \tag{6.144}$$

The subscript k represents the individual wave while the subscripts L, R represent the respective left and right values straddling the given interface. To extend this formulation to deal with the source term, van Leer *et al.* [168] also redefine the components of $|\mathbf{Z}|$ corresponding to the acoustic contributions as

$$|z_k^*| = \begin{cases} |z_k|, & |z_k| \geq \frac{1}{2}\delta z_k \\ \frac{z_k^2}{\delta z_k} + \frac{1}{4}\delta z_k, & |z_k| < \frac{1}{2}\delta z_k \end{cases}, \qquad k = 1 \text{ and } k = 3 \tag{6.145}$$

where

$$\begin{aligned} \Delta z_k &= z_{k,R} - z_{k,L} \\ \delta z_k &= 4\max(\Delta z_k, 0) \end{aligned}, \qquad k = 1 \text{ and } k = 3 \tag{6.146}$$

For the linear-degenerate wave and its source contribution, no special procedure is taken, i.e.

$$\begin{aligned} |\lambda_k^*| &= |\lambda_k^*| \\ |z_k^*| &= |z_k^*| \end{aligned}, \qquad k = 2 \tag{6.147}$$

Thus the entropy condition is enforced by use of the redefined variables $|\lambda_k^*|$ and $|z_k^*|$ substituted into equation (6.135). This eliminates any possible occurrence of expansion shocks and ensures a smooth transition from subsonic to supersonic flow.

Turning to boundary condition procedures, these are implemented using two fictitious or ghost cells at each boundary. For a transonic nozzle, the flow is subsonic both at the inlet and exit. At the inlet there are two incoming characteristics and one outgoing one, necessitating two physical boundary conditions and one numerical boundary condition. The former are prescribed by assuming isentropic flow and constant entropy. These relations provide the pressure and the density for the ghost cells. The numerical boundary condition is prescribed via extrapolation of velocity from the interior. At exit there is one incoming characteristic and two outgoing ones. To represent this situation, the pressure is prescribed while density and velocity are extrapolated from the interior. Similar boundary condition procedures may be found in Glimm, Marshall and Plohr [50]. A comprehensive analysis of boundary condition procedures for one-dimensional problems has been carried out by Gottlieb and Zhang [57].

Glaister [49] has used the same source-term splitting approach for the computation of time-dependent problems. His implementation differed from that described here, in as much as he used Roe's fluctuation splitting update rather than the formulation given by equations (6.134) and (6.135), in a formally second-order version of the algorithm, without the entropy fix described above. One of the problems he considered is a spherically infinite diverging shock wave. The numerical results obtained for this stringent test are in close agreement with the analytic solution and demonstrate higher accuracy than can be attained from more traditional methods. Toro [153] has also successfully used the source-term splitting technique and compared it with other source discretization techniques with application to combustion problems.

6.2.4 Shock Acceleration for Roe's scheme

The Basic Approach

To accelerate the convergence of Roe's scheme for the inhomogeneous Euler equations for duct flow, the approach proposed in section 6.1.5 for the scalar equation is adopted. Specifically, the algebraic single-point shock acceleration model is used. However, the chosen acceleration technique must first be extended to deal with the numerical scheme applied to a system of equations. The manner in which this is done is now presented.

In the steady-state solution of a system of hyperbolic conservation laws, the residual vector $\vec{\mathrm{R}}$ of Roe's first-order upwind scheme is given by

$$\begin{aligned}
&\vec{\mathrm{R}}_i = \Phi_{i-\frac{1}{2}}, && \text{for a non-shock point, } (\lambda_{k,i-\frac{1}{2}} \geq 0) \\
&\vec{\mathrm{R}}_i = \Phi_{i+\frac{1}{2}}, && \text{for a non-shock point, } (\lambda_{k,i+\frac{1}{2}} < 0) \\
&\vec{\mathrm{R}}_i = \Phi_{i-\frac{1}{2}} + \Phi_{i+\frac{1}{2}}, && \text{for a shock point, } (\lambda_{k,i-\frac{1}{2}} \geq 0 \geq \lambda_{k,i+\frac{1}{2}})
\end{aligned} \tag{6.148}$$

The subscript i denotes the cell to which the residual belongs. This equation is the counterpart to equation (6.67) which applies to the scalar case. The interface quantities $\Phi_{i\pm\frac{1}{2}}$ represent the residual contributions to cell i, being determined as a function of the wave speeds at each interface $\lambda_{k,i\pm\frac{1}{2}}$. Here the subscript k refers to an individual eigenvalue whereas $i \pm \frac{1}{2}$ denotes the spatial counter. For steady-state solutions, all residuals tend to zero, though they do not necessarily do so smoothly or at the same rate. A steady solution is reached when

$$\Phi_{i+\frac{1}{2}} = 0 \qquad \text{(for a non-shock point)} \tag{6.149}$$

$$\Phi_{i-\frac{1}{2}} + \Phi_{i+\frac{1}{2}} = 0 \qquad \text{(for a shock point)} \tag{6.150}$$

The single-point algebraic shock acceleration model is based on satisfaction of the equation (6.150) prior to its natural attainment in time. Forcing the residual in the shock cell to be zero is the mechanism that drives the acceleration process. At the abstract level, equation (6.150) is the acceleration model. The procedure has the effect of increasing the speed of movement of the shock to its steady-state position. This effectively damps the high-frequency error modes associated with discrete shock movement and ensures that these error modes decay at rates that are less disparate than the error modes from other parts of the flow. Hence, convergence of the numerical solution may be accelerated.

Formulating the Shock Acceleration Model

First, some notation must be explained. Recall that the diagonal matrix Λ contains the eigenvalues of the flux Jacobian in the ordering

$$\Lambda = \mathrm{diag}\,[\lambda_1, \lambda_2, \lambda_3] \tag{6.151}$$

$$= \mathrm{diag}\,[(\tilde{u} - \tilde{a}), \tilde{u}, (\tilde{u} + \tilde{a})] \tag{6.152}$$

These may be split into positive and negative contributions $\lambda_k^{\pm}$ using the definition

$$\lambda_k^{\pm}(\bar{\mathrm{Q}}_{i+\frac{1}{2}}) \equiv \lambda_{k,i+\frac{1}{2}}^{\pm} = \frac{\lambda_k(\bar{\mathrm{Q}}_{i+\frac{1}{2}}) \pm |\lambda_k(\bar{\mathrm{Q}}_{i+\frac{1}{2}})|}{2} \tag{6.153}$$

From this it follows that Λ^+ contains only positive eigenvalues while Λ^- contains only negative eigenvalues. As in the scalar case, quantities subscripted by the letter s are taken to represent the counter of the cell in which a shock is located. For systems of equations, the shock cell is defined as being the cell across which the acoustic wave speed changes sign. As there is also a change of sign in acoustic wave speed across a sonic expansion, care must be taken not to confuse an expansion with a shock wave. In fact, the shock cell is defined uniquely from the relationship

$$\lambda_k(\bar{\mathbf{Q}}_{i-\frac{1}{2}}) \geq 0 \geq \lambda_k(\bar{\mathbf{Q}}_{i+\frac{1}{2}}), \qquad k = 1 \text{ or } k = 3 \tag{6.154}$$

For flow from left to right, the shock is formed by the $u - a$ family of waves ($k = 1$), while for flow in the opposite direction it is formed by $u + a$ waves ($k = 3$). Where equation (6.154) holds, the shock cell is given by the cell counter $s = i$. Also, when moving through a one-dimensional shock, the wave speed may change sign only once, ensuring that the above condition is met in a single cell, even in instances when the numerical scheme captures the shock wave across more than one cell.

In the present wave decomposition scheme, the interface fluxes are computed according to the numerical flux defined by equation (6.135). The changes in the solution are brought about by the wave system at each interface where the Riemann problem is solved. The components in the numerical flux formulae that represents these changes may be defined at a generic interface as

$$\Phi_{i+\frac{1}{2}} = \sum_{k=1}^{3} \mathbf{r}_k \left[|\lambda_k| \alpha - |z_k| \Delta x \right]_{i+\frac{1}{2}} \tag{6.155}$$

in which $\Phi_{i+\frac{1}{2}}$ represents the sum of the individual residual components from each wave at the given interface. The residual components may contribute to one or both of the cells on either side of the interface depending on the upwind direction of each wave. For the purpose of the algebraic manipulations to follow, the above equation may be re-written in matrix form as

$$\Phi_{i+\frac{1}{2}} = \mathbf{R} \left[|\Lambda| \mathbf{R}^{-1} \Delta \mathbf{Q} - |\mathbf{Z}| \Delta x \right]_{i+\frac{1}{2}} \tag{6.156}$$

It will be remembered that $\mathbf{R}$ contains the matrix of right eigenvectors by column, $\mathbf{R}^{-1}$ the matrix of left eigenvectors by row, while $\Delta \mathbf{Q}$ is the jump in conserved variables across a cell interface. For a shock wave to form there must be a change in the sign of an acoustic wave. To understand how the acceleration procedure is to be applied, consider Figure 6.13 which shows the flow of information at the shock wave in flow moving from left to right. The interface to the left of the shock cell ($s = i$) is supersonic with all waves contributing to the residual at s. The interface to the right of the shock cell is subsonic with only one wave contributing to the shock cell, the $u - a$ wave. Thus all the waves at the left-hand interface contribute to the residual in the shock cell and have positive wave speed, while the single upstream-running wave at the right-hand interface that contributes to the shock residual has negative wave speed.

To satisfy the shock acceleration model given by equation (6.150), it is apparent that the expression

$$\mathbf{R} \left[\Lambda^+ \mathbf{R}^{-1} \Delta \mathbf{Q} - \mathbf{Z}^+ \Delta x \right]_{i-\frac{1}{2}} + \mathbf{R} \left[\Lambda^- \mathbf{R}^{-1} \Delta \mathbf{Q} - \mathbf{Z}^- \Delta x \right]_{i+\frac{1}{2}} = 0 \tag{6.157}$$

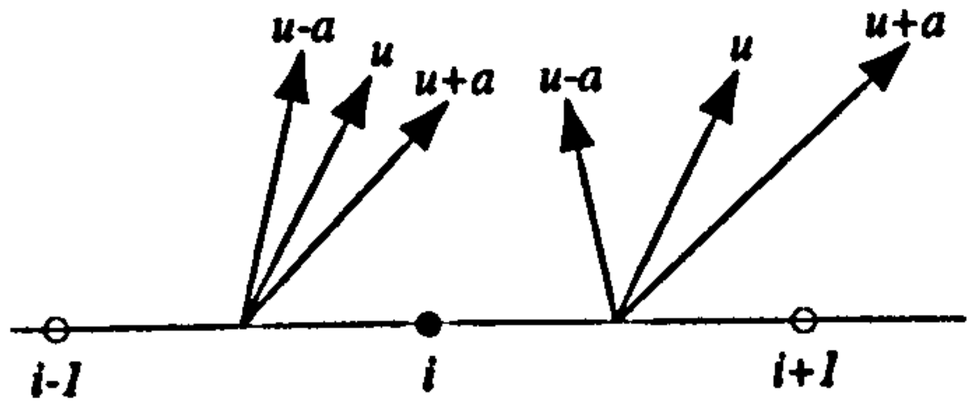


FIGURE 6.13. Flow of information at a shock wave for the Euler equations using Roe's method.

must hold. By definition, the above equation deals only with the residual components which feed the shock cell s and constitute its update. Hence it involves no absolute quantities. As such the source vector $\mathbf{Z}$ has also been split into positive and negative components corresponding to the sign of the eigenvalues. These components are defined as

$$\mathbf{Z}^{\pm} \equiv \text{sgn}\,(\Lambda^{\pm})\mathbf{Z} \tag{6.158}$$

Equation (6.157) is the expression used to obtain the solution in the shock cell. Although it is an algebraic relationship, it cannot be solved directly for the conserved variables at the shock even with the assistance of a symbolic algebra software package. This is because the interface quantities that it contains are Roe averages which make simplification in terms of the conserved variables intractable. Instead, the approach proposed here is to solve in terms of the primitive variables. It proceeds as follows. Define two matrices $\mathbf{B}^{\pm}$ such that $\mathbf{B}^{+}$ is the product of the right eigenvector matrix with the positive eigenvalues, while $\mathbf{B}^{-}$ is a similar product involving the negative eigenvalues. Thus

$$\mathbf{B}^{\pm} \equiv \mathbf{R}\Lambda^{\pm} \tag{6.159}$$

Equation (6.157) may now be re-written using the above definition as

$$\left[\mathbf{B}^{+}\mathbf{R}^{-1}\Delta\mathbf{Q} - \mathbf{Z}^{+}\Delta x\right]_{i-\frac{1}{2}} + \left[\mathbf{B}^{-}\mathbf{R}^{-1}\Delta\mathbf{Q} - \mathbf{Z}^{-}\Delta x\right]_{i+\frac{1}{2}} = 0 \tag{6.160}$$

Now recall that the wave strengths are composed of the jump in the conserved variables projected onto the left eigenvector matrix, viz. $\alpha = \mathbf{R}^{-1}\Delta\mathbf{Q}$. They may be composed equivalently in terms of the jumps in primitive variables $\Delta\mathbf{V}$ where

$$\Delta\mathbf{V} = \left[\Delta\tilde{\tilde{p}},\ \Delta\tilde{\tilde{\rho}},\ \Delta u\right]^{T} \tag{6.161}$$

in satisfaction of a new relationship

$$\mathbf{D}\Delta\mathbf{V} = \mathbf{R}^{-1}\Delta\mathbf{Q} \tag{6.162}$$

The above relation represents a change in variable from the jump in conserved variables to one in primitive variables. $\mathbf{D}$ is the transformation matrix for this variable change. It is straighforward to obtain $\mathbf{D}$ which is given by

$$\mathbf{D} = \begin{bmatrix} 1/2\tilde{a}^2 & 0 & -\tilde{\tilde{\rho}}/2\tilde{a}^2 \\ -1/\tilde{a}^2 & 1 & 0 \\ 1/2\tilde{a}^2 & 1 & -\tilde{\tilde{\rho}}/2\tilde{a}^2 \end{bmatrix} \tag{6.163}$$

Now equation (6.160) may be expressed in terms of the jump in primitive variables as

$$\left[\mathbf{B}^{+}\mathbf{D}\Delta\mathbf{V}-\mathbf{Z}^{+}\Delta x\right]_{i-\frac{1}{2}}+\left[\mathbf{B}^{-}\mathbf{D}\Delta\mathbf{V}-\mathbf{Z}^{-}\Delta x\right]_{i+\frac{1}{2}}=0 \tag{6.164}$$

Since the baseline scheme evaluates jumps in the conserved variables as

$$\Delta(\bullet)\equiv(\bullet)_R-(\bullet)_L$$

the equivalent jump in primitive variables should be treated similarly. Therefore assuming constant mesh spacing Δx, equation (6.164) may be expanded into

$$\left[\mathbf{B}^{+}_{i-\frac{1}{2}}\mathbf{D}_{i-\frac{1}{2}}(\mathbf{V}_s-\mathbf{V}_{s-1})+\mathbf{B}^{-}_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}}(\mathbf{V}_{s+1}-\mathbf{V}_s)\right]-\Delta x(\mathbf{Z}^{-}_{i+\frac{1}{2}}-\mathbf{Z}^{+}_{i-\frac{1}{2}})=0 \tag{6.165}$$

In more concise notation, this equation may be re-written as

$$\mathbf{P}\mathbf{V}_s=\mathbf{S} \tag{6.166}$$

in which

$$\mathbf{P}=\mathbf{B}^{+}_{i-\frac{1}{2}}\mathbf{D}_{i-\frac{1}{2}}-\mathbf{B}^{-}_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}} \tag{6.167}$$

$$\mathbf{S}=\left(\mathbf{B}^{+}_{i-\frac{1}{2}}\mathbf{D}_{i-\frac{1}{2}}\mathbf{V}_{s-1}-\mathbf{B}^{-}_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}}\mathbf{V}_{s+1}\right)+\Delta x(\mathbf{Z}^{-}_{i+\frac{1}{2}}-\mathbf{Z}^{+}_{i-\frac{1}{2}}) \tag{6.168}$$

Provided the matrix $\mathbf{P}$ is not singular, i.e.

$$\det\left(\mathbf{B}^{+}_{i-\frac{1}{2}}\mathbf{D}_{i-\frac{1}{2}}-\mathbf{B}^{-}_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}}\right)\neq 0 \tag{6.169}$$

equation (6.166), will have a solution. It may be re-written in the functional form

$$\mathbf{P}\mathbf{V}_s-\mathbf{S}=0 \tag{6.170}$$

and solved for $\mathbf{V}_s$ to produce an accelerated update for the shock cell in terms of the primitive variables.

The Order of Evaluation

A final word about the implementation is necessary. The order in which the acceleration procedure is carried out affects the potential benefit to be gained from shock acceleration. This issue has already been discussed for the model problem in section 6.1.5. It will be recalled that in that case, the shock point update is carried out after non-shock points since the shock point benefits more from the most recent update of its neighbours. Conversely with system acceleration, the update of a shock point is best carried out *before* the non-shock points are updated. This is because each of the eigenvalues at a given point in the flow are related by the same flow velocity. On updating the shock point, the speed of the upstream running acoustic wave at the subsonic shock interface is modified. This imparts a disparity between the modified wave and remaining downstream running waves at the same interface. Carrying out the update of the non-shock points after the shock point has been updated gives the other waves at the downstream interface the benefit of more recent information from the shock cell and hence a better chance to catch up with the accelerated upsteam-running wave.

Equation (6.170) is solved iteratively. Both Newton iteration and GMRES have been used for this purpose. In practice GMRES is preferred as it proved to be more efficient in the test cases that were attempted. Crucial to the success of both iterative methods is the way in which the derivatives are approximated numerically. In the computations carried out, a forward-difference was the most reliable approximation with the size of the interval used for the difference quotient being close to machine precision.

Admissibility of an Accelerated Update

As was discovered in the scalar case for Roe's method, if shock acceleration is applied at some arbitrarily early time during the solution procedure, divergence can be expected. This could be due to various reasons. Firstly when the iterative procedure is applied to solving the accelerated update given by equation (6.170), an unphysical root may be found. This could be caused by poor numerical conditioning or possibly even the existence of a genuinely spurious solution to the discrete approximation. Secondly, the acceleration procedure may be applied before the solution has fully attained its basin of attraction. Finally, the shock may be near the point of migrating into a new cell, in which case the acceleration model is no longer valid. Two tests are used to screen a potential accelerated update in order to preclude the possibility of divergence.

The first is a simple test which checks the density and pressure of the accelerated update to ensure that neither are negative. The second test is equivalent to the entropy criterion used in the scalar case, derived from the definition of the shock cell. For flow from left to right it is expressed as

$$\lambda_1(\bar{\mathbf{Q}}_{i-\frac{1}{2}}) \geq 0 \geq \lambda_1(\bar{\mathbf{Q}}_{i+\frac{1}{2}}) \tag{6.171}$$

in which the acoustic wave speed is $\lambda_1(\bar{\mathbf{Q}}_{i-\frac{1}{2}}) = \tilde{u} - \tilde{a}$. Using the definition for Roe-averaged quantities given by equations (2.40–2.42), the left-hand inequality may be expressed in terms of the shock cell counter as

$$(\omega_{s-\frac{1}{2}} u_s + u_{s-1})/(1+\omega_{s-\frac{1}{2}}) - \tilde{a}_{s-\frac{1}{2}} \geq 0 \tag{6.172}$$

for which

$$\omega_{s-\frac{1}{2}} = (\bar{\rho}_s/\bar{\rho}_{s-1})^{\frac{1}{2}} \tag{6.173}$$

with $\tilde{a}$ given by equation (2.39) in terms of Roe averages. The inequality may be re-arranged in terms of the velocity in the shock cell u_s to give the upper bound

$$u_s \geq \left[\tilde{a}_{s-\frac{1}{2}}(1+\omega_{s-\frac{1}{2}}) - u_{s-1}\right] \Big/ \omega_{s-\frac{1}{2}} \tag{6.174}$$

With similar manipulations for the right-hand inequality, the lower bound for u_s is found to be

$$u_s \leq \left[\tilde{a}_{s+\frac{1}{2}}(1+\omega_{s+\frac{1}{2}}) - \omega_{s+\frac{1}{2}} u_{s+1}\right] \tag{6.175}$$

in which

$$\omega_{s+\frac{1}{2}} = (\bar{\rho}_{s+1}/\bar{\rho}_s)^{\frac{1}{2}} \tag{6.176}$$

Re-combining these inequalities produces the final expression used to determine admissibility of an accelerated update, i.e.

$$\left[\tilde{a}_{s+\frac{1}{2}}(1+\omega_{s+\frac{1}{2}}) - \omega_{s+\frac{1}{2}} u_{s+1}\right] \geq u_s \geq \left[\tilde{a}_{s-\frac{1}{2}}(1+\omega_{s-\frac{1}{2}}) - u_{s-1}\right] \Big/ \omega_{s-\frac{1}{2}} \tag{6.177}$$

The admissibility tests are carried out as follows. Provided the trial accelerated update passes the first test (i.e. the density and pressure are not negative), the entropy test is carried out. To do this, each of the quantities in the above expression are computed using the accelerated trial values. Then the trial velocity in the shock cell u_s is checked against equation (6.177) to establish whether or not it is admissable. If it is, the accelerated values (primitive variables) are used to form the conserved variables and the accelerated update in the shock cell is carried out. If it is out of bounds, the trial update is rejected and the standard update from the baseline scheme is used instead.

Shock Transfer

The single-point algebraic shock acceleration model is only valid if the update that it produces maintains the shock in the same cell in which it was prior to the accelerated update. This is because it is based on a prescribed wave pattern, that indicated in Figure 6.13. If an update fails the admissibility test of equation (6.177) a third test may be carried out to establish whether a failure of the entropy criteria is indicative of the shock attempting to move into the downstream cell. This is done using the trial variables in the shock cell to compute the velocity of the $\tilde{u}-\tilde{a}$ wave at the downstream interface and checking its sign. If the sign changes to being positive, this indicates that the trial update is forcing the shock into the downstream cell.

Whether an accelerated update is inadmissable due to one reason or another is not particularly an issue with the one-point shock acceleration model. However, with a two-point model it does become an issue as it is then necessary to determine whether an update has failed due to shock transfer. This has already been discussed at some length with reference to the two-point scalar acceleration models for Roe's scheme. For present purposes, however, this study does not consider two-point models for systems of equations.

6.2.5 Performance of the accelerated Roe scheme

Computational results for transonic flow through a convergent-divergent nozzle are now presented to compare the accuracy, efficiency and robustness of the standard Roe method with that of its shock accelerated counterpart. The results for the accelerated and baseline schemes are obtained from the same computer code using local time stepping. To initiate an accelerated solution, the user need only specify the iteration number at which the procedure is to be initiated. Naturally, the soonest that acceleration can commence is on formation of a shock wave. This does not need to be known in advance as the acceleration procedure can only operate once a shock has been detected.

The present test case is one that was used by Steger [146]. The nozzle has an area ratio

$$S(x) = 1 - \frac{S_t}{S_e} x\,(1-x), \qquad 0 \le x \le 1 \tag{6.178}$$

where S_t/S_e is the ratio of the throat area S_t to the exit area S_e. In the present case $S_t/S_e = 0.8$. Being symmetrical about the throat, the entrance area ratio at $x = 0$ is equal to the exit area ratio at $x = 1$. Outside the interval $(0,1)$ the area ratio is constant at $S(x) = 1$. This has a bearing on the implementation of

the boundary condition procedures in the fictitious cells since the state vector $\bar{Q}$ contains areas. The form that the boundary conditions assume have already been described in section 6.2.3. Flowfield initializations are based on stagnation conditions. Computations were carried out using the baseline scheme without the entropy fix; with the entropy fix applied to convective terms; and, with the full entropy fix applied to both convective and source terms. Irrespective of whether an entropy fix was used or its type, a small but discernible two-cell plateau developed at the sonic point during the flow evolution. Without the entropy fix, the plateau remained in the steady solution. With the entropy fix applied to only convective terms, the plateau was smoothed quite well, particularly in the steady-state. However, with the full entropy fix, the plateau was completely removed at steady-state. Its presence was also found to improve convergence significantly. All the results presented are based on use of the full entropy fix.

For the throat to reach sonic conditions the incoming Mach number must attain a value of $M_\infty = 0.55332$. The position and strength of the shock is characterized by the exit conditions. Numerical solutions obtained from the baseline scheme contained single-point shocks that were indistinguishable from the analytically derived solution on a grid containing 80 cells.

To assess the acceleration method, initially a simple test was conducted to establish whether the acceleration procedure was consistent with the baseline scheme. This involved applying the acceleration procedure to several converged solutions obtained from the baseline scheme in which the l_2 norm had dropped by between 6-12 orders of magnitude. From these tests it was found that the acceleration procedure converged the solution to machine zero while maintaining the full accuracy of the non-accelerated solutions. A series of further tests were then carried out for different flow conditions and nozzle geometries in order to test the new acceleration procedure more rigorously. Some of the results obtained for the nozzle geometry of equation (6.178) are now presented. For each test, a set of plots is provided of the baseline solution in terms of the Mach number distribution along the length of the nozzle l, together with the solution obtained from the accelerated scheme. The convergence histories of the baseline and the accelerated schemes are shown beneath each Mach plot.

Figure 6.14 shows the first result for an exit Mach number $M_\infty = 0.57$ on a grid containing 40 cells. For this test case the baseline method develops a shock in cell 26 after about 600 iterations. Convergence to machine zero takes several thousand iterations of the baseline scheme, with the shock migrating downstream five cells before settling. In this test, acceleration was applied from the onset of shock formation. From the Mach number plot it is apparent that the acceleration procedure has inhibited the shock from migrating downstream from its formation cell. The solution is clearly unacceptable. Ironically, the convergence history of the accelerated solution, given in terms of $\log ||d\rho/dt||_2$, reflects a speed up. The second test shown in Figure 6.15 is conducted using the same flow conditions as the first test. The acceleration procedure is applied when the shock is one cell away from its terminal sell at iteration 1000. Again, the acceleration procedure prevents the shock from attaining its terminal cell, and so degrades the accuracy of the solution. Figure 6.16 shows the result when acceleration is initiated soon after the shock has reached its terminal cell. There is a small but noticeable difference between the accelerated solution and the baseline solution which is confined to the shock cell. Improvement in the rate of convergence is

significant. When acceleration is applied after a four orders of magnitude drop in the residual when the shock has attained its terminal cell and had a chance to settle, there is virtually no loss in accuracy, yet still a substantial improvement in the rate of convergence. Such a result is shown in Figure 6.17, in which the baseline and accelerated solutions are indistinguishable. In this particular case, the acceleration procedure was commenced at iteration 2000. On this grid, the acceleration procedure costs approximately 45% more in operation count than the standard scheme. Weighing this against the faster rate of convergence, there is an the overall savings in computational expense of almost 60% over the non-accelerated scheme, assuming convergence to machine zero is necessary. At this point, it is important to note that on finer grids, the overhead of the increased operation count of the acceleration procedure decreases with the increasing ratio of non-shock points to shock points.

Solutions from all the tests attempted, exhibit similar behaviour to the three tests already discussed: when acceleration is applied before the shock has reached its terminal cell, the single-point acceleration method prevents further shock motion. However, when it is applied once the shock has attained its final cell and settled down, there is little or no loss in accuracy but a substantial speed up in the rate of convergence. In view of this, the remaining test cases for the acceleration procedure deal exclusively with results for which acceleration is applied after the shock has settled in its terminal cell.

In the next test case the nozzle that has been discretized with 100 cells, but the Mach number is the same as the previous case, namely $M_\infty = 0.57$. Here the acceleration procedure is commenced at iteration 5500 and the results are given in Figure 6.18. Only the accelerated shock cell differs from the baseline solution exhibiting a small error. However there is an appreciable improvement in convergence. On this grid, the overhead of shock acceleration is approximately 20%. Off-setting this figure against the faster rate of convergence indicates a saving in computational expense of just over 50%.

Finally in Figure 6.19 the results of a computation are given using the same grid as the previous test case but with an increased exit Mach number, $M_\infty = 0.65$. Here acceleration is commenced at iteration 3000 and full convergence is reached after a further 620 iterations. The Mach number distribution of the baseline solution and the accelerated solution are indistinguishable to plotting accuracy. Compared to the baseline solution, the accelerated solution represents an improvement of 62% in the rate of convergence. Accounting for the increased operation count, the accelerated solution effects a savings still in excess of 60%.

6.3 Closing Discussion

Substantial success has been obtained using shock acceleration for two upwind methods applied to the model scalar equation. With Roe's scheme the acceleration procedures can only be applied once the solution has attained its basin of attraction. Using the two-point acceleration models, acceleration can be maintained while the shock is transferring from one cell to another. With the Engquist-Osher scheme, the admissibility criteria for screening accelerated updates are more stringent and therefore permit automatic commencement of the acceleration procedure from the earliest time possible. The efficacy of the shock ac-

celeration method has been shown to improve the rate of convergence of the upwind methods considered by an order of magnitude in terms of the number of iterations needed for convergence, with only small computational overhead.

In the case of systems of equations, it is apparent that the single-point algebraic model is inappropriate when applied to the solution of a rapidly evolving flow in which shock transfer is taking place. Although the prescribed wave pattern of this particular acceleration model is valid when it is applied –according to the admissibility criteria– its repeated application at times when there would otherwise be significant shock motion, introduces error. With natural shock transfer, as a shock moves downstream from one cell to the next, the sign of the left running acoustic wave at the downstream cell interface must change in order for it to become a right running wave. Without this happening, the shock cannot migrate into a new cell. In the prescription of its wave pattern, the single-point acceleration model prevents the natural wave pattern corresponding to shock transfer from developing. Thus when the acceleration model is applied before the shock reaches its final cell, the errors that it introduces violate the important property of conservation of the direction of wave propagation. In other words, the prescribed wave pattern of the single-point model becomes too restrictive. In contrast, when the same acceleration model is applied to the scalar equation using similar admissibility criteria, shock motion is not inhibited –the admissibility criteria rescue the situation whenever shock transfer is due. This is accomplished either by enforcing an update from the baseline scheme or one from a two-point model. Clearly there are particular difficulties unique to hyperbolic systems of equations which are not present in their scalar counterparts.

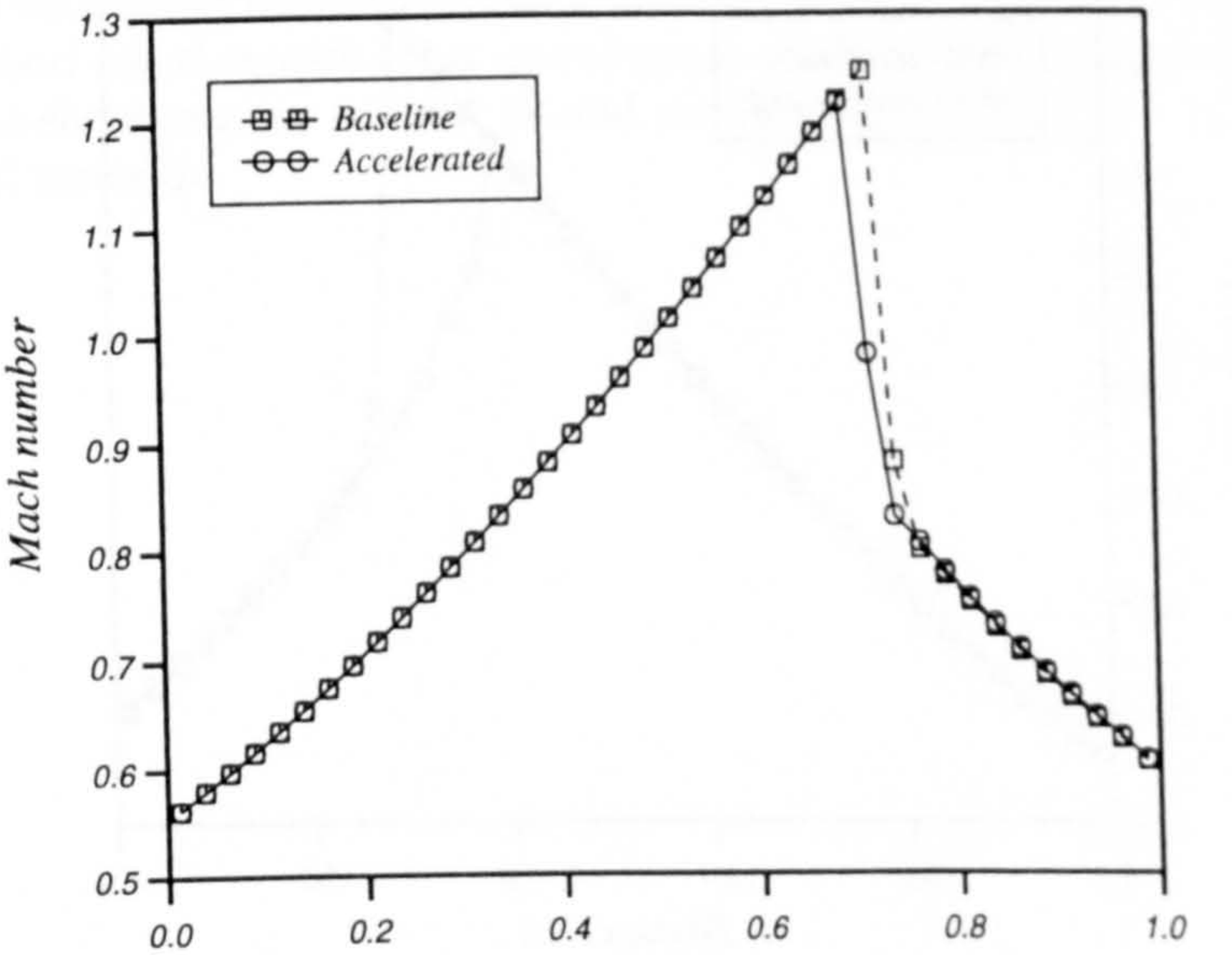
However, if acceleration is applied to the system of equations once the shock has settled in its terminal cell, the prescribed wave pattern of the single-point model is valid since there is no change in sign of the $\tilde{u} - \tilde{a}$ wave at the downstream interface. Consequently, if there is any loss in accuracy, it is only slight. Furthermore, the resulting improvement in the rate of convergence is entirely at the expense of high-frequency error.

The admissibility criteria for systems of equations prevented divergent solutions in all the tests attempted, irrespective of when acceleration was commenced. This contrasts to the scalar case using Roe's scheme. There the solution can be de-railed in spite of the admissibility criteria, if acceleration is applied at very early times during convergence.

The substantial success of shock acceleration in the model hyperbolic problem using both Roe's method and the Engquist-Osher method and, its limited success for the one-dimensional Euler equations with Roe's scheme suggest that further development of the shock acceleration method for systems of equations should be pursued. In that event, more sophisticated shock acceleration models based on two-point models are required. Such models would permit the flexibility needed for the selection of accelerated wave patterns in instances when shock transfer is taking place. As this objective has already been achieved successfully for the model equation in a single variable, much of the ground work is already in place.

The significance of the shock acceleration method lies in its ability to target the high-frequency errors arising from the singularity in discrete shock structure. The importance of developing approaches to convergence acceleration which deal with the most difficult high-frequency errors cannot be understated. Several of the acceleration techniques that are reviewed in chapter 4 tackle the error modes

that exist at lower frequencies, while high-frequency damping is only dealt with by optimal smoothing schemes and to some extent by local preconditioning. None of those methods deal specifically with the $M \rightarrow 1$ singularity. If the remaining difficulties with shock acceleration for systems of equations can be overcome, use of the method together with other convergence acceleration methods which work well in non-shock regions of flow, should produce significant gains in efficiency for upwind methods.



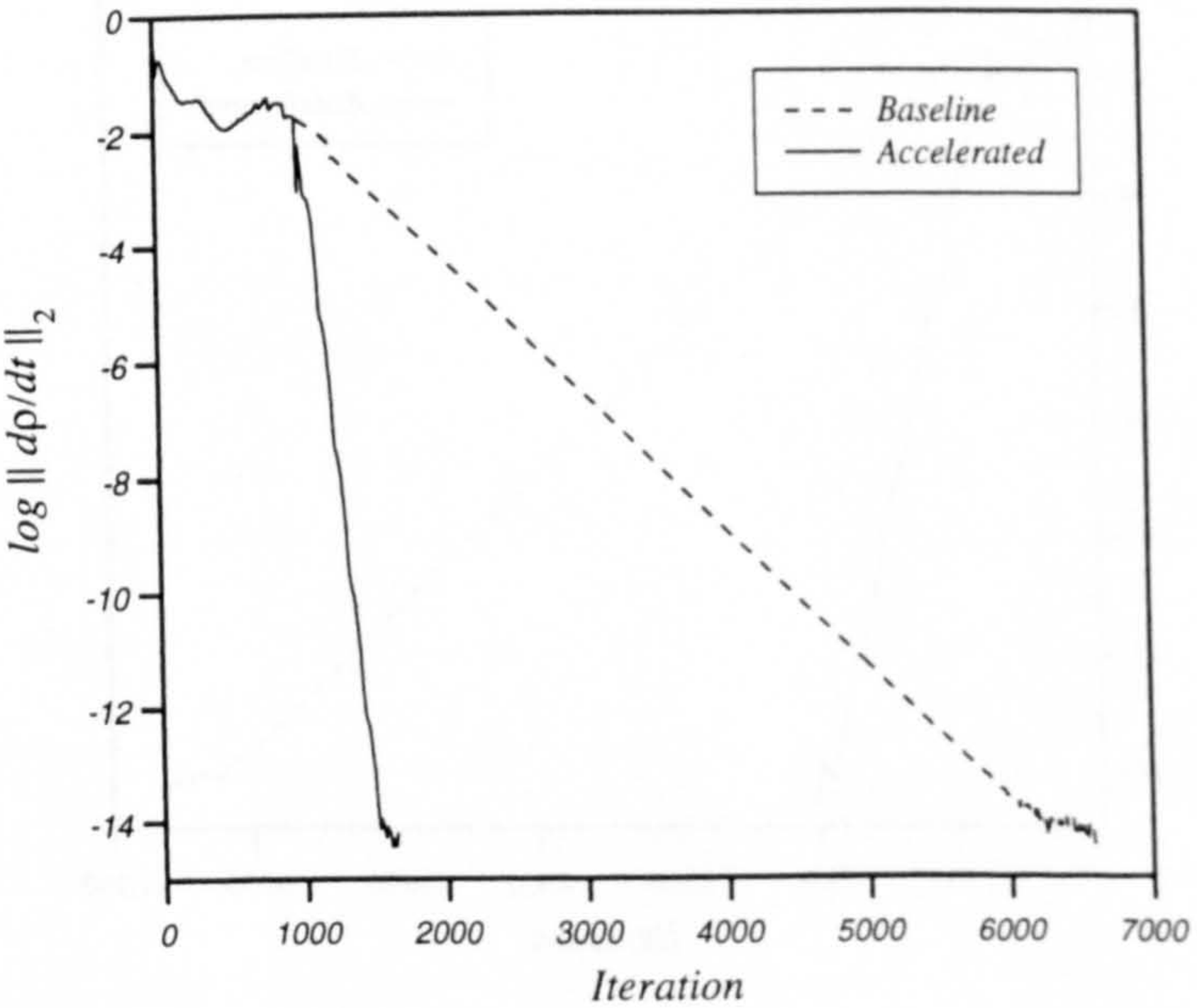
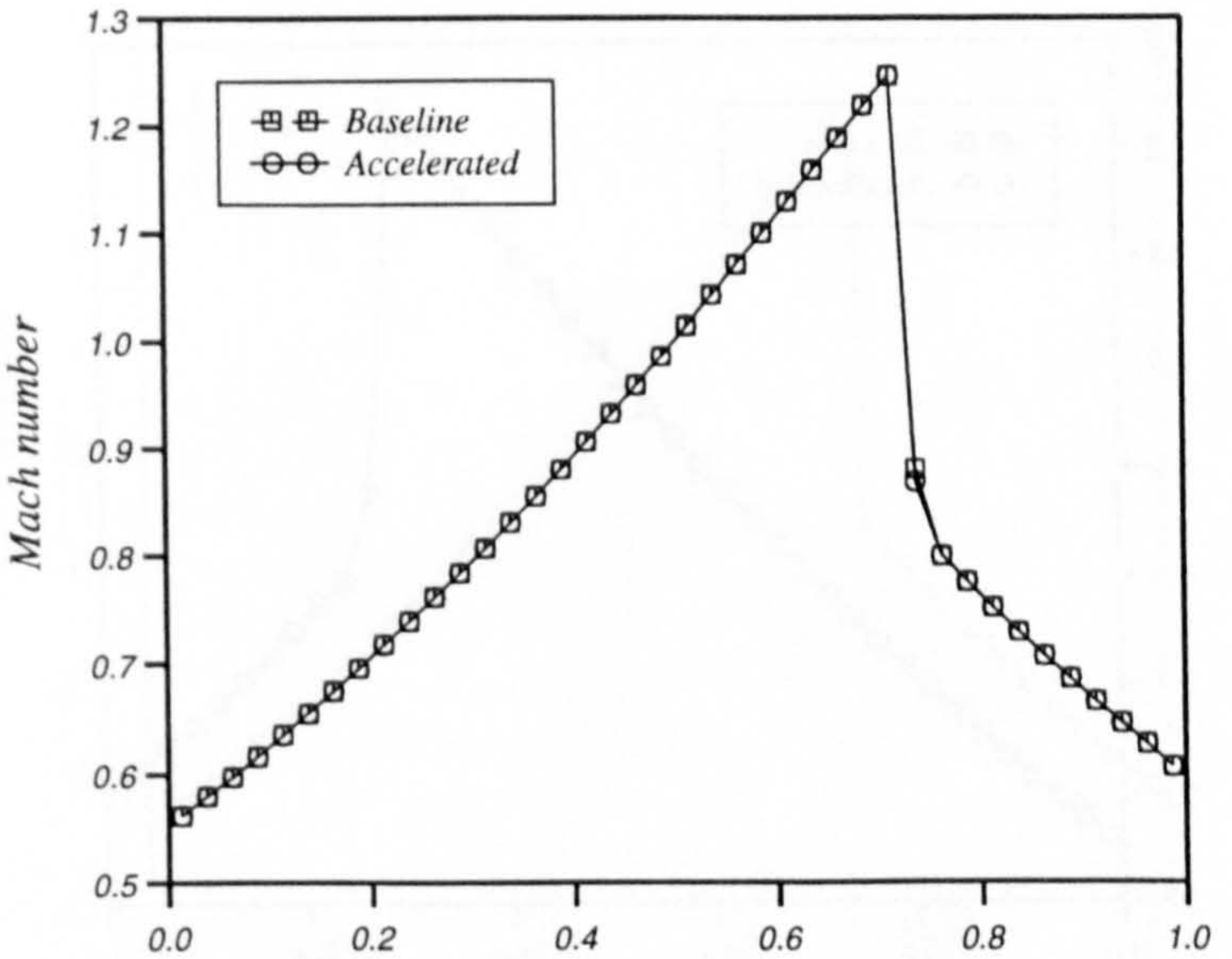


FIGURE 6.15. Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 1000. Grid size=40.



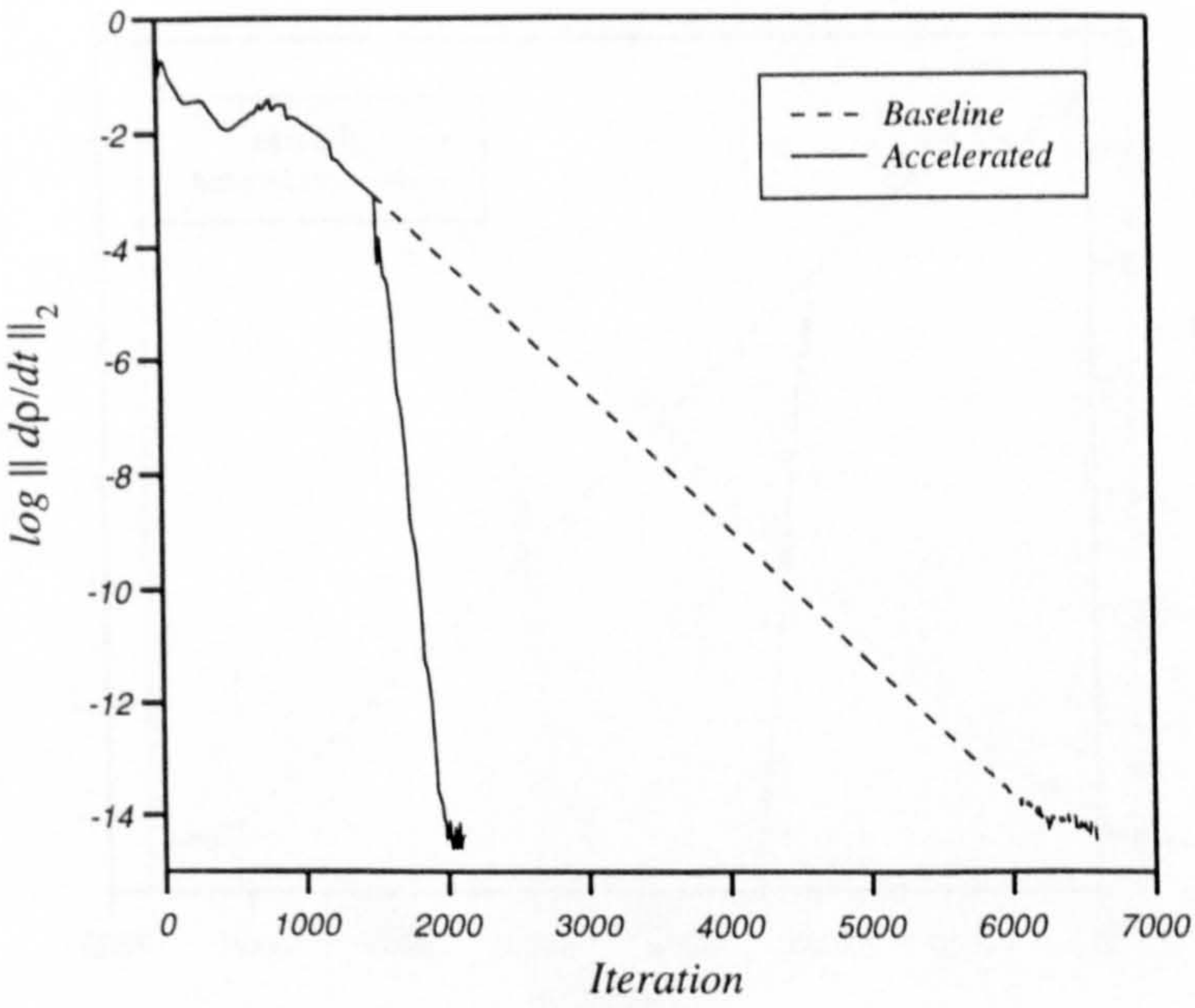
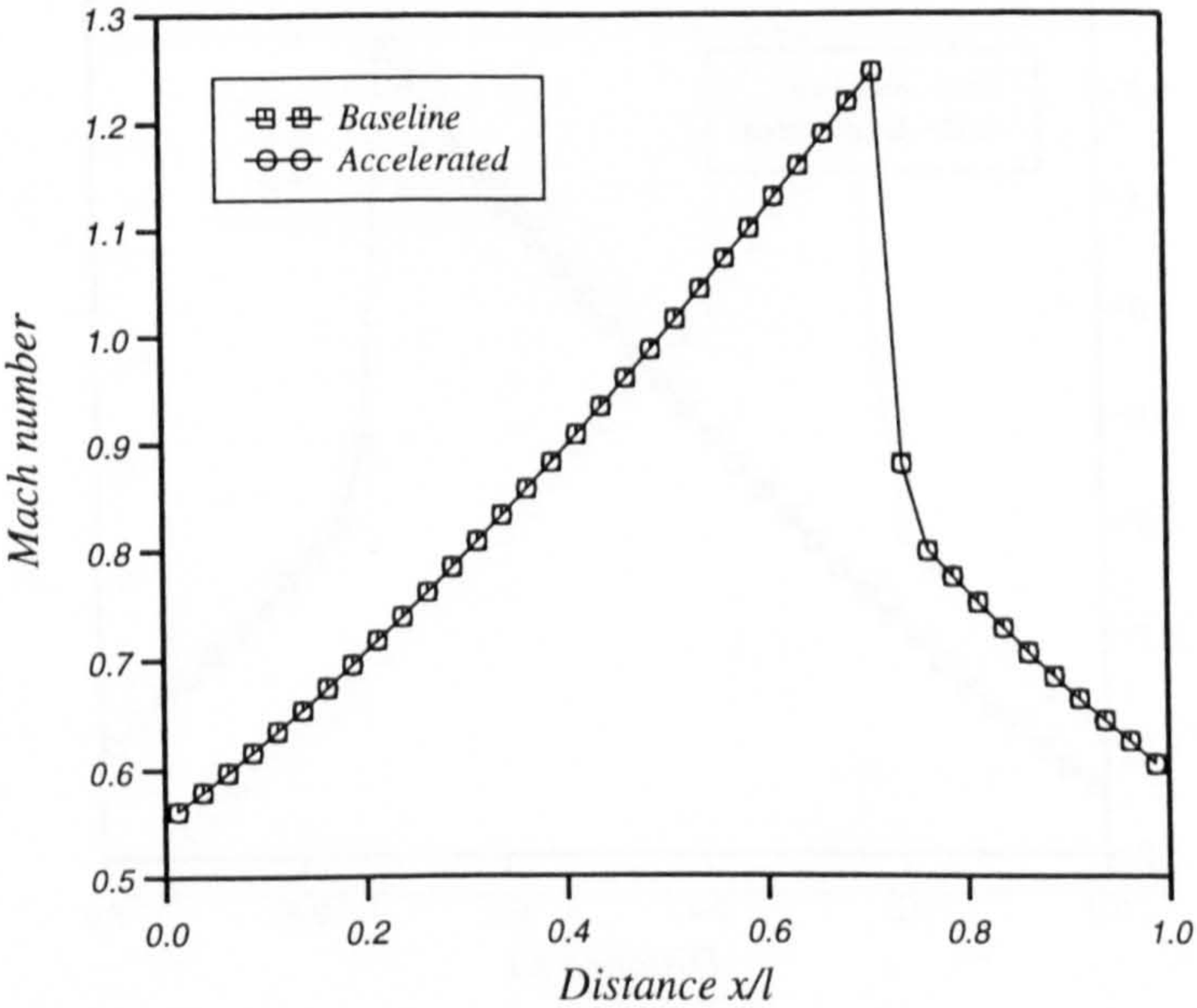


FIGURE 6.16. Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 1500. Grid size=40.



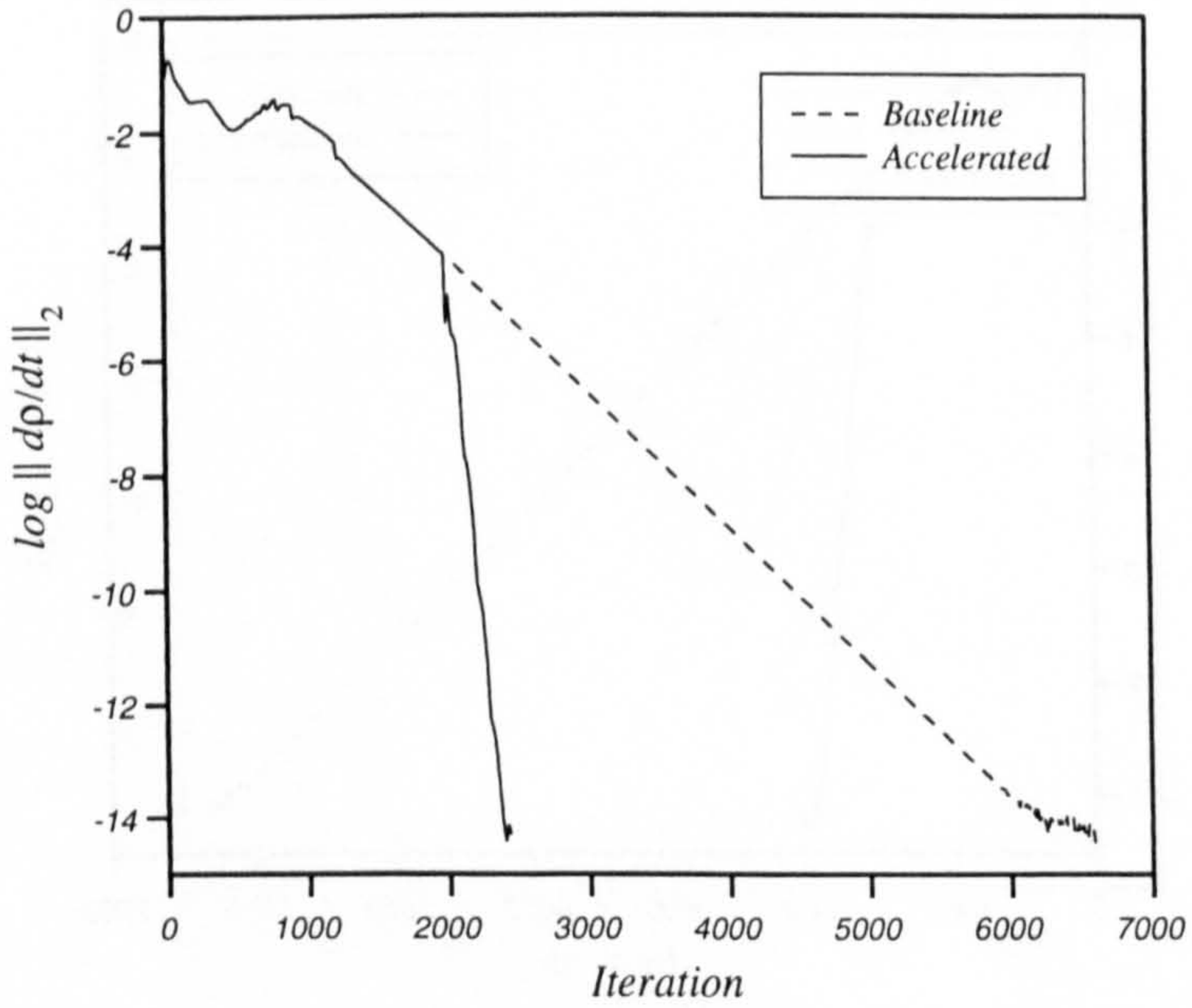
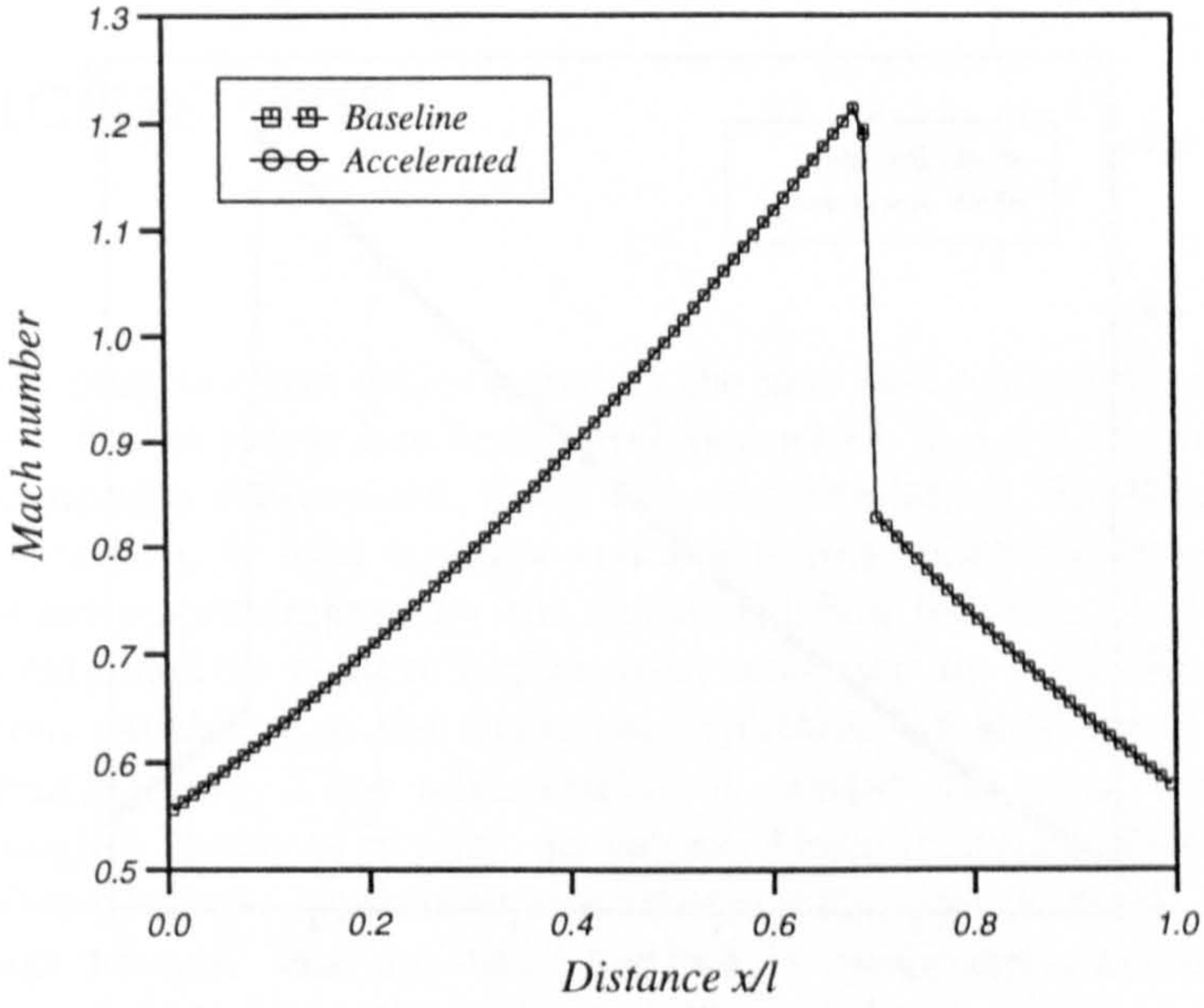


FIGURE 6.17. Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 2000. Grid size=40.



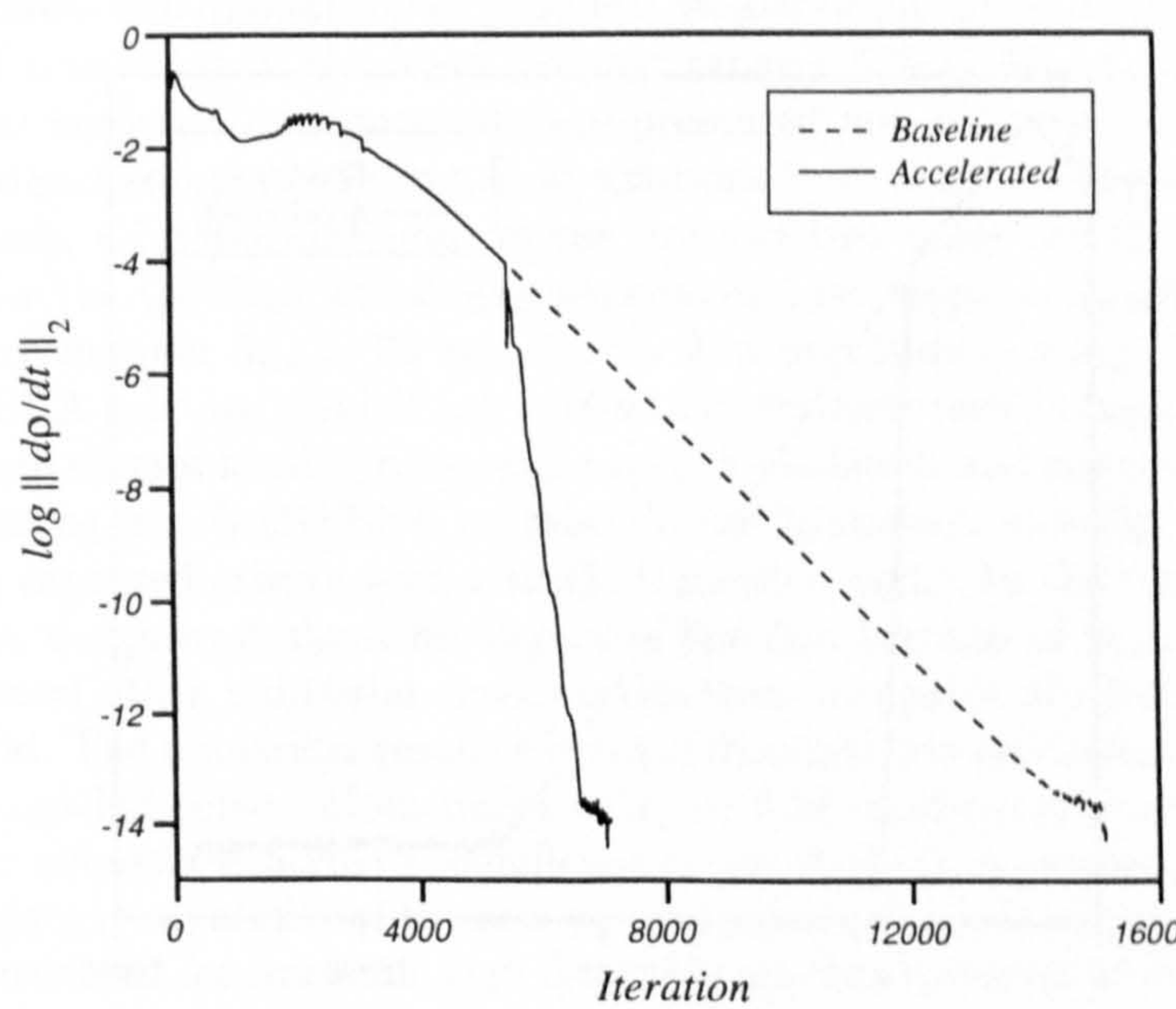
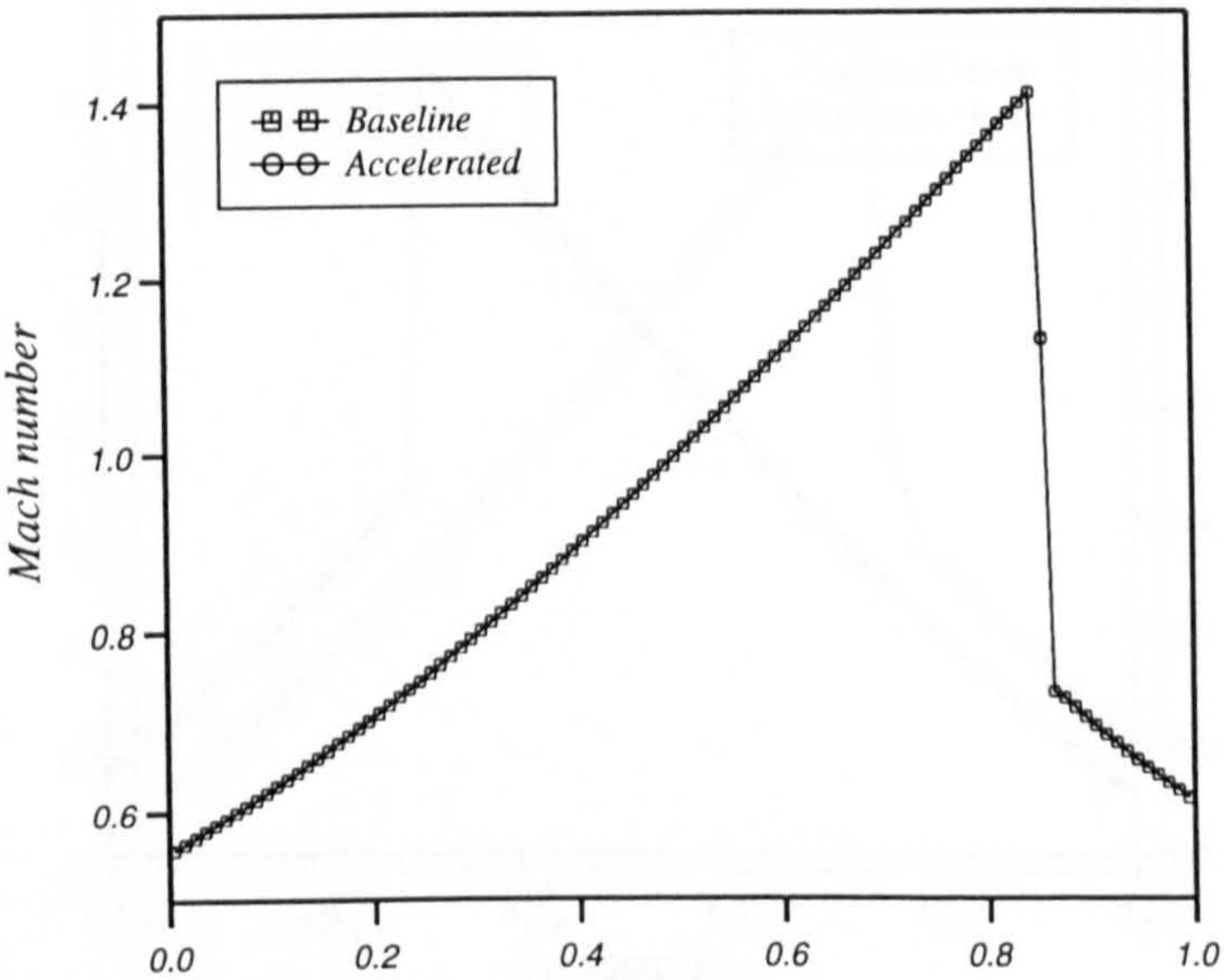


FIGURE 6.18. Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.57$. Accelerated solution commenced at iteration 5500. Grid size=100.



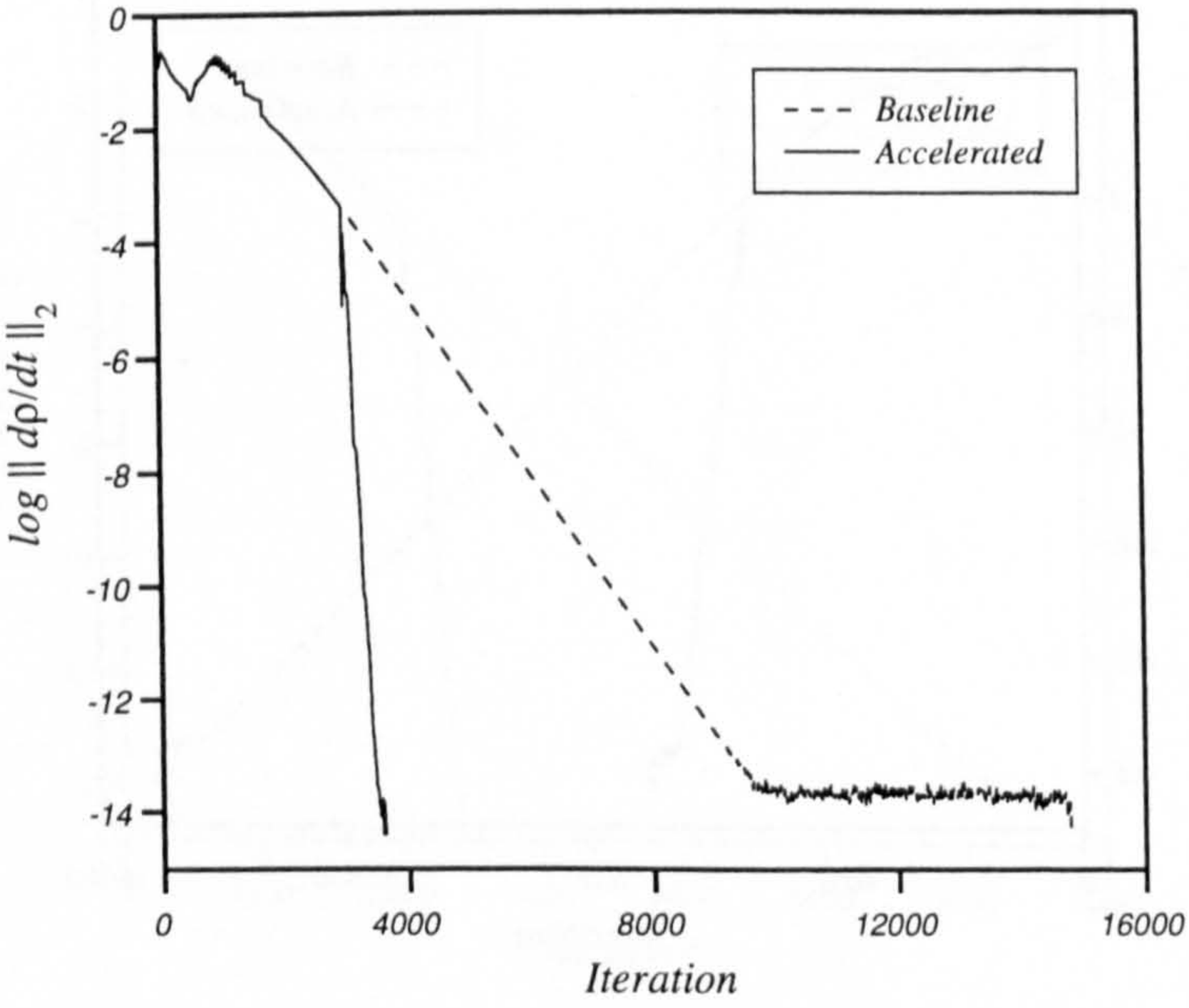


FIGURE 6.19. Comparison of Euler solutions and convergence histories of Roe's scheme for transonic nozzle flow at $M_\infty = 0.65$. Accelerated solution commenced at iteration 3000. Grid size=100.

7

Conclusions

This work comprises two major parts. In the first part a three-dimensional Euler and Navier-Stokes solver has been developed which is a structured grid upwind algorithm using a cell-centred, finite volume formulation. Based on the method of lines approach, it uses separate space and time discretizations. The spatial gradients are approximated by the numerical flux function of Roe. The algorithm is extended to achieve higher-order accuracy by algebraic flux limiting. The viscous capability of the algorithm facilitates the solution of both laminar and turbulent flows. Time advancement is carried out using local time stepping through a choice of explicit operators. These include a classical two-stage Runge-Kutta scheme, optimally smoothed multi-stage schemes and, a hybrid multi-stage scheme. The multigrid method is used together with the optimal smoothing schemes to accelerate convergence to the steady-state.

The three-dimensional solver has been validated for inviscid flows over a wide range of conditions in the Mach number range $0.675 \leq M_\infty \leq 25$. The results of several numerical computations are presented and discussed and in some instances direct comparison is made to solutions from other CFD solvers using the same mesh. Of particular note in the inviscid test cases are the solutions obtained for the ONERA blunt-nose supersonic case, hypersonic blunt-body flow at a Mach number $M_\infty = 25$ and, vortex flow over a delta wing. The results for the ONERA case are equal if not better than the best result obtained by participants from several leading European research institutes and aerospace companies participating in a GARTEUR exercise. In the hypersonic case the detached bow shock is captured crisply with a single transition point. In the vortex flow computation, the present algorithm captured fine flow features of the wing tip region that several other industrial design codes were incapable of resolving using the same grid. The numerical results obtained demonstrate the high accuracy of the present upwind solver, often under extreme flow conditions. Furthermore, they prove its robustness across a significant range of Mach numbers.

The viscous capability of the new upwind solver using convergence acceleration is demonstrated for transonic high Reynolds number turbulent flow about a wing. A grid convergence study has been carried out in which the solutions from the present upwind solver are compared to ones from an established industrial design code based on a central difference algorithm that uses scalar dissipation. For the upwind scheme, the variation of lift and drag with mesh refinement is found to be linear across the range of mesh considered, indicating full second-order accuracy. The central scheme does not possess such consistency. The upwind scheme also demonstrates faster grid convergence than the central scheme. In fact boundary layer profiles computed by the central scheme thicken substantially as mesh size is decreased. The study has found that the levels of dissipation in the central scheme are between one and two orders of magnitude greater than

the upwind scheme for realistic three-dimensional flow. This finding suggests that the central boundary layer profiles are severely contaminated by excessive numerical dissipation. It corroborates and extends the results of earlier studies. The results also indicate that accurate high Reynolds number turbulent flow solutions may be obtained from the upwind scheme on grids of approximately one-quarter the density of the central scheme. Consequently, by careful grid selection, the upwind solver can achieve superior accuracy to the central scheme at slightly less expense. Comparison between the standard upwind scheme and its fully accelerated counterpart indicate that when the solver is accelerated using an optimal smoothing scheme in conjunction with multigrid acceleration, it is over 20 times more efficient than the former for transonic flow computations. Thus the efficacy of the accelerated upwind scheme is demonstrated and its advantages over central schemes based on scalar dissipation models are quantified for transonic viscous flow.

In the second part of the study an original method for accelerating nonlinear inhomogeneous hyperbolic equations in one variable, is proposed. It specifically addresses the singularity that exists in shock waves. A model hyperbolic equation is examined in detail. Numerical methods based on the Roe scheme and the Engquist-Osher scheme are derived for the model problem. Several acceleration models are formulated, based on a localized treatment of the discrete shock structure using the principles of characteristic theory. The name *shock acceleration* has been given to this new approach to convergence acceleration, as the method accelerates the motion of the shock wave to its final location. For the Roe scheme, an acceleration model is formulated based on accelerating the update of a single transition point in the shock. Three other acceleration models based on updates of the shock point and its downstream neighbour are also developed for the Roe scheme. These two-point models are capable of maintaining acceleration when a discontinuity is moving between cells. An acceleration model is also developed for the Engquist-Osher scheme based on a two-point update. To prevent departure solutions occurring in instances when the acceleration is applied before the basin of attraction of the solution of the governing PDE has been attained and, to ensure that the acceleration procedures are automated, admissibility criteria have been developed for screening accelerated updates. With the Engquist-Osher scheme, they allow acceleration to be commenced automatically at the earliest possible time, whereas with the Roe scheme a single free-parameter is required to determine when acceleration is to be commenced. The numerical results for both the Roe and Engquist-Osher schemes demonstrate an order of magnitude improvement in the rate of convergence when measured by the number of iterations required to achieve convergence. The increased operation count of the acceleration procedures is relatively small and decreases as the number of non-shock points in a problem increases.

The Roe scheme has been applied to the quasi one-dimensional Euler equations for duct flow using source-term splitting. A single-point shock acceleration method has been formulated for Roe's method to accelerate the convergence of this system of equations. The solutions obtained demonstrate that when acceleration is applied before the shock has reached its terminal cell, the single-point acceleration procedure prevents further shock transfer. However, when it is applied once the shock has settled in its final cell, there is little or no loss in accuracy but a substantial speed up in the rate of convergence. For the quasi

one-dimensional Euler equations it is shown that the shock acceleration method affects a savings in computational expense of between 50%–60%. Considering such savings in isolation may not appear to be particularly significant. However, it is stressed that these savings are achieved entirely at the expense of some of the most stubborn high-frequency error modes. Furthermore, as these error modes are associated with the $M \rightarrow 1$ singularity that exists in discrete shock structure, they cannot be damped by other known convergence acceleration methods.

The substantial success of the shock acceleration method for two upwind schemes in the model hyperbolic problem and, its partial success for Roe's scheme with the one-dimensional inhomogeneous Euler equations make a compelling case for the continued development of the method for systems of equations. However, more sophisticated shock acceleration models based on two-point models are needed for situations in which there is significant shock movement. As this has been accomplished successfully for the model equation in a single variable, much of the ground work is already in place. In addition, the method would also need to be extended to higher dimensions for use in problems of engineering interest. Finally, if this approach to convergence acceleration can be developed further, its use in conjunction with other acceleration techniques, noted for their performance in expelling other types of error transients, is likely to produce significant improvements in the efficiency of upwind algorithms.

References

[1] S.R. Allmaras. Contamination of laminar boundary layers by artifical dissipation in Navier–Stokes solutions. Pre-print from ICFD Conference on Numerical Methods for Fluid Dynamics held in Reading on 7–10 April, 1992.

[2] J.D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill, 1984.

[3] W.K. Anderson, J.L. Thomas, and B. van Leer. Comparison of finite volume flux vector splittings for the Euler equations. *AIAA Journal*, 24(9):1453–1460, September 1986.

[4] A. Arnone, M.-S. Liou, and L.A. Povinelli. Multigrid calculations of three-dimensional viscous cascade flows. *Journal of Propulsion and Power*, 9(4), July–August 1993.

[5] B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flows. *AIAA Paper 78-0257*, 1978.

[6] J. Blazek, N. Kroll, R. Radespiel, and C-C. Rossow. Upwind implicit residual smoothing method for multi-stage schemes. *AIAA Paper 91-1533-CP*, pages 70–80, 1991.

[7] J. Blazek, N. Kroll, and C-C. Rossow. A comparison of several implicit smoothing methods. *Numerical Methods for Fluid Dynamics*, 4:451–460, 1993.

[8] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In H. Cabannes and R. Temam, editors, *Proceedings of Third Int. Conf. on Numerical Methods in Fluid Mechanics*, volume 1, pages 82–89. Springer-Verlag, 1973.

[9] A. Brandt. Multi-level adaptive solutions to boundary value problems. volume 31, pages 333–390, 1977.

[10] A. Brandt. Multilevel adaptive computations in fluid dynamics. *AIAA Journal*, 18:1165–1172, 1980.

[11] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods (Lecture Notes in Mathematics 960)*, pages 220–312. Springer-Verlag, 1982.

[12] C. Brezinski. Convergence acceleration methods: The past decade. *Journal of Computational and Applied Mathematics*, 12&13:19–36, 1985.

[13] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19, 577–593 1965.

[14] C.G. Broyden. The convergence of an algorithm for solving sparse nonlinear systems. *Math. Comp.*, 25(114), April 1971.

[15] J.C. Butcher. *The Numerical Analysis of Ordinary Differential Equations.* Wiley, New York, 1987.

[16] L.A. Catalano and H. Deconinck. Two dimensional optimization of smoothing properties of multistage schemes applied.to hyperbolic equations. Technical Note 173, von Karman Institute for Fluid Dynamics,Belgium, July 1990.

[17] T. Cebeci. Calculation of compressible turbulent boundary layers with heat and mass transfer. *AIAA Paper 78-741*, 1970.

[18] T. Cebeci and A.M.O. Smith. *Analysis of Turbulent Boundary Layers, Applied Mathematics and Mechanics*, volume 15. Academic Press, New York, 1974.

[19] M. Celestina and A. Sidi. Convergence acceleration for vector sequences and applications to computational fluid dynamics. *AIAA Paper 90-0338*, 1990.

[20] S.R. Chakravarthy and K-Y. Szema. Euler solver for three-dimensional supersonic flows with subsonic pockets. *Journal of Aircraft*, 24:73–83, February 1987.

[21] S.R. Chakravarthy, K-Y. Szema, U.C. Goldberg, J.J. Gorski, and S. Osher. Application of a new class of high accuracy TVD schemes to the Navier-Stokes equations. *AIAA Paper 85-0165*, 1985.

[22] R.V. Chima, E. Turkel, and S. Schaffer. Comparison of three explicit multigrid methods for the Euler and Navier-Stokes equations. *AIAA Paper 87-0538*, 1987.

[23] Y-H. Choi and C.L. Merkle. The application of preconditioning in viscous flows. *J. Comp. Phys.*, 105:207–223, 1993.

[24] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.*, 2:12–26, 1967.

[25] R. Courant and K.O. Friedrichs. *Supersonic Flow and Shock Waves.* Interscience Publishers, New York, 1948.

[26] A. Curtis, M.J.D. Powell, and J.K. Reid. On the estimation of sparse jacobian matrices. *Journal Inst. Maths. Applics.*, 13:117–120, 1974.

[27] A. Dadone and B. Grossman. Surface boundary conditions for the numerical solution of the euler equations. *AIAA Journal*, 32(2):285–293, February 1994.

[28] J.F. Dannenhoffer III and M.B. Giles. Convergence acceleration through the use of time inclining. *AIAA Paper 89-0096*, 1989.

[29] D. de Zeeuw and K.G. Powell. An adaptively-refined cartesian mesh solver for the Euler equations. *AIAA Paper 91-1542-CP*, pages 166–180, 1991.

[30] P.M. de Zeeuw. Developments of semi-coarsening and sparse-grid techniques. Department of Numerical Mathematics Report NM-R9504, CWI, March 1995.

[31] H. Deconinck, Ch. Hirsch, and J. Peutman. Characteristic decomposstion methods for the multidimensional Euler equations. *Lecture Notes in Physics*, 364:216–221, 1986.

[32] D. Degani and L.B. Schiff. Computation of turbulent supersonic flows around pointed bodies having cross-flow separation. *J. Comp. Phys.*, 66:173–196, 1986.

[33] F. Dubois and G. Mehlman. Nonparametized entropy fix for Roe's method. *AIAA Journal*, 31(1:Technical Note):199–200, 1991.

[34] J.R. Edwards and D.S. McRae. Nonlinear relaxation/quasi-Newton algorithm for the compressible Navier-Stokes equations. *AIAA Journal*, 31(1):57–60, January 1993.

[35] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Num. Anal.*, 25(2):294–318, 1988.

[36] B. Einfeldt, C.D. Munz, P.L. Roe, and B. Sjögreen. On Godunov-type methods near low densities. *J. Comp. Phys.*, 92:273–295, 1991.

[37] P. Eliasson and A. Rizzi. Hypersonic leeside flow computations using centred schemes for Euler equations. *Notes in Numerical Fluid Mechanics*, 29:109–118, 1990.

[38] A. Elsenaar and H.W.M. Hoeijmakers. An experimental study of the flow over a sharp-edged delta wing at subsonic and transonic speeds. In *AGARD conference proceedings 494: Vortex flow aerodynamics*, July 1991.

[39] P. Embid, J. Goodman, and A. Majda. Multiple steady states for 1-D transonic flow. Pre-print, Center for Pure and Applied Mathematics, University of California, Berkeley, February 1982.

[40] R. Enander. Improved implicit residual smoothing for steady state computations of first order hyperbolic systems. *J. Comp. Phys.*, 107:291–296, 1993.

[41] R. Enander and A.R. Karlsson. Implicit explicit residual smoothing in multigrid cycle. *AIAA Paper 95-0204*, 1995.

[42] B. Engquist and S. Osher. One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation*, 36:321–351, 1981.

[43] L.E. Eriksson and A. Rizzi. Computer-aided analysis of the convergence to steady state of discrete approximations to the Euler equations. *J. Comp. Phys.*, 57:90–128, 1985.

[44] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Comp. Math and Math. Phys.*, 4(3):227–235, 1964.

[45] S.A. Forth. Private communication, 1992.

[46] GARTEUR. Appendix 5 to GARTEUR AD(AG-15) report –Results presented at the third meeting held at Aerospatiale, Les Mureaux, France, 1991.

[47] M.B. Giles. Generalised conservation cells for finite volume calculations. *AIAA Paper 87-1118*, pages 187–197, 1987.

[48] M.B. Giles. Non-reflecting boundary conditions for Euler equation calculations. *AIAA Paper 89-1942-CP*, pages 143–153, 1989.

[49] P. Glaister. Flux difference splitting techniques for the Euler equations in non-cartesian geometry. Numerical Analysis Report 8/85, Department of Mathematics, University of Reading, 1985.

[50] J. Glimm, G. Marshall, and B. Plohr. A generalised Riemann problem for quasi-one-dimensional gas flows. *J. Comp. Phys.*, 5:1–30, 1984.

[51] A.G. Godfrey. Steps toward a robust preconditioning. *AIAA Paper 94-0520*, 1994.

[52] A.G. Godfrey, R.W. Walters, and B. van Leer. Preconditioning for the Navier-Stokes equations with finite-rate chemistry. *AIAA Paper 93-0535*, 1993.

[53] S.K. Godunov. Finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47(3):271–306, 1959.

[54] G. Golub and J.M. Ortega. *Scientific Computing. An Introduction with Parallel Computing.* Academic Press, 1993.

[55] J.B. Goodman and R.J. Leveque. A geometric approach to high resolution TVD schemes. *SIAM Journal on Numerical Analysis*, 25:268–284, 1988.

[56] J.J. Gottlieb, C.P.T. Groth, and D.S. Anderson. Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. In *Proceedings of Second International Meeting on Random Choice Methods in Gas Dynamics, Cranfield, England*, July 1987.

[57] J.J. Gottlieb and K.Y. Zhang. Dirichlet boundary conditions for the random-choice method for solving one-dimensional unsteady flows of a perfect gas. In *Proceedings of Second International Meeting on Random Choice Methods in Gas Dynamics, Cranfield, England*, July 1987.

[58] F. Grasso and M. Marini. Multigrid techniques for hypersonic viscous flows. *AIAA Paper 93-0771*, 1993.

[59] D.F. Griffiths, A.M. Stuart, and H.C. Yee. Numerical wave propagation in an advection equation with a nonlinear source term. *SIAM J. Num. Anal.*, 29(5):1244–1260, 1992.

[60] B. Gustafsson and P. Lötstedt. Convergence acceleration for first order systems. *Numerical Methods for Fluid Dynamics*, 4:191–206, 1993.

[61] W. Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20:291–306, 1978.

[62] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[63] M. Hafez, S. Palaniswamy, G. Kuruvila, and M.D. Salas. Application of Wynn's ε-algorithm to transonic flow calculations. *AIAA Paper 87-1143*, 1987.

[64] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357–393, 1983.

[65] W. Haase, F. Bradsma, E. Elsholz, M. Leschziner, and D. Schwamborn, editors. *EUROVAL –An European Initiative on Validation of CFD codes.* Vieweg, 1993.

[66] Ch. Hirsch. *Numerical Computation of Internal and External Flows. Volume 2: Computational Methods for Inviscid and Viscous flows.* John Wiley and Sons, 1990.

[67] M. Holt. *Numerical Methods in Fluid Dynamics.* Springer-Verlag, second edition edition, 1984.

[68] A. Jameson. The evolution of computational methods in aerodynamics. *Journal of Applied Mechanics*, 50:1052–1070, December 1983.

[69] A. Jameson. Transonic flow calculations. MAE Report 1751, Princeton University, Princeton, NJ, 1984.

[70] A. Jameson. Multigrid algorithms for compressible flow calculations. MAE Report 1743, Princeton University, Princeton, NJ, 1985.

[71] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1: Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. Preprint to appear in International Journal of Computational Fluid Dynamics, August 1994.

[72] A. Jameson and T.J. Baker. Solution of the Euler equations for complex configurations. *AIAA Paper 83-1929*, 1983.

[73] A. Jameson and T.J. Baker. Multigrid solution of the Euler equations for aircraft configurations. *AIAA Paper 84-0093*, 1984.

[74] A. Jameson and D.A. Caughey. A finite-volume method for transonic potential flow calculations. In *Proceedings of 3rd AIAA Computational Fluid Dynamics conference, Albuquerque, New Mexico*, pages 35–54, June 1977.

[75] A. Jameson, R.W. MacCormack, and W. Schmidt. Using Euler solvers. AIAA Professional Study Series Note book, January 1985. Seminar held in Stuttgart, Germany.

[76] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions for the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA Paper 81-1259*, 1981.

[77] U. Kaynak and J. Flores. Advances in the computation of transonic separated flow over finite wings. *AIAA Paper 87-1195*, 1987.

[78] B. Koren. Condition improvement for point relaxation in multigrid, subsonic Euler-flow computations. *Applied Numerical Mathematics*, 16:457–469, 1995.

[79] B. Koren and P.W. Hemker. Damped direction dependant multigrid for hypersonic flow computations. *Applied Numerical Mathematics*, 7:309–328, 1991.

[80] N. Kroll, D. Gaitonde, and M. Aftosmis. A systematic comparative study of several high resolution schemes for complex problems in high speed flows. *AIAA Paper 91-0636*, 1991.

[81] A. Lafon and H.C. Yee. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations, Part III: The effects of nonlinear source terms and boundary conditions in reaction-convection equations. Technical Report TM 103877, NASA, July 1991.

[82] M-H. Lallemand. Dissipative properties of Runge-Kutta schemes with Upwind spatial approximation for the Euler equations. Rapports de Recherche, No. 1173, INRIA, March 1990.

[83] P.D. Lax. Hyperbolic systems of conservation laws and the mathematical theory of shock waves. *SIAM Regional Conference Series in Applied Mathematics*, 11, 1972.

[84] P.D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure and Applied Math*, 13:217–237, 1960.

[85] D. Lee and B. van Leer. Progress in local preconditioning of the Euler and Navier-Stokes equations. *AIAA Paper 93-3328-CP*, pages 338–348, 1987.

[86] W-T. Lee. *Local preconditioning of the Euler equations.* PhD thesis, University of Michegan, 1992.

[87] A. Lerat. Une classe de schémas aux différences implicites pour les systèmes hyperboliques de lois de conservation. *Comptes Rendus Acad. Sciences, Paris*, 228 A(22):1033–1036, 1979.

[88] H.W. Liepmann and A. Roshko. *Elements of Gas Dynamics.* Wiley, New York, 1957.

[89] H-C. Lin. *Topics in Numerical Computation of Compressible Flow.* PhD thesis, College of Aeronautics, Cranfield Institute of Technology, 1990.

[90] H-C. Lin. Dissipation additions to flux-difference splitting. *J. Comp. Phys.*, 117:20–27, 1995.

[91] P. Lötstedt. Improved convergence to the steady state of the Euler equations by enhanced wave propagation. *J. Comp. Phys.*, 114:34–44, 1994.

[92] J.F. Lynn and B. van Leer. Multi-stage schemes for the Euler and Navier-Stokes equations with optimal smoothing. *AIAA Paper 93-3355-CP*, pages 642–656, 1993.

[93] R.W. MacCormack and A.J. Paullay. Computational efficiency achieved by time splitting of finite difference operators. *AIAA Paper 72-154*, 1972.

[94] A. Majda. *Compressible Fluid Flow and Systems of Conservation Laws in Several Space Dimensions. Applied Mathematical Sciences Series, Vol 53.* Springer-Verlag, 1984.

[95] M. Manna. Validation of central and upwind 3-D compressible flow solvers. In *Computational Fluid Dynamics Lecture Series 1992-04, von Karman Institute for Fluid Dynamics, Belgium*, March 1992.

[96] L. Martinelli. *Calculation of viscous flows with a multigrid method.* PhD thesis, MAE Department, Princeton University, 1987.

[97] L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. *AIAA Paper 88-0414*, 1988.

[98] D.J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the Navier-Stokes equations on mixed element meshes. NASA ICASE Report 95-53, July 1995.

[99] K. Mazaheri and P.L. Roe. New light on numerical boundary conditions. *AIAA Paper 91-1600-CP*, June 1991.

[100] P.W. McDonald. The computation of transonic flow through two-dimensional gas turbine cascades. ASME Paper 71-GT-89, 1971.

[101] C.Y. McNeil. The effect of numerical dissipation on high Reynolds number turbulent flow solutions. To appear at the 34th AIAA Aerospace Sciences Meeting. Reno, Nevada, January 1996.

[102] W.A. Mulder. Multigrid relaxation for the Euler equations. *J. Comp. Phys.*, 60:235–252, 1985.

[103] W.A. Mulder. A new multigrid approach to convection problems. *J. Comp. Phys.*, 83(2):303–323, August 1989.

[104] B. Müller. Simple improvements of an upwind TVD scheme for hypersonic flow. *AIAA Paper 89-1977-CP*, pages 455–463, 1989.

[105] E.M. Murman and J.D. Cole. Calculation of plane steady transonic flow. *AIAA Journal*, 9:114, 1971.

[106] R-H. Ni. A multiple grid scheme for solving the euler equations. *AIAA Journal*, 20(11):1565–1571, November 1982.

[107] S. Osher and F. Solomon. Upwind differencing schemes for hyperbolic systems of conservation laws. *Math. Comput.*, 38:339–374, 1982.

[108] H. Paillère, H. Deconinck, and P.L. Roe. Conservative upwind residual-distribution schemes based on the steady characteristics of the Euler equations. *AIAA Paper 95-1700*, June 1995.

[109] M. Pandolfi. A contribution to numerical prediction of unsteady flows. *AIAA Journal*, 22:602–610, 1984.

[110] K.M. Peery and S.T. Imlay. Blunt-body flow simulations. *AIAA Paper 88-2904*, 1988.

[111] K.G. Powell and B. van Leer. Tailoring explicit marching schemes to improve convergence characteristics. In *Computational Fluid Dynamics Lecture Series 1990-03, von Karman Institute for Fluid Dynamics, Belgium*, March 1990.

[112] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vettering. *Numerical Recipies: The Art of Scientific Computing.* Cambridge University Press, 1986.

[113] N. Qin and B.E. Richards. Sparse quasi-Newton method for high resolution schemes. *Notes in Numerical Fluid Mechanics*, 20:310–317, 1988.

[114] N. Qin and B.E. Richards. Sparse quasi-Newton method for Navier-Stokes solutions. *Notes in Numerical Fluid Mechanics*, 29:474–483, 1990.

[115] N. Qin, X. Xu, and B.E. Richards. Newton-like methods for fast high resolution simulation of hypersonic viscous flows. *Computing Systems in Engineering*, 3(1–4):429–435, 1992.

[116] J.J. Quirk. *An adaptive grid algorithm for computational shock dynamics.* PhD thesis, College of Aeronautics, Cranfield Institute of Technology, 1991.

[117] J.J. Quirk. A contribution to the great Riemann solver debate. NASA ICASE Report 92–64, November 1992.

[118] R. Radespiel, C. Rossow, and R.C. Swanson. Efficient cell-vertex multigrid scheme for the three-dimensional Navier-Stokes equations. *AIAA Journal*, 28(8):1464–1472, August 1990.

[119] R. Radespiel and R.C. Swanson. Progress with multigrid schemes for hypersonic flow problems. NASA ICASE Report 91–89, December 1991.

[120] G.D. Raithby. Skew upstream differencing schemes for problems involving fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 9:153–164, 1976.

[121] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. R. Soc. London, Series A*, 210:307–357, 1910.

[122] A. Rizzi. Numerical implementation of solid-body boundary conditions for the Euler equations. *ZAMM*, 58:301–304, 1978.

[123] T.W. Roberts. The behaviour of flux difference splitting schemes near slowly moving shock waves. In M.J. Baines and K.W. Morton, editors, *Numerical Methods for Fluid Dynamics III*. Oxford University Press, 1988.

[124] T.W. Roberts. Prospects for drag prediction using computational fluid dynamics. *SAE Technical Paper 901932*, 1990.

[125] P. L. Roe. Numerical algorithms for the linear wave equation. Royal Aircraft Establishment Technical Report 81047, 1981.

[126] P.L. Roe. The use of the Riemann problem in finite difference schemes. In W. C. Reynolds and R. W. MacCormack, editors, *Lectures in Physics No 141: Proc. 7th Int. Conf. on Num. Methods in Fluid Dynamics*, pages 354–359. Springer-Verlag, 1980.

[127] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, 43(2):357–372, October 1981.

[128] P.L. Roe. Some contributions to the modelling of discontinuous flows. In B. Engquist S. Osher and R. Somerville, editors, *Lectures in Applied Mathematics: Large Scale Computations in Fluid Mechanics*, volume 22, pages 163–193. American Mathematical Society, 1985.

[129] P.L. Roe. Characteristic-based schemes for the Euler equations. *Ann. Rev. Fluid Mech.*, 18:337–365, 1986.

[130] P.L. Roe. Discrete models for the numerical analysis of time-dependent multi-dimensional gas-dynamics. *J. Comp. Phys.*, 63(2):458–476, 1986.

[131] P.L. Roe. Upwind differencing schemes for hyperbolic conservation laws with source terms. Proc. 1st International Congress on Hyperbolic Problems, St. Etienne, France, January 13–17 1986.

[132] P.L. Roe. Sonic flux formulae. *SIAM J. Sci. Stat. Comput.*, 13(2):611–630, March 1992.

[133] P.L. Roe, H. Deconinck, and R. Struijs. Recent progress in multidimensional upwinding. In *Proc. of International Conference for Numerical Methods in Fluid Dynamics, Oxford*, July 1990.

[134] P.L. Roe and B. van Leer. Non-existence, non-uniqueness and slow convergence in discrete conservation laws. In M.J. Baines and K.W. Morton, editors, *Numerical Methods for Fluid Dynamics III*, pages 519–529. Oxford University Press, 1988.

[135] C-C. Rossow. Accurate solution of the 2-D Euler equations with and efficient cell-vertex upwind scheme. *AIAA Paper 93-0071*, 1993.

[136] J.W. Ruge and K. Strüben. Algebraic multigrid. In S. F. McCormick, editor, *Multigrid Methods, SIAM Frontiers in Applied Mathematics*, pages 73–131. SIAM, Philadelphia, June 1987.

[137] Y. Saad and M.H. Schultz. GMRES: A general minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Stat. Comp.*, 7(856–869), 1986.

[138] A.P. Saxer and M.B. Giles. Quasi-three-dimensional nonreflecting boundary conditions for Euler equation calculations. *Journal of Propulsion and Power*, 9(2):263–271, March–April 1993.

[139] H. Schlichting. *Boundary-Layer Theory.* McGraw-Hill, seventh edition edition, 1979.

[140] V. Schmitt and F. Charpin. Pressure distribution on the ONERA M6 wing at transonic mach numbers. In *AGARD Advisory Report 138: Experimental database for computer program assessment*, May 1979.

[141] L.K. Schubert. Modification of quasi-Newton method for nonlinear equations with a sparse jacobian. *Math. Comp.*, 24:27–30, 1970.

[142] C-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. NASA ICASE Report 87-33, May 1987.

[143] A.M.O. Smith. High lift aerodynamics. *Journal of Aircraft*, 12(6):501–530, June 1975.

[144] D.A. Smith, W.F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM Review*, 29(2), June 1987.

[145] G.R. Srinivasan, W.J McCroskey, J.D. Baeder, and T.A. Edwards. Numerical simulation of tip vorticies of wings in subsonic and transonic flows. *AIAA Paper 86-1095*, 1986.

[146] J.L. Steger. A preliminary study of relaxation methods for the inviscid conservative gasdynamics equations using flux splitting. Technical Report NASA CR 3415, March 1981.

[147] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Jornal of Numerical Analysis*, 21:995–1011, 1984.

[148] P.K. Sweby and M.J. Baines. On convergence of Roe's scheme for the general non-linear scalar wave equation. *J. Comp. Phys.*, 56:135–148, 1984.

[149] S. Ta'asan. Canonical forms of multidimensional steady inviscid flows. NASA ICASE Report 93-34, June 1993.

[150] S. Ta'asan. Canonical variables multigrid method for steady-state Euler equations. NASA ICASE Report 94-14, March 1994.

[151] C-H. Tai, J-H. Sheu, and B. van Leer. Optimal multi-stage schemes for Euler equations with residual smoothing. *AIAA Journal*, 33(6):1008–1016, June 1995.

[152] J.C. Tannehill, T.L. Holst, and J.V. Rakich. Numerical computation of two-dimensional viscous blunt body flows with an impinging shock. *AIAA Paper 75-0154*, 1975.

[153] E.F. Toro. Roe's method in gas dynamical problems associated with the combustion of high-energy solids in a closed tube. College of Aeronautics Report NFP86/18, Cranfield Institute of Technology, March 1986.

[154] E.F. Toro. A weighted average flux method for hyperbolic conservation laws. *Proc. Royal Soc. Lond.*, A 423:401–418, 1989.

[155] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comp. Phys.*, 72:277–298, 1987.

[156] E. Turkel. Review of preconditioning methods for fluid dynamics. ICASE Report 92-47, September 1992.

[157] E. Turkel, R.C. Swanson, V.N. Vasta, and J.A. White. Multigrid for hypersonic viscous two- and three-dimensional flows. ICASE Report 91-57, July 1991.

[158] E. Turkel and V.N. Vasta. Effect of artificial viscosity on three-dimensional flow solutions. *AIAA Journal*, 32(1):39–45, January 1994.

[159] M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *Journal of Aeronautical Sciences*, 23:805–824, 1956.

[160] G.D. van Albada, B. van Leer, and W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982.

[161] P.J. van der Houwen. *Construction of Integration Formulas for Initial Value Problems*. North-Holland Publishing Company, Amsterdam, 1977.

[162] B. van Leer. Towards the ultimate conservative difference scheme II. Monotinicity and conservation combined in a second order scheme. *J. Comp. Phys.*, 14:361–370, 1974.

[163] B. van Leer. Towards the ultimate conservative difference scheme V, A second-order sequel to Godunov's method. *J. Comp. Phys.*, 32:101–136, 1979.

[164] B. van Leer. On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe. *SIAM Journal of Scientific and Statistical Computing*, 5, 1984.

[165] B. van Leer. Upwind-difference methods for aerodynamic problems governed by the Euler equations. In B. Engquist S. Osher and R. Somerville, editors, *Lectures in Applied Mathematics: Large Scale Computations in Fluid Mechanics*, volume 22, pages 327–336. American Mathematical Society, 1985.

[166] B. van Leer. Euler solvers for transonic applications. In *NASA Transonic Symposium: Theory, Application and Experiment*, pages 217–230, 1989.

[167] B. van Leer. Progress with multi-dimensional upwind differencing. NASA ICASE Report 92–43, September 1992.

[168] B. van Leer, W-T. Lee, and K.G. Powell. Sonic-point capturing. *AIAA Paper 89-1945-CP*, 1989.

[169] B. van Leer, W-T. Lee, and P.L. Roe. Characteristic time-stepping or local preconditioning of the Euler equations. *AIAA Paper 91-1552-CP*, pages 260–282, 1991.

[170] B. van Leer, W-T. Lee, P.L. Roe, K.G. Powell, and C-H. Tai. Design of optimally smoothing multi-stage scheme for the euler equations. *Communications in Applied Numerical Methods*, 8:761–769, 1992.

[171] B. van Leer, C-H. Tai, and K.G. Powell. Design of optimally smoothing multi-stage schemes for the Euler equations. *AIAA Paper 89-1933-CP*, pages 40–59, 1989.

[172] B. van Leer, J.L. Thomas, P.L. Roe, and R.W. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations. *AIAA Paper 87-1104*, pages 36–41, 1987.

[173] V.N. Vasta, E. Turkel, and J.S. Abolhassani. Extension of multigrid methodology to supersonic/hypersonic 3-D viscous flows. ICASE Report 91-66, August 1991.

[174] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady-state solutions. *AIAA Paper 93-0880*, 1991.

[175] V. Venkatakrishnan. Preconditioned conjugate gradient methods for the compressible Navier-Stokes equations. *AIAA Journal*, 29(7):1092–1100, July 1991.

[176] V. Venkatakrishnan. Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters. *J. Comp. Phys.*, 118:120–130, 1995.

[177] M. Vinokur. An analysis of finite difference and finite volume formulations of conservations laws. *J. Comp. Phys.*, 81:1–52, 1989.

[178] H. Viviand. Pseudo-unsteady methods for transonic flow computations. In W.C. Reynolds and R.W. MacCormack, editors, *Lectures in Physics No 141: Proc. 7th Int. Conf. on Num. Methods in Fluid Dynamics*, pages 44–54. Springer Verlag, 1980.

[179] H. Viviand. Pseudo-unsteady systems for steady inviscid flow calculations. In F. Argand et. al., editor, *Numerical Methods for the Euler Equations of Fluid Dynamics*, pages 334–368. SIAM, 1985.

[180] Y. Wada and M-S. Liou. A flux splitting scheme with high-resolution and robustness for discontinuities. *AIAA Paper 94-0083*, 1994.

[181] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992.

[182] L.B. Wigton and R.C. Swanson. Variable coefficient implicit residual smoothing. *Lecture Notes in Physics*, 371:313–317, 1990.

[183] L.B. Wigton, N.J. Yu, and D.P. Young. GMRES acceleration of computational fluid dynamic codes. *AIAA Paper 85-1494*, pages 67–74, 1985.

[184] D.C. Wilcox. *Turbulence Modelling for CFD.* DCW Industries, 1993.

[185] B.R. Williams, W. Kordulla, M. Borsi, and H.W.M. Hoeijmakers. Comparison of solution of various Euler solvers and one Navier-Stokes solver for the flow about a sharp-edged cropped delta wing. In *AGARD conference proceedings 494: Vortex flow aerodynamics*, July 1991.

[186] X. Xu, N. Qin, and B.E. Richards. α-GMRES: A new parallelizable iterative solver for large sparse non-symmetric linear systems arising from CFD. *Int. J. Num. Meth. in Fluids*, 15:613–623, 1992.

[187] H.Q. Yang and A.J. Przekwas. A comparative study of advanced shock-capturing applied to Burgers' equation. *J. Comp. Phys.*, 102:139–159, 1992.

[188] H.C. Yee. A class of high-resolution explicit and implicit shock-capturing methods. Technical Report TM 101088, NASA, February 1989.

[189] H.C. Yee and P.K. Sweby. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations, Part II: The dynamics of numerics of systems of 2×2 ODEs and its connection to finite discretizations of nonlinear PDEs. Technical Report RNR-92-008, NASA, March 1992.

[190] H.C. Yee, P.K. Sweby, and D.F. Griffiths. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations, Part I: The ODE connection and its implications for algorithm development in computational fluid dynamics. *J. Comp. Phys.*, 97:249–310, 1991.

[191] H.C. Yee, P.K. Sweby, and A. Lafon. Basins of attraction and the time-dependent approach to obtaining steady-state numerical solutions. Preprint to appear in Proceedings of ICFD conference on Numerical Methods for Fluid Dynamics, Reading, England, April 1992.

[184] D.C. Wilcox. *Turbulence Modelling for CFD.* DCW Industries, 1993.

[185] B.R. Williams, W. Kordulla, M. Borsi, and H.W.M. Hoeijmakers. Comparison of solution of various Euler solvers and one Navier-Stokes solver for the flow about a sharp-edged cropped delta wing. In *AGARD conference proceedings 494: Vortex flow aerodynamics*, July 1991.

[186] X. Xu, N. Qin, and B.E. Richards. α-GMRES: A new parallelizable iterative solver for large sparse non-symmetric linear systems arising from CFD. *Int. J. Num. Meth. in Fluids*, 15:613–623, 1992.

[187] H.Q. Yang and A.J. Przekwas. A comparative study of advanced shock-capturing applied to Burgers' equation. *J. Comp. Phys.*, 102:139–159, 1992.

[188] H.C. Yee. A class of high-resolution explicit and implicit shock-capturing methods. Technical Report TM 101088, NASA, February 1989.

[189] H.C. Yee and P.K. Sweby. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations, Part II: The dynamics of numerics of systems of 2×2 ODEs and its connection to finite discretizations of nonlinear PDEs. Technical Report RNR-92-008, NASA, March 1992.

[190] H.C. Yee, P.K. Sweby, and D.F. Griffiths. Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations, Part I: The ODE connection and its implications for algorithm development in computational fluid dynamics. *J. Comp. Phys.*, 97:249–310, 1991.

[191] H.C. Yee, P.K. Sweby, and A. Lafon. Basins of attraction and the time-dependent approach to obtaining steady-state numerical solutions. Preprint to appear in Proceedings of ICFD conference on Numerical Methods for Fluid Dynamics, Reading, England, April 1992.